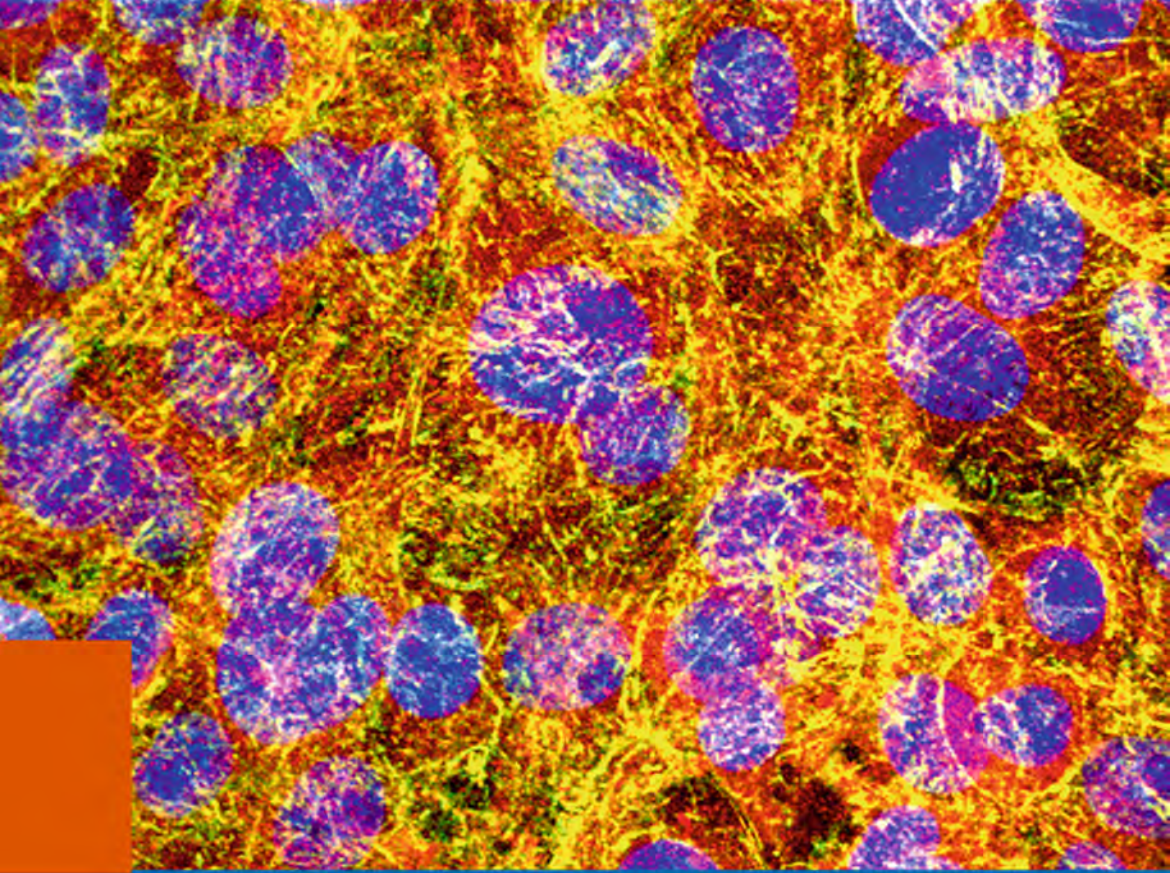


Biomedical Informatics for Cancer Research

Michael F. Ochs
John T. Casagrande
Ramana V. Davuluri
Editors

Springer

Biomedical Informatics for Cancer Research

Michael F. Ochs · John T. Casagrande
Ramana V. Davuluri
Editors

Biomedical Informatics for Cancer Research

Springer

Editors
Michael F. Ochs
Division of Oncology Biostatistics
and Bioinformatics
Sidney Kimmel Comprehensive
Cancer Center
John Hopkins University
Baltimore, MD 21205-2011
USA
mfochs@mac.com

John T. Casagrande
Cancer Research Informatics Core
University of Southern California Kenneth
Norris Jr.
Comprehensive Cancer Center
Los Angeles, CA 90089
USA
John.Casagrande@med.usc.edu

Ramana V. Davuluri
The Wistar Institute
Philadelphia, PA 19104
USA
rdavuluri@wistar.org

ISBN 978-1-4419-5712-2 e-ISBN 978-1-4419-5714-6
DOI 10.1007/978-1-4419-5714-6
Springer New York Dordrecht Heidelberg London

Library of Congress Control Number: 2010924049

© Springer Science+Business Media, LLC 2010
All rights reserved. This work may not be translated or copied in whole or in part without the written permission of the publisher (Springer Science+Business Media, LLC, 233 Spring Street, New York, NY 10013, USA), except for brief excerpts in connection with reviews or scholarly analysis. Use in connection with any form of information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed is forbidden.
The use in this publication of trade names, trademarks, service marks, and similar terms, even if they are not identified as such, is not to be taken as an expression of opinion as to whether or not they are subject to proprietary rights.

Cover Illustration: An image of matrix production by fibroblasts, showing the extracellular matrix and cell adhesion structures that play a key role in cancer cell behavior and metastasis. Indirect immunofluorescent images, of NIH-3T3 cells (murine fibroblasts), were acquired during the cellular process of extracellular matrix deposition. Nuclei are shown in blue while extracellular fibronectin matrix fibers (red) and cellular adhesion structures (integrin in green) can be seen co-localizing in yellow. Image was acquired after 4 days of matrix production. Image kindly provided by Dr. Edna Cukierman of the Fox Chase Cancer Center in Philadelphia.

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Michael F. Ochs
dedicates this volume to
Erica Golemis
Over half of our lives together so far, forget the asymptotic limit, let's go for all of our lives together.

John T. Casagrande
dedicates this volume to
Yolee Casagrande
Her continuous support and encouragement in all aspects of my life are greatly appreciated. I would like to especially thank her for encouraging me to pursue my "creative" side, which has resulted in this volume.

Ramana V. Davuluri
dedicates this volume to
Adinarayana Davuluri
I owe a lot to my father, whose commitment encouraged me to go to school. His loss to cancer inspired me to pursue medical research in cancer.

Acknowledgments

Many people dedicated significant effort to making this volume possible. I would like to thank Alisa Moore, who keeps my endless paperwork moving through the impressive NIH and Hopkins bureaucracies. I would also like to thank my coeditors, John and Ramana, for their efforts and especially for their patience – emails from me tend to arrive in fits and starts, almost always proposing an imminent deadline when at least a few weeks notice would have been more appropriate. They (almost) always responded rapidly. Finally, my thanks to all the authors who produced these chapters, despite the deluge of ARRA submissions and paperwork.

–Michael F. Ochs

I would like to thank Michael Ochs for inviting me to work on this project with him and Ramana; although we collaborated cross-country via e-mail, it has been a very gratifying and satisfying experience and a testament to the use of technology. Finally, I would like to thank those of you who are reading this volume, I hope you find it informative and helpful.

–John T. Casagrande

I would like to thank Michael Ochs for having me on this project, along with John. I know that I tested the patience of Michael, who constantly had to send me repeated emails to keep up with the deadlines. I am glad that the volume came out so well, I hope many of the readers will find the information very useful.

–Ramana V. Davuluri

Contents

Contributors

Waqas Amin
Department of Biomedical Informatics, UPMC Cancer Pavilion, 5150 Centre Avenue, Suite 301, Pittsburgh, PA 15232, USA

Anand Basu
Center for Biomedical Informatics and Information Technology, National Cancer Institute, 2115 East Jefferson Street, Suite #5000, Rockville, MD 20852, USA

Vicki Beal
St. Jude Children's Research Hospital, 262 Danny Thomas Place, Memphis, TN 38105, USA

Michael J. Becich
Department of Biomedical Informatics, UPMC Cancer Pavilion, 5150 Centre Avenue, Suite 301, Pittsburgh, PA 15232, USA

J. Robert Beck
Fox Chase Cancer Center, 333 Cottman Avenue, Philadelphia, PA 19111, USA

Ghislain Bidaut
Inserm, UMR891, CRCM, Integrative Bioinformatics, Marseille 13009, France
Institut Paoli-Calmettes, Marseille 13009, France
Université Méditerranée, Marseille 13007, France

Amanda Blackford
Division of Oncology Biostatistics and Bioinformatics, Johns Hopkins University, 550 North Broadway, Suite 1103, Baltimore, MD 21205, USA

Vaibhav Bora
Cancer Research Informatics Core, University of Southern California Kenneth Norris Jr. Comprehensive Cancer Center, Harlyne Norris Cancer Research Tower MC 9601, Los Angeles, CA 90089, USA

Walter Bosch
Department of Radiation Oncology at Washington University, 921 Parkview Place, Saint Louis, MO 63110, USA

Daniel Brat
Pathology and Laboratory Medicine, Emory University School of Medicine, 1648 Pierce Drive, Atlanta, GA 30322, USA

Vincent J Carey
Channing Laboratory, Brigham and Women's Hospital, Harvard Medical School, 181 Longwood Avenue, Boston, MA 02115, USA

John T. Casagrande
Cancer Research Informatics Core, University of Southern California Kenneth Norris Jr. Comprehensive Cancer Center, Harlyne Norris Cancer Research Tower MC 9601, Los Angeles, CA 90089, USA

Scott Catherall
Financial and Business Services, University of Southern California, University Gardens Building MC 8010, Los Angeles, CA 90089, USA

Reed Comire
Cancer Research Informatics Core, University of Southern California Kenneth Norris Jr. Comprehensive Cancer Center, Harlyne Norris Cancer Research Tower MC 9601, Los Angeles, CA 90089, USA

Ramana V. Davuluri
The Wistar Institute, 2601 Spruce Street, Philadelphia, PA 19104, USA

Mark DeTomaso
Cancer Research Informatics Core, University of Southern California Kenneth Norris Jr. Comprehensive Cancer Center, Harlyne Norris Cancer Research Tower MC 9601, Los Angeles, CA 90089, USA

Nilesh Dosooye
St. Jude Children's Research Hospital, 262 Danny Thomas Place, Memphis, TN 38105, USA

Yingliang Du
St. Jude Children's Research Hospital, 262 Danny Thomas Place, Memphis, TN 38105, USA

David Ervin
Department of Biomedical Informatics, Ohio State University, 333 W 10th Avenue, Columbus, OH 43210, USA

Richard Evans
University of California at Los Angeles, Transportation Services, Los Angeles, CA 90095, USA

Paul Fearn
Division of Biomedical and Health Informatics, Department of Medical Education and Biomedical Informatics, University of Washington, 1959 NE Pacific St #B509, Seattle, WA 98195, USA

T.J. Fitzgerald
Quality Assurance Review Center, 272 West Exchange Street, Suite 101, Providence, RI 02903, USA

Aris Floratos
Center for Computational Biology and Bioinformatics, Columbia University, 1130 St. Nicholas Avenue, New York, NY 10032, USA

Ian Fore
Center for Biomedical Informatics and Information Technology, National Cancer Institute, 2115 East Jefferson Street, Suite #6000, Rockville, MD 20852, USA

Robert A. Gatenby
Departments of Radiology and Integrative Mathematical Oncology, Moffitt Cancer Center, 12902 Magnolia Drive, Tampa, FL 33612, USA

Rama Gundapaneni
St. Jude Children's Research Hospital, 262 Danny Thomas Place, Memphis, TN 38105, USA

Shannon Hastings
Department of Biomedical Informatics, Ohio State University, 333 W 10th Avenue, Columbus, OH 43210, USA

Kristina Holton
Dana-Farber Cancer Institute, 44 Binney Street, Smith 822A, Boston, MA 02115, USA

Eleanor Howe
Dana-Farber Cancer Institute, 44 Binney Street, Smith 822A, Boston, MA 02115, USA

Hyunseok Peter Kang
Department of Pathology, Roswell Park Cancer Institute, Elm and Carlton Streets, Buffalo, NY 14263, USA

Juli Klemm
Center for Biomedical Informatics and Information Technology, National Cancer Institute, 2115 East Jefferson Street, Suite #6000, Rockville, MD 20852, USA

George Komatsoulis
Center for Biomedical Informatics and Information Technology, National Cancer Institute, 2115 East Jefferson Street, Suite #6000, Rockville, MD 20852, USA

Tahsin Kurc
Center for Comprehensive Informatics, Emory University, 201 Dowman Drive, Atlanta, GA 30322, USA

Duncan Temple Lang
Department of Statistics, University of California at Davis, 4210 Mathematical Sciences Building, One Shield Avenue, Davis, CA 95616, USA

Stephen Langella
Department of Biomedical Informatics, Ohio State University, 333 W 10th Avenue, Columbus, OH 43210, USA

Frank J. Manion
University of Michigan Comprehensive Cancer Center, 1500 East Medical Center Drive, Ann Arbor, MI 48109, USA

James McNamee
School of Medicine, University of Maryland, 655 West Baltimore Street, Baltimore, MD 21201, USA

Clayton Naeve
St. Jude Children's Research Hospital, 262 Danny Thomas Place, Memphis, TN 38105, USA

Sarita Nair
Dana-Farber Cancer Institute, 44 Binney Street, Smith 822A, Boston, MA 02115, USA

Joyce C. Niland
City of Hope National Medical Center, 1500 East Duarte Road, Duarte, CA 91010, USA

Deborah Nolan
Department of Statistics, University of California, 367 Evans Hall MC 3860, Berkeley, CA 94720, USA

Michael F. Ochs
Division of Oncology Biostatistics and Bioinformatics, 550 North Broadway, Suite 1103, Johns Hopkins University, Baltimore, MD 21205, USA

Scott Oster
Department of Biomedical Informatics, Ohio State University, 333 W 10th Avenue, Columbus, OH 43210, USA

Tony Pan
Center for Comprehensive Informatics, Emory University, 201 Dowman Drive, Atlanta, GA 30322, USA

Andrew Pappas
St. Jude Children's Research Hospital, 262 Danny Thomas Place, Memphis, TN 38105, USA

Giovanni Parmigiani
Department of Biostatistics and Computational Biology, Dana-Farber Cancer Institute, 44 Binney Street, Boston, MA 02115, USA

Philip Payne
Department of Biomedical Informatics, Ohio State University,
333 W 10th Avenue, Columbus, OH 43210, USA

Roger D. Peng
Department of Biostatistics, Johns Hopkins Bloomberg School of Public Health,
615 North Wolfe Street, Baltimore, MD 21205, USA

Justin Permar
Department of Biomedical Informatics, Ohio State University,
333 W 10th Avenue, Columbus, OH 43210, USA

Jeet Poonater
Cancer Research Informatics Core, University of Southern California Kenneth Norris Jr. Comprehensive Cancer Center, Harlyne Norris Cancer Research Tower MC 9601, Los Angeles, CA 90089, USA

James Purdy
Department of Radiation Oncology, UC Davis Cancer Center,
4501 X Street, Sacramento, CA 95817, USA

John Quackenbush
Dana-Farber Cancer Institute, 44 Binney Street, Smith 822A, Boston,
MA 02115, USA

Srinivas Raghavan
St. Jude Children's Research Hospital, 262 Danny Thomas Place,
Memphis, TN 38105, USA

Layla Rouse
City of Hope National Medical Center, 1500 East Duarte Road,
Duarte, CA 91010, USA

Joel Saltz
Center for Comprehensive Informatics, Emory University,
201 Dowman Drive, Atlanta, GA 30322, USA

Momodou Sanyang
St. Jude Children's Research Hospital, 262 Danny Thomas Place,
Memphis, TN 38105, USA

Daniel Schlauch
Dana-Farber Cancer Institute, 44 Binney Street, Smith 822A, Boston,
MA 02115, USA

Frank Sculli
Biodigital Systems, 594 Broadway, New York, NY 10012, USA

Sundeep Shakya
St. Jude Children's Research Hospital, 262 Danny Thomas Place,
Memphis, TN 38105, USA

Ashish Sharma
Center for Comprehensive Informatics, Emory University, 201 Dowman Drive, Atlanta, GA 30322, USA

Roshan Shrestha
St. Jude Children's Research Hospital, 262 Danny Thomas Place, Memphis, TN 38105, USA

Frank Siebenlist
Mathematics and Computer Science Division, Argonne National Laboratory, 9700 South Cass Avenue, Argonne, IL 60439, USA

Raktim Sinha
Dana-Farber Cancer Institute, 44 Binney Street, Smith 822A, Boston, MA 02115, USA

John Speakman
Center for Biomedical Informatics and Information Technology, National Cancer Institute, 2115 East Jefferson Street, Suite #6000, Rockville, MD 20852, USA

Matt Stine
St. Jude Children's Research Hospital, 262 Danny Thomas Place, Memphis, TN 38105, USA

Victoria Stodden
Yale Law School, 127 Wall St, New Haven, CT 06511, USA

Aarti Vaishnav
Cancer Research Informatics Core, University of Southern California Kenneth Norris Jr. Comprehensive Cancer Center, Harlyne Norris Cancer Research Tower MC 9601, Los Angeles, CA 90089, USA

William Weems
University of Texas Health Science Center at Houston, 7000 Fannin, Suite 1200, Houston, TX 77030, USA

Section 1
Concepts, Issues, and Approaches

Chapter 1
Biomedical Informatics for Cancer Research: Introduction

Michael F. Ochs, John T. Casagrande, and Ramana V. Davuluri

Abstract Biomedical informatics encompasses a set of disciplines focused on developing, implementing, and perfecting the use of informatics and computational tools in biomedical research and clinical care. In this volume, we focus on a number of areas crucial to the establishment of state-of-the-art informatics methods and systems to support cancer research. We provide motivation for undertaking such developments and deployments, a quick overview of the field, and hopes for the impact on cancer treatment and survival in this introduction.

1.1 The Goals of Biomedical Informatics for Cancer Research

Biomedical informatics is a field that focuses on the leveraging of computational and informatics resources for improving medical care and research. The increasing use of computers for biological research, for instance through the analysis of sequence similarity between genes and proteins or annotation of high-throughput data, led to the development of the field of bioinformatics. This followed the earlier emergence of medical informatics, which has resulted in the development of ontologies for disease and treatment, standards for medical and pharmacological data exchange, and techniques in effectiveness analysis for medical decision making. In addition, ontologies and terminologies have been developed to codify areas of knowledge, permitting searching of the vast medical literature. These fields form the basis of biomedical informatics.

M.F. Ochs (✉)
Division of Oncology Biostatistics and Bioinformatics, Johns Hopkins University,
550 North Broadway, Suite 1103, Baltimore, MD 21205, USA
e-mail: mfo@jhu.edu

M.F. Ochs et al. (eds.), *Biomedical Informatics for Cancer Research*,
DOI 10.1007/978-1-4419-5714-6_1, © Springer Science+Business Media, LLC 2010

1.1.1 The Burden of Cancer

Cancer is a depressingly prevalent disease, with 40% of all individuals expected to develop cancer during their lifetimes (SEER statistics 2004–2006, NCI, Bethesda, MD). In addition, in contrast to the equally prevalent heart disease and diabetes, treatments have not succeeded in significantly reducing morbidity. This reflects the complexity of the disease, with cancer showing great heterogeneity at the molecular level. However, all cancers share the traits of uncontrolled growth, leading to a single set of cells spreading throughout the body and absorbing all resources or putting pressure on normal organs until death occurs. Early detection of some cancers can lead to effective lifetime cures; however, many cancers are detected only at late stages or can recur regardless of early treatment.

Despite the complexity of treating cancer, at an individual patient level there have been some hopeful signs for the future. New targeted therapies, for example, imatinib mesylate, have completely changed the prognosis in specific cancers like chronic myelogenous leukemia (Druker 2001). This success has led to an explosion in the development of molecularly targeted therapeutics, including sorafenib and sunitinib, which have added significantly to progression-free survival in renal cell carcinoma (Favaro and George 2005). However, such progress remains far short of a cure, with improved survival measured in months or years, not decades. Our goal must be vastly different from our accomplishments to date, with a focus on making cancer a treatable if not curable disease. Biomedical informatics coupled to computational modeling and statistical analysis promises to accelerate the achievement of this goal through the codification of knowledge from the clinic and the bench (ontologies), the development of models (computational biology), and the testing of emerging hypotheses (statistics). In support of this goal, biomedical researchers are developing a substantial infrastructure to capture data, mine and analyze it, and present results in meaningful ways to clinical and bench researchers in order to have a significant impact on the disease process.

1.1.2 Treating the Individual Cancer: The Role of Informatics

The revolution in molecular biology has led to a much deeper understanding of the etiology of cancer in the last two decades. Importantly, the large investment in cancer research led to the clarification of the role of genetic mutation (Cho and Vogelstein 1992) and the identification of specific molecular processes (Hanahan and Weinberg 2000) as the fundamental drivers of cancer. With these discoveries, we finally understood the deep heterogeneity in cancer, with phenotypically identical behavior in patients arising from different molecular aberrations. Recent studies validated this

view, showing that multiple molecular pathways must be affected for cancer to develop, but with different specific proteins in each pathway mutated or differentially expressed in a given tumor (The Cancer Genome Atlas Research Network 2008; Parsons et al. 2008). Different studies demonstrated that while widespread mutations exist in cancer, not all mutations drive cancer development (Lin et al. 2007). This suggests a need to target only a deleterious subset of aberrant proteins, since any treatment must aim to improve health to justify its potential side effects.

Treatment for cancer must become highly individualized, focusing on the specific aberrant driver proteins in an individual. This drives a need for informatics in cancer far beyond the need in other diseases. For instance, routine treatment with statins has become widespread for minimizing heart disease, with most patients responding to standard doses (Wilt et al. 2004). In contrast, standard treatment for cancer must become tailored to the *molecular phenotype* of an individual tumor, with each patient receiving a different combination of therapeutics aimed at the specific aberrant proteins driving the cancer. Tracking the aberrations that drive cancers, identifying biomarkers unique to each individual for molecular-level diagnosis and treatment response, monitoring adverse events and complex dosing schedules, and providing annotated molecular data for ongoing research to improve treatments comprise a major biomedical informatics need.

Each individual also has a specific genetic background and environmental insults, which encourages a unique path in the development of each cancer leading to the diverse molecular phenotypes. There are examples where this is not the case, primarily in pediatric cancers that often have single driving aberrations or in specific types of cancer that follow a certain sequence of aberrations in many cases (Cho and Vogelstein 1992). However, even in these cases there exist many exceptions, far in excess of the fraction that fall outside standard treatment in heart disease or diabetes.

The role of genetics suggests a need for knowledge of family histories in cancer. Such histories are already very valuable for counseling and decision making in cases where a strong genetic basis for a significant portion of the population risk exists (Parmigiani et al. 1998; Berry et al. 2002). In addition, genotyping coupled to modeling of cancer risk will improve our ability to advise patients. However, successful use of this knowledge will require appropriate integration into the informatics framework, both to gather and provide information to the patient and to protect privacy.

1.1.3 Evidence-Based Medicine

For the above knowledge to be beneficial to the patient, it will also be necessary to make it available to practicing oncologists and other cancer care givers so they can tailor an appropriate treatment plan based on all the available information. As defined by Sackett et al. (1996), evidence-based medicine (EBM) is the "*integration of individual clinical expertise with the best available external clinical evidence*

from systematic research." The definition of systematic research will likely need to be broadened from randomized clinical trials or observational studies to accommodate the newly obtained personal profiles. In addition, the obstacle of disseminating or "diffusing" EBM in a timely and efficient manner to all providers, as discussed by Shojania and Grimshaw (2005), will need to be overcome before it can evolve into evidence-based management (Shortell et al. 2007).

1.1.4 Electronic Records

There is a long history of utilizing computers in cancer research. In the late 1960s into the 1970s, they were used to perform computational analyses and enumeration and management of cancer incidence and mortality statistics for population-based cancer registries. There was also a growing interest in utilizing computers to track and record information in the hospital and clinic settings. Many of these early systems were event based, since these systems were largely intended for billing or accounting activities in a hospital. In one early approach of designing a computerized medical record, Hammond et al. determined that if they captured all relevant activities, services, or resources that an outpatient received during a visit and associated these with the patient rather than a "department," a more beneficial "electronic record" system could be developed (Hammond et al. 1980). One of the editors (JTC) implemented the TMR system in 1983 in a Comprehensive Cancer Center and it served as the operational system of the Center for over 20 years before it was replaced.

The term "electronic records" now has several different connotations as noted in Chap. 2. The fundamental advantage to a researcher having access to such a system is that first, it can serve as a subject selection repository for investigations and second, it can also be a source of rudimentary information on the study population (discussed in detail in Chap. 2). Despite these advantages, there are also some who would argue that the cost and uniformity of the information gathered in the care setting is not sufficient for research purposes and that they rely on redundant or duplicative data collection systems for each research study. As mentioned in Chap. 2, the best approach is probably a mixed approach of centralized records and specialized additions.

1.2 The Components of Biomedical Informatics and Their Allies

1.2.1 The Electronic Medical Record and Data Warehouse

In the early 1970s, there was an emerging interest in using computers to assist in the management and care of patients (Collen 1991). Several of these early systems were targeted to the specific needs of oncology and were unique in that

they were patient focused rather than event- or transaction-focused. The most notable was the Oncology Clinical Information System developed at Johns Hopkins University by Blum, Lenhard, and others and summarized by Enterline et al. (1994). To some extent, these early systems were a victim of their time. Computer technology was rapidly changing, and with the advent of the microcomputer and the more sophisticated man–machine interfaces, these early systems became difficult to sustain due to user dissatisfaction or the cost of adapting to more modern technologies. In addition, many of these early systems had originated as research endeavors and were difficult to justify in the developing managed care environment. Another shortcoming of these early systems was their architecture, which was a "monolithic" approach attempting to encompass all the functionality needed in a single or modular system from a single vendor on a single hardware platform. This approach did not prove successful due to the amount of functionality needed and the time needed to complete development and implementation. In addition, the variability of scope and functionality across the modules from each vendor complicated the selection process. Rada and Finley (2004) provide an organizational evaluation of an aging oncology system and enumerate a number of the issues an institution must face when considering the replacement of a legacy information system.

In the early 1980s, a new approach emerged. The intent of this approach was to allow an organization to purchase the individual departmental systems that suit their business needs and then utilize a common "health bus" providing network and data standards to allow for data interchange and interaction between these "best of breed" systems (Collen 1991). This approach allowed for facile data interchange between various departmental subsystems, but it provided little in the way of coordinated system functionality. If the underlying message content could act as the "trigger" to fire off the required processes in the receiving system, coordinated functionality could be regained. A secondary problem was the flexibility in the HL7 standards, which provided the "health bus," that resulted in the unintended result that vendors were still able to introduce variability in their interface implementations based on their interpretation or preferences.

With "best of breed" approaches, it became necessary to aggregate and merge the distributed data into one database to ease analysis. Although there was no technological need to coalesce all organization's data assets into a single database or data warehouse, this greatly reduced the security concerns and performance issues that could occur if all the distributed databases were available to those needing to do cross-departmental searches. More recently, Ambite et al. (2001) has taken a more distributed approach to this problem that removes the requirement for all data to be merged into a single database. The current architectural approach to designing an HIS is based on object-oriented principles and a component- or services-oriented approach (Geissbuhler 2003). In this approach, the individual units, components, or services that are needed for the system are built separately but adhere to specific interfaces and context dependencies that can be deployed independently of each other (Szyperski 1997). Similar techniques are the basis of caGrid and of the caBIG® infrastructure (see Chaps. 4 and 9).

1.2.2 The Computational Grid and Access to It

The computational requirements both for data management and for computation will be substantial. In addition, much of the data that will be useful for an analysis will not be local in any sense, as it will comprise the knowledge codified in national data resources, such as those maintained by NCBI and NLM (Maglott et al. 2007), and the data gathered by the field as a whole (e.g., Edgar et al. 2002; Thorisson and Stein 2003; Parkinson et al. 2005). While a data warehouse could gather this data, this may not provide a scalable approach, as the data volume continues to increase exponentially in many molecular domains (sequence, transcripts, proteins, metabolites, etc.).

The use of data federation and its computational counterpart, a computational grid, provide another approach to the problem of the analysis of large integrated data sets. This approach, which is used by the caBIG® consortium (see Chap. 9), relies on bringing data sets together as needed from multiple sources, applying analytical tools to the data using resources on the grid (see Chap. 4), and returning results to the requesting party. This approach requires substantial informatics infrastructure both in terms of interoperability through a shared interface (syntax) and integratability of the data in a meaningful way (semantics). The issues involved are discussed in Chaps. 4 and 6.

Once resources are presented for use on the Internet, security becomes a major issue, especially as public and private health information will be involved. As such, grid technologies rely heavily on distributed authentication and authorization tools, which identify an individual as having access to the grid and detail those resources that the individual may access. The issues involved in such systems are discussed in detail in Chap. 5 and an example system is presented in Chap. 16.

1.2.3 Making Sense of Large Data Sets

The volume of data now being generated in even routine biological experiments exceeds that seen in traditional studies. The development of microarrays introduced biological and medical researchers to complex, dynamic high-dimensional data for the first time, and the result was the rediscovery of the need for statistical reasoning. This was most obvious in the poor quality of initial analyses of data from the new technology, which was followed by development of statistically validated methods for normalization of the arrays (Irizarry et al. 2003), identification of differential expression (Kerr et al. 2002), and discovery of patterns (Lukashin and Fuchs 2001; Moloshok et al. 2002).

The technologies now in routine use, including SNPchips, MS–MS proteomic analysis, next generation sequencing, and those under development, including metabolic profiling, miRNA arrays, and protein arrays, will provide unparalleled massive amounts of data. Analysis of this data will require the infrastructure of grid

computing and data warehousing, but it depends critically on the development of novel statistical approaches for analysis. The computational infrastructure for implementation of the methods is discussed in Chap. 6.

1.2.4 Modeling the Disease

Statistics can only reach so far in the determination of the causes of cancer and the identification of potential treatments. The strong nonlinearity and dynamic nature of the cancer system makes a mathematical description essential for a deep understanding of an individual cancer. While statistical analysis can identify potential drivers of carcinogenesis (Carter et al. 2009), the response to therapeutic treatment of the multiple changes or mutations that must have occurred for cancer development (Hanahan and Weinberg 2000) and an understanding of the interactions of the cancer cell with the external environment require a mathematical model due to the nonlinearity of the processes (Strogatz 2001).

Modeling cancer initiation and progression in mathematical models that replicate the transformation of normal cells to tumor cells can provide deep insight into molecular mechanisms that could provide targets for therapeutic development. Although mathematical modeling has dramatically impacted some areas of biomedical research (e.g., cardiac function and prosthetic design), major efforts in developing predictive mathematical models to guide experiment have not kept pace with analytical and statistical developments in cancer research. However, mathematical modeling combined with tailored experiments could lead to improved cancer treatment, as discussed for the targeting of aberrant signaling (Ventura et al. 2009). Issues in the mathematical modeling of cancer are discussed in Chap. 7.

1.2.5 The Goal of Reproducibility

The foundation of scientific research is the ability to reproduce previous results of experiments in multiple laboratories or previous treatment successes among different institutions. Remarkably, the development of high-throughput measurements in biology led to not only a failure in this regard, but also a reluctance to even provide adequate access to data and statistical code to allow fellow researchers to demonstrate reproducibility. While this is a result of the demands placed on modern researchers to produce progress in the form of publications and grant funding, it nevertheless reduces the validity of the overall scientific effort and draws this effort into question in a society that is often already suspicious of claims of medical advances. Advances in cancer treatment can reduce morbidity and mortality, but only if the patient population trusts that the novel treatments have a sound scientific basis.

The failure of so many high-profile, high-throughput studies in terms of reproducibility (Baggerly et al. 2004, 2005; Coombes et al. 2007) has led to a demand for reproducibility in analysis. In an era of highly computational analysis, this requires sharing of the data and code that led to a discovery. Such sharing can be done with blinded data, as demonstrated by patient-based data in national repositories (Edgar et al. 2002), and with code that both produces a textual description of the methodology and permits reanalysis of the data (see Chap. 8).

1.2.6 Reaching the Oncologist and the Patient

The framework for biomedical informatics outlined in this chapter and in this volume provides a technical and cultural solution, albeit with substantial financial and institutional demands, for cancer researchers. However, advances in treatments must reach the community physicians who care for the vast majority of cancer patients and the patients themselves. While computerized medical records (see Chap. 2) are becoming a reality in many areas, these alone will not keep physicians up-to-date on the latest treatment and biomarker developments.

Standards-based organizations will play a critical role in the improvement in community care through EBM. Fortunately for cancer treatment, the National Comprehensive Cancer Network (NCCN) has already established the basis for dissemination of guidelines for treatment (Miller 2000). If such guidelines can be automatically integrated into EMRs through workflows and order sets, and these guidelines updated to handle the demands of personalized tumor treatments, this could provide a major improvement in patient care.

1.3 National Efforts

1.3.1 National Centers for Biomedical Computing

The National Centers for Biomedical Computing (NCBCs) were established under the NIH Roadmap initiative, and the first centers were established in 2004. The goal of the NCBCs is to establish the national infrastructure for biomedical computing. Present NCBCs focus on Computational Biology, Informatics for Integration of Basic Research and Clinical Research, Medical Imaging, Biological Structure Simulation, Biomedical Ontologies, Integrative Informatics, and Multiscale Network Analysis. The centers support cooperative research with individuals outside the NCBC host institution, providing infrastructure and tools to accelerate research.

The tools developed in the NCBCs provide infrastructure and test-beds for many areas useful for cancer research. To provide just one example, the National Center

for Biomedical Ontology maintains numerous ontologies suitable for encoding data on model organisms, genes and proteins, experimental protocols, and biological structures. Such ontologies will play a crucial role in data integration and analysis, and this is discussed in Chap. 6.

1.3.2 The Cancer Biomedical Informatics Grid

The Cancer Biomedical Informatics Grid (caBIG®) is a further national informatics initiative. The National Cancer Institute (NCI) initiated caBIG® in 2003 with visits to the NCI-supported cancer centers. The focus of caBIG® is on an informatics infrastructure tailored to the needs of cancer research, which makes it an obvious source for biomedical informatics infrastructure for cancer researchers. A major focus of the initial efforts was the establishment of vocabulary and interoperability standards, similar to the NCBC centers focused on ontologies and integrative and bench to clinic informatics. In addition, the initiative identified early needs among cancer centers, allowing identification of a first round of tools to be built upon the emerging infrastructure. Because of the strong connection to cancer research, we include Chap. 9, devoted to the caBIG® project, and Chaps. 10, 14, and 16 on available caBIG® tools and infrastructure.

Successful creation of the needed biomedical informatics solutions for a cancer research center will rely on the purchase, development, implementation, and maintenance of a number of systems, combined with the training and culture-modification necessary to transform the enterprise and leverage the new infrastructure. The NCBCs and caBIG® project will provide some tools; however, they can also provide insight into successful methods of attacking informatics problems and expertise in deploying informatics systems for cancer research.

1.4 The Payoff

The investment to develop a biomedical informatics infrastructure that can support cancer research at a level necessary for the development of personalized medicine will be substantial. Spending within the caBIG® program alone in the 3-year pilot phase was $60 million, and this focused mostly on establishing an initial grid and vocabulary infrastructure with a few modifiable open-source tools. Integrating the EMR, establishing searchable data repositories with molecular and clinical data, and developing analytical and modeling tools will require substantial local and national investments. However, the result should be a vast improvement in outcomes as individuals are treated for the specific molecular aberrations driving their cancers.

1.4.1 Cancer as a Treatable Disease

The key to improving survival and quality of life for cancer patients lies in identifying the root cause of the disease. This root cause will be different at the level of individual molecular driving aberrations in each cancer, while retaining commonality at the level of overall cellular phenotypes and potentially pathways. The tracking of potential molecular triggers will be done at the population level and therefore will require the creation of shared national resources, such as maintained presently at the National Center for Biotechnology Information (NCBI) and the National Library of Medicine (NLM). These resources will need to be searchable both by individual researchers and by computational algorithms and therefore require the use of ontologies and vocabularies for data encoding. Much of this effort is already underway (Humphreys and Lindberg 1993; Rubin et al. 2006).

The population level data will provide a framework for understanding cancer. The measurements made on an individual tumor will then need to be integrated into this framework, requiring the integration of the clinical and laboratory records with population-based data. Vocabularies and ontologies will be essential for this integration. The data will need to be analyzed with new statistical methods capable of taking point measurements on individuals and interpreting them within the distributions from a population (e.g., Katz et al. 2006). This will require computational modeling of cancer, since the complexity of the interactions in biological systems makes them highly nonlinear, thus requiring modeling to determine robust and sensitive components prior to treatment.

Once the drivers of the individual cancer are identified and sensitive points for disruption of aberrant processes found, treatment must be designed. The hope is that future treatment will not involve broad cytotoxic regimens, such as standard platin-based chemotherapy or radiation, which damage all cells with substantial adverse side effects. Instead, each individual cancer will be modeled, the aberrant proteins that drive that cancer will be identified, the weak points where the tumor cell can be driven to cell death will be found, and the application of multiple-targeted therapeutics will be planned. Treatment will still result in adverse events and side effects; however, these should be minimal while disruption of tumor processes is maximized. Ideally, this could result in a cure; however, cancer is a robust system once established, and it may succeed in recovering from even such a targeted treatment. Thus, this process of analysis, modeling, and treatment may be ongoing, especially for advanced cancers.

Naturally this must all take place with an understanding that cancer is one aspect of the system – that the individual must be treated, not the tumor. In the end, the tools we build will enable treatment, but the doctor–patient relationship will still play the central role. Each patient will face hard choices, just as today, and patients and physicians will need to discuss options concerning the patient as a whole. Knowledge must guide the choices to maximize benefit, so methods to present the results of the complex interactions of data and analysis will be needed.

1.4.2 Building the Research Community

The vision of cancer research and treatment presented here requires a much greater integration of researchers and clinicians than we have today. Physicians will always remain the prime point for patient care; however, they will need a vast array of collaborators to bring the vision of personalized cancer care to fruition. The gathering and annotating of data, both clinical and biological, require close cooperation of informaticists and scientists. The maintenance of the data resources requires traditional information technology specialists. Analysis requires computational scientists and statisticians working closely with biologists to properly model the data and systems. The results must be presented in a meaningful way, relying on visualization techniques developed by computer scientists. The underlying infrastructure must be robust, which requires computer systems analysts and networking specialists. Applications must be built, which will require application programmers and collaborations with commercial vendors.

However, biomedical research in general and cancer research in particular have been slow to adopt the methodologies and technologies developed outside medicine, and this may reflect the difficulties of bringing together expertise from diverse fields. An example of successful integration of cultures is provided by high-energy physics, where theorists, experimentalists, computer scientists, statisticians, and engineers must all work together to accomplish the discovery and validation of a new subatomic particle. A prerequisite for this is a mutual respect for the contributions and difficulties faced in each subfield and the expertise developed by leaders in those fields. Unfortunately, many computational scientists of significant renown feel that mutual respect is not yet the norm in biomedical research, and this may well slow discoveries of fundamental significance given the overwhelming amounts of data now being generated. Addressing the cultural issues impeding collaboration may in the end be more important than addressing technical issues.

1.5 Conclusion

We are on the verge of major leaps forward in our understanding of cancer and its treatment. The discoveries derived from molecular biology and early targeted therapeutics promise a fundamental shift in treatment, away from general cytotoxic approaches to the targeting of the cancer cells. However, this vision brings with it a need for mathematical and computational resources in excess of any past experience of clinical and biological researchers. A substantial investment will be needed for cancer medicine and research to match the leaps forward in other areas of society, like banking and communications. In addition, an appreciation for the talents of all individuals involved in the process will be essential for progress.

References

Ambite JL, Knoblock CA, Muslea I, Philpot A (2001) Compiling source descriptions for efficient and flexible information integration. J Intell Inf Syst 16(2):149–187

Baggerly KA, Morris JS, Coombes KR (2004) Reproducibility of SELDI-TOF protein patterns in serum: comparing datasets from different experiments. Bioinformatics 20:777–785

Baggerly KA, Morris JS, Edmonson SR et al (2005) Signal in noise: evaluating reported reproducibility of serum proteomic tests for ovarian cancer. J Natl Cancer Inst 97:307–309

Berry DA, Iversen ES Jr, Gudbjartsson DF et al (2002) BRCAPRO validation, sensitivity of genetic testing of brca1/brca2, and prevalence of other breast cancer susceptibility genes. J Clin Oncol 20:2701–2712

Carter H, Chen S, Isik L et al (2009) Cancer-specific high-throughput annotation of somatic mutations: Computational prediction of driver missense mutations. Cancer Res 69:6660

Cho KR, Vogelstein B (1992) Genetic alterations in the adenoma–carcinoma sequence. Cancer 70:1727–1731

Collen MF (1991) A brief historical overview of hospital information system (HIS) evolution in the United States. Int J Biomed Comput 29:169–189

Coombes KR, Wang J, Baggerly KA (2007) Microarrays: retracing steps. Nat Med 13:1276–1277

Druker B (2001) Signal transduction inhibition: results from phase I clinical trials in chronic myeloid leukemia. Semin Hematol 38:9–14

Edgar R, Domrachev M, Lash AE (2002) Gene expression omnibus: NCBI gene expression and hybridization array data repository. Nucleic Acids Res 30:207–210

Enterline JP, Lenhard RE, Blum BI et al (1994) OCIS: 15 years experience with patient-centered computing. MD Comput 11:83–91

Favaro JP, George DJ (2005) Targeted therapy in renal cell carcinoma. Expert Opin Invest Drugs 14:1251–1258

Geissbuhler A (2003) Building man-man-machine synergies. Experiences from the Vanderbilt and Geneva clinical information systems. Int J Med Informatics 69:127–133

Hammond WE, Stead WW et al (1980) Functional characterisitics of a omputerized medical record. Methods Inf Med 19:157–162

Hanahan D, Weinberg RA (2000) The hallmarks of cancer. Cell 100:57–70

Humphreys BL, Lindberg DA (1993) The UMLS project: making the conceptual connection between users and the information they need. Bull Med Libr Assoc 81:170–177

Irizarry RA, Bolstad BM, Collin F et al (2003) Summaries of affymetrix genechip probe level data. Nucleic Acids Res 31:e15

Katz S, Irizarry RA, Lin X et al (2006) A summarization approach for affymetrix genechip data using a reference training set from a large, biologically diverse database. BMC Bioinformatics 7:464

Kerr MK, Afshari CA, Bennett L et al (2002) Statistical analysis of a gene expression microarray experiment with replication. Stat Sin 12:203–218

Lin J, Gan CM, Zhang X et al (2007) A multidimensional analysis of genes mutated in breast and colorectal cancers. Genome Res 17:1304–1318

Lukashin AV, Fuchs R (2001) Analysis of temporal gene expression profiles: clustering by simulated annealing and determining the optimal number of clusters. Bioinformatics 17:405–414

Maglott D, Ostell J, Pruitt KD et al (2007) Entrez gene: gene-centered information at NCBI. Nucleic Acids Res 35:D26–D31

Miller SJ (2000) The national comprehensive cancer network (NCCN) guidelines of care for nonmelanoma skin cancers. Dermatol Surg 26:289–292

Moloshok TD, Klevecz RR, Grant JD et al (2002) Application of Bayesian decomposition for analysing microarray data. Bioinformatics 18:566–575

Parkinson H, Sarkans U, Shojatalab M et al (2005) Arrayexpress – a public repository for microarray gene expression data at the EBI. Nucleic Acids Res 33:D553–D555

Parmigiani G, Berry D, Aguilar O (1998) Determining carrier probabilities for breast cancer-susceptibility genes BRCA1 and BRCA2. Am J Hum Genet 62:145–158

Parsons DW, Jones S, Zhang X et al (2008) An integrated genomic analysis of human glioblastoma multiforme. Science 321:1807–1812

Rada R, Finley S (2004) The aging of a clinical information system. J Biomed Informatics 37: 319–324

Rubin DL, Lewis SE, Mungall CJ et al (2006) National center for biomedical ontology: advancing biomedicine through structured organization of scientific knowledge. OMICS 10:185–198

Sackett DL, Rosenberg WMC et al (1996) Evidence based medicine: what it is and what it isn't. BMJ 312:71–72

Shojania KJ and Grimshaw JM (2005) Evidence-Based Quality Improvement: The State of the Science. Health Affairs 24(1):138–150

Shortell SM, Rundall TG et al (2007) Improving Patient Care by Linking Evidence-Based Medicine and Evidence-Based Management. JAMA 298(6):673–676

Strogatz SH (2001) Exploring complex networks. Nature 410:268–276

Szyperski C (1997) Component Software, 1st Edition. ACM, New York

The Cancer Genome Atlas Research Network (2008) Comprehensive genomic characterization defines human glioblastoma genes and core pathways. Nature 455:1061–1068

Thorisson GA, Stein LD (2003) The SNP consortium website: past, present and future. Nucleic Acids Res 31:124–127

Ventura AC, Jackson TL, Merajver SD (2009) On the role of cell signaling models in cancer research. Cancer Res 69:400–402

Wilt TJ, Bloomfield HE, Macdonald R et al (2004) Effectiveness of statin therapy in adults with coronary heart disease. Arch Intern Med 164:1427–1436

Chapter 2
Clinical Research Systems and Integration with Medical Systems

Joyce C. Niland and Layla Rouse

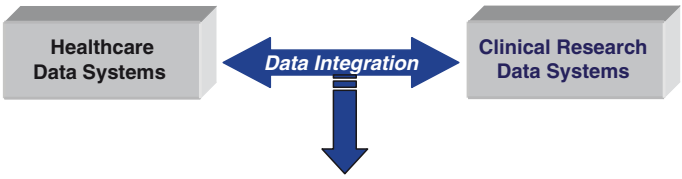
Abstract Integration of the Electronic Medical Records (EMR) with clinical research systems has the potential to greatly enhance the efficiency, speed, and safety of cancer research. New hypotheses could be generated through mining of EMR data, observational studies may be conducted more rapidly, and clinical trial recruitment and conduct could be greatly facilitated. Such enhancements will be accomplished through secondary use of EMR data for research and the development of automated decision support systems that rely on EMR data. In this chapter, we define the various types of EMR and clinical research data systems in use and describe the goals and rationale for integrating these two types of systems to enhance research as well as quality of care. The various approaches and benefits to integrating EMR and clinical research systems are discussed. While major benefits are conferred by such system integration, many challenges exist as well, such as the need for stringent data quality assurance, appropriate granularity, metadata and person index management, and extremely careful handling of data access and security issues. Furthermore, the movement toward the EMR within the USA has been slow to date, hampering these data integration efforts. However, recent legislation to incentivize the adoption of EMRs will make the feasibility and utility of EMR data integration to support clinical research more promising in the near future.

2.1 Introduction

It is critical that the efficiency, speed, and safety of cancer research be continually enhanced to make more rapid inroads and progress in battling this devastating disease. One approach to achieving these goals is to ensure that when conducting clinical research, full advantage is taken of the emerging role of electronic medical records (EMRs) in the field of cancer care. Yet an aspect of EMRs that has received

J.C. Niland (✉)
City of Hope National Medical Center, 1500 East Duarte Road, Duarte, CA 91010, USA
e-mail: JNiland@coh.org

M.F. Ochs et al. (eds.), *Biomedical Informatics for Cancer Research*,
DOI 10.1007/978-1-4419-5714-6_2, © Springer Science+Business Media, LLC 2010



	Secondary Use of Data	*Automated Decision Support*
I. ***Speed***	- Identification of subjects for trial - Import of pre-existing data into e-forms	- Algorithms to cull out ineligible subjects - Metadata-driven mouseover information
II. ***Efficiency***	- Hypothesis generation via data mining - Prevalence data for study planning	- Pattern dectection rules applied to data - Graphical contour plots of prevalence
III. ***Safety***	- Import of data to identify toxicities - Use of data to develop care guidelines	- Automatic grading of adverse events - Rule-based guideline adherence data

Fig. 2.1 Synergies between clinical research and medical data systems

little attention to date is the potential benefit of these systems to clinical research (Powell and Buchan 2005).

The integration of EMRs with clinical research systems enables two key forms of functionality: *secondary use of data* and *automated decision support*. Through the former, integration of these two types of data systems can facilitate the efficiency and speed with which cancer clinical research can be conducted. Through the latter, such integration can greatly improve patient safety, as well as efficiency, as clinical research is being conducted. The synergistic nature of these systems and the goals of each are depicted in Fig. 2.1. In this chapter, we will discuss the approaches, benefits, and challenges of integrating clinical research systems with medical care systems. First we introduce and define the terms and processes that will frame our discussion.

2.2 Electronic Systems to be Integrated

2.2.1 Clinical Research Data Systems

Clinical research data systems take on several different forms and functions. One of the most frequently deployed clinical research systems can be defined as a Clinical Data Management System (CDMS) which is used in clinical research to

manage the data of a clinical trial (i.e., an experimental interventional study conducted with human subjects), as well as other forms of clinical research such as observational, outcomes, or epidemiological trials (Summerhayes 2002; Tai and Seldrup 2000; Greenes et al. 1969; Clinical Data Management System Wikipedia 2009). The data to be stored in the CDMS may be gathered on paper forms, such as Case Report Forms (CRFs) in the case of a clinical trial, or on survey forms, questionnaires, and other data capture forms for observational research studies.

Another form of clinical research system more specific to the area of interventional clinical trials is known as a Clinical Trial Management System (CTMS). A CTMS consists of a customizable software system to manage large amounts of data involved with the operation of a clinical trial (Choi et al. 2005; Payne et al. 2003; see Chaps. 10–12). Such a system not only provides a data capture interface and data storage, but also provides additional functionality, such as maintaining and managing the clinical trial planning, preparation, performance; tracking deadlines, data expectations, and milestones; and reporting of clinical trials for regulatory and analysis purposes. Modules for handling trial budgeting and patient study calendars may be included in the CTMS as well. Compatibility with other data management systems is a highly desirable feature of any CTMS or related study management software tool.

Clinical research data collected during the investigation of a new drug or medical device is collected by physicians, nurses, and research study coordinators in medical settings (offices, hospitals, and universities) throughout the world. Historically, this information was collected on paper forms, which were then sent to the research sponsor (e.g., a pharmaceutical company) for entry into a database and subsequent statistical analysis. However, this process has a number of shortcomings, including that data are copied multiple times, producing errors that may not be caught until weeks later. To alleviate such issues, another type of clinical research system that has evolved within biomedical research is known as a Remote Data Entry (RDE) system (Electronic Data Capture Wikipedia 2009).

RDE systems allow research staff to enter data directly at the medical setting, particularly useful when a multicentered study is being conducted with many institutions participating. By moving data entry directly into the clinic or other facility, data checks can be implemented during data entry, preventing some errors altogether and immediately prompting for resolution of suspicious entries. Early RDE systems often used “thick-client” software installed on a laptop computer, such that the system needed to be deployed, installed, and supported locally at every participating site. This process becomes quite expensive for the study sponsor and complicated for the research staff. For Cancer Centers that typically participate in many research studies simultaneously, this deployment model for RDE results in a proliferation of different systems being installed, leading to complexity for the users along with space constraints.

In recent times, the user interface for RDE has shifted to Web-based deployments, for entry of data by the research team member directly into the system. EDC systems do not require local installation initially or with each software upgrade, but rather can be deployed centrally by the study sponsor for immediate and seamless

access by users. Although these systems are better than thick-client approaches, there are still cross-browser dependencies that need to be dealt with to make these Web-based systems truly universal. Typically an EDC system will include not only the graphical user interface (GUI) component for data entry, but also imbedded validation algorithms to rapidly check data for errors or suspicious entries and a reporting tool for synthesis and display of the collected data (Electronic Data Capture Wikipedia 2009). Such functionality formerly would be made available as separate software solutions within the CDMS or CTMS; however, integrated end-to-end solutions are evolving more recently. While EDC systems are primarily designed for the collection of data for clinical trials, there is no prohibition for this type of system to become equally popular and useful for observational research studies as well.

The term "electronic data capture" also may encompass several types of technology, beyond an electronic replacement for the CRFs that are completed at the enrolling site (Handleman 2005). EDC systems can include data capture technologies such as interactive voice response (IVR) systems, for example, to allow patients to report information over the phone (e.g., "press a key from 1 to 5 to describe your current pain level, with 5 being the highest"). Patient-reported outcomes collected via electronic diaries, for example using a Personal Digital Assistant (PDA) such as a Palm Pilot or similar device to record information best captured at home, also may be considered a form of EDC (Handleman 2005).

For simplicity, within this chapter we will use the more global term of "CDMS" to encompass any form of electronic clinical research data system to be integrated with medical systems.

2.2.2 Electronic Healthcare Data Systems

There are many limitations of paper medical records, including unavailability at the point-of-care (a given medical record cannot be in multiple places at once), inconsistent legibility, duplication of information, poor indexing of information, and inconsistency of information (Winkelman and Leonard 2004). To help alleviate such deficiencies, electronic healthcare data systems have been evolving. The National Cancer Institute (2009) defines an Electronic Medical Record (EMR) as "a collection of a patient's medical information in a digital (electronic) form that can be viewed on a computer and easily shared by people taking care of the patient." Though often used interchangeably, the terms EMR and Electronic Health Record (EHR) have different meanings in medical informatics. An EHR is defined as a "a longitudinal electronic record of patient health information generated by one or more encounters in any care delivery setting; including information on patient demographics, progress notes, problems, medications, vital signs, past medical history, immunizations, laboratory data, and radiology reports" by the Health Information and Management System (Electronic Health Record Wikipedia 2009). While increasing familiarity with the term "EHR" is being engendered by the 2009

Health Information Technology for Economic and Clinical Health (HITECH) Act (see below), we will use more technical informatics term of EMR for purposes of this chapter.

A related but distinct form of electronic system for the capture, management, and reporting of health information is the Personalized Health Record (PHR), defined as an electronic system to allow individuals to enter and manage their own private health information. Because the data come directly from the person him/herself, the advantages are that the information may be more completely and accurately captured from a personal view point. However, a disadvantage is that lay persons may not fully comprehend or enter data that is fully correct medically. Generally the term Health Information System (HIS) is reserved for electronic systems that go beyond even the EMR/EHR functionality to include features such as automated decision support (see below), alerting, and/or lifetime cumulative records. Another term that may be encountered is The Medical Record (TMR), designed to be a truly comprehensive personal health record, including a birth-to-death, time-oriented database of all parameters related to a person's well-being. Integrating data from all points of delivery and from all medical specialties, the TMR is envisioned to create a historical view of the health-related course of events in a person's life (Hammond et al. 1997).

Again for simplicity within this chapter, we will use the term "EMR" to encompass the several types of electronic healthcare systems defined above that could potentially be integrated with clinical research data systems.

To be considered a "full" EMR, typically a minimum of three functional components must be included in the system: computerized physician order entry (CPOE), both for computerized prescription orders and orders for tests; reporting of test results; and capture of caregiver notes (Electronic Health Record Wikipedia 2009). One of the largest national EMR projects has been implemented by the United Kingdom's National Health Service (NHS) that will include 60,000,000 patients within a centralized EMR by 2010 (Electronic Health Record Wikipedia 2009). As another example, Alberta Province in Canada has deployed Alberta Netcare, a large-scale operational EMR system (Electronic Health Record Wikipedia 2009). The United States (US) Department of Veterans Affairs has deployed the largest enterprise-wide health information system that includes an EMR, the Veterans Health Information Systems and Technology Architecture (VistA) (Electronic Health Record Wikipedia 2009). This system allows healthcare providers to review and update a patient's EMR at any of the more than 1,000 VA facilities around the country. The New York City Health and Hospitals Corporation, serving over 1.3 million patients in the largest urban US healthcare agency, is another positive example of a successfully implemented EMR (Electronic Health Record Wikipedia 2009).

The National Center for Health Statistics (2006) has indicated that the overall adoption of EMRs has been slow within the USA, in spite of a study showing revenue gains after implementation of a new billing technology. US healthcare industry spends only 2% of gross revenues on information technology compared to upwards of 10% within other information intensive industries such as finance (CDC

National Center for Health Statistics 2006). If all medical payment transactions were handled electronically, it has been estimated that America could save $11 billion annually (Medicare Part B Imaging Services 2008). Yet, the vast majority of healthcare transactions in the USA still take place on paper. Data from the 2005 National Ambulatory Medical Care Survey indicated that only about 25% of office-based physicians reported using EMRs. While this represented a 31% increase from the 18% reporting use of such systems in 2001, only 9.3% of the responding physicians reported having a "complete" EMR in place as of 2005 (CDC National Center for Health Statistics 2006).

Beginning in 2005, a private nonprofit branch of the US Department of Health and Human Services, the Certification Commission for Healthcare Information Technology (CCHIT), was established and charged with developing a set of EMR standards, in order to certify vendors who are able to meet these standards. Hopefully such product certification will provide US physicians and hospitals with the mandate and justification needed to make the significant investment of EMR implementation. By July 2006, CCHIT had released its first list of 22 certified ambulatory EMR products, and starting in early 2007, EMR vendors began utilizing these certification criteria in building their systems (Certification Commission for Health Information Technology 2009; Certification Commission for Healthcare Information Technology Wikipedia 2009). Additional barriers to adopting an EMR, beyond the daunting cost, include the complexity of such systems and the necessary change management and training to allow widespread adoption. Furthermore, the lack of a national standard for interoperability among competing software options is a major hindrance to widespread adoption of such tools (National Archives and Records Administration 2008).

In 2009, President Obama signed into law an economic stimulus package known as the "HITECH Act": Medicare and Medicaid Health Information Technology; Title IV of the American Recovery and Reinvestment Act. One aim of this legislation is to incentivize more medical practices to implement EMRs, by providing a financial subsidy for physicians who adopt and meaningfully use certified systems. Using a "carrot and stick" approach, the bill also progressively reduces Medicare reimbursement to any physicians who have not implemented an EMR by 2015 (Health Information Technology for Economic and Clinical Health Act 2009; Center for Medicare and Medicaid Services Fact Sheet 2009).

2.3 Goals to be Achieved Through CDMS-EMR System Integration

2.3.1 Secondary Use of Data

A major goal in integrating clinical research systems with electronic healthcare data systems is to achieve "secondary data use." Safran et al. (2007) documented that secondary use of data can be defined as "non-direct care use of

personal health information (PHI), including but not limited to use of such data for analysis, research, quality/safety measurement, public health, payment, provider certification or accreditation, and marketing and other business including strictly commercial activities" (Safran et al. 2007). The first few uses listed above touch on this important intersection of clinical care and biomedical research. Individuals and organizations involved in cancer research that may benefit from secondary data use from medical records include health services researchers and clinical investigators, disease registries, health data organizations, healthcare technology developers, and research or policy centers (Anonymous 1993).

The Institute of Medicine also has identified that two types of patient records exist, emphasizing that all users should not have access to all parts of patient records, so that patient confidentiality can be maintained (Institute of Medicine 1991):

(a) *Primary records* are those used by healthcare professionals while providing patient care services to review previously recorded data or to document their own observations, actions, or instructions.
(b) *Secondary records* are derived from primary records and contain data elements to aid nonclinical users in supporting, evaluating, or advancing patient care.

Such secondary record usage includes biomedical research to advance the evaluation and discovery of new treatments, better methods of diagnosis and detection, and prevention of symptoms and recurrences. Cancer clinical trial research can be enhanced and informed by some of the data collected during the practice of care, such as comorbid conditions, staging and diagnosis, treatments received, recurrence of cancer, and vital status and cause of death. Analytic observational studies may involve the use of valuable standard of care data available in the EMR from the routine practice of medicine. Quality/safety measures can be gleaned from the EMR in support of outcomes and comparative effectiveness research to determine whether new clinical trial findings are being adopted into the community of all cancer patients, what the most effective strategies are in the general cancer population, identify population groups, and conduct epidemiological studies.

Secondary data on health-related subjects extends beyond only clinical medical information and may also include administrative records; statistical reports of governments and other agencies; political/legal documents such as voting records, wills, contracts, laws, and statutes; organizational minutes; proceedings and reports; poll returns; survey data; commercial, industrial, and institutional records; historical documents; personal documents such as letters; and communications in the mass media (Brown and Semradek 1992). While several of these data types can be instrumental in supporting various forms of research (e.g., epidemiological investigations into disease etiology, case–control studies with neighborhood controls matched on socioeconomic factors), for the purposes of this chapter we will restrict our discussion of research uses of data to the clinical information arising within EMR systems.

2.3.2 *Automated Decision Support Systems*

Another potential benefit to clinical cancer research that can be conferred by integrating the CDMS with the EMR is automated decision support to enhance patient safety and study conduct efficiency. Automated Decision Support System (ADSS) can be defined as a rule-based system that is able to automatically provide solutions to repetitive management problems (Turban et al. 1997). Software components of an ADSS include rules engines, mathematical and statistical algorithms, and workflow applications. While the ADSS is frequently found in business settings, such systems can play a crucial role in the continual struggle to improve the quality and efficiency of patient care. A healthcare ADSS is based on rules or algorithms that trigger an automatic decision; however, unlike in business informatics, such rules typically are not automatically acted upon without final review and acceptance by the medical caregiver to provide the human interaction, adjudication, and expert knowledge layer needed for safety reasons.

An ADSS is most useful in situations that require solutions to repetitive problems that mostly involve electronically available information (Automated Decision Support System Wikipedia 2009). For the ADSS to be useful, the problem situation at hand must be clear and well understood, and the required knowledge and relevant decision criteria must be very clearly defined and structured, requirements that are particularly challenging to achieve in the medical field. Particularly in the conduct of interventional clinical trials, and to some extent within observational research, the healthcare ADSS can be important for improving the safety and efficiency of clinical research.

2.4 Rationale for Integrating the CDMS with the EMR

An ideal solution for leveraging the EMR to support clinical cancer research would be to extract patient data directly from the EMRs, as opposed to collecting the data in a separate data collection software application (Electronic Medical Record Wikipedia 2009). The convergence between patient care EMR systems within the broader healthcare ecosystem is expected to continue and perhaps could one day reach the point where separate CDMS and EMR systems would not be needed. However, in today's world this combined usage of a single electronic system to fully serve both patient care and clinical research needs is not yet tenable and would be extremely challenging on several levels.

First, both EMRs and CDMSs represent "transactional" database systems, built to support a specific business process and set of use cases. Medical records are structured primarily for the clinicians and administrators (Electronic Medical Record Wikipedia 2009). An EMR is a dynamic entity, affording greater efficiency and quality control to the work processes of clinicians by providing data entry at the point of care, logistical information access capabilities, efficient information

retrieval, user friendliness, reliability, information security, and a capability for expansion as needs arise (Electronic Medical Record Wikipedia 2009).

Patient care systems can streamline the many daily interactions with thousands of patients to avoid slowing the healthcare process while using an EMR. Because they resemble paper-based formats, these highly structured data formats encourage a greater standardization of data entry, thus, promoting collaborative and goal-directed treatment planning (Stam and van Ginneken 1995). Within EMR systems, structured entries (e.g., codes, classifications, and nomenclatures) are more frequently used over paper-based records (Thiru et al. 2003). However, much of the patient care information still is not entered using close-ended standardized coding schemas, as is needed for research and data analytic purposes.

In addition, the transactional data records of an EMR are indexed by patient and often by account/visit numbers, unlike the need to index by protocol and subject within research data systems. Further, the large research data queries that need to be conducted could greatly impede the daily performance of the EMR and interfere with patient care, if performed directly within these healthcare-driven systems.

Therefore, given the current state of EMRs, the varied complexity of patient care vs. clinical research, and the different nature of the transactional databases that support the two processes, this convergence into a single shared-purpose electronic data system is not yet on the horizon. Instead at this juncture, the goals of secondary use of EMR data and automated decision support for clinical cancer research can best be achieved through the integration of EMR and CDMS data systems. The potential approaches to patient care–clinical research data integration, along with the many benefits conferred and challenges faced, are discussed in the following sections.

2.5 Approaches to Integrating CDMS and EMR Systems

2.5.1 Point-to-Point Data System Integration

One technical approach to integrating an EMR with the CDMS in order to support clinical research would be a “point-to-point” data integration solution. In this instance, the exported data from the EMR would be directly imported into the CDMS, most often as a scheduled “batch” file update, for example, nightly. First, a detailed systems analysis needs to be conducted to determine what data elements exist within the EMR that would be useful for research purposes and that exist in an appropriate form. Ideally, the data would be in coded or numeric format (e.g., M = Male, F = Female, numeric laboratory data results, etc.) and at an appropriate level of granularity or specificity to suit the research purpose at hand. While open-ended text-based data could be imported into the CDMS as well, this form of unstructured data requires substantial manual curation on the clinical research side before it could be readily used for research purposes. An intermediate level of data between coded and open text would be structured text, for example, arising from

physicians conducting dictations using formatted standardized templates, so that consistent information is obtained with each dictation, in a predictable order.

When only two systems are involved, a single EMR and single CDMS, the point-to-point data technical integration approach would be quite reasonable. However, more frequently there are several source systems that could provide data to support research, for example, financial systems for cost–benefit analyses, ancillary healthcare systems not linked into the EMR, etc. When more than the two systems are involved, point-to-point data integration solutions quickly begin to break down, and it becomes intractable to manage the numerous interfaces and synchronization of data across all systems. Integrated biomedical data not only enhances clinical research, but could also benefit hospital quality assurance, accreditation reporting, caseload and volume analyses, as well as genotype–phenotype correlative research if the "omics" forms of data are integrated as well. Therefore, a much more flexible scalable technical approach to this data integration problem is the data warehouse, as described in Sect. 2.5.2.

2.5.2 Data Warehousing

As shown in Fig. 2.2, the data warehousing approach to data integration, while challenging, provides a highly extensible, large dimension data integration solution (see Chap. 3). In this approach, there can be many "feeder" data systems that provide valuable source data to be exported to and stored in the data warehouse. These systems could include ancillary care systems (laboratory, pathology, etc.) that may pass through the EMR itself to the warehouse or may represent stand-alone data systems that pass data into the warehouse.

Additional source systems could consist of the observational and/or clinical trial data systems into which data are collected specific to research, that are not available through the patient care systems. Such data might include graded adverse events, best response to treatment according to the protocol definition, and outside medical care records pertinent to the research project, but existing only on paper and not coded in the internal EMR system. In addition, the "omics" data arising from genomics and/or proteomics experiments, and stored in systems such as those described in Chaps. 13 and 14, could be synthesized and imported into the warehouse in an aggregated reduced-dimensionality format, to be merged with the treatment and biological "phenomic" data on the patients. As with the point-to-point solution above, a detailed systems analysis and data dictionary (i.e., metadata, data defining data) development is a critical prerequisite to a successful data integration project such as data warehousing.

The process of extracting the specific subset of data from the source systems into the data warehouse is called the "Extract-Transform-Load" or ETL process (Adelman and Moss 2000). Via an automated, scheduled routine, the required data elements are exported from the feeder systems, typically nightly or weekly, transformed to meet the data model of the warehouse, and loaded into the data warehouse data

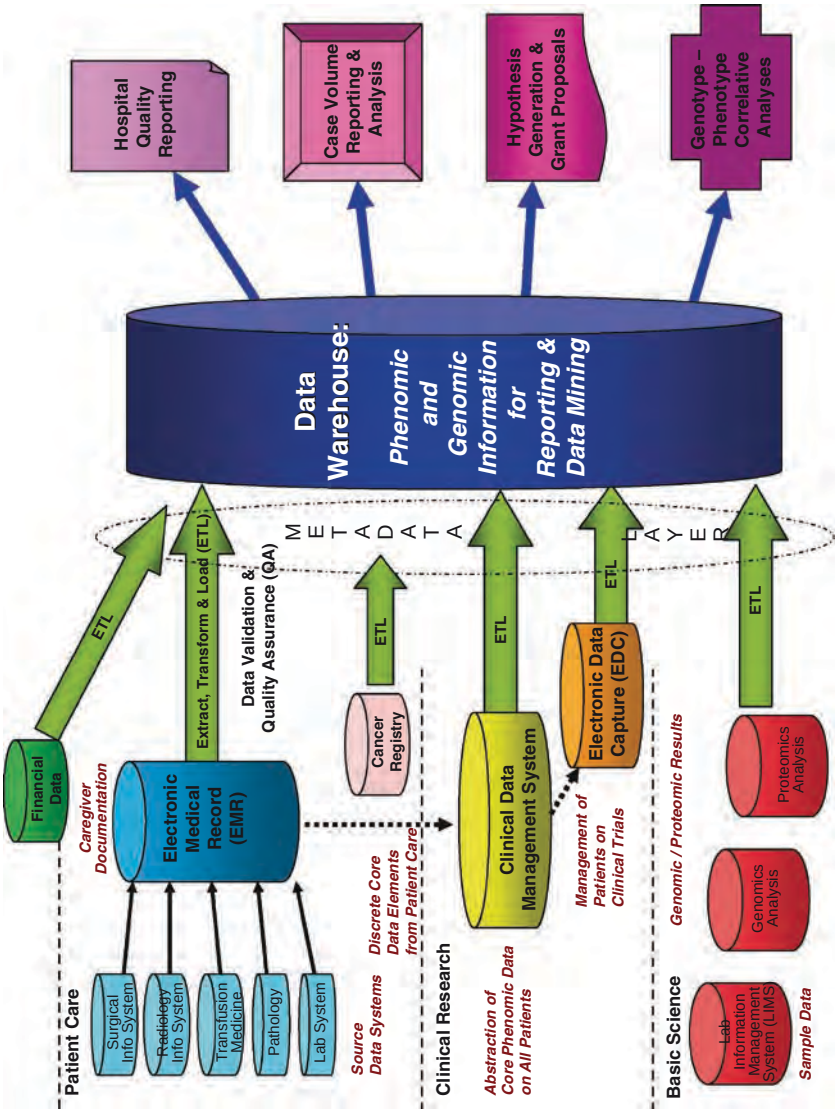


Fig. 2.2 Data warehouse schematic overview

structure. The underlying data model is usually specified as a "star schema" in order to provide the most efficient storage mode for data integration across the sources and subsequent abstraction for data mining purposes (Gray and Watson 1998).

As also shown in the Fig. 2.2, regardless of the technical integration solution, data quality assurance and validation are critical, as is metadata management, as described below. Once data are populated and integrated through a data warehousing approach, several types of "data marts" or subsets can be spun off from the main data store to meet different analytic and reporting purposes. These might include hospital quality assurance reports, evaluating whether complications of care and comorbidities are within acceptable ranges or case volume analyses to determine trends and plan for hospital beds and staffing. On the research side, clinical trials and observational research can be greatly facilitated through the integrated data, and genomic–phenomic correlative research facilitated through this highly valuable integrated data store.

2.5.3 Utilization of Standards

Regardless of which technical approach to data integration is utilized, it is crucial to follow existing and emerging data standards to ensure high-quality results and the ability to integrate across institutions, organizations, pathways, and diseases. Only through such standards will clinical research be advanced in a rapid highly organized manner, along with multicenter studies that are required to make more rapid biomedical discoveries.

Although few standards exist today for EMR systems as a whole, a number of standards exist relating to specific aspects of the EMR (Electronic Medical Record Wikipedia 2009). Adoption of several of these standards would greatly enhance the ability to conduct research on a global multidisciplinary scale when integrating data from the EMR for research. For example, the American Society for Testing and Materials (ASTM) International Continuity of Care Record (CCR) is a patient health summary standard based upon XML. The CCR can be created, read, and interpreted by various EMR systems, allowing easy interoperability between otherwise disparate entities (Electronic Medical Record Wikipedia 2009).

Standards for billing and financial purposes are available to potentially enhance data compatibility for research purposes, particularly because of their mandatory nature. The ANSI ASC X12 (EDI), a set of transaction protocols used for transmitting virtually any aspect of patient data, is in use in the USA for transmitting billing information, particularly as several of the transactions are required by the Health Insurance Portability and Accountability Act (HIPAA) (American National Standards Institute Accredited Standards Committee X12 Wikipedia 2009; Accredited Standards Committee X12 2009; Health Information Privacy 2009; Health Insurance Portability and Accountability Act 2009). Digital Imaging and Communications in Medicine (DICOM) standards are in widespread use for representing and communicating radiology images and reporting (Digital Imaging and Communications in Medicine Wikipedia 2009).

Interoperability can be defined as the ability of different information technology systems and software applications to communicate, to exchange data accurately, effectively, and consistently, and to use the information that has been exchanged (Electronic Medical Record Wikipedia 2009). The Health Level 7 (HL7) messaging standard is in use for interoperability among data from hospital, physician, EMR, and practice management systems (Health Level Seven 2009; Health Level 7 Wikipedia 2009). HL7 Version 2 has conveyed "syntactic" interoperability among these vendor-based systems, such that data can be physically imported from one HL7 compliant system to another. The next advance, HL7 Version 3, not only provides syntactic interoperability, but also provides, very importantly for research usage, "semantic" interoperability. Although adoption of this HL7 version has been relatively slow by vendors and others, once in place it will allow for meaningful standardized understanding and interpretation of the data being exchanged across data systems.

Additionally standard information models for clinical data and research are being developed at this time as well. The Clinical Data Interchange Standards Consortium (CDISC) is a voluntary initiative to develop standards for clinical data across the Food and Drug Administration (FDA), pharmaceutical companies, and research institutions, ideally worldwide (Clinical Data Interchange Standards Consortium 2009; Clinical Data Interchange Standards Consortium Wikipedia 2009). The Biomedical Research Integrated Group (BRIDG) model is collaboration among HL7, CDISC, and the National Cancer Institute (NCI) to provide a common integrated data model for clinical research (Biomedical Research Integrated Domain Group 2009). These standard-setting initiatives, some of which are described in Chap. 9, will greatly enhance and support the ability to integrate EMR and CDMS data for research in the future.

2.6 Benefits of Integrating CDMS and EMR Systems

The integration of electronic records arising from the EMR and the CDMS could facilitate new interfaces between care and research environments, leading to great improvements in the scope and efficiency of research (Powell and Buchan 2005). Clinical narrative information, captured electronically as structured data or as transcribed "free text," when combined with other existing data, can dramatically increase the breadth and depth of information available for nonclinical applications (Safran et al. 2007).

Clinical trials, outcomes research, survival analyses, survey studies, and epidemiological research in cancer could all benefit from secondary use of EMR data for research purposes. Secondary uses of health data can expand knowledge about cancer diagnoses and treatments, strengthen understanding of healthcare systems' effectiveness and efficiency, support public health and security goals, and aid businesses in meeting customers' needs (Safran et al. 2007). Possible research benefits range from systematically generating hypotheses for research to eventually undertaking entire studies based only on electronic record data. Information for planning studies, such as prevalence and variance of conditions in local contexts, could be collected with relative ease (Powell and Buchan 2005).

Researchers can utilize secondary data to supplement their own data, to expand on or check the findings of the original studies, to test hypotheses or analyze relationships quite different from those analyzed and reported in the original study (Brown and Semradek 1992). Using longitudinal patient care data, they may discover or identify trends in relation to changes in the social and physical environment (Brown and Semradek 1992). Another evolving use of patient records data is to support clinical practice for the development of guidelines for clinical practice (Anonymous 1993). Such usage of EMR data also facilitates outcomes research, in which guideline performance and success of patient care can be evaluated and correlated, much as is being carried out within the National Comprehensive Cancer Network (NCCN) outcomes research project (Niland 1998).

Vital statistics are essential for determining the health needs of the population and for program planning and evaluation. Disease-specific mortality rates help pinpoint the major health problems of the population and target at-risk groups for interventions, and natality and infant mortality data help in planning maternal and child health programs (Brown and Semradek 1992). The crucial survival analyses required for such research can be greatly facilitated through the mortality data available through the EMR. In addition to utilizing information available through the EMR, national registers of diseases and treatments could be established more easily and economically with a coherent approach to security across agencies (Robertson 2003). This process could accelerate and expand epidemiological research, via disease registries encompassing well-characterized populations (Robertson 2003).

In the course of providing cancer care, practitioners with access to an EMR rely on this system to monitor patient progress, provide continuity of care, maintain patient care standards, and monitor quality of care. Another major benefit of secondary usage of clinical care data within research is that automated decision support could be incorporated into the conduct of interventional research studies to help ensure the safety of patients as they are being treated with highly experimental drugs. As an example, City of Hope Cancer Center has developed and incorporated into their monitoring of cancer clinical trials a system called the Cancer Automated Lab-based Adverse Event Grading Service (CALAEGS). The CALAEGS is fed laboratory results and normal ranges for clinical trial patients from the City of Hope EMR to provide automated grading of lab-based adverse events (AEs). The CALAEGS system has been proven to greatly improve the accuracy and completeness of AE reporting for the many thousands of lab tests that must be assessed for a given trial, compared with the former manual method (Niland et al. 2007).

2.7 Challenges of Integrating CDMS and EMR Systems

Rapidly evolving nationwide efforts for more widespread health information exchange must include work to address pressing issues of secondary health data usage (Safran et al. 2007). However, there are many challenges associated with achieving this complex and difficult goal. Secondary use of health data poses technical,

strategic, policy, process, and economic concerns related to the ability to collect, store, aggregate, link, and transmit health data broadly and repeatedly for legitimate purposes (Safran et al. 2007). The current lack of coherent policies and standard "good practices" for secondary use of health data impedes efforts to transform the US healthcare system (Safran et al. 2007). As new record systems are designed, records and record-keeping habits need to be studied to improve our processes and to identify redundancies that can be eliminated in the future (Institutes of Medicine 1991). Extreme care must be taken and failsafe processes put in place to ensure that the appropriate record linkage is occurring both across the various EMR systems that may contain data on the same patient and between the EMR data and the clinical research data. Some of the critical factors for meeting the challenges of EMR-clinical research system integration are described here.

2.7.1 Metadata Management

Metadata or "data about the data" are critical to successfully document, interpret, and analyze patient care or clinical research data. Two general forms of metadata exist, the "technical metadata" utilized by the programming staff and database architects to define the structure of the database, including the field types, lengths, table storage locations, etc. The technical metadata generally arise from the creation of the database itself and are therefore readily available and accessible from the database management system.

The other form is the "business metadata," including the data definitions, directives for collection, allowable code lists, creation date, sunset date, etc. The business metadata is critical from the data user's perspective, but is not so readily available, as it takes a major human manual curation effort to diligently create and maintain the business metadata for any given electronic data system. Tools for business metadata management are not widely accepted and standardized, and it is tempting and all too easy to create a database system and fail to document this critical information in a timely manner or at all. Best practices would dictate that the database elements cannot be created, changed, or deleted without requiring the attendant business metadata to be documented. Only through such documented information can the integrated EMR and CDMS information be valid or meaningful as it is analyzed and reported.

2.7.2 Data Quality Assurance

Whether data are entered into an EMR or CDMS, or integrated via a data warehouse, data quality checking is a mandatory process to ensure valid, accurate, complete data, particularly as in most cases the data entered into these systems are several steps removed from the original source of the information. In the case of

interventional research such as clinical trials, the original source of the data includes the caregiver generating the observations on patients, or the laboratory, blood bank, or other healthcare application that processes a patient's sample results, or at times the patients themselves, for example, via completion of home diaries. In observational research, the data may be provided directly by the patient, for example, surveys, but still could contain inaccuracies or be incomplete due to recall issues, or misunderstanding of his/her medical condition. The data also could arise from a secondary source once removed from the primary subject, such as a family member or caregiver, who may not have full accurate knowledge of the desired information. While billing and financial information may be quite useful for research purposes, the quantitative data of administrative records often are imprecise and unreliable (Brown and Semradek 1992).

Data quality assurance is a laborious and imperfect process. When data entry is involved in capturing the data within a CDMS, a traditional but time-consuming method to decrease data entry errors is the process of double data entry. This process may be carried out by the same person who initially keyed in the data or preferably by a second independent party. Once data have been screened for typographical errors, the entries can be further validated to check for logical errors, such as mistakenly entering the patient's year of birth as the current year. In addition, process errors may be detected, for example through a check of the subject's age to ensure that they are within the inclusion criteria for the study. These instances are flagged for review to determine if there is an error in the data, an incorrect process has occurred within the study conduct, or further medical clarification from the investigator or caregiver is required.

2.7.3 Data Completeness

To achieve linkages and the ability to aggregate data, several conditions must be met. A set of core data elements will need to be defined and recorded for all patient records, ideally including problem lists with current status and clinical rationale, as well as standard data within future patient records that can be drawn upon for research (Institute of Medicine 1991).

One investigation found that many items of information that a researcher might desire frequently are not available. For example, while sex and age were routinely noted in over 90% of cases, other basic demographic information was less frequently available: marital status in 79% of cases, race 40%, occupation 40%, religion 36%, and education 35% (Brown and Semradek 1992). The absence of such core data elements clearly will handicap certain research, such as efforts to relate illness to environmental factors. Clinicians have recognized that data collection is more accurate and complete when accomplished while the patient is still in the hospital, rather than through retrospective chart review, as missing elements could be obtained from physicians and definitions could be more consistently applied (Robertson 2003). Because data can be reviewed on a daily basis, omissions or

errors can be identified and corrected while the patient and their records are still immediately available (Robertson 2003).

Those who elect to use secondary data, whether researchers, practitioners, educators, administrators, or policy makers, have the obligation to evaluate the data they employ and to demand high quality and completeness. Otherwise, based on unsound data, research will be compromised and end, if not in failure, in less than optimal success (Brown and Semradek 1992).

2.7.4 Data Coding and Granularity

Coding of data is a critical process for the capability of generating analyzable information (Rangachari 2007). Two key areas that are not widely available in coded manner in the EMR, but are required within the CDMS are adverse event terms and medication names. In cancer the Common Terminology Criteria for Adverse Events (CTCAE) is the most common grading scale, and standard dictionaries of these terms can be loaded into the CDMS. Then the data items containing the adverse event terms or medication names can be linked to one of these dictionaries. An emerging standardized coding system for drugs is the RxNorm system (NLM 2009). Some systems allow for the storage of synonyms to allow the system to match common abbreviations and map them to the correct term. As an example, ASA could be mapped to Aspirin, a common notation.

Because every medical practice has distinct requirements, EMR systems usually need to be custom tailored (Electronic Medical Record Wikipedia 2009). The majority of EMR systems are based on templates that are initially general in scope. These templates can then be customized in cooperation with the system developer to better fit data entry based on a medical specialty, environment, or other specified needs. These templates tend to be customized individually by each organization, with few reusable standards in place. There are also EMR systems available that do not use templates for data entry and therefore can be easily personalized by each individual user. While this is advantageous in terms of flexibility for individualized patient care, the process leads to silos of information and lack of standardized information that can be shared across data systems and integrated with the CDMS. Further, secondary data often are aggregated to a less granular level, and this fact, or the unit by which data are aggregated, may render the information unusable for research purposes (Brown and Semradek 1992).

Risk adjustment is required not only to account for differences in patient characteristics across hospitals to enable comparison of hospitals' outcomes (such as mortality rates or the complication rates), but also to adjust risks within research analyses (Iezzoni 1997). Hospital coding accuracy is critical for ensuring accurate risk adjustment and, correspondingly, reliable comparative quality ratings (Rangachari 2007). Existing studies on hospital coding accuracy have viewed coding from a purely reimbursement perspective rather than a quality-measurement perspective or for research purposes (Rangachari 2007).

2.7.5 Data Access and Security

Secure management of electronic records from either the CDMS or EMR is a major concern to protect the confidentiality of the individuals involved. Such concerns are magnified further with regard to the potential privacy risk additionally posed by integrating information across the CDMS and the EMR. There is a potential lack of protection of PHI when used by entities not explicitly covered by HIPAA legislation or regulations (Safran et al. 2007). While providing a reasonable solution to this problem is not difficult, providing a perfect solution to the problem currently is impossible (Hammond et al. 1997). Patients must be reassured that no personally identifiable information will be used for research without the consent of the individual (Robertson 2003). Establishing role-based security can help achieve protection of the information by restricting access to particular types of information within the system based on the individual's need to access the data and then providing access only to the necessary types of data (Niland et al. 2006).

2.8 Conclusions

It can be seen that there are many advantages to secondary use of healthcare data for the purposes of clinical and translational research. Many different forms of cancer research can benefit from the integration of the EMR with the CDMS (Niland and Rouse 2006). Observational studies and case series may be conducted more rapidly, and new hypotheses generated through data mining. In epidemiological research, previously undetected patterns of response or toxicity could be detected more readily if a core set of uniform high-quality data were available for all patients. Clinical trials could be greatly expedited by using the EMR data to screen for potentially eligible subjects and to document their presenting characteristics if they enter into the trial. During the trial conduct, test results could be imported electronically from the EMR, so that automated decision support could help guard the safety on patients receiving highly experimental treatment. Outcomes research analyses could be facilitated by the availability of coded data on subjects' past history, comorbidity, treatments, and long-term outcomes.

However, there are also many challenges to achieving the full benefits of integrated data across the CDMS and the EMR. Quality, consistency, and standardized coding of the EMR data must be in place both within an institution and among institutions. Care must be taken to fully safeguard the integrated data, as computerized databases of personally identifiable information may be accessed, changed, or deleted more easily and by more people than with paper-based records. Metadata that carefully documents the definitions, conditions under which data arise, coding schemas available, etc. must be complete and readily available to the users of the integrated information.

Yet there is no doubt that the emerging EMR holds great promise for speeding biomedical discoveries through integration with the CDMS data. It is hoped that EMR adoption and standardization will proceed rapidly throughout the USA, and other countries worldwide, so that this promise can be realized.

References

Accredited Standards Committee (ASC) X12. Retrieved 17 August 2009. http://www.x12.org/

Adelman S, Moss LT (2000) Data warehouse project management. Addison-Wesley, Upper Saddle River, NJ

American National Standards Institute Accredited Standards Committee X12 (ANS ASC X12). In: Wikipedia, the free encyclopedia. Retrieved 12 August 2009 from http://en.wikipedia.org/wiki/ASC_X12

Anonymous (1993) Users and uses of patient records. Report of the Council on Scientific Affairs, American Medical Association. Arch Fam Med 2(6):678–681

Automated Decision Support Systems (ADSS). In: Wikipedia, the free encyclopedia. Retrieved August 4, 2009 from http://en.wikipedia.org/wiki/Automated_decision_support

Biomedical Research Integrated Domain Group (BRIDG). Retrieved 17 August 2009. http://bridgmodel.org/

Brown JS, Semradek J (1992) Secondary data on health-related subjects: major sources, uses, and limitations. Public Health Nurs 9(3):162–171

CDC National Center for Health Statistics (2006) More physicians using medical records. http://www.cdc.gov/media/pressrel/a060721.htm?s_cid=mediarel_a060721. Accessed 20 August 2009

Center for Medicare and Medicaid Services (CMS) Fact Sheet (2009) Medicare and medicaid health information technology: Title IV of the American recovery and reinvestment act. Retrieved 12 August 2009. http://www.cms.hhs.gov/apps/media/fact_sheets.asp

Certification Commission for Health Information Technology (CCHIT). Retrieved 17 August 2009. http://www.cchit.org/

Certification Commission for Healthcare Information Technology (CCHIT). In: Wikipedia, the free encyclopedia. Retrieved August 4, 2009 from http://en.wikipedia.org/wiki/Certification_Commission_for_Healthcare_Information_Technology

Choi B, Drozdetski S, Hackett M, Lu C, Rottenberg C, Yu L, Hunscher D, Clauw D (2005) Usability comparison of three clinical trial management systems. AMIA Annu Symp Proc 2005:921

Clinical Data Interchange Standards Consortium (CDISC) (2009) Retrieved 15 August 2009. http://www.cdisc.org/

Clinical Data Management System (CDMS). In: Wikipedia, the free encyclopedia. Retrieved 4 August 2009 from http://en.wikipedia.org/wiki/Clinical_data_management_system

Digital Imaging and Communications in Medicine. In: Wikipedia, the free encyclopedia. Retrieved 12 August 2009 from http://en.wikipedia.org/wiki/Digital_Imaging_and_Communications_in_Medicine

Electronic Data Capture (EDC). In: Wikipedia, the free encyclopedia. Retrieved 4 August 2009 from http://en.wikipedia.org/wiki/Electronic_data_capture

Electronic Health Record (EHR). In; Wikipedia, the free encyclopedia. Retrieved 4 August 2009 from http://en.wikipedia.org/wiki/Electronic_health_record

Electronic Medical Record (EMR). Electronic Data Capture (EDC). In: Wikipedia, the free encyclopedia. Retrieved 4 August 2009 from http://en.wikipedia.org/wiki/Electronic_medical_record

Gray P, Watson HJ (1998) Decision support in data warehouse. Prentice Hall, Upper Saddle River, NJ

Greenes RA, Pappalardo AN, Marble CW, Barnett GO (1969) Design and implementation of a clinical data management system. Comput Biomed Res 2(5):469–485
Hammond WE, Hales JW, Lobach DF, Straube MJ (1997) Integration of a computer-based patient record system into the primary care setting. Comput Nurs 15(2 Supp 1):S61–S68
Handleman D (2005) Electronic data capture: when will it replace paper? Retrieved 4 August 2009. http://www.sas.com/news/feature/hls/sep05edc.html
Health Information Privacy. In: U.S. Department of Health & Human Services. Retrieved 12 August 2009. http://www.hhs.gov/ocr/privacy/
Health Information Technology for Economic and Clinical Health (HITECH) Act (2009) H.R.1 – American Recovery and Reinvestment Act of 2009. Retrieved 20 August 2009. http://www.opencongress.org/bill/111-h1/show
Health Insurance Portability and Accountability Act. In: Wikipedia, the free encyclopedia. Retrieved 12 August 2009 from http://en.wikipedia.org/wiki/HIPAA
Health Level Seven (HL7). Retrieved 17 August 2009. http://www.hl7.org/
Health Level Seven (HL7). In: Wikipedia, the free encyclopedia. Retrieved 15 August 2009 from http://en.wikipedia.org/wiki/Health_Level_7
Iezzoni L (1997) Risk adjustment for measuring health care outcomes. Health Administration Press, Chicago, IL
Institute of Medicine (1991) The computer-based patient record: an essential technology for health care. National Academy Press, Washington, DC
Medicare Part B Imaging Services (2008) United States government accountability office report to congressional requesters. http://www.gao.gov/new.items/d08452.pdf. Accessed 20 August 2009
National Archives and Records Administration (2008) Long-term usability of optical media. Retrieved 20 August 2009. http://206.180.235.135/bytopic/electronic-records/electronic-storage-media/critiss.html
National Cancer Institute. Electronic Medical Record, Dictionary of Cancer Terms. Retrieved 19 August 2009. http://www.cancer.gov/Templates/db_alpha.aspx?CdrID=561399
National Center for Health Statistics (2006) Electronic medical record use by office-based physicians, United States, 2005. http://www.cdc.gov/nchs/products/pubs/pubd/hestats/electronic/electronic.htm. Accessed 20 August 2009
National Library of Medicine (2009) National Institutes of Health, Unified Medical Language System, RxNorm. Retrieved 19 August 2009. http://www.nlm.nih.gov/research/umls/rxnorm/
Niland JC (1998) NCCN internet-based data system for the conduct of outcomes research. Oncology 12:11
Niland J, Rouse L (2006) Clinical research needs. In: Lehmann H, Abbott P, Roderer N et al (eds) Aspects of electronic health record-system, 2nd edn. Springer, New York
Niland JC, Rouse L, Stahl DC (2006) An informatics blueprint for healthcare quality. J Am Med Assoc 13(4):402–417
Niland JC, Pannoni S, Neat J, Sarbora R, Lee J (2007). Cancer Automated Lab Adverse Event Grading Service (CALAEGS). American Medical Informatics Association 'Clinical Research Informatics EXPO' Posters and Demonstrations
Payne PR, Greaves AW, Kipps TJ (2003) CRC Clinical Trials Management System (CTMS): an integrated information management solution for collaborative clinical research. AMIA Annu Symp Proc 2003:967
Powell J, Buchan I (2005) Electronic health records should support clinical research. J Am Med Internet Res 7(1):e4
Rangachari P (2007) Coding for quality measurement: the relationship between hospital structural characteristics and coding accuracy from the perspective of quality measurement. Perspect Health Inf Manag 4:3
Robertson J (2003) Cardiovascular point of care initiative: enhancements in clinical data management. Qual Manag Health Care 12(2):115–121

Safran C, Bloomrosen M, Hammond WE, Labkoff S, Markel-Fox S, Tang PC, Detmer DE (2007) Toward a national framework for the secondary use of health data: An American informatics association white paper. J Am Med Assoc 14:1–9

Stam H, van Ginneken AM (1995) Computer-based patient record with a cardiologic extension. Medinfo 8(Pt 2):1666

Summerhayes S (2002) CDM regulations procedures manual. Blackwell, London

Tai BC, Seldrup J (2000) A review of software for data management, design and analysis of clinical trials. Ann Acad Med Singapore 29(5):576–581

Thiru K, Hassey A, Sullivan F (2003) Systematic review of scope and quality of electronic patient record data in primary care. Br Med J 326:1070

Turban E, Leidner D, McLean E, Wetherbe J (1997) Information technology for management: transforming organizations in the digital economy, 6th edn. Wiley, Danvers, MA

Winkelman WJ, Leonard KJ (2004) Overcoming structure constraints to patient utilization of electronic medical records: a critical review and proposal for an evaluation framework. J Am Med Inform Assoc 11:151–161

Chapter 3
Data Management, Databases, and Warehousing

Waqas Amin, Hyunseok Peter Kang, and Michael J. Becich

Abstract This chapter provides a discussion on data management, databases, and data warehousing with particular reference to their utilization in cancer research. The section on data management describes the special requirements of data for research purposes. It discusses policies, ethics, and protocols involved in data collection, standardization, confidentiality, data entry and preparation, storage, quality assurance, and security. We have focused on the unique issues pertaining to data uniformity and consistency facilitating multi-institutional data sharing, data transfer, and collaboration. The section on Databases elaborates on the architecture and components of database systems. It also discusses various types of database systems with emphasis on the more commonly employed relational model of databases, database functions, and properties. In Data Warehousing the concept of data warehouses, along with warehouse architecture, technology, tools, and applications are discussed. A section on existing data resource systems has been detailed focusing on systems currently employed at the University of Pittsburgh to facilitate translational cancer research. There is a brief discussion on issues and approaches related to both databases and warehouses, which emphasizes their individual strengths and attributes.

3.1 Data Management

Data management can be defined as the development, execution, and supervision of plans, policies, programs, and practices that control, protect, deliver, and enhance the value of data and information assets.

M.J. Becich (✉)
Department of Biomedical Informatics, UPMC Cancer Pavilion, Room 305, 5150 Centre Avenue, Pittsburgh, PA 15232, USA
e-mail: becich@pitt.edu

M.F. Ochs et al. (eds.), *Biomedical Informatics for Cancer Research*,
DOI 10.1007/978-1-4419-5714-6_3, © Springer Science+Business Media, LLC 2010

3.1.1 Data Requirements in Cancer Research

Cancer research is an ever growing and advancing field in which there has been considerable investment of time and resources on institutional, national, and international data sharing. The demand for high quality, accurate, and comprehensive data to support genomic, proteomic, biobanking, clinical, and translational research is increasing. The data requirements in this field present novel challenges, which are summarized below:

- The data must first be acquired in accordance with legal and ethical (human subjects research) policies.
- The data must be acquired from sources that are ethically consented.
- The data must be accurate and verified.
- The data must be comprehensive and pertinent to the needs of the resource, biorepository, trial, or research that it is being collected for.
- This data must be standardized, consistent, and uniform to facilitate transfer, sharing, and interoperability across multiple institutes to enhance the pace of research.
- The data must be made secure and the highest level of importance has to be placed on the confidentiality of patient health information.
- This data must be quality assured and maintained.
- Data ownership issues need to be addressed.
- Data access, transfer, availability, and update processes must be adequately managed.

These issues are discussed in greater detail in the following sections.

3.1.2 Policies and Ethical Issues

For management of any kind of research data, clear and definite parameters should be defined regarding who is to be involved in providing and granting access as well as whom will be able to receive the data and to what extent access to the data will be provided. This is the domain of the Institutional Review Board (IRB). The IRB defines the institutes and individuals involved in data collection, handling, and utilization. It also approves the design and purpose of the intended research. No one is authorized to deal with the data unless preapproved by the IRB; also the data cannot be utilized for any purpose other than that explicitly presented in the research proposal and approved by the IRB. All sources of data including patients should be properly consented before participating in data collection. The IRB also takes into consideration all ethical matters related to cancer research before laying out its boundaries.

3.1.3 Sources

Data can be collected by authorized and trained personnel including research nurses, research associates, investigators, cancer registrars, and physicians, etc. Integrating data collection into the routine workflow of the hospital or research

facility streamlines the process and helps make it self-sustaining. The sources include patient interviews, questionnaires, patient health records and treatment charts, existing databases, consultation with referring physicians, archived data, and pathology reports. Data can also be acquired through automated electronic import from standardized sources such as synoptic (standardized) clinical records (see Chap. 2), Anatomic Pathology Laboratory Information System (AP-LIS), Clinical Pathology Laboratory Information System (CP-LIS), and other databases.

3.1.4 Levels of Data Collection

Data pertaining to various types of information can be collected. These include:

- Molecular level data
- Genomic level data
- Cell and tissue level data
- Pathology block level data
- Demographic data
- Patient clinical data
- Patient treatment data
- Biochemical data
- Outcome and follow up data

Other types of data collection can also be considered by the researchers' community, to fulfill the requirements of individuals, as well as in anticipation of the ever increasing needs in the field of cancer research.

3.1.5 Data Element Development

In order to ensure that collected data is comprehensive and easy to understand, facilitating sharing across multiple institutions, it must be uniform and standardized. To serve this purpose, data is collected in the form of structured (common) data elements. These structured data elements help in the integration of data from various clinical, pathological, and molecular resources into one design, supporting basic science and clinical as well as translational research. The coordinated use of such data elements can provide semantic and syntactic interoperability across multiple institutions and hospital data resources.

For translational cancer research standards such as the North American Association of Central Cancer Registry (NAACCR) (North American Association of Central Cancer Registries), Association of Directors of Anatomic and Surgical Pathology (ADASP) (Association of Directors of Anatomic and Surgical Pathology 2007), American Joint Committee on Cancer (AJCC) (Greene et al. 2002) and College of American Pathologists (CAP) (College of American Pathologists 2009) should be considered as starting points. These data elements are developed by a

committee from mutual consensus between all parties with a stake in data element development and data collection. This includes oncologists, research specialists, informaticians, surgeons, pathologists, molecular pathologists, microbiologists, genomics core directors, epidemiologists, occupational health specialists, biobankers, and experts from all fields involved in specific cancer research. This committee designs the structured data elements to fulfill the current needs of the research community as well as the projected needs of the data resource in the future.

Semantic interoperability can be achieved by describing the content, quality, condition, and other characteristics of structured data elements in the form of metadata or data descriptors and by using controlled vocabulary and ontology, making the data understandable and sharable for end users and flexible for the system. Each data element is associated with an object or concept, attribute, and valid value(s). For example, "patient age at diagnosis" is a data element that is made up of "patient" (object), "age at diagnosis" (property), and the representation (value domain) in "years." Specifically for each of the approved data elements, the data collectors need to know (1) the fundamental definition of the data element (i.e., date of diagnosis), (2) how that data element will be collected (e.g., 11/2003 vs. Nov. 2003 vs. 11/03, etc.), (3) what are the consensus acceptable values or codes for the data element (e.g., precise date of birth, not calculated from clinical records where the "patient appears to be a well developed 75-year-old"), and (4) what the acceptable data format is for inclusion into the database (e.g., dates as integers not character strings). Although the concept of formalized metadata is fairly straightforward, it has rarely been incorporated by clinical and research groups building databases (Mohanty et al. 2008).

The structured data elements developed can be ISO/IEC 11179 compliant, which helps to define the structure and format of the metadata. This defines a number of fields and relationships for metadata registries including a detailed metamodel for defining and registering items, of which the primary component is a data element (Patel et al. 2005; Mohanty et al. 2008).

3.1.6 Data Confidentiality

Before entry the data is primed to protect patient privacy and confidentiality according to IRB regulations and Health Insurance Portability and Accountability Act (HIPPA) approved protocols. The database should disclose only deidentified patient information and display no links to patient identifiers (name, date of birth, procedure date, therapy date, etc.). The only linkage should be kept within the institution/resource and the database should generate deidentified datasets upon query by the end users (the so-called safe harbor approach to HIPPA-compliance). The "safe harbor" approach involves exclusion of all identifiers itemized according to the HIPPA protocols. Thus, for example, a participant's age is presented as age range, rather than the date of birth, and therapy dates are provided in months from first positive tissue diagnosis to therapy start date rather than presenting a precise calendar date. These are some of the measures adopted to protect the identity of patients while still providing sufficient information for research purposes.

The deidentification process is performed by an honest broker that acts as a filter between completely identified confidential clinical patient information and the completely deidentified data made available to the research community. An honest broker is an individual, organization, or system acting on behalf of the covered entity to collect and provide health information to the investigators in such a manner whereby it would not be reasonably possible for the investigators or others to identify the corresponding patients–subjects directly or indirectly. The honest broker cannot be one of the investigators or researchers. A researcher may use an honest broker service to obtain the Protected Health Information in a deidentified manner. The honest broker service will deidentify medical record information by automated or manual methods. All honest broker services are approved in advance by both the IRB and research committee. The honest brokers may even be individuals who have clinical responsibilities, such as tissue bankers, postdoctoral fellows who manage the pathology data, or cancer registry specialists. Based on their clinical job duties, their educational backgrounds and experience may vary. Depending on the nature of the projects, these honest brokers can work autonomously or collaboratively to meet data needs (Dhir et al. 2008).

3.1.7 Data Entry and Data Preparation

Once the data has been *extracted* from the various sources it undergoes a process of *cleansing* (quality control measures) before it can be entered into the database. Cleansing operations classically include correction of typographical mistakes, other data entry errors, completing missing values, making sure the data is standardized etc. Any data that is obviously erroneous is excluded from entry into the database. When data from different sources is being merged, it undergoes a process of *consolidation*; relationships between the data from different sources are defined and data is synchronized. The data is then *loaded* into the system and undergoes consistency checking and integrity checks (Berry and Linoff 1997; Date 2000).

Data entry is the responsibility of data managers or authorized data entry personnel. Sometimes Web-based entry applications are utilized allowing data managers from different sites or collaborating institutions to enter data independently. When this strategy is used for data entry, secure protocols such as https: should be employed, so clear text data is not exposed on the network.

3.1.8 Data Storage

Data can be stored in a variety of manners depending on the requirements of the organization, resource, or institute involved. Data may be stored in relational, object-oriented, or other types of databases or in a data warehouse allowing for easier sharing and automated electronic data transfers and accruals. Back up copies of the data are maintained as both local electronic versions online, remote online, and tape backups stored in disaster recovery locations depending upon the institute/resource.

3.1.9 Data Quality Assurance

Once the data has been loaded into the database and the database is active, it must undergo repeated quality checks to maintain accuracy and validity. Data quality maintenance is the responsibility of a data manager. After importing the multimodal data into the database, accuracy is assessed by trained and certified personnel, using policies, variable constraints, and logical tests established by the resource. The evaluation of the collected data can be done using the following approach.

The first step is to evaluate discrepancies between the database quality check curators, such as the data managers and entry personnel. The primary focus of data accuracy assessment is on tumor record, staging, histology, diagnosis, treatment, recurrence, and risk factor exposure data. The error rate for each case is calculated from the number of discrepant entries and the number of fields evaluated for a case. Evaluated numbers of fields and numbers of discrepant entries for selected cases are used to find the error rate for each discrete priority level data field.

The second step evaluates the accuracy of database entries by comparing them to the electronic data source from which data is collected. The number of deviations from the entry per total number of fields assessed will yield an estimate of the error rate. Initially, 1% of the subjects are evaluated. If discrepancies are within error rate guidelines for fields, a further 5% of subjects will be randomly selected using the same strategy, and estimated database error rates will be calculated. If the error rate guidelines are not met for the 1% initial evaluation, a careful analysis of the differences will be performed and discrepancies identified. A quality check of all database subjects will be performed, focusing on discrepant fields. After the quality check has been completed, a second sampling of 1% of subjects will be performed. This sampling will exclude subjects sampled in the prior evaluations. Error rates will be determined, and if error rate guidelines are met, a further 5% of subjects will be evaluated using the same criteria.

The third step involves comparing the data in the database to that in primary sources such as clinical charts and pathology reports. Subject sampling will be performed and data field error rates will be calculated as above (Patel et al. 2005; Mohanty et al. 2008).

3.1.10 Data Security

Data security means *protecting the data against unauthorized utilization, disclosure, or damage*. This can be enforced by Discretionary Control or Mandatory Control. Discretionary control employs levels of privileges or access rights for the authorized users: a user can set the access permissions on resources that they own, modifying the groups or users permitted to access the data, together with the level of access allowed. In mandatory control, every unit of data itself is tagged with a specific classification for which a certain clearance level is mandatory to allow access. Discretionary control provides a much more flexible

environment, but mandatory control is a much more secure access system. However, mandatory control also imposes a high system management overhead, so the needs of the system and the advantages of each method must be evaluated carefully to select the one most appropriate for a given situation (Bell and Padula 1974). Also the actual data should be maintained in an encrypted form, so that it remains secure even in case of security bypasses. The original data is known as plaintext. This plaintext is encrypted by processing it through an encryption algorithm with an encryption key; the algorithm produces the cipher text, which is the encrypted data. This encryption is only as secure as the encryption key, access to which must be limited to designated personnel. The cipher text will be incomprehensible to anyone who does not possess the encryption key (Denning 1982; Date 2000).

There are other general considerations that are paramount in maintaining the security of data. These include:

- Deciding who should have access to the data and to what extent.
- Physical controls such as the security of the computer terminal and hardware, for instance security of the room containing the computer terminal or database server.
- Frequency of changing passwords, and maintaining secrecy of the passwords.
- Complexity and length.
- Security features built into the hardware such as storage protection keys or protected operation modes.
- Security features built into the operating systems, such as automatic deletion of disk files and temporary storage at completion of task (Bell and Padula 1974).

Another concept is data integrity. This refers to the accuracy and validity of the stored data. This is maintained by removal of erroneous data and regular quality checks. Database systems also have intrinsic integrity constraints to maintain the accuracy of data at every data transfer and upload.

3.2 Databases

A database is *a computerized system that is designed to store information and permits a user to obtain and update the stored information on demand*. This "on demand" nature of retrieving and modifying data assists greatly in the routine operating process of an individual or organization.

3.2.1 Architecture

Database architecture is based on the multiple levels that are described as follows (Fig. 3.1).

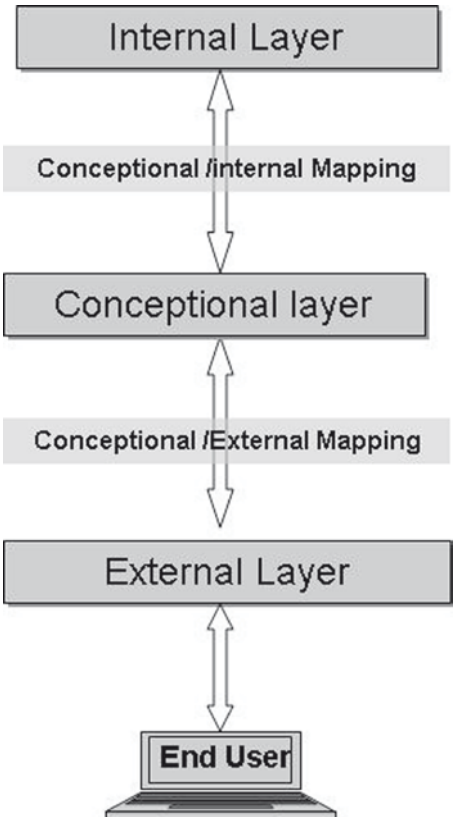


Fig. 3.1 Presents three different layers of database architecture and mapping for data integration among them

3.2.1.1 External Level

This level is also called the individual user level. The user can be a programmer involved in application development, an end user, or a database administrator. Each user approaches the application with a specific language, either traditional program languages like JAVA and C++, or specialized languages for end users such as query language, special purpose language or forms, or menu driven language developed to user requirement and handled by online applications. The languages for end users include the data sublanguage, which is a subset of the entire language dealing with particular database objects and operations. The data sublanguage is closely knitted to the parallel host language that deals with a variety of nondatabase utilities like computational operations, branching logic, local variables, etc. One particular data sublanguage that is compatible with most database systems is Structured Query Language (SQL). Theoretically, a data sublangauge is a combination of data definition language (DDL) that maintains definition of database objects and data manipulation language (DML) that supports handling and processing of data objects.

The external level provides an external view, a distinct view of some portion of the data that is physically stored in the database. Each external view is defined in an external schema, which basically consists of descriptions of each of the various types of external record in that external view (Date 2000).

3.2.1.2 Conceptual Level

This is also called the community logical level and it is an abstract representation of the total information content of the database independent of how the data is physically stored. The conceptual view is designed to present the real data to the users overcoming the limitations on their data visualization imposed by hardware or software restrictions. This view consists of multiple episodes of many types of conceptual records, defined in the conceptual schema, which is written in conceptual DDL. These records are not necessarily similar to external records or stored data. In order to achieve physical data independence, the conceptual DDL definitions should be limited to definitions of information content and must not involve physical structure or access techniques. In addition, the conceptual schema must not hold any reference to stored field representation, stored record sequence, indexes, hashing schema, pointers or any other storage, or access information. In this manner the conceptual schema is developed independent of the data and the external level is described in reference to the conceptual schema.

In essence, the conceptual view is a visualization of the database in its entirety, and the conceptual schema is a description of this conceptual view. The aim of the conceptual schema is to detail the entire project, for instance the flow, utilization, and regulation, etc., of data, and not just elaborate the data individually (van Griethuysen 1982).

3.2.1.3 Internal Level

This is also called the physical layer because it exists close to physical data storage and deals with how data is actually stored. This deepest level of a database holds repetitive stored records. The internal view is described by the internal schema, which defines a variety of accumulated record types, indices, the representation of stored fields, and the physical sequence that records are stored in. The internal schema is written in internal DDL (Tsatalos et al. 1994).

3.2.1.4 Mapping

In addition to the three levels of architecture, specific mappings are required among these levels. The conceptual/internal mapping describes the communication between the conceptual and internal data storage levels, and the representation of conceptual records and fields in the internal level. The conceptual/internal mapping changes according to any structured data change in the internal level allowing the conceptual schema to stay unchanged. The modifications are kept below the

conceptual level, which is vital to maintaining physical data independence. The external/conceptual mapping communicates between the external and conceptual levels. Through this mapping, fields and record titles can be changed and many conceptual fields can be aggregated into a sole external view. It also provides many external views that can be present at the same time to any number of users. Changes in the conceptual level are reflected in this mapping, with the external schema unmodified, achieving logical data independence (Date 2000).

3.2.2 Components of Database Systems

There are four essential components of database system, which are briefly discussed below.

Data: The data in database system is considered to be both integrated and shared, offering a huge advantage in large environments where multiple users are utilizing the data. Integrated refers to the fact that the database is an amalgamation of numerous discrete files, and redundancy between these files is somewhat removed. Shared means data in a database is shared among various users who have access to identical parts of the dataset but are utilizing it for different purposes.

Hardware: This component of the system consists of secondary data storage mainly in magnetic disks that are employed to hold the stored data in association with Input/Output devices, device controllers, Input/Output channels, etc. The hardware processor and connected main memory are employed to maintain the implementation of the database system software.

Software: The Database management system (DBMS) is a layer of software that lies between physical data storage and the users of the system. It may also be referred to as a data manager or database server. The DBMS manages all interactions with the database, such as adding, removing, or editing records and tables, and retrieving or updating data. This protects users from the specific physical details that these processes involve.

Users: There are three wide categories of users. The first is programmers who are responsible for database application programming using a variety of programming languages like VB, SQL, C++, and Java, etc. The second is end users who interact with the system from online work stations, and lastly there are database administrators who are responsible for the entire database system at a technical level.

3.2.3 Database Models

3.2.3.1 Relational Database Model

A relational database is a database that is apparent to the user in the form of a collection of relation variables known as relvars or tables. A relational system is

a system that sustains relational databases and processes executed on these databases. Most databases in use today are based on the relational model of data; this is a groundwork or theory that deals with data from the following three aspects:

Structural Aspect: This means that the data is perceived by the user in the form of tables only. The essences of these tables are meant to be self-evident.

Integrity Aspect: The data available in these tables must be an accurate representation of the reality that it represents. These tables must fulfill certain integrity constraints.

Manipulative Aspects: The user can maneuver and manage these tables utilizing operators that develop tables from tables. The three most important operators are "restrict," "project," and "join." The *restrict* task selects particular rows from within a table. The *project* task mines particular columns from the table and the *join* task links two tables with each other according to shared values in a common table.

A relational database has five components:

1. An unrestricted assortment of scalar or discrete, single *types*. (Types are clusters of objects that we can talk about.)
2. A *relation type generator* and a proposed explanation for such generated relation types. (Relations are sets of things we say about *types*.)
3. Provisions for describing *relation variables* of above-mentioned generated relation types. (Relation variables play the important role of representing the persistent database values.)
4. A relational assignment function for conveying relation values to such *relational variables*.
5. An unlimited collection of standard *relational operators* for obtaining relation values from other relation values (Date and Warden 1990).

For the management of a relational database system it is imperative to have a detailed, comprehensive, and efficient *catalog* or *dictionary* function. This catalog maintains in depth and meticulous *descriptor information* also known as *metadata* (discussed elsewhere in greater detail) concerning a range of items, relvars, indexes, users, integrity constraints, and security constraints and so on. Metadata is absolutely essential to the proper functioning of the system (Date and Warden 1990). In most relational DBMS's this metadata is stored internally in system tables.

Database operations support logical units of work known as "Transactions." In relational databases, these transactions follow a principle that they are implemented completely as a whole or they do not execute at all. These transactions are also durable, meaning that once successfully executed they will definitely be applied to the system even if the system fails at any point after their implementation. These transactions are also independent of each other. Once successfully implemented synchronized transactions become serialized, which means that they are guaranteed to produce the same result even if reexecuted in an unstipulated order (Date and Warden 1990).

One of the main advantages of a relational database is the property known as *automatic navigation*. The function of navigating around the stored data in response to a request by the user is performed automatically by the system. This is in contrast

to nonrelational systems where navigation is the user's responsibility. The relational database is also very beneficial with regards to maintenance of data integrity and support of ad hoc queries (Date and Warden 1990; Date 2000).

3.2.3.2 Object Database Model

Object database systems have originated from attempts to apply concepts from object-oriented programming languages, like C++ and Smalltalk, to databases; this is in contrast to relational databases that are based on relational algebra. Interest in object-oriented databases has been increasing over the past few years although in the fields of cancer research most databases are predominantly relational in nature. The main theory behind their development is to assist the user so that they do not have to deal with machine-oriented assemblies such as bits and bytes but rather work with objects and tasks and operations involving these objects that are more familiar on a human level and more similar to their equivalents in reality (Cox 1986).

These systems have proven to be effective for application programs, where they are designed with the purpose of performing a specific task. In this field they are rapidly gaining popularity. However, databases may be called upon to perform tasks that are not anticipated at the time of development, in this case object-oriented databases may simplify some tasks but may also make other functions difficult to fulfill. Object-based databases require entrenching a large degree of "intelligence" into the database and this also makes it difficult to ensure data integrity and security. Early object-based databases did not support ad hoc query functions. The newer systems have included this feature but have also placed limitations on the type and nature of query performed thereby reducing their usefulness (Date 2000).

3.2.3.3 Object/Relational Database Model

A recent development is the production of Object/Relational databases also known as "Universal Servers." Their development is strongly based on the foundation of the relational model with incorporation of the beneficial features of object databases. Such systems are in actuality just relational systems that support the relational domain concept (i.e., types), which allows users to characterize their own types (Date 2000).

3.2.4 Database Types

In this section, we will describe various specialized types of database implementations relevant to cancer research.

3.2.4.1 Distributed Databases

A distributed database is a sole application that works transparently on data that is dispersed across a number of databases managed by different database management systems operating on separate machines at sites connected by a communicating network. Each site has its own autonomy; however, with mutual agreement users at any site can access data that reside anywhere on the network in the same manner that they access data stored locally. In other words, a distributed database is a virtual database whose data is stored in different databases located at physically different sites. In addition, they can also access the data at their own site exactly as if the user database did not participate in the distributed system. The distributed database functions as a kind of affiliation among the individual DBMS at the local sites. Historically speaking, distributed databases have been employed in multiple sites in the same building through Local Area Network (LAN). With the rapid development of national and international collaboration in cancer research, multiple sites in geographically distant locations are participating in distributed database utilization with the implementation of Wide Area Networks (WANs).

In this day and age, deployment of distributed databases is obviously desirable. Corporations, institutes, and research collaborations are distributed not only logically into departments and divisions, but also on a geographical scale into multiple sites. Therefore, this distributed layout of a database facilitates not only fast and easy accessibility but also updating and processing of data. The disadvantage to this system is that WANs are usually slow, putting a limit on the number of and volume of messages deployed through the system, and thereby minimizing its utilization. Other issues of concern are quality assurance of data replication and update processes (Date 2000).

The data is protected during transmission through a process that stores and transfers sensitive information in an encrypted format. Plaintext (original data) is encrypted through an algorithm using an encryption key, which converts the original data into ciphertext. The encryption key is kept secret and encryption algorithm is made public. The ciphertext is not understandable to anyone who does not possess the encryption key. During the encryption process plaintext is formatted into a string, after which the encryption key is applied. The plaintext is broken down into blocks of the same length as the encryption key string. The characters of plaintext and encryption key are replaced by integers, and then the integers of both the plaintext and encryption key are added together to obtain a sum. Each integer in the sum is further replaced by its character equivalent, completing the encryption process. The decryption is then fairly simple for any authorized person holding the encryption key. This is a substitution type encryption process, which can be made further secure by incorporating a process of permutation in which the original plaintext characters are rearranged in a different sequence. Such an Encryption algorithm is the basis for the Data Encryption Standard that was originally developed by IBM and adopted as the United States Federal Standard in 1977 (1997 January 15; Date 2000).

A relatively newer concept is Public Key encryption that is deemed to be more secure in the face of very fast highly parallel processors. It is based on the presence of two distinct encryption and decryption keys with corresponding signatures that cannot be forged (Date 2000).

3.2.4.2 Temporal Databases

A temporal database is a database that contains historical data as well as current data. In some of these databases data is only inserted, it is not deleted or updated; in such a case the database will sustain historic data only. The counterpart of such a historic database is a snapshot database that houses only current data that is updated or deleted as facts change. The information in temporal databases is an encoded version of time stamped facts; hence all of the data is temporal (Date 2000).

3.2.4.3 Statistical Databases

A statistical database is a database that supports statistical queries, such as those that obtain cumulative information like sums, averages, percentages etc. The issues that such databases face are those of security. The risk is of *deduction of confidential information by inference*, this means that sensitive data may be reconstructed by administering a sufficiently large number of queries (Date and Warden 1990).

3.3 Data Warehousing

W. H. Inmon defined a data warehouse as "*A subject oriented, time variant, non-volatile collection of data in support of management decisions*" (Inmon 1992). This has also been described more broadly as joining two or more software tools to pull out derived datasets from any data structure.

Essentially, a data warehouse is *a database developed for data analysis from multiple data source applications to facilitate a large number of users with temporally extended interactions*. It maintains recent and archived data to offer a past perspective of collected information.

Data warehouses offer a method of isolating operational vs. informational data processing. One rationale for this is that operational systems are optimized for recording data and preserving its integrity: only a small amount of data is affected with each transaction. A data warehouse, on the other hand, is optimized for speed of data retrieval from the application perspective and ease of designing queries from the user perspective. Separating the two allows the efficient retrieval of data without slowing down operational systems. Data collected from the operational environment resides in the warehouse as a unilateral flow from the source database

(see Chaps. 2 and 10–14 for examples of source systems). The warehouse processes the raw data from the operational database and provides procedures that collect, reconcile, and summarize data to make it applicable and useful for end users. It holds subsets of the entire data that an organization possesses, providing a common model for data originating from a variety of primary sources such as operational databases. Data warehouses function in a multidimensional integrated manner, therefore reducing errors in data structure, semantics, and utility across multiple operational databases. They can collect vast amounts of information from operational databases and provide an efficient way to navigate across it. The collected data is mapped to the appropriate application within the warehouse and is used for decision-making processes. In addition, they facilitate automated data mining, data refreshing, data consolidation, and replication of data for remote sites. They can also sustain the data replication process to make certain that remote sites are coordinated with the events at the central sites (Berson and Smith 1997).

3.3.1 Data Warehouse Architecture

Data warehouse architecture is designed on a relational database management system (RDMS) that operates as a midlevel source for informational data. The architecture is designed to be completely separate from operational data and processing. Data flows into the data warehouse from operational applications and is converted into a common structure and layout. The data transfer process involves data conversion, summarization, condensation, and filtering. A data warehouse is expected to be able to accommodate and manage huge volumes of archived data in addition to tracking and recording changes in a database over a period of time. The different components of a data warehouse are described in Fig. 3.2.

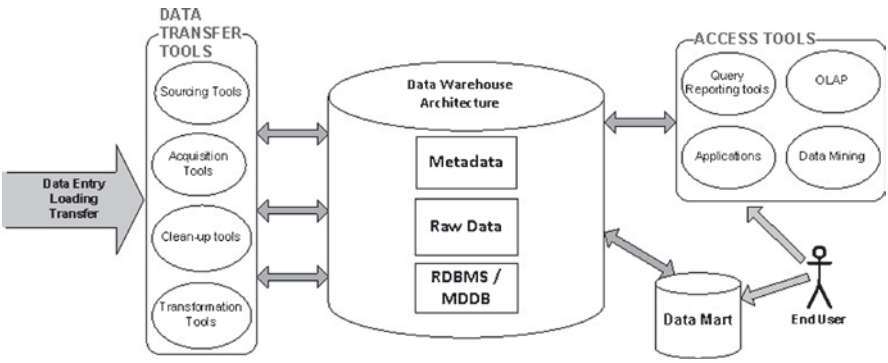


Fig. 3.2 Presents a data warehouse system that comprises of data transfer tools, user access tools and central data warehouse architecture component

3.3.1.1 Software Tools for Data Sourcing, Acquisition, Clean-up, and Transformation

An important step in data warehouse development is the transfer of data from operational systems and converting it into the appropriate structure for informational applications. This job is carried out by the data sourcing, acquisition, clean-up, and transformation tools, also known as extract/transform/load (ETL) tools, that eliminate redundant data, translate data into common data names/definitions, compute summaries, set up defaults for missing data, house alterations to source data definitions, and convert discrete data into a suitable format that can be incorporated into decision support tools. These tools can conserve a substantial amount of time and effort. The major vendors in this field are Prism Solution, Evolutionary Technology Incorporation, Vality, Praxis, and Carleton. There are also open source tools available, such as Kettle, Octopus, and Clover ETL. Finally, although a prepackaged solution is best if it can be adapted to the specific situation, most of these assume a certain degree of integrity in the source data that is often not present. If the source data is too disorganized, it may be easier to write a program for this task (Berson and Smith 1997).

3.3.1.2 Metadata

Metadata can be defined as *data about data*. In a data warehouse, it is required for managing, maintaining, building, and using the system and can be categorized into technical and business metadata. The technical metadata documents hold information about data sources, transformation descriptions, data definitions, data quality assurance, data mapping operation, etc., and this is utilized by the developers and administrators to perform warehouse development and management tasks. The business metadata documents help users to understand data stored in the warehouse. They describe subject areas and information object types (queries, report, images, etc.), as well as warehouse operational information, Internet Web sites, and information on handling warehouse components (subscription, scheduling, business query objects, etc.).

3.3.1.3 Data Warehouse Technology

The central database in a data warehouse is traditionally developed using a RDMS. This is limited by the lack of key data warehouse features such as ad hoc query processing and requirement for development of flexible user views, as well as being less suited to very large databases. There are a variety of technological measures that address these issues, such as parallel relational database design, which requires a parallel computing platform utilizing symmetric multiprocessors (SMPs), massively paralleling processors (MPPs), or groups of uni-processor or multiprocessor technologies.

Another measure is the development of multidimensional databases (MDDBs), which remove many of the restrictions set on data warehouses by the relational data model. These are combined with online analytical processing (OLAP) tools that are classified as data query, reporting, analysis, and mining tools and architecturally fit into "data mart" component of data warehouse (Berson and Smith 1997).

3.3.1.4 Front End Tools (Access Tools)

The fundamental aim of the data warehouse is to provide information for strategic decision making to an end user who interacts with the warehouse through access tools. This highly efficient system supports analyses that can be predefined or dynamic. Joins, précis, and regular updates are examples of predefined outcomes, which are made readily available to the end user. The main analyses performed in a warehouse are carried out by the ad hoc query, regular reports, and custom applications. The majority of development efforts in data warehouses are in the area of exception reporting also known as "alerts" that inform the user when a particular event occurs in the warehouse. This functionality provides a great advantage to the user when it is harmonized with business objectives. The access tools use metadata definitions to gain access to captured data in the data warehouse; some of them use additional or mediator data stores, for example MDDBs. These supplementary data stores either act as specialized data stores for a particular end user tool or a subset of the data warehouse encompassing a particular subject area, such as a data mart. These access tools can be categorized as follows: Data query and reporting, Application development, Executive information system, Online analytical processing, and Data mining.

Query and Reporting Tools: These tools can be broadly classified into two types: Reporting tools and Managed query tools. Reporting tools allow the organization to conduct high-volume group jobs including calculations and printing paperwork, etc., facilitating the production of day-to-day task reports. A specific type of reporting tool known as a "report writer" is a simpler desktop tool for end users. Managed query tools act as a user friendly interface between the intricacies of the SQL software and the users. The managed query tools support point and click functionality, generate query results, and facilitate user navigation.

OLAP: Online analytical tools utilize the multidimensional data model to quickly answer analytical queries. OLAP tools can support the sophisticated analysis of data and provide detailed, multidimensional, and compound views.

Applications: Applications are additional tools that support the integral query and reporting tools to sustain increasing user requirements. These applications can be developed internally using graphical data access milieu. These application development platforms incorporate well with OLAP tools and can be integrated with major database systems.

Data Mining: Data mining can be defined as the process of utilizing artificial intelligence, statistical and mathematical methods to discern meaningful new

associations, patterns, and trends by exploring and analyzing large amounts of data stored in warehouses. The main benefit of data mining is the capacity to analyze on a predictive level rather than just retrospective analysis. Data mining allows the resource and its users to discover sequestered patterns, associations, and interactions from the stored data. Data mining also allows the data to be visualized and displayed in different and comprehensive manners, making very large amounts of data easily understandable. Previously overlooked errors and discrepancies in the data are also brought to notice and easily corrected during the data mining process (Berson and Smith 1997).

3.3.1.5 Data Marts

A data mart implies different things to different people. Most commonly, it is a data store that is supplementary to a data warehouse. It is also presented as a low-cost substitute for data warehouse, or a way to start a data warehousing project that can deliver results relatively quickly. In this scenario, it is very important to keep in mind the overall design and requirements for metadata consistency in the final data warehouse system. Otherwise, as additional data marts are added, it is very easy to end up with an unintegratable series of data silos that replicate various processes and contain redundant information. A data mart presents a subset of data that is designed to answer specific questions for particular users and might be a set of denormalized, summarized, or cumulative data. In most instances, the physical data is stored in a separate location, although occasionally it may reside on the data warehouse server (Berson and Smith 1997).

3.3.1.6 Administration and Management

Data warehouses can be up to several fold larger than linked operational databases depending on the amount of archived information that is stored. The data warehouse is not synchronized in real time with the operational database, so the data must be updated periodically according to the requirements of the application.

Most products use gateways to transparently access multiple data sources to obviate the need for redundant applications to interpret and utilize the data. In a heterogeneous data warehouse environment, there is an additional requirement for networking technologies to address the various databases in place on disparate systems.
In summary, the following efforts are required to manage a data warehouse:

- Data security and priority management
- Monitoring data/metadata updates and quality checks from different data sources
- Auditing and reporting on data warehouse utilization and status
- Reproducing, removing, dividing, distributing, and purging data
- Storage management and back-up and recovery protocols

3.3.1.7 Information Delivery Component

This allows the process of subscribing to data warehouse information and delivering to more than one user based on a user-specified scheduling algorithm. In other words, this system exports data and information objects from one data warehouse to other data warehouses as well as local databases and applications such as excel spreadsheets. The logic behind the data warehouse information delivery system is that once the data warehouse is established and functional, the end user should be able to generate reports and see analytic views of data at particular times based on significant events without knowing the location and maintenance issues of the data warehouse. The development of Internet/intranet and the World Wide Web delivery system has enabled vast number of users to browse data warehouse information (Berson and Smith 1997).

3.4 Existing Systems

The increasing demand of cross-institutional translational research and advancement in the development of tissue banking informatics tools has increased the need of the research community for high quality and well-annotated biospecimens. To fulfill this need the Department of Biomedical Informatics (DBMI) at University of Pittsburgh (http://www.dbmi.pitt.edu/index.cfm) has established and integrated various organ specific and federated tissue banking query tools. These systems are constructed on an underlying architecture of common data elements (CDEs) for characterization of tissue samples and clinical follow-up data, supported by an essential quality assurance process. In addition to the development and implementation of tissue banking databases, various Web-based query tools have been designed to help investigators query the annotated biospecimens (Table 3.1).

3.4.1 Tissue Banking Informatics Models

3.4.1.1 Organ Specific Database

The informatics model used in the Cooperative Prostate Cancer Tissue Resource (http://www.cpctr.info), Pennsylvania Cancer Alliance for Bioinformatics Consortium (http://www.pcabc.upmc.edu/main.cfm), Early Detection Research Network (EDRN), Colon and Pancreatic Neoplasm Virtual Biorepository (http://www.cancer.gov/prevention/cbrg/edrn), Shared Pathology Informatics Network (SPIN), and Specialized Program of Research Excellence (SPORE) Head and Neck Neoplasm Virtual Biorepository (http://spores.nci.nih.gov/) is a relational database built on multitiered Oracle Database Server 10.2.0.2 and Oracle Application Server 10.1.0.2, with both the database server and the application server configured as virtual hosts on IBM Power6 Series 570 hardware. The hardware and system software are centrally supported for backup and routine maintenance.

Table 3.1 Presents a comparison among different Web-based tissue banking initiatives developed at department of biomedical informatics, University of Pittsburgh

				User Web interfaces			
Human tissue repositories	Funding organizations	Information model	Data accessibility	Public statistical query interface	User clinical database interface	Data entry interface plus electronic data import/export	Biospecimen types available to research community
NMVB	CDC/NIOSH	Federated	U. Pitt, collaborators and IRB/ SRCB approved investigator	Yes	Yes	Both	Mesothelioma (paraffin, fresh frozen, blood product, and TMA)
PCABC	Pennsylvania department of health	OSD	U. Pitt, collaborators and IRB/ SRCB approved investigator	Yes	Yes	Both	Breast cancer, prostate cancer, melanoma (paraffin, fresh frozen, Blood products)
EDRN, Colorectal and Pancreatic Neoplasm Virtual Biorepository	NCI	OSD	U. Pitt EDRN researchers only	No	Yes	Both	Colon and Pancreatic Cancer (paraffin, fresh frozen, blood products)
SPORE Head and Neck Neoplasm Virtual Biorepository	NCI	OSD	U. Pitt SPORE researchers only	No	Yes	Both	Head and neck cancers (paraffin, fresh frozen, blood products)
CPCTR	NCI	OSD	U. Pitt, collaborators and IRB/ SRCB approved investigator	Yes	Yes	Electronic data import/export only	Prostate cancer (paraffin, fresh frozen, blood product) Six different TMAs

NMVB National Mesothelioma Virtual Bank (http://mesotissue.org/), *PCABC* Pennsylvania Cancer Alliance Bioinformatics Consortium (http://pcabc.upmc.edu/main.cfm), *CPCTR* Cooperative Prostate Cancer Tissue Resource (http://www.cpctr.info/), *EDRN* Early Detection Research Network (http://edrn.nci.nih.gov/), *SPORE* Specialized Program of Research Excellence (http://spores.nci.nih.gov/current/hn/index.htm), *CDC* Center for Disease Control and Prevention,

The Organ Specific Database (OSD) model consists of the following integrated application layers that maintain data query/entry, data deidentification, and the user management module (Melamed et al. 2004; Patel et al. 2006, 2007a) (Fig. 3.3).

Presentation Layer: This contains the following components: *metadata curation* is used by data administrators and data element curators for registering new data elements or editing definitions of existing data elements. The *administrator security system* is used by the application administrators to grant, revoke, or limit privileges to new and existing users. *Manual annotation* is used by honest brokers or domain experts for collecting information regarding patients registered for the study. *Data query* is used by the honest brokers and research community to run criteria-based queries. The query results show identified and deidentified outputs depending on the individual roles and privileges granted by application administrators. This tool provides two levels of access to researchers. The first level of access is the honest broker view of the consented patients for their own study and second is a deidentified view on all the patients for other studies for which they do not have access but want to study and analyze overall trends. The *data import/export* component provides users an option to electronically import preformatted data from existing systems or export data for analysis on their own computers.

Metadata Engine: The Metadata Engine is based on the development of *Common Data Elements* that are used to hold application data structure for data elements/fields as defined by the research project working group. The *HELP builder* is used for each data element/field with its detailed definition of business rules and usage. The *business rules engine* constitutes business rules for how multiple elements can be combined with simple numerical and algorithmic techniques to report complex values for decision support and statistical time sensitive outputs. The *mapping engine* maps logical and physical layers of design that facilitate data retrieval and storage.

Security Engine: The security engine secures the application at three levels: the first is *registration* of new user accounts and requesting application roles. Second is *authentication* by adding/editing user information, and lastly, *authorization* is granting or revoking user roles and privileges.

Physical Data: The physical database tables are presented in the data warehouse in a three-step fashion. First is the *application database* that holds case data contents in a metadata coded format. Second is the *metadata database*, which holds metadata definitions and descriptions for all the attributes and values in the system. The third one is a *security database* that drives the security and authorizations definitions and assignments.

The data administrator can provide a user name and password for approved researchers or nursing coordinators to access the tool. The query tool access to the central database is through a highly structured “click and point” interface that allows queries on approved data elements and also is based on the researcher’s IRB

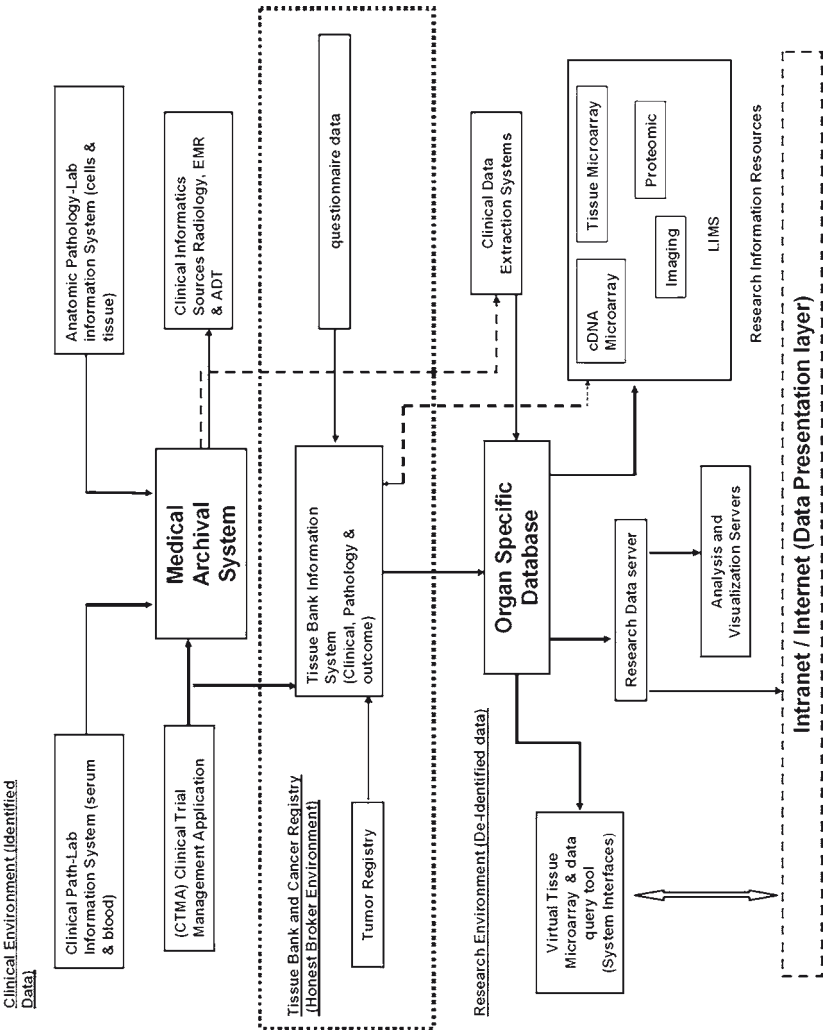


Fig. 3.3 Presents the flow of multimodal datasets from the different clinical and research data sources into database and structure of organ specific database

approval. The specificity of the data returned depends on the user's profile. There are three user profiles:

Approved Investigator Query Tool is a password-protected application, which is distributed to those research investigators who have approved research protocols. It allows users to refine and compile case lists for their application and also to mine and modify the datasets on the cases where they have received biospecimens. The query tool provides search capability on all the annotated data associated with each subject through multiple predefined standard views of the dataset and also allows users to customize their own views and save them under their account.

Data Administrator Query Tool is a password-protected tool available only for the internal data administrators. It is meant to be used by data administrators to address quality assurance issues regarding the data. The main difference between this and the approved investigator tool is that this tool allows the user to search by "all Subjects" or "limited Subjects" based on consent for a particular study.

Public Query Tool is available to the general public and accessible through the main home page. The output display of a public query will be the accrued number of cases, specimens in the database that meet the criteria of the query and general statistics on a limited number of data elements. It is designed to allow interested investigators to see if the resource will be applicable to their research needs (Melamed et al. 2004; Patel et al. 2006).

3.4.1.2 Federated Model Database

The National Cancer Institute (NCI) (http://www.cancer.gov/) launched the caBIG® (Cancer Bioinformatics Grid) (https://cabig.nci.nih.gov/) initiative in February 2004, under the leadership of the NCI Center for Bioinformatics, with the goal to create a "World Wide Web" of cancer research (see Chap. 9). Systems developed for caBIG® are interoperable both in the methods by which data is transmitted (syntactic interoperability) and in the meaning of the data itself (semantic interoperability). caBIG® is organizing itself to engage a broad spectrum of the oncology research community to ensure that its products are widely adopted.

The functionality and implementation of the Tissue Banking and Pathology Tools (TBPT) Workspace will be discussed in the following subsections. The goal of the TBPT suite (https://cabig.nci.nih.gov/workspaces/TBPT) is to standardize biospecimen-associated information by utilizing novel pathology informatics tools, so that individual collections of biospecimens can be shared across the cancer centers. The ultimate strategic vision is to enable automated systems interfaced to disparate clinical cancer informatics environments to communicate with each other. A successful outcome will result in a single point of entrance to federated tissue banks and pathology systems across the institutions, allowing for a more effective mechanism for researchers to locate and analyze biospecimens for use in cancer research based on anatomic pathology, laboratory medicine, patient data, and experimental results. The Tissue Banks and Pathology Tools Workspace is focused on the development of a well-designed and adequately formatted set of caBIG®-compliant

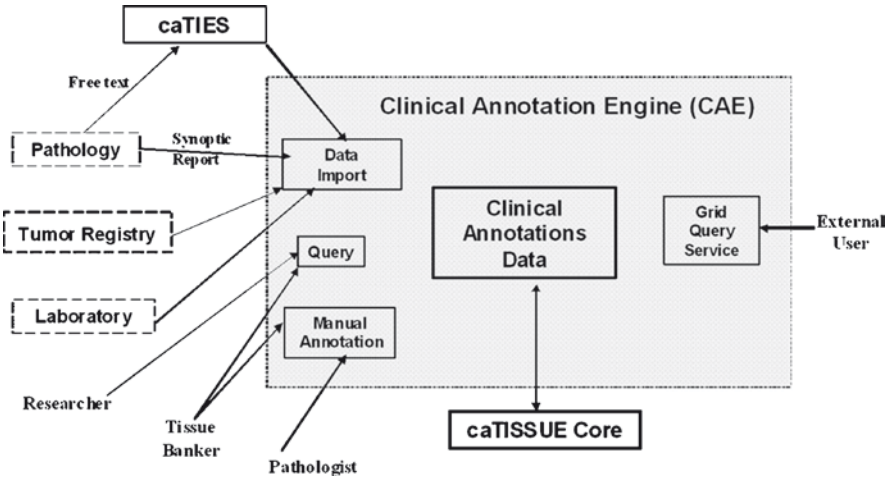


Fig. 3.4 Presents data import, integration, and query interface in a Web-based environment of federated model architecture of caBIG tools

tissue banking tools (https://cabig.nci.nih.gov/workspaces/TBPT/TBPT_Software/). Three primary components that significantly strengthen translational research on the caBIG® are caTISSUE Core (http://ncicb.nci.nih.gov/NCICB/infrastructure), Cancer Text Information Extraction System (caTIES) (http://caties.cabig.upmc.edu), and Cancer Tissue Clinical Annotation Engine caTISSUE CAE (https://cabig-stage.nci.nih.gov/tools/cae) (Fig. 3.4).

caTISSUE CAE

Informatics Architecture

The various components of CAE include Web tier, business tier, data tier, and integration tier. The Web tier serves up static HTML, images, style sheets, and dynamically generated Web pages via a standard JSP/Servlet engine. Dynamic requests will be managed through a Model-View-Controller (MVC) mechanism. This mechanism manages the processing of individual requests and the flow between requests. The Spring Framework is being evaluated for this purpose. The controller object makes requests to the business tier that results in the return of model objects that represent the information that must be presented back to the user. Based upon the result of the controller actions, the model objects are forwarded to an appropriate view (JSP) that renders them into a displayable page.

The business tier consists of a set of functional components, an Object-Relational (O/R) mapping mechanism, a metadata interrogation mechanism, a caCORE-compliant Application Programming Interface (API) (http://cabio.nci.nih.gov/NCICB/infrastructure/cacore_overview) and a set of shared services. These components act together to implement the principal functionality of the system. The functional components consist of a series of service objects that provide a consistent interface

to Web tier controllers. There are three primary functions such as "Query," "Manual Annotation," and "Import Management." The object layer of the business tier consists of services required for managing domain objects. The principal functions of this layer are to provide O/R mapping capabilities via caCORE Software Development Kit (SDK) generated and custom code (http://ncicb.nci.nih.gov/NCICB/infrastructure/cacoresdk). The resulting objects present a unified model-based interface of the domain to the functional components so that they need not be concerned with the physical database implementation. In addition to the O/R mapping capabilities, the object layer also provides an API into the domain objects as required by the caBIG® silver-level compliance specification. This API is generated using the caCORE SDK. Metadata interrogation capabilities can also be accessed from the object layer. The required metadata are available from the Cancer Data Standard Repository (caDSR). However, there may be some additional metadata necessary for the rendering of user interface components that is application-specific. This tier also provides shared services for logging and audit capabilities, authentication, and authorization services via the NCI Common Security Module (CSM) and caDSR access via the NCI Clinical Infrastructure Application Framework (CIAF) module. The services will be implemented generically so as to potentially be reusable by other caBIG® components (Piwowar et al. 2008; Niland et al. 2007).

The data tier consists of a domain database that houses the clinical annotations data, the security data (users, groups, roles, protected elements, etc.), and a staging area for import data. The database is in Oracle with a MySQL implementation.

The integration tier combines other systems with caTISSUE CAE in one of the following two ways:

Data Import: Anatomic Pathology Lab Information Systems, Cancer Registries, and caTIES that hold tissue-related data can add cases or annotation data to the system by exporting their data into a published, XML-based format. The data can then be imported into the annotations database using the Web-based import management capabilities provided by the CAE system.

Application Programming Interface (API): The CAE system also provides a caBIG®-compliant API for accessing domain-object data, which can be utilized by caTISSUE Core and other future caBIG®-compliant systems to directly access and perform searches on annotations data.

Utility

caTISSUE CAE is an efficient data annotation system devised to address the following objectives:

- Manual annotation of biospecimens with pathology, tumor marker, staging, grading, and clinical outcome data using a Web-based user interface.
- HIPAA compliant de-identification, secure sharing, and high quality annotation of specimens to facilitate researcher query of collected materials for translational research.
- Data managers and other knowledgeable users can map their local data model by a user interface to the caBIG® standard metadata, represented in the CAE as CDEs.

- An administrator's interface to support the import of structured data from clinical information systems such as AP-LIS, CP-LIS, and other specimen registries, which may be employed in the life cycle of clinical trials or clinical research.
- Integration of annotations from multiple sources within the translational research center. As data from multiple clinical systems and from manual annotation is added to CAE, the application will record the source of each data value. Using algorithms tested during system development, CAE will identify data that originates in different systems but pertains to the same patient, so that users can retrieve a more complete picture of a patient's disease.
- The clinical annotations may be attached either to a participant/patient, a pathology accession, or to a specimen (part) or subspecimen (block) in the latest version of CAE. Altogether these entities form a hierarchy or backbone that encapsulates the entire datasets of annotation for a case. Annotations can be entered manually using the provided user interface or imported electronically.

National Mesothelioma Virtual Bank: The NMVB Web-based query tool (www.mesotissue.org) is based on the caTISSUE CAE application. The NMVB database allows researches to search clinically annotated Mesothelioma biospecimens via a Web interface in real time. The database is made available through a publicly available Web site. The database facilitates standardized clinical annotation structure and incorporates a variety of datasets from different data sources (Amin et al. 2008).

caTIES

The caTIES project deals with the information extraction from free text and tissue accessions in biospecimen resources. It is a general purpose text information extraction tool to automate the process of converting free text surgical pathology reports (SPRs) into structured data, storing those data in a federated capacity, and to facilitate retrieval, advanced query, and further analysis of the pathology information. The data extracted from these reports can directly populate caBIG® data structures. It is an extension of the SPINties application, developed through the SPIN. caTIES builds on the SPINties foundation by expanding the pathology vocabularies drawn from Enterprise Vocabulary Services (EVS) instead of the Unified Medical Language System (UMLS) (Drake et al. 2007; Patel et al. 2007b). Additionally, caTIES structures data based on ISO 11179 compliant CDEs, which can be accessed from NCI's cancer Data Standards Repository (caDSR) (Tobias et al. 2006). The information models for prostate, breast, and melanoma are available through the caBIG® Web site.

Informatics Architecture

Within the single logical data model, caTIES houses all its data using three primary data stores. The private data store maintains the original data derived from the clinical systems, such as the AP-LIS, containing identified free text, dates, patient medical

record numbers, and specimen accession numbers. The private data store is only available for access by Honest Brokers within the hosting institution. Typically, the research data store resides on a separate machine and houses the deidentified reports along with unrestricted information (gender, age if less than 90), which are targeted by the Natural Language Processing (NLP) Pipeline Service to create and store conceptual annotations. The Collaborative Tissue Resource Manager (CTRM) is the third data store, which manages the collaborative construction and manipulation of tissue studies so that researchers can build tissue order sets by electronically interacting with honest brokers at their respective organizations.

The data preparation phase functions as a succession of operating system-based services that convert data from SPRs stored in an AP-LIS setting to concept-annotated deidentified documents stored in the research data store. The corresponding services perform in the following order: acquisition, deidentification, concept-coding, and indexing. The initial transfer of data from AP-LIS to the private data store is executed through an assortment of acquisition services, which may consist of existing tools, provided by vendors, or internally developed transfer mechanisms targeting the caTIES logical schema. For compliance with HIPAA regulations, the deidentification service removes the required 18 identifiers, replacing dates with an offset to preserve temporal relationships and names with symbols consistently throughout the document to preserve nominal relationships. The caTIES coding pipeline service constructs conceptual annotations on free text documents using a sequence of modular processing resources preconfigured with the NCI MetaThesaurus whose use is a condition of participation in caBIG®. For fast entry to the documents based on the characteristics of the text and conceptual codes, the caTIES indexing service makes use of a text search engine library.

The NCI MetaThesaurus is the central reference terminology within the NCI EVS integrated suite of resources and services designed to meet the controlled terminology needs of the NCI and its partners, as well as within the caBIG®/caCORE bioinformatics architecture (Fragoso et al. 2004; Sioutos et al. 2007). It possesses a key role in the design of the project along with its integration with other resources as part of the caGrid architecture (see Chap. 4). Using description logic to enforce logical consistency, this concept-based terminology system provides a formal model with computationally tractable semantics (Hartel et al. 2005). Each concept represents a single specific meaning, and includes multiple terms, codes, text definitions, and other properties that reflect that meaning. Also, a concept can be defined by formal description criteria that logically make it a subtype of its parent concept(s), and distinguish it from sibling concepts with the same parents. Concepts are arranged in disjointed subsumption hierarchies under 18 root nodes, such as *Activity* and *Gene*, with each step down from parent to child concept representing some added specialization of meaning.

Three Role-Based Perspectives Comprise the caTIES User Interface (UI): Researcher, honest broker, and administrator. If a user is registered with more than one role within the system, he or she can switch between views. The researcher perspective supports query construction and execution, and order management for the distribution protocol. The user can query in caTIES by text (entering text strings

which the system will search for within the documents), or by concept (entering strings that are mapped to candidate vocabulary concepts). Temporal searches allow researchers to seek specific characteristics in a given patient's diagnosis timeline based upon the timing of diagnostic reports. The administrator perspective is used by both honest brokers and system administrators to perform administrative functions, such as user account creation, registration of new IRB-approved studies or a new institution as a provider, or addition of honest brokers from the administrator's local organization. Administrators also contribute to quality assurance of deidentification and concept-coding. The honest broker perspective enables impartial individuals, such as cancer registrars and tissue bankers, to fulfill requests for tissue and clinical data. Brokers are only able to view information from the private data store from their own institution.

Utility

The principal objectives for caTIES include the extraction of concept-coded information from free text SPRs using controlled terminologies to populate caBIG®-compliant data structures, and the overall pioneering of research for distributed text information abstraction within the context of caBIG®. An additional goal would provide researchers with the ability to query, browse, and create orders for annotated tissue data and physical material across a network of federated sources. As a result through caTIES, the SPR acts as a locator to tissue resources.

caTISSUE Core

The caTISSUE Core suite (https://cabig.nci.nih.gov/tools/catissuecore) is developed by the Siteman Cancer Center (Washington University School of Medicine, St. Louis, MO) to manage data related to biospecimen inventory, tracking, quality assurance, basic annotation, annotation and regulatory issues. This permits users to track the collection, storage, quality assurance, and distribution of specimens as well as the derivation and aliquoting of new specimens from existing ones (e.g., for DNA analysis). It also allows users to browse and request specimens that may then be used in correlative molecular studies.

Informatics Architecture

The overall workflow of caTISSUE Core is described below:

- This is based on open source architecture, controlled vocabularies, and CDEs in keeping with caBIG® principles.
- Initial development at "Silver" level caBIG®-compliance, with eventual evolution to "Gold" level compliance concurrent with grid development.
- Object model design is derived from use cases generated from review of current specimen banking systems.
- Modular architecture to allow future developers to expand functionality without system redesign.

- Rapid deployment and testing at several institutions with significant needs for enhanced biospecimen informatics.
- Contingencies for mapping objects and data elements from existing legacy systems.

Utility

caTISSUE Core is a foundation data system for biospecimen inventory, tracking, and basic annotation that may be used by biospecimen resource facilities, regardless of the nature of biospecimen transactions that occur or the type of biospecimens involved in the transaction. Key features of the caTISSUE Core system include:

- caTISSUE Core captures data related to clinical studies collecting biospecimens, research studies that utilize biospecimens, locations for biospecimen collection, storage and utilization, and system user information.
- Participant, Accession, Specimen, Segment, and Sample objects and their corresponding basic attributes are represented.
- Tracking of individual biospecimens from accession to storage with a flexible inventory configuration are incorporated into the system.
- Both simplified and complex queries for biospecimens based on any attribute are possible using an intuitive interface.
- Semiautomated requests for distribution based on queries.
- Every user action is tracked by the system and information stored for future audit.

3.5 Issues and Approaches

3.5.1 Database

Data Sharing: Data can be shared by multiple applications accessing the same database. In addition, new applications can be developed to function parallel to these, utilizing existing data. In many cases, the database can fulfill the data requirements of new applications without importing new data.

Redundancy Control: In a nondatabase environment, each application carries its own data files: this leads to data redundancy and wastage of storage space. In databases, different files can be integrated and issues of redundancy can be controlled by the data administrator evaluating the data requirements of various applications. Redundancy can also be controlled if the files need to be maintained in separate applications. This can be achieved by ensuring that when one file is updated the other identical file is updated at the same time; this is the concept of *propagating updates*.

Consistency Control: The presence of inconsistencies in a database is to some extent the direct result of the presence of redundant information. If one data item is updated

and the other identical item existing in a separate file or application is not updated at the same time, two conflicting results to the same query will be generated regarding that data item. This means that the database has lost its consistency. This issue can be overcome by controlling redundancy and ensuring propagating updates.

Transaction Support: It is imperative to ensure secure and complete transactions because usually one transaction is in actuality a series of more than one transaction which need to be processed in their entirety to be successful. Databases possess the quality of transaction atomicity, which guarantees that either all components of a transaction are executed or none of them are, even in the face of system failure during the process.

Data Integrity: It is necessary to ensure that the data in the database is exact and valid. Centralized control of a database can facilitate maintenance of data integrity by establishing integrity constraints to be applied whenever data is uploaded, transferred, or updated. Data integrity is vital to database systems because data is shared. If one user enters inaccurate data into the system it can corrupt the queries of other users by producing bad data.

Data Security: This is a necessary component of databases and has been discussed in greater detail in Sect. 3.1.10. Security can be enforced by ensuring that all access to the data is through proper and authorized channels, and security constraints should be employed whenever there is a need to access sensitive data.

Requirement Management: It is important to cater to all the needs and requirements of different users especially because some needs may be conflicting in nature. This is balanced by offering a holistic service that is in the best interest of the organization as a whole.

Standard Preservation: In a centralized model of database system, it is important to maintain all the applicable standards for data entry, storage, transactions, utilization, and data representation; especially since more than one site is involved in data sharing and interchange. The preservation of all these standards is the responsibility of the database administrator (Date 2000).

3.5.2 Data Warehouse

Advantages and disadvantages of data warehouses as compared to relational databases are discussed below.

As previously described, operational databases are designed and optimized for recording data and preserving its integrity. Because a data warehouse is optimized to respond to analysis questions, separating the two allows users to retrieve and analyze data without slowing down operational systems.

Data warehouses provide a platform to perform complete analysis of transactions and processes and have the ability to integrate data from all over the organization. This supports decision making on the basis of information of the entire organization in contrast to crude approximations from individual data. In addition, they offer the capability to concurrently comprehend and handle both the macro

and microworkings of the resource/organization, which can easily be extended into strategic decision making, yielding very useful and significant results.

There is a high cost associated with maintaining the hardware, software, and storage capacity for data warehouses, which can become very large. However, with the rapid development in technology they are becoming more cost-effective.

User Satisfaction: It is important to adequately address the issues of the users and identify their needs, trends, and patterns of data utilization. Information regarding these requirements and trends is used to improve the data warehouse system and its utilization.

Metadata Issues: An issue pertaining to metadata is that the ability of many data extraction tools to accrue metadata is still developing; they are not very effective. This results in a necessity to develop a metadata interface for the users. This can be achieved fairly easily with some effort required for the duplication process.

Web-Enabled Information Delivery: To maximize the benefit of data warehousing it is important to enable the individuals who require the data with access to the data irrespective of their geographical location and time constraints. This is especially challenging when individuals are at distant sites from the data warehouse location. The efficiency of the data warehouse depends on the reliability of Web-enabled information delivery systems; these systems support the collective decision making and critical data analysis on a global level.

Data Integration: Data integration is an issue that arises in data mart applications. Initial data marts quickly acquire data and grow rapidly in multiple dimensions. This problem is overcome by developing an overall scalable data warehouse structural design and recognizing and employing the common dimensions. Focus should be given to system scalability, data reliability, uniformity, and ease of management (Berson and Smith 1997).

3.6 Conclusions

Databases and data warehouses are a cornerstone of modern advances in the area of translational research. They have facilitated and simplified data collection, data management, and data sharing, while removing redundant processes and ensuring consistency, transaction support, data integrity, data security, and the preservation of standards. With appropriate informatics support they allow user friendly and secure data retrieval while ensuring its confidentiality. Data can be uploaded, managed, quality assured, retrieved, and analyzed, from anywhere around the globe, while maintaining stringent security protocols. The philosophy of data sharing that is realized in these systems is essential to the growth and development of translational research: the sharing and dissemination of knowledge does not diminish its value, but creates new synergies and applications by increasing its utilization. By making it possible to provide high quality, comprehensive, and pertinent data in a secure and confidential manner, databases open up multiple avenues of data analysis and research, accelerating the rate of advances in that field.

Acknowledgement This work was supported by the following grants: Cooperative Prostate Cancer Tissue Resource – NCI U01 CA86735; Pennsylvania Cancer Alliance Bioinformatics Consortium – PA DOH – ME 01-740; Cancer Center Support Grant – NCI – P30 CA47904; Shared Pathology Informatics Network – NCI – U01 CA091338; caBIG – NCI Contracts – 94125DBS47 and 28XS210; NMVB – CDC NIOSH – 5 – U19 OH009077-02 and U24 OH009077-03; Clinical and Translational Science Award – NCRR – (UL1 RR024153-03).

References

Amin W, Parwani AV et al (2008) National Mesothelioma Virtual Bank: a standard based biospecimen and clinical data resource to enhance translational research. BMC Cancer 8:236

Association of Directors of Anatomic and Surgical Pathology (2007) Recommendations for the reporting of pleural mesothelioma. Am J Clin Pathol 127(1):15–19

Bell DE, Padula LJL (1974) Secure computer systems: mathematical foundation and models. MITRE Technical Report M74-243

Berry MJA, Linoff G (1997) Data mining techniques for marketing, QTY, and customer support. McGraw-Hill, New York

Berson A, Smith SJ (1997) Data warehousing, data mining, and OLAP. McGraw-Hill, New York

College of American Pathologists (2009) "Cancer protocols and checklists." Retrieved July 23, 2009, from http://www.cap.org/apps/cap.portal?_nfpb=true&cntvwrPtlt_actionOverride=/portlets/contentViewer/show&_windowLabel=cntvwrPtlt&cntvwrPtlt{actionForm.contentReference}=committees/cancer/cancer_protocols/protocols_index.html&_state=maximized&_pageLabel=cntvwr

Cox BJ (1986) Object-oriented programming: an evolutionary approach. Addison-Wesley, Reading, MA

Date CJ (2000) An introduction to database systems. Addison-Wesley, Reading, MA

Date CJ, Warden A (1990) Relational database writings, 1985–1989. Addison-Wesley, Reading, MA

Denning DER (1982) Cryptography and data security. Addison-Wesley, Reading, MA

Dhir R, Patel AA et al (2008) A multidisciplinary approach to honest broker services for tissue banks and clinical data: a pragmatic and practical model. Cancer 113(7):1705–1715

Drake TA, Braun J et al (2007) A system for sharing routine surgical pathology specimens across institutions: the Shared Pathology Informatics Network. Hum Pathol 38(8):1212–1225

Fragoso G, de Coronado S et al (2004) Overview and utilization of the NCI Thesaurus. Comp Funct Genomics 5(8):648–653

Greene FL, Page DL et al (2002) AJCC cancer staging manual. Springer, New York

Hartel FW, de Coronado S et al (2005) Modeling a description logic vocabulary for cancer research. J Biomed Inform 38(2):114–129

Inmon WH (1992) Building the data warehouse. QED Technical Publication Group, Boston, MA

Melamed J, Datta MW et al (2004) The cooperative prostate cancer tissue resource: a specimen and data resource for cancer researchers. Clin Cancer Res 10(14):4614–4621

Mohanty SK, Mistry AT et al (2008) The development and deployment of Common Data Elements for tissue banks for translational research in cancer – an emerging standard based approach for the Mesothelioma Virtual Tissue Bank. BMC Cancer 8:91

Niland JC, Townsend RM, Annechiarico R, Johnson K, Beck JR, Manion FJ, Hutchinson F, Robbins RJ, Chute CG, Vogel LH, Saltz JH, Watson MA, Casavant TL, Soong SJ, Bondy J, Fenstermacher DA, Becich MJ, Casagrande JT, Tuck DP (2007) The Cancer Biomedical Informatics Grid (caBIG): infrastructure and applications for a worldwide research community. Medinfo 12(Pt 1):330–333

North American Association of Central Cancer Registries. "Data Standards for Cancer Registries." Retrieved July 23, 2009, from http://www.naaccr.org/index.asp?Col_SectionKey=7&Col_ContentID=122

Patel AA, Kajdacsy-Balla A et al (2005) The development of common data elements for a multi-institute prostate cancer tissue bank: the Cooperative Prostate Cancer Tissue Resource (CPCTR) experience. BMC Cancer 5:108

Patel AA, Gilbertson JR et al (2006) An informatics model for tissue banks – lessons learned from the Cooperative Prostate Cancer Tissue Resource. BMC Cancer 6:120

Patel AA, Gilbertson JR et al (2007a) A novel cross-disciplinary multi-institute approach to translational cancer research: Lessons learned from Pennsylvania Cancer Alliance Bioinformatics Consortium (PCABC). Cancer Inform 3:255–273

Patel AA, Gupta D et al (2007b) Availability and quality of paraffin blocks identified in pathology archives: a multi-institutional study by the Shared Pathology Informatics Network (SPIN). BMC Cancer 7:37

Piwowar HA, Becich MJ et al (2008) Towards a data sharing culture: recommendations for leadership from academic health centers. PLoS Med 5(9):e183

Sioutos N, de Coronado S et al (2007) NCI Thesaurus: a semantic model integrating cancer-related clinical and molecular information. J Biomed Inform 40(1):30–43

Tobias J, Chilukuri R et al (2006) The CAP cancer protocols – a case study of caCORE based data standards implementation to integrate with the Cancer Biomedical Informatics Grid. BMC Med Inform Decis Mak 6:25

Tsatalos OG, Solomon M et al (1994) The GMAP: a versatile tool for physical data independence. University of Wisconsin-Madison, Computer Sciences Department, Madison, WI

U.S Department of Commerce/National Bureau of Standards: Data Encyption Standard. Federal Information Processing Standards Publication 46. In.; 1997 January 15

van Griethuysen JJ (1982) Concepts and terminology for the conceptual schema and the information base: International Organization for Standardization

Chapter 4
Middleware Architecture Approaches for Collaborative Cancer Research

Tahsin Kurc, Ashish Sharma, Scott Oster, Tony Pan, Shannon Hastings, Stephen Langella, David Ervin, Justin Permar, Daniel Brat, T.J. Fitzgerald, James Purdy, Walter Bosch, and Joel Saltz

Abstract As our ability to capture and generate large biomedical datasets improves, researchers increasingly need to synthesize information using a variety of data types, data systems, and analysis tools. The need for informatics support to facilitate coordinated and federated access to disparate data and analysis resources is more pronounced in collaborative basic, clinical, and translational research studies spanning multiple institutions. This chapter presents a high-level overview of several middleware architecture frameworks and technologies and discusses how these approaches can be employed to address the informatics requirements of large-scale and collaborative cancer research.

4.1 Introduction

The nature of cancer research has been transformed in the past decade thanks to the increasing availability of high-throughput and high-resolution instruments. These instruments enable investigators to capture imagery at cellular, organ, and tissue levels, measure genes and proteins expressed in cells and tissues, and map molecular interactions under a variety of experimental and disease conditions. As a result, basic, translational, and clinical cancer studies are increasingly driven by analysis and integration of information from a wide range of data sources and data types. Integrative cancer research studies, for example, investigate complex interrelationships among different biological entities and across multiple biological scales in order to understand the function and structure of biological phenomena in normal and disease states. Datasets in these studies may be captured from high-throughput molecular analyzers (such as data from microarray analysis, mass spectroscopy measurements, and measurements from real-time PCR platforms), from high-resolution

J. Saltz (✉)
Center for Comprehensive Informatics, Emory University, 36 Eagle Row, Atlanta, GA 30322, USA
e-mail: jhsaltz@emory.edu

M.F. Ochs et al. (eds.), *Biomedical Informatics for Cancer Research*,
DOI 10.1007/978-1-4419-5714-6_4, © Springer Science+Business Media, LLC 2010

imaging of tissues and organs (using high-power light and confocal microscopy scanners, high-resolution MR and PET scanners), and from phenotypic observations. Translational research studies, on the other hand, apply knowledge obtained from basic research in order to develop new approaches for diagnosing, treating, and preventing cancer as well as to create best community practices. These studies test many different kinds of hypotheses and carry out a wide range of experiments. Data types captured and referenced in translational research studies include clinical information on patients in clinical trials, outcome data, radiology and pathology reports, data from laboratory tests, molecular data, and imaging data.

There are, however, a number of factors that limit researchers from taking full advantage of these advances and integrating them in research studies optimally. A major challenge is that a researcher has to access and analyze large volumes of semantically complex datasets to synthesize biologically meaningful information. This problem is compounded by the fact that complementary datasets are often captured and managed in different systems. The process of manually browsing individual data systems and downloading the data is inefficient and labor intensive: the researcher has to figure out how to access each data source/system; he/she has to implement tools for translating the format of each data source to formats used by different analysis methods; and he/she needs to deploy systems to store and index the downloaded datasets so that they can be managed and integrated efficiently. In collaborative studies spanning multiple institutions, the issues of data discovery, access, analysis, and integration become more challenging. Such studies require access to geographically disparate, heterogeneous data and analytical resources. Access to resources goes across multiple administrative security domains, further complicating the utilization of the resources. All of these factors limit the impact of research studies, in particular that of collaborative research projects, as well as the extent to which the research can be carried out on a national and international scale.

In this work, we look at a number of distributed middleware architecture frameworks and technologies developed by the information technology community and discuss how these approaches can be employed to address the informatics requirements of large-scale and collaborative cancer research. In Sect. 4.2, we identify and illustrate some of the common informatics needs of cancer research studies using example research pattern templates (Saltz et al., 2008a, b, c). Section 4.3 describes the architecture frameworks and middleware technologies in the context of these requirements. We use caGrid(Saltz et al. 2006; Oster et al. 2008) and the cancer Biomedical Informatics Grid (caBIG®) (caBIG 2009) as well as examples from other biomedical informatics efforts to show the application of these frameworks. We conclude in Sect. 4.4.

4.2 Informatics Requirements of Cancer Research Studies

The particular approach employed by a specific cancer study, the set of experiments carried out, the datasets captured, and how the datasets are analyzed are largely determined by the specific research questions targeted by the study. Nevertheless, a common set of principles and processes are employed by studies investigating

similar problems. We refer these principles, processes, requirements, and constraints on families of research studies as *research pattern templates* (Saltz et al., 2008a, b, c). In this section, we present example pattern templates for integrative cancer research and translational research studies in order to identify and illustrate the common informatics requirements of cancer research studies. We should note that a specific study may involve elements of multiple research pattern templates depending on the scope of the scientific questions and the scale of the study.

4.2.1 Multiscale Deep Integrative Investigation

Studies represented by the multiscale deep integrative investigation pattern template aim to measure and quantify biomedical phenomena at multiple biological scales (e.g., molecular, cellular, macroanatomic scales). Datasets in these studies are obtained from experimental measurements and, in some cases, from simulations. The work being carried out by the newly initiated In Silico Center for Translational Neuro-oncology Informatics (In Silico Brain Tumor Research Center; ISBTRC) provides an excellent example of a multiscale integrative investigation. This research leverages complementary molecular, pathology and radiology brain tumor data obtained in The Cancer Genome Atlas (TCGA),[1] Rembrandt,[2] and Vasari[3] studies. These studies involve collection and generation of radiology; full-slide digital pathology; high-throughput genetic, genomic, and epigenetic analyses for patient populations accrued at a large number of clinical sites. The ISBTRC also plans to generate additional data by carrying out image-analysis-based laser-captured microdissection on institutional tissue specimens followed by targeted high-throughput genetic and genomic studies.

The research work at the center will initially explore the relationship between tumor genetics, gene expression necrosis, and degree and type of vascular hyperplasia. The degree and pattern of necrosis/vascular hyperplasia are variable with a given tumor, so molecular analyses should be interpreted in the context of histopathology. Glioblastoma Multiforme (GBM), for instance, is characterized by necrosis and vascular hyperplasia, which are tightly correlated with the presence of hypoxia. There is large variation in gene expression patterns between regions of severe hypoxia and normoxic regions. Thus, categorization of GBM gene expression patterns should take into account the tumor microenvironment to ensure that the clustering of GMBs within gene expression families is not altered on computer-assisted algorithms. One of the efforts undertaken by the ISBTRC is investigating whether the presence and degree of necrosis within the frozen section slides correlates with

[1] TCGA is a large-scale community resource project co-funded by the NCI and the National Human Genome Research Institute, http://cancergenome.nih.gov

[2] Rembrandt is a community resource that hosts and integrates clinical and functional genomics data from clinical trials involving patients suffering from gliomas, http://caintegrator-info.nci.nih.gov/rembrandt

[3] Vasari datasets consist of the Rembrandt data collection with the addition of characterized MR images, https://wiki.nci.nih.gov/display/Imaging/VASARI

specific gene expression patterns. This effort requires the use of both human experts and image analysis algorithms to identify, markup, and quantify the size, shape, and distribution of necrotic regions. Another effort involves linking nuclear shape and texture brain tumor biological and clinical behavior by posing and answering questions about (1) relationship between nuclear shape and texture to gene expression category defined by clustering analysis of Rembrandt data sets and (2) relationship between nuclear morphometry and gene expression to neuroimaging features. Answering these questions involves correlation of imaging characteristics defined by feature sets (such as the Vasari feature set) with pathologic grade, vascular morphology, and underlying gene expression profiles. Figure 4.1 illustrates the high-level

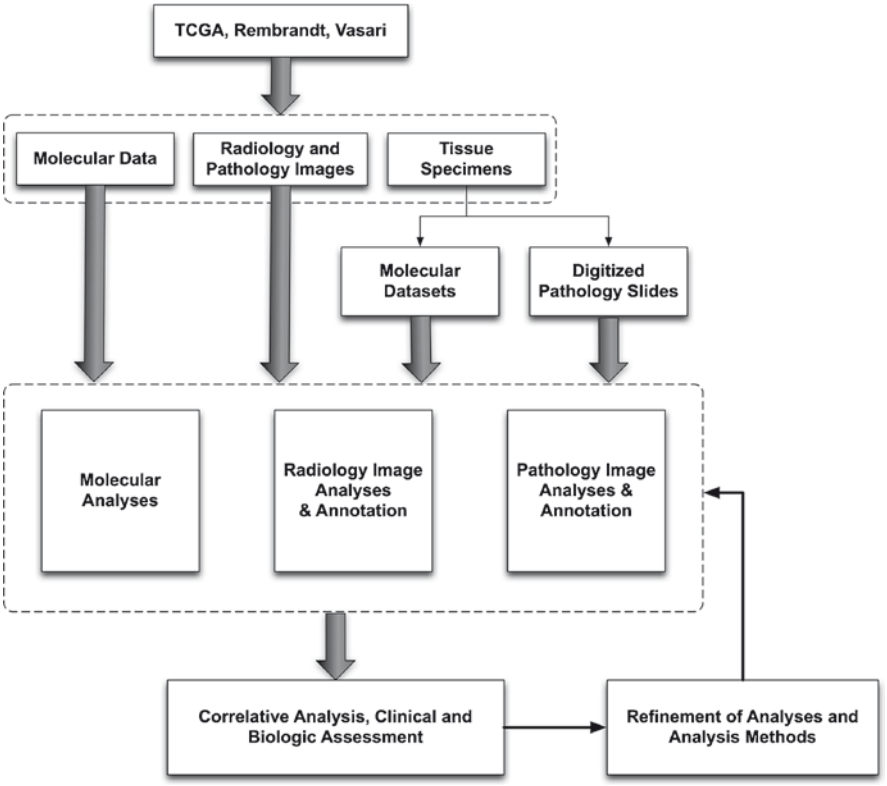


Fig. 4.1 High-level analysis, query, and data integration workflow for the in silico studies. Datasets come from TCGA, Rembrandt, and Vasari studies as well as data generated from tissue specimens at collaborating institutions. Data types include molecular data, radiology and pathology images, and anonymized clinical information. Images are analyzed and image features are classified and annotated. Image analysis workflows may involve simple and complex operations such as normalization, feature segmentation, and feature classification. Molecular data is analyzed via a series of bioinformatics analysis methods and programs. Bioinformatics analysis results, image features, and classifications are stored in databases for correlative analyses involving queries on shape, texture, and gene expression categories as well as integration with clinical information. The results of these queries and analyses may suggest refinement of image and molecular analyses

workflows in these efforts. These workflows involve execution of networks of bioinformatics and image analysis operations and creation and query of semantically complex datasets representing analysis results (molecular analyses and image annotations). Moreover, analysis programs, primary datasets, and results of various analyses may be hosted at multiple collaborating institutions requiring queries and workflow executions to take place in a distributed environment.

4.2.2 Prospective Clinical Research

This template involves studies in which a group of patients are systematically followed over a period of time. Prospective studies are designed to understand risk factors for development or progression of disease and to perform quality control in disease classification and assess treatment effectiveness. Clinical studies that rely on biomedical imaging as both an indicator of disease progression and an assessment of treatments are examples of this pattern template. Cooperative groups conducting prospective studies collect imaging and clinical data as well as treatment reports and outcomes data from patients at multiple institutions. Interpretation of radiology, treatment, and pathology information are crucial factors for reproducible disease classification and assessment of treatment response. In radiation oncology, for example, digital imagery plays an important role in defining the tumor volume and outlining areas that receive radiation and in excluding the portions of healthy tissue that must be spared. There is, however, a high interobserver variability among reviewers of image data. One strategy that is increasingly used to reduce this variability and improve protocol compliance is a central review of imaging objects. In central review, multiple expert radiologists at different institutions review image and clinical data, and an independent adjudicator incorporates their reviews into a consensus review. Figure 4.2 illustrates an example of data submission, review, and analysis in radiation oncology trials. Expert reviewers at different institutions access a remote database of images and may use different workstations to view and annotate the images. Image annotations are then stored in a results database, which is secured to enforce access control.

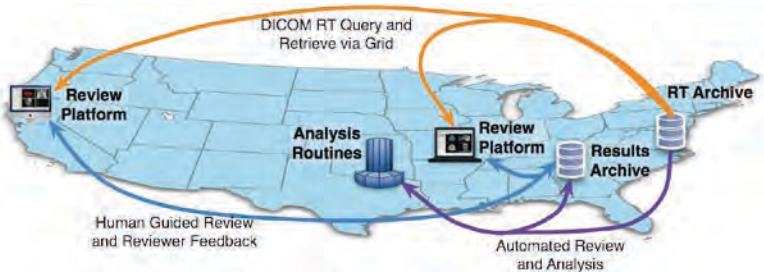


Fig. 4.2 An example of data submission, review, and analysis in radiation oncology clinical trials

4.2.3 Informatics Architecture Requirements

Successful implementation in a multi-institutional setting of the example pattern templates is impacted by (1) how effectively investigators can discover information that is available and relevant to the research project and (2) how efficiently they can query, analyze, and integrate large volume of information from different, potentially distributed, resources. One obstacle is the fact that distributed data sources are fragmented and oftentimes are not interoperable. Datasets vary in size, type, and format and are managed by different types of database systems. Naming schemes, taxonomies, and metadata used to represent the structure and content of the data are heterogeneous and managed in isolation. Two databases may employ completely different data structures, naming schemes, and metadata in order to represent the same information. Informatics architectures supporting the types of studies presented in this work should enable interoperability among multiple systems, facilitate coordinated and secure access to remote resources, and support integration of information from multiple data types and data sources.

The pattern templates provide motivating requirements for the development of support for deep semantic integration of complementary types of information. Genetic expression, protein expression, and cellular structure need to be interpreted, represented, and modeled as highly interrelated phenomena in multiscale integrative studies. The prospective pattern template also has a huge semantic scope. There is a vast span of possible diseases, treatments, symptoms, and radiology and pathology findings in imaging-based studies.

The prospective template provides motivation for architecture support for semantic interoperability and interfacing with existing institutional systems. It is very expensive to develop a purpose built information system for a particular prospective study. From the point of view of economics, logistics, and quality control, it is more efficient to share a core information architecture for many prospective trials. Prospective template information architectures need to interoperate with existing institutional systems in order to better support prospective trial workflow, and to avoid double entering of data and manual copying of files arising from Radiology and Pathology. Subsystems that need to be interacted with may be commercial or open-source systems that adhere to varying degrees to a broad collection of distinct but overlapping standards. Data in these systems are represented, exchanged, and accessed through a set of architectures and standards, such as HL7 (HL7 2009), IHE (IHE 2009), DICOM (DICOM 2009), developed by different communities. A software system developed on top of a particular standard will not be able to readily interoperate with systems developed on top of other standards.

The main objective of gathering data in a scientific study is to better understand the problem being studied and to be able to predict, explain, and extrapolate potential solutions and possible outcomes. This process requires complex problem-solving environments that integrate on-demand data access and processing of very large databases. Studies in the example templates involve querying and assimilating

information associated with multiple groups of subjects from multiple data sources, comparing and correlating the information about the subject under study with this information, and classifying the analysis results. Information systems to support these studies should enable federated queries across potentially heterogeneous datasets and systems. Analysis workflow requirements also arise from the example templates. Multiscale template information systems, for example, should be able to support analysis of data by a series of simple and complex image analysis operations expressed as a data analysis workflow.

Support is needed for workflows comprised of data and analytic services that will allow investigators to iteratively refine complex multidisciplinary analyses as well as to make the in silico research results and processes publicly available to the research community. Analysis results may further be integrated with other data types for additional analyses; for example, genetic and cellular information can be integrated with biological pathway information to study how genetic, epigenetic, and cellular changes may impact major pathways.

Protection of sensitive data and intellectual property is an important component in many design templates. The prospective template in particular has strong requirements for authentication and controlled access to data because of the fact that prospective clinical research studies capture, reference, and manage patient-related information. While security concerns are less stringent in the other pattern template, protection of intellectual property and proprietary resources is important.

4.3 Middleware Architectures for Collaborative Cancer Research

The computer science community and the information technology industry have developed architecture frameworks to address the informatics requirements of loosely coupled, heterogeneous, and (geographically) distributed applications in science, engineering, and business. In this section, we look at several architecture approaches and discuss how they can address the requirements of collaborative cancer research studies. We should note that these architecture frameworks share characteristics and overlapping sets of principles and requirements. We present implementation examples from biomedical informatics efforts.

4.3.1 Grid Computing

Grid computing broadly refers to the notion of accessing and using resources hosted at multiple institutions to support distributed applications in science and engineering (Foster and Kesselman 1999; Foster et al. 2001; Berman et al. 2003).

The concept of Grid computing was originally motivated by large-scale applications in Physics, Astronomy, Earth Systems Sciences, and Engineering that required access to supercomputers at multiple supercomputing centers. Earlier Grid tools and infrastructures, thus, focused on supporting execution of applications on storage and computing systems across multiple security domains. Through community efforts such as the Open Grid Forum (formerly known as the Global Grid Forum) (OGF 2009), Grid computing has evolved into an architecture framework for sharing and federating data and analytical resources as well as computation and storage systems.

Grid computing provides the foundational architecture framework to address several of the core requirements imposed by multi-institutional studies in the example pattern templates. A Grid computing system enables (1) *Remote and coordinated access to decentralized resources.* Collaborative studies in the example templates draw information from multiple systems – for example, health information records, lab information management systems, picture archival and communication systems (PACS), as well as genetic, genomic, epigenetic, microscopy databases – potentially hosted at different institutions. It can be expensive or infeasible, because of security and ownership concerns, to copy data to a single, centralized database management system. Moreover, centralized database solutions are not flexible and scalable. A group of research studies may involve access to the same or overlapping sets of resources; however, each study may make use of different functions provided by these resources or access different subsets of data. Thus, a centralized solution designed for a specific study will not likely address the requirements of other studies. Grid computing, on the other hand, allows the owner of a resource to manage the resource locally. It enables remote access to the resource via open, standard, general-purpose information and data exchange protocols. The resource owner has to make his/her resource available to the environment through these protocols, but he/she does not need to relinquish the ownership of the resource or port it to a centralized system. Different research studies can coordinate access to resources using these protocols and associated tools and create "virtually centralized" solutions for their needs. (2) *Secure and controlled access to resources across administrative boundaries.* Protection of sensitive data and intellectual property is an important component in many design templates. The prospective template in particular has strong requirements for authentication and controlled access to data because of the fact that prospective clinical research studies capture, reference, and manage patient-related information. These issues become more challenging in a distributed environment, because requests to access resources will have to travel across institutional administrative boundaries (see Chaps. 5 and 16). It is not likely that all institutions in a multi-institutional study will have the same type of security infrastructure; it will also be expensive and in some cases infeasible to dynamically create and manage accounts for researchers in multi-institutional projects at every institution participating in the projects. Grid technologies provide the infrastructure for the core security requirements: privacy (i.e., information exchanged

between two entities can only be read by the two entities), integrity (i.e., when information is sent from one entity to another, the information received by the receiving entity is the same as the information sent by the sending entity), and authentication (i.e., being able to verify that an entity involved in information exchange is who he/she claims to be) (Foster et al. 1998; Welch et al. 2003; Langella et al. 2008). Using the core infrastructure support, higher level functions for account management and provisioning, authorization and access control, and secure information sharing can be implemented (Langella et al. 2008).

In cancer research, one of the most prominent efforts is the NCI-funded cancer Biomedical Informatics Grid (caBIG®) program (caBIG 2009) (see Chap. 9). The objective of this program is to develop enabling informatics technologies for collaborative, multi-institutional biomedical research and to create a voluntary network of cancer centers and research laboratories with the overarching goal of accelerating translational cancer research. caGrid is the core Grid architecture of this program (Saltz et al. 2006; Oster et al. 2007, 2008). caGrid is designed to provide the core infrastructure to support federated access to data and analytical resources and applications deployed at different institutions and to enable researchers to both query, integrate, and synthesize information from distributed resources. caGrid leverages Grid computing technologies and tools, including the Globus Toolkit (Foster and Kesselman 1997; Foster 2006) and Mobius (Hastings et al. 2004), to create a biomedical research Grid environment. The caGrid infrastructure provides a common runtime environment and Grid-enabled tools to support the deployment, discovery, and invocation of data and analytical resources, metadata management, management of Grid-wide security, federated query across multiple data sources, and composition of resources into analysis workflows.

Requirements associated with the need to access geographically dispersed resources have been a key motivation behind several other large scientific projects. The Biomedical Informatics Research Network (BIRN), funded by NIH, provides shared access to medical data in a Grid environment (Santini and Gupta 2003; Grethe et al. 2005). The BIRN initiative focuses on data and analysis tools developed in neuroscience research studies. The CardioVascular Research Grid (CVRG) (CVRG 2009) project creates a Grid environment and resources to facilitate research efforts that span multiple research groups and institutions. The CVRG employs Grid computing as the underlying core architecture framework to provide an open-source, extensible software infrastructure enabling discovery, federation, and sharing of cardiovascular data and tools. MammoGrid is a multi-institutional project funded by the European Union (EU) (Amendolia et al. 2003; Solomonides et al. 2003). The objective of this project is to apply Grid middleware and tools to build a distributed database of mammograms and to investigate how it can be used to facilitate collaboration between researchers and clinicians across the EU. eDiamond targets deployment of Grid infrastructure to manage, share, and analyze annotated mammograms captured and stored at multiple sites (Brady et al. 2003; Solomonides et al. 2003).

4.3.2 Service-Oriented Architecture

Grid computing provides an architectural platform for coordinated and secure access to remote resources. In order to facilitate more effective federation and integration of information across multiple resources, informatics systems should also enable *syntactic* and *semantic* interoperability. Syntactic interoperability enables a consumer (e.g., a client program) to programmatically access the components and functionality of a resource (e.g., a service). It deals with the heterogeneity of the programming and messaging interface syntax. Semantic interoperability, on the other hand, is concerned with the use of a resource – that is, semantically correct, unambiguous interpretation, and consumption of a resource. We will elaborate on semantic interoperability in greater detail in Sect. 4.3.3. In this section, we describe an architecture approach, service-oriented architecture (SOA), to facilitate syntactic interoperability.

SOA is an architectural framework in which the functionality of a software component is exposed to the environment as a service via well-defined and published programming interfaces; these programming interfaces are referred to here as service interfaces. A service-oriented architecture environment consists of software components (e.g., applications, tools, and databases) that are loosely coupled to each other and exchange information by invoking service interfaces. For example, a gene expression database, stored in a relational database system, may be wrapped as a service with two interfaces: query and insert. The query interface allows a client program to issue queries for the gene data. The insert interface can be used to insert data into the database. With the service-oriented interface, the client program does not directly interact with the relational database system.

The caGrid infrastructure is an SOA. It builds on the Web Services Resource Framework (WSRF) standards (Foster et al. 2005; Humphrey and Wasson 2005). The WSRF draws from the Web services standards (Graham et al. 2002), but extends them with such concepts as stateful services, service lifetime, service context, etc. These extensions enable more efficient and richer services to be implemented for scientific application scenarios. For example, a query to a large image dataset may take long time to execute and may return many images to the client. In a simple request–response implementation supported by Web services, the client would be blocked waiting for the image data service to finish executing the query and return all of the results at once. A stateful service implementation, on the other hand, could instantiate a *resource*, which would process the query and maintain the results and return a resource handle to the client. The client could then interact with the resource to check for query completion and retrieve the query results (in multiple images at a time). In caGrid, databases are exposed to the Grid environment as caGrid data services with well-defined interfaces. Similarly, analysis methods and programs are accessible as caGrid analytical services. To simplify service development and deployment, caGrid provides a graphical development environment called the Introduce toolkit (Hastings et al. 2007).

A basic characteristic of the SOA is the notion of request–response. That is, the consumer of a resource interacts with the functionality of the resource by submitting requests via well-defined and published service interfaces and receiving responses to these requests from the resource. The request–response pattern defines the interaction between a pair of entities (i.e., the resource and the consumer). However, as we discussed in Sect. 4.2.1, the process of synthesizing information from complex datasets requires researchers to compose and execute analysis workflows involving multiple data and analytical services. These workflows require coordination and execution of interactions (requests and responses) between multiple pairs of services and flow of data and control information between groups of services. Thus, systems building on the SOA should implement support for execution of distributed workflows. caGrid provides a workflow management service that supports the execution and monitoring of workflows expressed in the Business Process Execution Language (BPEL) (Kloppmann et al. 2004; Sarang et al. 2006). The use of BPEL in caGrid facilitates easier sharing and exchange of workflows. There are other workflow systems that are employed in SOA implementations. WEEP (Janciak et al. 2008; WEEP 2009), for example, provides high-level API and runtime support for management and execution of workflows. The Taverna Workflow Management System implements support for composition of service components, shims to mediate between incompatible services, and a graphical workflow development GUI (Hull et al. 2006). The myExperiment site provides a forum for sharing Taverna workflows (De Roure et al. 2009).

An aspect of the request–response pattern is that it is a *pull-based* model. That is, information is transferred from a resource to a consumer upon a request from the consumer. Studies in the prospective template can benefit from messaging and event-driven systems in which information can be *pushed* to one or more consumers. Prospective template studies follow groups of patients in clinical trials. The process of managing clinical trials and data collection requires interaction with a range of systems and instruments, which may interact with each other via exchange of messages. In addition, information about any major events (e.g., adverse events) should be sent to the respective clinical trials management systems for timely handling of events to protect the safety of patients. Data in these systems are represented, exchanged, and accessed through a set of standards, such as HL7 and IHE. Architecture support is needed to facilitate efficient exchange of messages among heterogeneous sets of producers and consumers as well as enable efficient mappings between different messaging standards and data structures. The Enterprise Service Bus (ESB) has recently gained popularity as a software architecture and system integration framework. The ESB provides fundamental functionality and services to implement complex, loosely coupled systems using event-driven and standards-based message middleware. An example of a software system that makes use of the ESB framework is the Clinical Data Exchange (caXchange) tool in the caBIG® program (caXchange 2009). caXchange provides a configurable service and messaging hub built on open-source ESB technologies. An application of the caXchange tool is the mapping of nonstandard laboratory data from clinical care systems into standard formats and delivery of this data via messages to clinical trial databases.

4.3.3 Model-Driven Architecture

The model-driven architecture (MDA) paradigm has gained popularity in recent years as an architecture approach for development of semantically interoperable software systems. As we described in Sect. 4.3.2, the SOA framework provides a platform for syntactic interoperability of disparate systems. Web services standards address the interoperability problem by specifying language-independent access to distributed resources. Research studies that involve information integration and analysis using heterogeneous data and analytical resources have to tackle several additional problems. One problem is that resources may have different data representations. Each resource may define the same conceptual data type using a different data structure and database schema than other resources. Another problem is that the meaning of a resource and the attributes of data structures representing the content of the resource can be named and described differently by different groups. Two data elements representing the same entity might have been defined using different terms; more importantly, two data elements representing different concepts may have the same attributes and attribute names. To address these issues, the contents and meaning of a resource need to be explicitly defined using terms from a vocabulary, which is agreed upon by the community (resource providers and resource consumers). Thus, in addition to the standards imposed and employed by SOAs, controlled vocabularies, common data elements (CDEs), and published information models are necessary to enable interoperability among resources.

In MDA, the information structure and interface specifications of a software component are expressed as models, generally using the Unified Modeling Language (UML). These models can then be mapped to specific architecture or technology platform realizations. The MDA promotes the use of object-oriented design practices and rich metadata in order to facilitate implementation of interoperable systems.

caGrid adopts an MDA approach to enable interoperability through object-oriented abstractions, CDEs, and controlled vocabularies. That is, client and service APIs in caGrid are object oriented. These objects, in turn, are defined using CDEs and controlled vocabularies registered on the Grid. For example, the names of an object's fields are terms from the controlled vocabularies. In addition, the type of a field (Integer, String, etc.) matches the type specified in a CDE. The benefit of this approach is that resources are defined in one location (the vocabulary or CDE) and used to generate all Grid artifacts, preventing any issues with remodeling (the same) data at each Grid layer. A caGrid data service abstracts data as objects. Similarly, an analytical resource (e.g., an analysis program) implemented as a caGrid analytical service provides methods that input objects and return objects. caGrid leverages existing data modeling infrastructure to manage, curate, and employ domain models. Specifically, a Grid developer creates UML class diagrams to model data that will be shared on the Grid. Using UML tools and NCI data modeling infrastructure, the domain models are converted into CDEs in the form of ISO/IEC 11179 administered components and registered in the Cancer Data

Standards Repository (caDSR) (Covitz et al. 2003; Phillips et al. 2006). These data elements are annotated by terms and concepts drawn from vocabulary registered in the Enterprise Vocabulary Services (EVS) (Covitz et al. 2003; Phillips et al. 2006). In the Grid environment, clients and services communicate using messages encoded in XML. When an object is transferred between clients and services, it is serialized into a XML document that adheres to a registered XML schema. The requirement for use of registered data models and XML schemas is to ensure syntactic and semantic interoperability between two end-points exchanging information. With a published model and schema, the receiving end-point can parse the data structure and interpret the information correctly. XML schemas corresponding to CDEs and object classes are registered in the Global Model Exchange (GME) service (Hastings et al. 2004). In summary, the caDSR and EVS define the properties and semantics of caBIG® data types, and the GME defines the syntax of their XML materialization.

4.3.4 Semantic Web Technologies

Semantic Web technologies aim to provide a framework and supporting infrastructure that allows management, query, sharing, and integration of information via machine interpretable semantic representations. Along with enabling tools and middleware infrastructures, standards for representation and storage of semantic information have been developed over the years, including the Resource Description Framework (RDF) (RDF 2009), RDF Schema (RDFS) (RDFS 2009), and the Web Ontology Language (OWL) (OWL 2009). RDFS and OWL are ontology languages, which enable greater functionality for expressing domain knowledge in machine interpretable form.

Domain knowledge and semantic information are critical components in all of the example templates. Multiscale integrative studies, for example, aim to model genomic expression, protein interactions, cellular structure, and other phenotypic observations as interrelated functions of biological systems. It is desirable to express data collected and referenced in such studies using ontologies that represent the domain knowledge. As we stated earlier, the prospective pattern template also has a huge semantic scope; there is a vast span of possible diseases, treatments, symptoms, and radiology and pathology findings in imaging-based studies. Semantic Web technologies can be employed to express, manage, and integrate information using ontologies so that the interrelationships can be captured and represented in a biologically meaningful framework. In addition to facilitating semantic exploration of data, semantic Web technologies can enable semantic interoperability of heterogeneous resources as well as more efficient discovery of resources in a distributed environment. In caGrid, for instance, services register metadata about themselves to the environment (using the caGrid Indexing Service). This metadata can contain terms from controlled vocabularies. The caGrid discovery application programming interfaces allow searches on these terms,

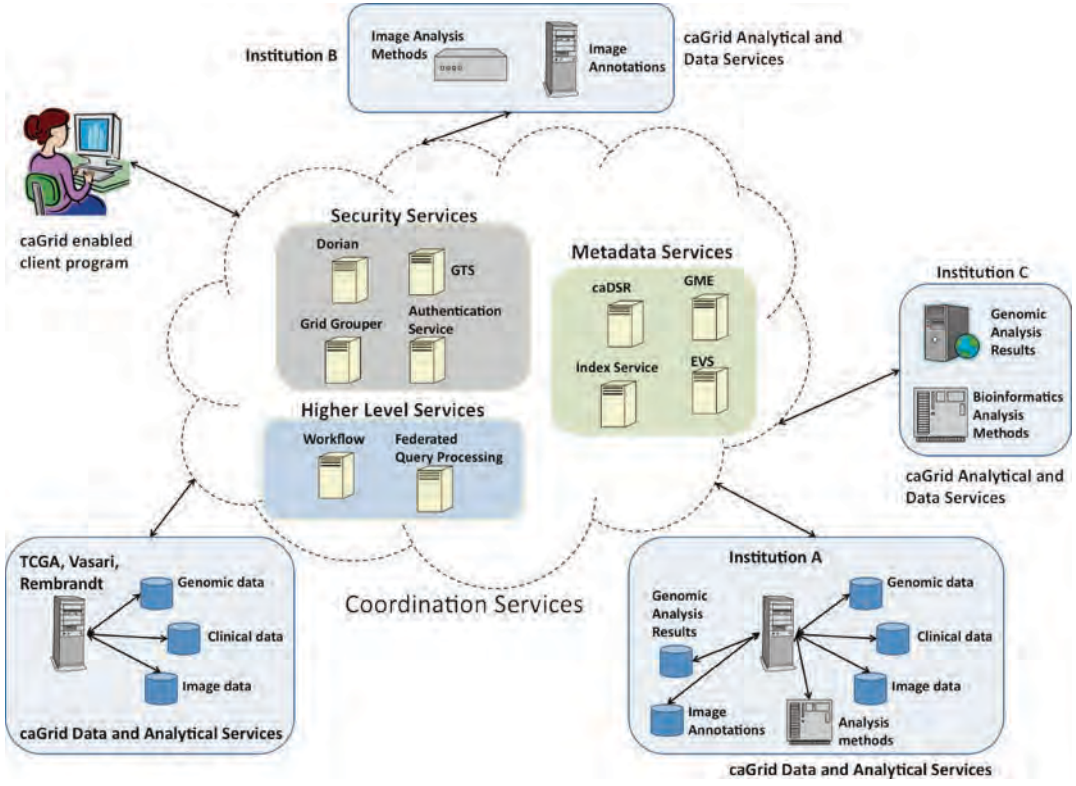


Fig. 4.3 A possible realization of the multiscale integrative investigation template example in a collaborative, multi-institutional environment. In this implementation, the caGrid infrastructure is employed, illustrating the combined use of Grid computing, SOA, MDA, and Semantic Web technologies. The caGrid infrastructure provides a set of coordination services to support service discovery, federated query execution, workflow execution, and security

enabling semantic discovery of services – a client may search for services that manage "Gene" data, for example. The Neuroscience Information Framework project makes use of semantic Web technologies to support representation and discovery of neuroscience research resources, including data sources and tools, hosted at different institutions (Gardner et al. 2008; Gupta et al. 2008). The myGrid project (Stevens et al. 2003, 2004) implements a suite of tools and middleware components, based on semantic Web technologies, that allow researchers to discover bioinformatics services and compose them into bioinformatics analysis workflows.

A possible implementation of the multiscale integrative investigation template example (see Sect. 4.2.1) using the architecture approaches described in this section is illustrated in Fig. 4.3. In this example, caGrid is employed as an example of software infrastructure based on Grid computing, SOA, MDA, and Semantic Web technologies. Grid computing technologies enable access to remote resources across multiple administrative domains (Institutions A, B, and C). The implementation employs an SOA approach: TCGA, Rembrandt, and Vasari datasets are accessible as data services with well-defined service interfaces; similarly, datasets generated locally at each institution, databases of analysis results (bioinformatics results and image annotations), as well as bioinformatics and image analysis methods at collaborating institutions (Institutions A, B, and C) are all exposed to the environment as services. In order to ensure interoperability among these services, the data models used by the services are harmonized and registered in the MDA components such as caDSR, EVS, and GME. Each service advertises itself to the environment by registering service metadata, which includes information about where the service is hosted, the contact information about the service provider, which data models the service exposes, etc. Since the metadata can contain terms from controlled vocabularies, the discovery of services can be done using this semantic information. Grid security services can be used by a service provider to enforce authentication and access control policies to restrict access to a service (see Chap. 5). Grid-enabled clients can discover services and submit federated queries across multiple services or execute workflows involving multiple data and analytical services.

4.4 Conclusions

A critical factor in the advancement of cancer research is the efficiency with which clinical, imaging, molecular, and tissue data can be integrated, disseminated, and analyzed both within and across functional domains. Informatics systems supporting cancer research studies need to address several challenging issues arising from common requirements of research patterns. Systems integration architectures developed by the information technology community have potential to facilitate better interoperability of heterogeneous resources and enable more effective and

efficient utilization of disparate data and analytical resources. Shared domain semantics manifested as published information models, common/controlled terminologies, and standards for data type bindings are clearly at the heart of interoperability. Harmonization of security and policies is another important element in resource sharing and federation. Moreover, tools and services are needed to enable efficient mappings between different messaging standards, controlled vocabularies, and data types associated with many communities and between different messaging and resource invocation protocols. We believe that additional research and development in these and other areas such as interoperability between systems building on standards developed by different communities will further promote and facilitate a wider use of information technologies in science, biomedicine, and engineering.

Acknowledgments This work is supported in part by the National Cancer Institute, National Institutes of Health, through the caBIG® program with Contracts 28X193/N01-CO-12400, HHSN261200800001E, 94995NBS23, and 85983CBS43; by Grant R01LM009239 from the National Library of Medicine; by Grant R24HL085343 from the National Heart, Lung, and Blood Institute; by PHS Grant UL1RR025008 from the Clinical and Translational Science Award Program, NIH; and by Grants CNS-0403342, CNS-0426241, CSR-0615412, CCF-0342615, CNS-0615155, CNS-0406386 from the National Science Foundation. The content of this publication does not necessarily reflect the views or policies of the Department of Health and Human Services, nor does mention of trade names, commercial products, or organizations imply endorsement by the US Government. The In Silico Brain Tumor Research Center is a multi-institutional collaborative effort, funded by the National Cancer Institute, between Emory University (Investigators: Joel Saltz, Daniel Brat, Carlos Moreno, Erwin Van Meir, Chad Holder, Tahsin Kurc, Ashish Sharma), Henry Ford Health System (Investigator: Tom Mikkelsen), Stanford University (Investigator: Daniel Rubin), and Thomas Jefferson University (Investigator: Adam Flanders).

References

Amendolia S, Brady M, Mcclatchey R et al (2003) Mammogrid: large-scale distributed mammogram analysis. Stud Health Technol Inform 95:194–199

Berman F, Hey A, Fox G (eds) (2003) Grid computing: making the global infrastructure a reality. Wiley, New York

Brady M, Gavaghan D, Simpson A et al (2003) eDiamond: a grid-enabled federated database of annotated mammograms. In: Berman F, Fox G, Hey A (eds) Grid computing: making the global infrastructure a reality. Wiley, New York

caBIG (2009) The cancer biomedical informatics grid. Retrieved September 2009 from https://cabig.nci.nih.gov

caXchange (2009) The caXchange project. From https://cabig.nci.nih.gov/tools/LabIntegrationHub

Covitz P, Hartel F, Schaefer C, Coronado S, Fragoso G, Sahni H, Gustafson S, Buetow K (2003) caCORE: a common infrastructure for cancer informatics. Bioinformatics 19:2404–2412

CVRG (2009) The CardioVascular Research Grid (cvrg) 2009. From http://www.cvrgrid.org

De Roure D, Goble C, Stevens R (2009) The design and realisation of the myExperiment virtual research environment for social sharing of workflows. Future Generation Comput Syst 25:561–567

DICOM (2009) The digital imaging and communications in medicine standard 2009. From http://medical.nema.org/

Foster I (2006) Globus toolkit version 4: software for service-oriented systems. J Comput Sci Technol 21:523–530

Foster I, Kesselman C (1997) Globus: a metacomputing infrastructure toolkit. Int J High Perform Comput Appl 11:115–128

Foster I, Kesselman C (eds) (1999) The grid: blueprint for a new computing infrastructure. Morgan Kaufmann, San Francisco, CA

Foster I, Kesselman C, Tsudik G et al (1998) A security architecture for computational grids. In: Proceedings of the 5th ACM conference on computer and communications security conference. ACM, San Francisco, CA, pp 83–92

Foster I, Kesselman C, Tuecke S (2001) The anatomy of the Grid: Enabling scalable virtual organizations. Int J Supercomput Appl 15:200–222

Foster I, Czajkowski K, Ferguson D et al (2005) Modeling and managing state in distributed systems: the role of OGSI and WSRF. Proc IEEE 93:604–612

Gardner D, Akil H, Ascoli G et al (2008) The neuroscience information framework: a data and knowledge environment for neuroscience. Neuroinformatics 6:149–160

Graham S, Simeonov S, Boubez T et al (2002) Building web services with Java: Making sense of XML, SOAP, WSDL, and UDDI. SAMS Publishing, Indianapolis, IN

Grethe J, Baru C, Gupta A et al (2005) Biomedical informatics research network: building a national collaboratory to hasten the derivation of new understanding and treatment of disease. Stud Health Technol Inform 112:100–109

Gupta A, Bug W, Marenco L et al (2008) Federated access to heterogeneous information resources in the neuroscience information framework (NIF). Neuroinformatics 6:205–217

Hastings S, Langella S, Oster S et al (2004) Distributed data management and integration: the mobius project. In: Proceedings of the Global Grid Forum 11 (GGF11) semantic grid applications workshop, Honolulu, Hawaii, USA, pp 20–38

Hastings S, Oster S, Langella S et al (2007) Introduce: an open source toolkit for rapid development of strongly typed grid services. J Grid Comput 5:407–427

HL7 (2009) Health Level Seven 2009. From http://www.hl7.org

Hull D, Wolstencroft K, Stevens R et al (2006) Taverna: a tool for building and running workflows of services. Nucleic Acids Res 34(Web Server issue):729–732

Humphrey M, Wasson G (2005) Architectural foundations of WSRF.NET. Int J Web Serv Res 2:83–97

IHE (2009) Integrating the healthcare enterprise 2009. From http://www.ihe.net

Janciak I, Kloner C, Brezany P (2008) Workflow enactment engine for WSRF-compliant services orchestration. In: The 9th IEEE/ACM international conference on grid computing, pp 1–8

Kloppmann M, Konig D, Leymann F et al (2004) Business process choreography in websphere: combining the power of BPEL and J2EE. IBM Syst J 43:270–296

Langella S, Hastings S, Oster S et al (2008) Sharing data and analytical resources securely in a biomedical research grid environment. J Am Med Inform Assoc (JAMIA) 15:363–373

OGF (2009) The open grid forum 2009. From http://www.ogf.org

Oster S, Hastings S, Langella S, Ervin D, Madduri R, Kurc T, Siebenlist F, Foster I, Shanbhag K, Covitz P, Saltz J (2007). Cagrid 1.0: a grid enterprise architecture for cancer research. In: Proceedings of the 2007 American medical informatics association (AMIA) annual symposium, Chicago, IL

Oster S, Langella S, Hastings S et al (2008) Cagrid 1.0: an enterprise grid infrastructure for biomedical research. J Am Med Inform Assoc (JAMIA) 15:138–149

OWL (2009) The web ontology language 2009. From http://www.w3.org/TR/owl-features/

Phillips J, Chilukuri R, Fragoso G et al (2006) The caCORE software development kit: Streamlining construction of interoperable biomedical information services. BMC Med Inform Decis Mak 6:2

RDF (2009) The resource description framework standard 2009. From http://www.w3.org/RDF/

RDFS (2009) The resource description framework schema standard 2009. From http://www.w3.org/TR/rdf-schema/

Saltz J, Oster S, Hastings S et al (2006) caGrid: design and implementation of the core architecture of the cancer biomedical informatics grid. Bioinformatics 22:1910–1916

Saltz J, Hastings S, Langella S et al (2008a) A roadmap for cagrid, an enterprise grid architecture for biomedical research. Stud Health Technol Inform 138:224–237

Saltz J, Kurc T, Hastings S et al (2008b) e-Science, caGrid, and translational biomedical research. IEEE Comput 41:58–66

Saltz J, Oster S, Hastings S et al (2008c) Translational research design templates, grid computing, and HPC. In: The 22nd IEEE international parallel and distributed processing symposium (IPDPS'08). IEEE, Miami, FL

Santini S, Gupta A (2003) The role of Internet imaging in the biomedical informatics research network. In: Santini S, Schettini R (eds) Proceedings of SPIE, vol 5018. Internet Imaging IV, San Jose, CA

Sarang P, Juric M, Mathew B (2006) Business process execution language for Web services BPEL and BPEL4WS, 2nd edn. Packt Publishing, Birmingham

Solomonides A, Mcclatchey R, Odeh M et al (2003) Mammogrid and eDiamond: grids applications in mammogram analysis. In: Proceedings of the IADIS international conference: e-Society 2003, Lisbon, Portugal, pp 1032–1033

Stevens R, Robinson A, Goble C (2003) myGrid: personalised bioinformatics on the information Grid. Bioinformatics 19:302–304

Stevens R, Mcentire R, Goble C et al (2004) myGrid and the drug discovery process. Drug Discov Today: BIOSILICO 2:140–148

WEEP (2009) The workflow enactment engine project 2009. From http://weep.gridminer.org

Welch V, Siebenlist F, Foster I et al (2003) Security for grid services. In: 12th international symposium on high performance distributed computing (HPDC-12). IEEE, Washington, DC

Chapter 5
Federated Authentication

Frank J. Manion, William Weems, and James McNamee

Abstract Federated Authentication and Authorization is an emerging technology with the potential to facilitate seamless access to information from a variety of providers. Within this chapter we summarize the key concepts, technologies, protocols, and national and even international structures that are being developed to support federated security. We start with the environmental drivers that are stimulating this technology to develop. We then discuss two major approaches to federated security: those based on assertion-based identity and assurance and those based on public key infrastructure. In the second part of the chapter, we discuss the three major components required for development of federated authentication systems: the representation of identity in cyberspace, the manner in which credentials or identity tokens are made available to users, and the required governance processes supporting these concepts. The chapter concludes with a brief overview of the emerging national-scale infrastructure in the form of identity federations, and we present a brief background on these initiatives and the tools and local infrastructure required for joining them.

5.1 Introduction

The topic explored in this chapter is federated authentication, which we define as the ability of a person or entity to rely on, at a particular level of trust, the identity and associated identity metadata asserted by a second entity. The situation is not limited to two-party arrangements; identity federations can be quite large, incorporating millions of individuals from hundreds of different companies and institutions.

Note: Portions of this material are abstracted from an unpublished AAMC/GIR report entitled "Challenges and Opportunities for New Collaborative Science Models: Report from the AAMC Task Force on Information Technology Infrastructure Requirements for Cross-Institutional Research. 2010. Washington, DC: Association of American Medical Colleges. (to be published).

F.J. Manion (✉)
University of Michigan Comprehensive Cancer Center, 1500 East Medical Center Drive, Ann Arbor, MI 48109, USA
e-mail: fmanion@umich.edu

M.F. Ochs et al. (eds.), *Biomedical Informatics for Cancer Research*,
DOI 10.1007/978-1-4419-5714-6_5, © Springer Science+Business Media, LLC 2010

Although this chapter treats a topic in computer security, note that it is neither a general treatise on information security, nor is it intended as a primer for computer or network security in general.

There are a variety of reasons to use common, well-defined legal and technical architecture strategies to allow authentication and authorization practices and technology within and between academic medical center settings. Current models of authentication and authorization have been developed for specific use cases or narrow areas of focus that, almost exclusively, have been concerned with the security needs internal to one organization or corporate structure. While these mechanisms have been reasonably effective within the domains in which they have been developed, to date a comprehensive consideration of an optimal security model to support collaboration across the broad academic medical community has not been done. Such an effort involves consideration of the scientific and clinical workflows that routinely occur *between* institutions and the development of structures – technical, legal, and procedural – to support these exchanges in a repeatable, scalable, and secure fashion.

Development of a multi-institutional common model of authentication and authorization structures would allow teams of clinicians and scientists working either between or within institutions to exchange data and research results easily and effectively in a secure and well-controlled fashion, with the potential to meet the needs of the research, patient, bench science, regulatory, and administrative communities (see Chap. 9 for a discussion of the caBIG® initiative as an example). It would enable the development of workflows that cross institutional boundaries, resulting in increased laboratory throughput, particularly for team-based science initiatives such as those emerging from the NIH Clinical Translational Science Awards (CTSA) community. Development of a well-defined common model agreed upon by the academic medical community and one which is congruent with similar trends in broader academia and government will facilitate the use of national and even international scale shared resources, such as the TeraGrid [http://www.teragrid.org], at the local, state, and national scale. Taken together, these factors would facilitate clinical and basic research by providing an agreed infrastructure for securing clinical data exchanges with the research community.

Two use cases serve to illustrate these points and the type of research and complex informatics environment we find ourselves in today.

> Use Case #1: A cancer center has developed a partnership with a major pharmaceutical company to share tumor tissue and clinical data associated with each tumor. To achieve the numbers of tumors required by the project, the cancer center is partnering with several other sites to collect tumors and the corresponding clinical data. An Institutional Review Board (IRB) has approved the protocol to allow for dates (patient visit date, treatment date, drug release from pharmacy date, lab test dates, perhaps gathered in an electronic medical record as described in Chap. 2) to be released as part of the clinical data to the pharmaceutical company. The cancer center is acting as a clearinghouse for the data transfer between the tumor collection site partners and the pharmaceutical company. Additionally, the pharmaceutical company is performing genomics testing on each tumor sample and research data is being transferred to the cancer center for data distribution to local and

partner researchers (the data warehouse, see Chap. 3). The data warehouse will distribute data based on the number of tumors that each tumor collection site has contributed to the project while the pharmaceutical company will have access to the entire deidentified data (that will include dates) and the cancer center will be able to access the entire dataset.

Use Case #2: A researcher wants to collect all microarray data available from all collaborating centers involved with a particular project from patients with bladder or ovarian cancer that were part of any clinical trial protocol using cisplatin within the past 5 years (which could be stored in laboratory information management systems as described in Chaps. 13 and 14). In addition, the researcher wants to know all available tissue samples, cancerous and noncancerous (normal) tissue localized within 10 mm of tumor site from this patient group such that she/he can perform Affymetrix gene expression studies to include with previously performed studies that were identified by the query. Finally, the researcher needs all the data for any severe adverse events for the group of patients identified that had a severity rating of 3–4 and are likely linked to cisplatin administration (see Chap. 2). The query would be one or a series of federated queries across multiple institutions, clinical and research databases, or data warehouses (potentially using grid technology as described in Chap. 4). All data needs to be deidentified and the results would be sent back to the researcher as an aggregated report that contained the information needed to request expression data, tissues samples, and clinical data. Data would then be analyzed using statistical techniques, potentially using a pipeline (see Chap. 6).

Both of these examples highlight the difficulty of securely transferring data between teams of scientists, clinicians, and other individuals working across institutional boundaries. For information and materials to be reliably and securely transferred between the different parties, a number of key facts about the actors involved must be established. Such an analysis shows that there are more actors involved in each of these use cases than is originally apparent. Consider what is needed to uniquely identify a given investigator who is part of the project, say a "John Smith," and for the moment, remove technology from the picture and consider just a paper-based world. Even in this simple world, to grant a request for information using the protocol outlined in use case 1, above, we would need to know the following facts; that Dr. Smith is in fact who he says he is (the binding of real, physical individual with a name), that he works where he says he does (assume he works at one of the partner sites of the cancer center), and that one or more IRB's have approved the protocols under which he is requesting.

As can be seen from the above, the two related processes of authentication and authorization involve the retrieval and analysis of information documenting the occurrence of a series of prior mandatory manual steps. These steps include the establishment of trust via common policies, procedures, and contractual agreements between the various parties; definitions of the level of trust needed between the parties; appropriate mechanisms for vetting or otherwise establishing the physical identity of the subjects in question; and well-defined and unambiguous definitions of the criteria needed to allow a given subject access to a given resource. Specifically, when an organization allows access to a system based upon a database entry in their database that says person (or possibly, system) X satisfies criteria Y, this works because they trust the manual steps necessary to get that entry into the database. It is not the information in the database that matters for this trust, but the

assurance that the manual process for putting it there is correct and is accurately executed. If one fails to recognize that, then one is likely to make the mistake that equivalent information from a different database will be (or should be) adequate for granting automatic access – it will not be, and never will be, unless there is equal knowledge of, and trust in, the associated manual processes.

All these highlight the need to reach consensus on a variety of topics, both technical and nontechnical, before such an infrastructure can be effectively employed. Elements of this infrastructure include development of a common trust fabric,[1] the development of common naming conventions for institutions involved, common definitions of roles, titles, and practice credentials (such as RN, LPN, CCRP, etc.), and the legal, policy, financial, and governance structures to support the effort.

Later sections of this document discuss aspects of the governance, legal, and technical implications of choosing such a course of action. We also review the existing efforts in developing common models of authentication and authorization practices for use in the academic community.

5.2 Cross-institutional Authentication and Authorization

Increasingly, collaborative tools and other "shared" information resources located at specific institutions require user authentication and authorization for access. However, since many collaborators wishing to utilize these restricted resources may not be affiliated with the hosting institutions, the question arises as to how these external potential collaborators should be identified, authenticated, and authorized to use these resources. This is problematic because few individuals have authentication credentials and other digital identities that can be recognized, evaluated, and trusted by information systems hosted by multiple organizations. This results in the implementation of a variety of chaotic and duplicative identity-management work-arounds that are frustrating for users, are inefficient to manage, and are often insecure.

In the physical, or non-cyber world, we authenticate the *physical identity* of another person we meet by using our five senses – usually by sight and sound. When someone we never met comes into the range of our senses, we mentally register his or her physical appearance. Depending on how soon and/or often we subsequently see the person, we may develop a mental image of the individual and hence are able to recognize that person with considerable confidence upon seeing him or her again. This process, however, is not infallible. We could fail to recognize someone because the individual might grow a beard, dye her hair, lose weight, get

[1] Or perhaps agreed upon mechanisms to allow such fabrics to be negotiated in real time, as this is an emerging area of research.

older, or we might have memory loss regarding previous encounter(s). Or, we might falsely "recognize" the identical twin of the intended person.

We now need to extend our physical beings into the cyber world such that each of us has one or more virtual existences that can be at least as easily recognized (i.e., authenticated) and trusted in cyberspace as our physical appearance is in the face-to-face world. This virtual existence must be readily recognizable and trusted at defined levels of assurance by both humans and digital information systems. This can be accomplished by establishing an identity management infrastructure (IdM) that spans across institutions. This infrastructure provides individuals with authentication credentials and digital identity profiles that dynamically permit the formation of trusted relationships not only among people, but also between people and digital information systems and among digital information systems themselves.

Currently, no generally accepted identity management infrastructure exists to support large-scale, cross-institutional collaboration where various aspects of each collaborator's identity and documented attributes must be known and trusted, although attempts are being made to develop such infrastructure in the caBIG® project (Langella et al. 2007 and Chap. 16). Such a common infrastructure is indispensable when authentication, authorization, utilization of digital signatures, information integrity, and individual accountability must dynamically occur across institutional boundaries and among large numbers of relying parties. *Operationally, electronic transactions requiring knowledge of personal identities must be appropriately secure and protect privacy, but be virtually as easy to use as public Web pages*. However, since most individuals have not experienced such an environment, many have trouble appreciating why this is required and envisioning the resulting benefits.

5.2.1 Identity in Cyberspace

There are two necessary and distinct aspects of a *person's identity in cyberspace*:

1. *Physical identity*. Each participating human being must be vetted, assigned a globally unique, persistent identifier, and credentialed by a *credentialing authority* (CA). The authentication credential issued to a vetted physical person must be designed such that it can only be activated by that individual.
2. *Personal identity attributes*. It must be possible to verify certain key attributes of each authenticated physical person, so that attribute-dependent authorization can occur. These attributes often change over time and so must be capable of real-time verification from sources that are trusted to be both accurate and current. Validated data are verified by one or more *sources of authority* (SOAs), maintained in *systems of record* (SORs), and often distributed by trusted *attribute authorities* (AAs) to approved requestors.

Within institutions, the management of verified identity and verified attributes is often handled through similar or identical processes in the HR or security department. Because these are so effectively conflated within institutions, people often have great

difficulty conceptually separating the process of *authenticating* a person as a *physical entity* vs. determining the *personal attributes* of that person needed for *authorizing* specific privilege(s). Yet such a separation is required in a federated environment.

1. *Authentication of physical identity* is a process whereby every time relying parties receive the same authentication credential, the relying parties can trust at a defined *level of assurance* (LOA) that the certified physical person is actually the individual presenting the credential.
2. *Authorization* is the process whereby a relying party determines if an authenticated physical person has the appropriate attributes to be qualified to conduct specific activities, to qualify for access to specific resource(s), etc.

5.2.2 Federated Authentication and Authentication Credentials

The concept of an "authentication credential" as used above is crucial to authenticating a claimant's physical identity across organizations. An "authentication credential" must minimally provide the following functions and information:

1. It can only be activated by the certified person to whom it was issued
2. It positively identifies the certifying authority (CA) – that is, the credentialing authority
3. It positively identifies the physical claimant – that is, physical identity is vetted by the CA
4. It provides a certified globally unique identifier issued to the vetted individual and registered with the CA
5. It asserts a defined LOA that the credential is presentable only by the physical person it authenticates

Authentication credentials must be issued by trusted *credentialing providers* (CPs), also referred to as *certifying authorities* (CAs) or *credential services providers* (CSPs). These credentialing authorities must follow well-defined and mutually agreed upon policies and procedures.

Figure 5.1 illustrates the basic process that a credential provider must follow to identity, register, and credential a person.

1. A person presents him/herself, either virtually or physically depending on the LOA to be asserted, before a credential provider (CP).
2. The CP vets the person's physical identity according to explicitly defined levels of assurance (LOA) polices and procedures. (Note that at low levels of LOA, "physical identity" may be inferred from online and "out-of band" exchanges of "private" information between a CP and the person being vetted and not explicitly confirmed by direct physical contact between the CP and the claimant.)
3. The CP validates any required personal attributes (e.g., date of birth, legal name, etc.) required by policy, assigns an ever-lasting CA-specific identifier, and records this information in a person registry.

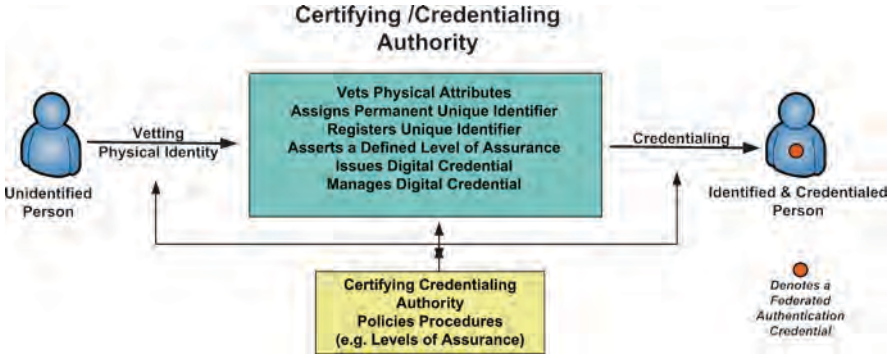


Fig. 5.1 Certifying/credentialing authorities

4. The CA, according to explicitly defined policies and procedures, issues to the person a strongly bound authentication credential that has a specific LOA.
5. The CA must have well-defined policies and procedures for revoking an issued credential.
6. The CA must have well-defined policies and procedures for credential renewal.
7. The CA must provide an online process for checking the status of a credential, for example, whether the credential is valid, revoked, or expired.

The exact details of these steps that must be followed by all CAs vary from federation to federation, but they are spelled out in the policy and procedure documents and in general will follow the above outline.

5.2.3 Significance of Federated Authentication Credentials

It is important to realize that a digital authentication credential has far more functionality than the traditional application-specific username/passwords issued to individuals. A person having a federated authentication credential can use that credential to initiate de novo interactions with relying parties – both people and machines. In the new collaborative milieu within cyberspace, millions of intertwined people and digital systems must safely "introduce" themselves to each other when they need to meet and appropriately interact – not all that unlike face-to-face meetings in the non-cyber world. Widely trusted authentication credentials enable these just-in-time introductions.

Functionally, in a federated collaborative context, the following can occur:

1. Every physical person has a single, certified authentication credential with a permanent personal digital identifier issued by a credentialing authority. (The requirement that every physical person have a *single*, *permanent* personal digital

identifier argues strongly in favor of third-party-issued credentials and strongly against employer-issued credentials. If credentials are issued by employers, then whenever a person changes employment he/she must obtain a new credential, thereby violating the requirement of a single, permanent identifier.)
2. All relying parties, both persons and digital information systems, trust the certified credentials issued by the credentialing authorities
3. Relying parties "know" a physical person by the certified authentication identifier.
4. Every person can authenticate their physical identity to any relying party willing to "trust" a credential. (Remember that authentication does not itself generally grant access, and an authentication credential likely provides few if any personal attributes other than the assigned personal identifier.)
5. A relying party, if permitted, obtains specific required personal attributes of an authenticated person from certified SOAs.
6. If the obtained personal attributes meet conditions specified by a relying party, the authenticated person is trusted and privileged.
7. To protect individual privacy, personal attributes of an identified physical person can only be released to specific relying parties if approved by the authenticated person and/or by legal agreement.

Figure 5.2 illustrates a situation where an individual affiliated with Institution A must access a restricted service provider (SP) at Institution B. *The requestor has never attempted to access the resource before*. Since both institutions are members of the same identity federation, the following can occur:

1. The requestor presents her federated authentication credential issued by institution A to the service provider (SP) hosted at institution B.
2. The SP authenticates the requestor using her credential and accepts the person's certified identifier – that is, the SP now recognizes the physical identity of the claimant.

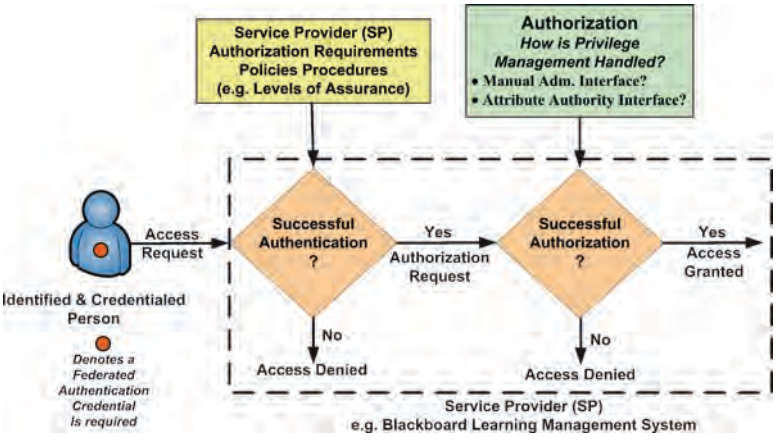


Fig. 5.2 Access across institutional boundaries

3. The SP follows a defined authorization process to determine if the authenticated and identified individual has the proper personal attributes to be granted appropriate access to the system.
4. Access is authorized for specific privileges only after the SP has somehow determined that the authenticated physical person has the required personal attributes.

5.2.4 Federated Credential Providers

Academic institutions must decide how they will provide federated authentication credentials to their faculty, staff, students, and possibly other affiliates such that they can be authenticated by relying parties at other institutions. Since there is not a single, centralized authentication credential provider (CP), institutions are currently faced with the following choices:

1. Implement one or more credential providers at each institution
2. Use an external, possibly commercial, credential provider to issue authentication credentials
3. Utilize some combination of the preceding two choices

Most institutions today have implemented an internal identity management (IdM) infrastructure that allows physical and attribute identity to be managed. This in turn may support single sign-on (SSO) authentication within an institution so that each user has a single authenticator such as a password or a one-time password device. Once used, the authenticator grants access to multiple restricted information resources until the user signs off. These IdM infrastructures in most cases were not designed to provide users with authentication credentials that could be used to authenticate their physical identity across organizational boundaries. However, these infrastructures may be refined to adhere to specific federated policies and procedures and to implement technologies that support management and use of authentication credentials across institutions. Some institutions may decide that they do not have the resources to become credential providers and instead chose to use authentication credentials issued to individuals by an external credential provider (CP).

In order for relying parties to "trust" authentication credentials issued by multiple credential providers, the credential providers must adhere to well-defined policies and procedures that are known to and verifiable by the relying party. To accomplish this goal, identity management federations comprised of member organizations are forming in which the membership defines and agrees to follow specific policies and procedures. Section 5.5, entitled *Emerging National Infrastructure to Support Federated Identity and Anticipated Impact on Research Organizations*, examines the current state of supporting policy and assessment frameworks currently being used and built upon by the United States Federal Government and several large-scale identity federations.

Figure 5.3 depicts three institutions, a commercial service provider, a commercial credential provider, and three credentialed persons. The institutions and the two commercial entities are all members of a single identity management federation. Institutions A and B implemented institutional credential providers, whereas Institution C does not have an institutional CP. Instead, institution C requires its personnel to have authentication credentials issued by the commercial credential provider. The three institutions and the commercial service provider (SP) all host restricted information resources that accept federated authentication credentials issued by the three CPs. The directed lines denote that the three individuals have presented their authentication credentials to various resources, all of which in turn have authenticated the individuals and identified their physical identities, that is, they have accepted their certified unique identifiers. However, the figure does not indicate if the successfully authenticated individuals have in fact been authorized for access.

Systematic use of federated identities is still in its infancy. Therefore, there is still some discussion about whether it is best for institutions to implement internal credential service providers and issue their own authentication credentials or to utilize credentials issued by external, commercial providers. However, as noted above, the use of employer-provided credentials must, over time, result in a violation

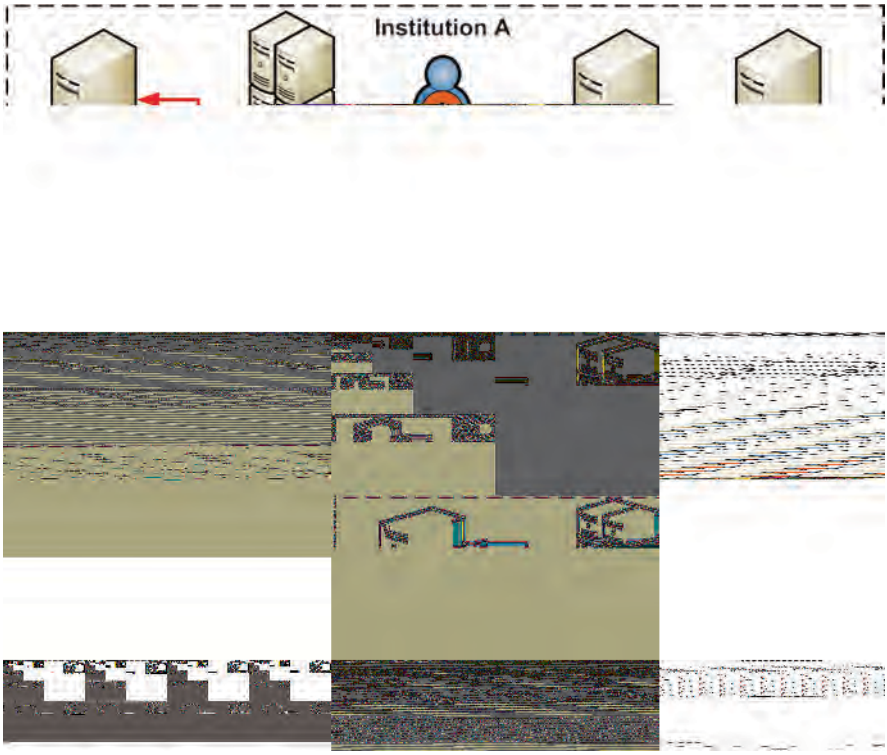


Fig. 5.3 Hybrid local and commercial credentials

of the goal of every person having one and only one permanent digital identifier to be used in the federated environment.

Despite the stability advantage of third-party credentials, the initial trend is for many universities to establish their own CPs, since this more closely matches their traditional methods. Clearly, the technical, policy, and procedural aspects of operating an institutional CP are resource intensive. Thus, several institutions, particularly smaller organizations are considering utilizing commercial CPs. Because of this, some IdM federations are considering accepting commercial CPs as federation members.

A quick examination of Fig. 5.3 illustrates the problems associated with employer-issued credentials. For example, if the person credentialed by Institution A moved to Institution C, then he would lose his Institution A credential and be issued a new authentication credential from the Commercial CP. That physical person would no longer be recognized by the IRB Protocol Management System hosted by institution A, nor the Pre-grant Award System hosted by the commercial SP or the LMS Application hosted by Institution B, because he now has a different authentication credential and associated unique identifier. Also, the credential previously issued by institution A is no longer guaranteed to be currently associated with an appropriately vetted individual. This loss of association could lead some institutions to reuse the now released identifier, creating even more problems in the federated environment.

5.2.5 Authorization

Service providers (SPs), as noted in Fig. 5.2, can implement authorization (i.e., privilege management) in multiple ways. Some SP applications may have an internal role-based system with an associated administrative interface that administrators use to manually assign roles/privileges to physically recognized persons. These administrators must determine from "trusted" SOA if authenticated physical persons have the appropriate personal attributes to be authorized for specific access privileges. In order to accomplish this, an administrator must somehow know an individual having a certified unique identifier is the same person for which one or more SOAs are providing personal attribute information.

Other SP applications may implement automated authorization by interfacing with attribute authority (AA) systems and/or other "trusted" systems of record (SOA). This allows the SP application to query an approved SOA if an authenticated individual has the appropriate personal attributes required to be granted specific privileges. In some case, an SP may need to query attribute authorities at multiple institutions. Note that without permanent, globally unique digital identifiers for individuals, obtaining trustworthy attribute information from third-party sources will be extremely difficult, if not impossible.

Figure 5.4 illustrates autoprovisioning of a learning management system (LMS). In this case, successful authentication of a physical person leads to automated

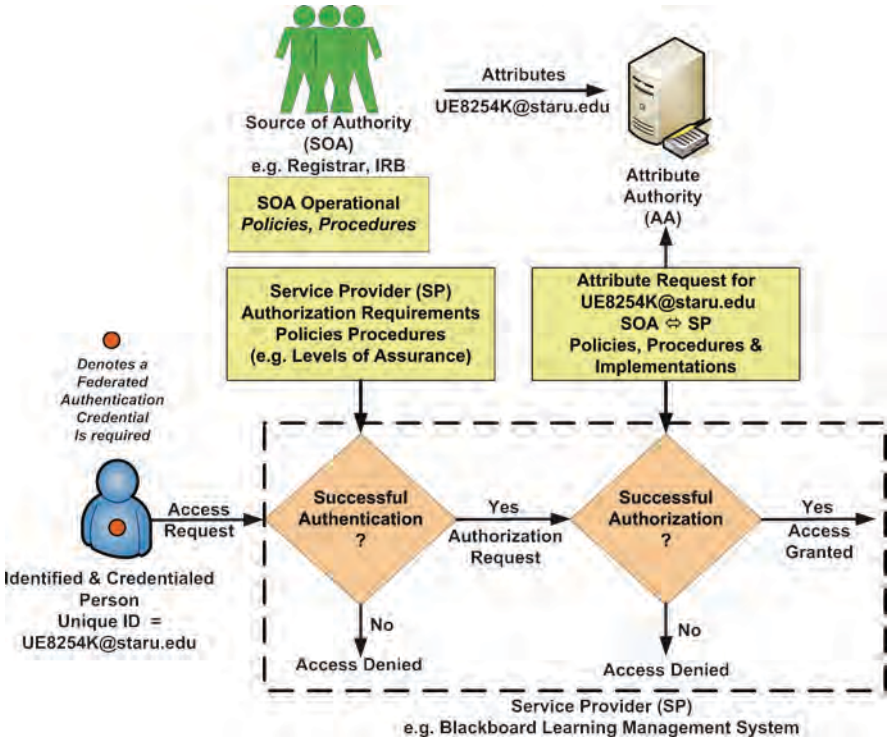


Fig. 5.4 Autoprovisioning of identity into applications

authorization resulting in the claimant being granted access with appropriate privileges. The following context applies:

1. Claimant requesting access is credentialed by a commercial credential provider.
2. The LMS is hosted by Institution B.
3. The personal attribute source of authority and attribute authority is hosted by Institution C.
4. Claimant presents his authentication credential to the LMS.
5. LMS authenticates the claimant and obtains the certified unique ID from the credential.
6. LMS sends an attribute request to the AA to determine if the claimant has entitlements for roles and hence access to the LMS.
7. The AA sends an entitlement attribute for the claimant indicating, for example, a coordinator role for a specific collaborative working group.
8. The claimant is granted access with the coordinator role.

This example clearly indicates the importance of a permanent, unique identifier obtained from an authentication credential in the authorization process. If the source of authority (SOA) "knows" the physical identity of the claimant by an identifier different from that in the authentication credential, the automated authorization and

subsequent granting of access cannot occur. Thus, it is critical that authorization polices and procedures ensure that personal attribute profiles maintained by SOA for a specific physical person are truly the profiles belonging to the authenticated person. Otherwise, the wrong personal attributes will be sent from attribute authorities to relying parties. It is equally critical that the digital personal identifiers used as "keys" to retrieve the personal attributes be permanent and globally unique.

Privacy protection demands that personal attributes of an identified individual be sent to requesting applications and other relying parties only if the identified person has formally consented to the release of those specific attributes or if legal requirements specifically permit or require their release. When appropriate, applications and workflows can be created that enable an authenticated person to be asked to electronically sign a consent statement for the release of specific attributes to specific relying parties.

5.3 Emerging Authentication and Authorization Technologies

Up to this point, the discussion of identity, identity management, authentication, and authorization has focused on high-level functional requirements for a cross-institutional IdM infrastructure. The National Institute of Standards and Technology published NIST Special Publication 800-63 version 1.02 entitled *Electronic Authentication Guideline* (Burr et al. 2006), which provides specific technical requirements for the issuance, management, and use of authentication credentials. This and other related documents are covered in the later sections of this chapter. As a result of these documents, authentication credentials are progressively being viewed as falling into four levels of assurance – Levels 1–4, proportional to the consequences of the authentication errors and misuse of credentials. As the consequences of an authentication error becomes more serious, the required LOA increases.

5.3.1 SAML

Authentication credentials having levels of assurance (LOA) 1 and 2 are assertion-based credentials that use network transmitted passwords (i.e., password tokens) as authenticators. One emerging standard for transmitting such assertions is the Security Assertion Markup Language (SAML) created by the Organization for the Advancement of Structured Information Standards (OASIS). SAML 2.0 [http://saml.xml.org/saml-specifications] is an XML standard used to exchange authentication credentials and authorization attributes between an identity provider (IdP) (i.e., a producer of assertions) and a service provider (i.e., a consumer of assertions).

A credential provider, wishing to assert Level 1 and/or Level 2 authentication credentials and/or specific personal attributes for persons it certifies, implements an identity provider (IdP) that uses SAML. At the request of a credentialed individual, the IdP sends SAML identity assertions to a service provider. The service provider then uses the information received to make access decisions.

5.3.2 *Shibboleth*

Shibboleth [http://shibboleth.internet2.edu/] is a SAML-based, open-source implementation of an infrastructure for federated identity-based authentication and authorization that was developed as part of the Internet2 Middleware Initiative. It provides both identity provider (IdP) and service provider (SP) components.

The basic Shibboleth components are illustrated in Fig. 5.5. In this example, Institution A is an authentication credential provider and issues authentication credentials to its affiliated personnel. It uses a Shibboleth IdP to assert both the authentication credential and specific personal attributes for an authenticated user to an LMS hosted at Institution B. The following are the basic functional steps in this process:

1. The credentialed user contacts the Shibboleth protected LMS.
2. The Shibboleth SP redirects the user to the Shibboleth discovery service.
3. From a menu provided by the discovery service, the user selects his institutional IdP.
4. The IdP prompts the user for a username/password, authenticates the user, and sends his authentication credential to the Shibboleth SP.
5. The SP sends an attribute request to the IdP, which in turn returns the specific personal attributes to the SP.
6. The user is then granted access to the LMS with appropriate use privileges.

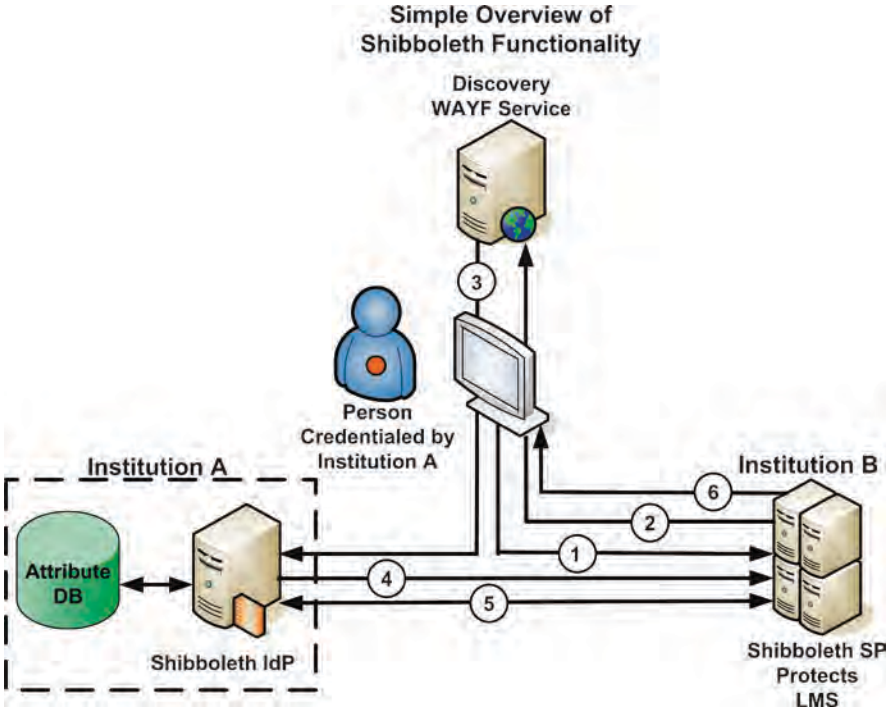


Fig. 5.5 Shibboleth

Current Shibboleth implementations do not permit an SP to request and receive personal attributes from IdPs other than the IdP which asserted the authentication credential. Thus, an SP cannot aggregate personal attributes belonging to an authenticated physical person from other sources of authorities using the Shibboleth infrastructure. If additional personal attributes are required, they must be manually entered in the protected application or automatically obtained via a non-Shibboleth data exchange.

Shibboleth is the most widely used authentication/authorization infrastructure currently employed by academic institutions worldwide for cross-institutional collaboration.

Authentication credentials having Level 3 and Level 4 assurance levels (LOAs) cannot use network transmitted passwords as token, but instead must use cryptographic tokens – for example, certified private/public key pairs with X.509 certificates.

5.3.3 Grid Authentication/Authorization Infrastructures

While not directly discussed in this chapter, grid computing in the sciences is becoming an important topic in many e-Science projects (see Chap. 4). Projects such as Enabling Grids for E-Science in Europe (EGEE) [http://www.eu-egee.org/] and the Cancer Bioinformatics Grid (caBIG®) [https://cabig.nci.nih.gov/] (see Chap. 9) in the USA, make use of grid computing frameworks such as the Globus Toolkit (Foster 2006). The Globus toolkit contains a variety of components to enable identity sharing across institutional boundaries. Further, the grid community in general has established a framework, the International Grid Trust Federation [http://www.igtf.net/], consisting of four member *Policy Management Authorities* representing various geographic regions. These organizations serve to coordinate grid certificates and access to allow interoperation between various grid projects worldwide. In addition, the caBIG® project has developed specific infrastructure to coordinate trust and multiple levels of authority across a data and computation grid (see Chap. 16).

5.3.4 Authorization Across Institutional Boundaries

Authorization involves establishing that currently valid information about a person's membership (university employee), role (clinical faculty), attributes (MD), credentials (Board certified in Pathology), or other significant qualifications (currently NIH-funded) meet predefined criteria for access to an information system. Systems serving dozens of users have successfully relied on manual authorization procedures. But as a resource becomes open to hundreds or thousands of users, it can become advantageous to automate the authorization step.

Automating authorization is straightforward when one database holds all the necessary information about all people who can possibly use that resource. Systems operating strictly within the scope of one organization, for example, employee electronic time sheets, might only need to consult one database, HR/Payroll, to determine whether an authenticated individual needs to track his or her work hours.

A system intended for access by employees at any one of the several organizations might need to consult databases at those or other organizations to validate that an authenticated person's qualifications meet stipulated criteria. An example of this might be a clinical research database. Prospective users must demonstrate employment in a clinical department at an accredited medical school, hold an MD degree, be board-certified in pathology, and be the PI on an active NCI grant before receiving access to this resource. Several databases located at independent SOAs (university, board certification registry, NIH, etc.) may have to be consulted before these criteria could be automatically verified.

Many formal and ad hoc research collaborations will be created in coming years. If there is any hope for interoperability among them, it will be necessary to establish standards and governance processes to define which attributes a particular SOA will be trusted to provide, to identify the timeliness of these data, and to provide a process for resolving inconsistent information between SOAs. This will be necessary before decisions can be made about which SOA to contact for which personal attributes. In so far as governmental SOAs such as NIH are involved in providing personal attributes, certain practices can flow from existing interagency and intergovernmental initiatives (e.g., US Federal Government e-Authentication initiative). Other practices now in use by federations of organizations (e.g., InCommon) should also provide a shared blueprint for the standards and governance process within and between research federations. The governance concept must extend beyond sharing identity and attribute information. Scientific data, physical samples, and personnel must flow between organizations and federations to achieve greater benefits. Because no single governance committee can have sufficient expertise to address each of these areas, we expect that subcommittees will evolve for each of them and well as others as needs arise.

Ease of maintenance and speed of provisioning services are important properties for research systems. Automating authentication and authorization will make significant improvements in both. Having met automatically verifiable objective criteria such as the one illustrated above, there could be other attributes that cannot be inferred from data available from traditional SOAs. Is the person who seeks access a research colleague of a previously authorized system user? Is that person a recognized expert in their field? If system access is a scarce resource, did the person demonstrate sufficiently great scientific need for it? Are they under a sanction for previous scientific misconduct? Even though such additional attributes may be crucial, existing SOA databases may not be capable of representing these facts with sufficient detail or currency to meet every need.

Even though manual authorization steps may not be completely eliminated by middleware infrastructure, the number of those steps could be substantially reduced

by incorporating SOAs into design and operation of research systems that require authentication and authorization.

5.4 Policies, Procedures, and Standards to Ensure the Integrity of Research Data and Networks

Careful research requires tracking data from its collection, through its initial publication and for the duration of its future uses. As the size and lifespan of research data sets grow beyond an investigator's capacity to manage them, the biomedical research community looks to central information technology organizations for help. Building systems that assure data integrity, availability, and interoperability is challenging not only technically but also from a policy perspective as well. Several of these issues concerning the use of collaborative tools and authentication/authorization were raised previously but policy concerns were not specifically addressed. Here, we consider some policy issues that are especially relevant in a federated environment.

Systems are composed of technology, processes, and people. Technology encompasses the computers, storage devices, and databases linked through a data network. To achieve interoperability, each piece of technology must support widely adopted standards so that it can easily and unambiguously exchange information with other information systems. Process includes the methods used to acquire and accumulate research data, the application software that analyzes them, and the metadata that allows the data to be indexed, saved, and readily retrieved. Technicians who operate the technology as well as investigators who create, use, or reuse this information are the people who determine the goals and define the utility of an information system. Just as technical standards can assure that data can be transferred seamlessly among devices and locations, so policies permit processes and people to share information in ways that promote its reuse.

The Data Sharing and Intellectual Capital (DSIC) workgroup of caBIG® established policy guidelines for working with cancer research data. They recommend assessing four characteristics of a data set to classify its sensitivity: Economic/Proprietary/IP Value, Privacy/Confidentiality/Security Considerations, IRB or Institutional Restrictions, and Sponsor Restrictions. Though crafted with patient data in mind, these considerations are a useful model for deciding whether to share a data set and in determining the strength of safeguards that protect it. The presumption is that data believed to be less sensitive or valuable will have fewer protections and can be made more widely accessible.

DSIC's guidelines do not address whether others must provide protection to the data set once it has been shared. The guidelines are also silent on controlling future uses and attribution of any data shared. These are important matters to negotiate when deciding why, when, and with whom to share research data. Matters become more complex for a research consortium in which researchers from many organizations contribute to a common data warehouse. If it becomes necessary to negotiate

hundreds of data protection and use agreements among a two dozen consortium partners, absence of a single commonly adopted agreement will impede research progress considerably.

Using the DSIC's three-tiered classification scheme, a Low level of control would require an agreement not to redistribute the data set and to give attribution of the original data source; a Medium level would add a nonuse period to prevent competing publication or patent claims; a High level might permit data sharing only with named colleagues and only for the scope of a joint research project.

We recommend broadening DSIC's guidelines to address standardizing security for data that is shared outside an organization. A common data protection agreement might simply stipulate that once transferred, a shared data set must receive a comparable level of physical and technical security. For society to reap the greatest benefits from research data sharing, organizations must avoid imposing excessive restrictions on data reuse. Too many restrictions could prevent researchers and organizations that have limited resources from contributing effectively to team discoveries.

IRBs have the responsibility of protecting the rights and safety of human research subjects. They must also assure that investigators take appropriate steps to maintain confidentiality of patient information. IRBs can supplement their own expertise with that of information technology specialists to assess the adequacy of proposed data protections.

A well-accepted way to address privacy concerns when working with patients and human subjects is to use deidentified data exclusively. Other forms of research data can be highly sensitive if it contains proprietary information, harbors trade secrets, or has significant intellectual property value. There may be no way to desensitize such data. Therefore, strong data security protections must be afforded under a variety of circumstances.

Barriers to automated authentication and authorization were previously discussed in this chapter. In addition to establishing a process for who may access research data collections, investigators and organizations must reach consensus on permitted uses of those data. Interinstitutional agreements might deny external investigators' access to a particular set of research data until it is published by its creator or until a patent is applied for. Policy decisions like these become more complex when they involve a database to which investigators from many organizations contribute. Questions regarding information ownership and stewardship must be resolved for research data sharing to realize its full potential.

People participate in various capacities when creating biomedical research data. A laboratory technician or graduate student may execute experimental procedures, make measurements, and document observations. A statistician may analyze the data and test hypotheses constructed by the investigator. A team of collaborators may integrate these data with those from their laboratories prior to submitting it for publication. Presumably, all contributors to this project will work with the data to produce a high-quality data set. Metadata containing a record of who contributed what should be made a lasting part of the data set for two reasons. It can establish a pedigree for the data fostering trust in and credibility of it. Professional credentials of personnel relevant to the project can be logged. Metadata will also help

assure proper attribution when the data are reused by someone not part of the original team (see Chap. 8 for an introduction to reproducible research methods).

Research data may have economic value separate from its scientific merit. Loshin (2002) lists several parties for whom research data has value and urges their interests to be considered when making decisions about protecting it. Fishbein (1991) encourages institutions to state their policies on this value proposition clearly and offers useful guidelines for those policies.

Technology, process, and people thus share data ownership, stewardship, and management responsibilities that are necessary to make research collaborations function well. None by itself can do everything that is required. The three must mesh to create the framework that yields a hospitable environment where research collaborations can flourish. Until recently, these matters were addressed ad hoc within a university or between a few handfuls of organizational partners. As the volume of research data, the number of investigators, and the diversity of organizations rises, one-to-one agreements become unwieldy to execute and enforce. There is a need and an opportunity to forge a consensus for common expectations around research data sharing.

We recommend organizations build on their existing policies and standards and extend them as necessary to address new collaboration challenges. Reconcile policies and practices to achieve a broad consensus whenever possible. A lack of consensus will slow data sharing, research collaboration, discovery, and the benefits that society expects to reap from them.

5.5 Emerging National Infrastructure to Support Federated Identity and Anticipated Impact on Research Organizations

We earlier noted that authentication credential providers must adhere to clear, well-defined policies and procedures when issuing and managing identity credentials. Organizations and individuals will only trust credentials from a provider when the provider explicitly demonstrates it has adhered to commonly agreed upon standards. Even then, additional legal requirements such as liability assumption and indemnification may be required before full trust is extended. Since there is not a "universal" single credential provider, as is the case in some countries, identity management federations are forming. Within such federations, members agree to adhere to specified open standards and/or openly published frameworks.

In December 2003, the Office of Management and Budget published OMB Memorandum M-04-04 entitled "E-Authentication Guidance for Federal Agencies." This document defines four levels of assurance, Levels 1 through 4, proportional to the consequences of the authentication errors and misuse of credentials. As the consequences of an authentication error becomes more serious, the required LOA increases. Thus, for example, Level 1 requires no identity vetting, whereas Levels 2 through 4 require progressively stronger vetting of identity,

stronger credential binding to the identified individual, as well as progressively stronger authentication tokens.

In April 2006, the National Institute of Standards and Technology published NIST Special Publication 800-63 version 1.02 entitled Electronic Authentication Guideline (Burr et al. 2006). This document provides specific technical requirements for tokens (typically a cryptographic key or password) for proving identity; identity proofing, registration, and delivery of credentials which bind an identity to a token; remote authentication mechanisms, that is the combination of credential, tokens, and authentication protocols used to establish that a claimant is in fact the subscriber he or she claims to be; and assertion mechanisms used to communicate the results of a remote authentication to other parties. Subsequently, the US Authentication Identity Federation provided an Authentication Credential Assessment Suite [http://www.idmanagement.gov/eauthentication/drilldown_ea.cfm?action=ea_credsuite] to be used in assessing compliance of a Credential Service Provider (CSP) and their Credential Services (CS) to E-Authentication Levels of Assurance. This suite consists of the Guide to Preparing for a Credential Assessment; Certificate Credential Assessment Profile, v2.0.0; Password Credential Assessment Profile, v2.0.0; Credential Assessment Framework, v2.0.0 and the Entropy Spreadsheet, v2.0.0.

The Password Credential Assessment Profile evaluates Level 1 and Level 2 authentication credentials, whereas the Certificate Credential Assessment Profile evaluates Level 3 and Level 4 credentials. The reason for the two profiles is that assurance Levels 1 and 2 apply to assertion-based credentials that use password tokens, for example, Shibboleth SAML assertions, whereas Levels 3 and 4 require cryptographic tokens.

In academic medicine, there is a movement toward adopting these federal standards. The development of the Shibboleth Authentication and Authorization infrastructure by the Interent2 Middleware Initiative makes the implementation of Level 1 and Level 2 authentication credentials relatively easy. The InCommon [http://www.incommonfederation.org/] identity management federation is the first nationwide US federation for higher education. It utilizes the Shibboleth infrastructure and it is attempting to adhere to the Level 1 and Level 2 assurance levels. The University of Texas System Identity Management Federation is an example, where multiple organizations have taken a common approach with success.

The National Institutes of Health announced on August 14, 2007 a Memorandum of Agreement (MOA) for interfederation with the US Higher Education's InCommon Identity Management Federation (InCommon). This announcement noted NIH's "goal is for researchers to use their institutional identity credentials to authenticate to NIH online applications and services. All NIH online [computer] applications have been assessed and assigned one of the four Federal Level of Assurance (LOA) identity requirements for authentication." Subsequently, both the NIH Cancer Biomedical Information Grid (caBIG®) and the NIH Clinical Translation Science Award (CTSA) Consortium have announced plans to comply with the same requirements.

We believe that the embracing of the InCommon Federation by the major scientific funding agencies in the USA, as well as a number of major science awards will

have a major impact on research organizations. These organizations will, over the next several years, find it convenient, if not necessary, to join the InCommon Federation. For this reason, we recommend that research organizations consider establishing a Shibboleth Identity Provider (IdP) to assert Level 1 and Level 2 authentication credentials for at least some of their workforce. In addition, they should consider enabling various information resources as Shibboleth-enabled Services Providers (SPs) that can accept Level 1 or Level 2 authentication credentials. It should also be noted that a number of institutions are requiring providers of commercial software applications to Shibboleth-enable their products. The most difficult task for an organization will be implementing an institutional IdP that meets the federal assurance standards for Levels 1 and 2. Joining an existing Identity Management Federation whose members are collectively working to meet the assurance requirements is likely to be easier than doing it all on one's own. As with any auditable process, it will be crucial to have all security-related policies and procedures documented. Creating or adopting a complete set of Standard Operating Procedures is a critical step in this process.

The NIH MOA with InCommon recognizes InCommon's Bronze Profile as a Level 1 LOA. InCommon's Silver Profile is designed to meet Level 2 requirements; however, not all policies and procedures have been agreed upon for the membership. Institutions that already have a solid institutional Identity Management System in place should be able to easily implement and integrate a Shibboleth into that infrastructure. If local resources do not permit a locally hosted IdM infrastructure, consider using a commercial Shibboleth credential service provider. This approach can also be used to credential individuals who are not institutional personnel. Some IdM federations are considering accepting commercial IdPs as federation members in recognition of these needs.

These recommendations notwithstanding, we again note the long-term advantages that accrue when third-party credentials are used. Despite the momentum that is growing for InCommon and other federations based on employer-issued credentials, we note that there may yet be a shift toward third-party credentials. If that happens, institutions that have made significant investments in federated systems that depend upon employer-provided credentials, and which cannot easily be retooled to support third-party credentials, may be faced with the need to rebuild their security infrastructure. For this reason, we suggest that, whenever possible, institutional systems be designed to evolve gracefully in the face of technical and social changes in the remote credentialing process.

5.6 Conclusion

We believe that we have presented a cogent argument for why application designers, particularly those building systems to support multi-institutional partnerships, should consider the developing or joining existing identity federations. Below we present a series of recommendations to assist individuals in developing a strategy relevant to their organizations.

5.6.1 Prepare for Cross-Institutional Workflow and Transactional Data Exchanges

Necessary steps to prepare for this process include developing a strategy for identity management that recognizes the growing importance of federations and extrainstitutional sources of authentication and authorization as part of the process. This strategy should closely examine the US Federal Government e-Authentication initiative and NIST standards. A central consideration is if the use of digital signatures is required, such as for medical records applications. Digital signatures require digital credentials that are based on a public key infrastructure and some form of cryptographic key management systems. These measures will likely become necessary for maintaining the security and privacy of medical records, clinical trials, or other documents. This will evolve from traditional username/password approaches that are commonly in use.

5.6.2 Build Directory-Aware Systems

From a technical roadmap standpoint, consider developing a list of core business and scientific applications that are "aware" of directory services (such as LDAP or Active-Directory). Develop a single point authentication service – at a minimum do this for all Web-based applications. New applications that are acquired should be required to be aware of directory services, and this should be treated as a high-priority functional requirement – that is, it should be in the deal-stopper category for systems acquisition. Examine existing applications to determine which key applications are the important SOA for key information involved in authentication, such as employee status. Many campuses have found that there is often more than one system involved in such decisions for different groups of individuals, such as employees, students, contractors, pool nurses, etc. Developing a list of these key applications and documenting the processes involved in provisioning systems with this information will allow these systems to be both better controlled and interfaced to control and provision the directory services in a more efficient manner.

5.6.3 Become Familiar with Shibboleth and InCommon

Incorporate Shibboleth capability into the campus identity management infrastructure. Nearly all emerging federations, including the InCommon Federation, the United Kingdom's JISC, NCI's caBIG® effort, and a number of European grid computing projects have Shibboleth technology at their core. Shibboleth is open-source software that emerged from the NSF Cyberinfrastructure project and has been developed by the Internet2 as part of that effort. Additional components

(Grouper, Signet) of that infrastructure project support management of authentication and authorization. A number of open-source collaboration platforms are well integrated with this software. Some of these are discussed in the first section of this report and are available at the Internet2 Web site. A growing number of enterprise software and other application systems are becoming Shibboleth aware as well, as the 117 InCommon member campuses (as of September 2009), six research agencies and laboratories (including the NIH), and 42 sponsored participants continue to promote adoption of this technology with major providers of commercial software to higher education.

Similarly, the underlying protocols for security are based on the use of SAML version 2.0. We advise system selection teams to use this as criteria when specifying and evaluating technology.

5.6.4 Adopt Standard Forms for Personal Attributes

Examples of attributes are employee status, date of hire, functional role, or any other piece of data deemed important in inter- or intrainstitutional authentication or authorization decisions. These and other attributes have been agreed on by InCommon and other organizations. They are defined in the InetOrgPerson and EduPerson schemas.

5.6.5 Issues of Incorporation and Governance

We suggest that institutions consider integrating with the Internet2 InCommon effort. InCommon is currently working with the NIH, the NSF, and a large number of university systems in the USA. Consequently it is evolving into a large identity federation geared toward academia. By joining an existing identity federation, institutions will enjoy the benefits of a common incorporation framework, common policies, common procedures, and common auditing standards. Many of the existing identity federations are cross-certifying with the Federal Bridge Certificate Authority, which will ultimately allow members of each federation to cross-certify with other federations at specific levels of assurance.

A related issue involves how individuals with legitimate need could gain access to a federation's resources if they are not already affiliated with one of its members. Some individuals may have valid identity credentials available for use from government or other sources acceptable to the federation. InCommon, SAFE-BioPharma, and others already have agreements and infrastructure in place to allow this. The more difficult issue is how to issue identity credentials to individuals unaffiliated with institutions. One approach may be to contract with an outside provider of identity credentials.

A further, more detailed treatment of the topic of governance in federations involving academic medical centers can be found in Manion et al. (2009) and Weems et al. (2007).

5.6.6 *Identification of Missing Technical and Infrastructure Components Required to Support Academic Medicine and Research*

Current authentication and authorization efforts, whether conducted locally or between institutions, have yet to develop all facets of academic medicine. Chadwick (2006) has shown in clinical contexts that authorization decisions may require a computerized application to consult multiple SOAs. Some of those SOAs do not yet exist or have no standardized form for such information. Further research on these topics and leadership to develop these components through national infrastructure efforts are needed.

Other areas that require effort will be the development of a variety of taxonomies and ontologies, which include:

- Common forms for institutional names, including their constituent parts.
- The development of a full spectrum of high-level use cases, analysis to common abstractions, and development of specific role definitions (or attributes) needed for nonclinical uses. Note that Health Level 7 (HL7) and the American Society for Testing and Materials (ASTM) have already done substantive work on this topic in the clinical area that could be extended into the translational and basic science research domains.
- The development of object class definitions incorporating the roles and attributes suitable for use by directory services and interfederations.

The development of these areas is outside the scope of what a single institution can accomplish. There remains an ongoing need for concerted, sustained effort by standards organizations, government, and other stakeholders in this area. A further, detailed treatment of the technical implications of governance and infrastructure needs can be found in Robbins et al. (2007).

5.6.7 *Plan for Change*

The development of federated authentication and authorizations systems is a dynamic, rapidly evolving area. As we have pointed out above, much current work has been based upon extensions to institutional security systems, including the use of employer-provided credentials. However, the strong technical and theoretical advantages to third-party credentials may yet lead to significant changes in the technology of federated authentication and authorization. When making investments in these technologies, institutions are advised to develop systems that will be forgiving if significant changes occur in the underlying technologies and processes.

References

Burr WE, Dodson DF, Polk WT (2006) NIST Special Publication 800-63 Version 1.0.2. Electronic authentication guideline. US Department of Commerce, National Institute of Standards and Technology. http://csrc.nist.gov/publications/nistpubs/800-63/SP800-63V1_0_2.pdf. Accessed 28 September 2009

Chadwick DW (2006) Authorisation using attributes from multiple authorities. In: Proceedings of the Fifteenth IEEE international workshops on enabling technologies: infrastructure for collaborative enterprises, pp 326–331. doi:10.1109/WETICE.2006.22

Fishbein EA (1991) Ownership of research data. Acad Med 66(3):129–133

Foster I (2006) Globus toolkit version 4: software for service-oriented systems**.** In: IFIP international conference on network and parallel computing, LNCS 3779. Springer, Berlin, pp 2–13

Langella S, Oster S, Hastings S, Siebenlist F, Phillips J, Ervin D, Permar J, Kurc T, Saltz J (2007) The Cancer Biomedical Informatics Grid (caBIG) security infrastructure. AMIA Annual Symp Proc 2007:433–437

Loshin D (2002) Knowledge integrity: data ownership (Online) June 8, 2004. http://www.datawarehouse.com/article/?articleid=3052. Accessed March 2009

Manion FJ, Robbins RJ, Weems WA, Crowley RS (2009) Security and privacy requirements for a multi-institutional cancer research data grid: an interview-based study. BMC Med Inform Decis Mak 9:31. doi:10.1186/1472-6947-9-31

Robbins RJ, Crowley R, Weems WA, Whitney D, Ransom M, Mathew G, Olivastro D, Chisti A, Manion FJ (2007) Technical implications generated by requirements discovered in caBIG™ security, privacy, and IRB interviews. Available at http://gforge.nci.nih.gov/frs/download.php/1972/DSIC_Security_Deliverable_6.pdf

Weems WA, Robbins, RJ, Whitney D, Crowley R, Manion FJ (2007) caBIG™ Major governance and policy areas. Available at http://gforge.nci.nih.gov/frs/download.php/1975/DSIC_Security_Deliverable_13.pdf

Chapter 6
Genomics Data Analysis Pipelines

Michael F. Ochs

Abstract Data size and flow are rapidly increasing in cancer research, as high-throughput technologies are developed for each molecular type present in the cell, from DNA sequences through metabolite levels. In order to maximize the value of this data, it must be analyzed in a consistent, reproducible manner, which requires the processing of terabytes of data through preprocessing (normalization, registration, QC/QA), annotation (pathways, linking of data across molecular domains), and analysis (statistical tests, computational learning techniques). The demands on data processing are, therefore, enormous in terms of computational power, data storage, and data flow. In this chapter, we address some of the issues faced when developing a data analysis pipeline for this high-dimensional, high-volume data. We focus on a number of best practices important for the implementation of the pipeline, including use of software design patterns, tiered storage architectures, ontologies, and links to metadata in national repositories.

6.1 Introduction

With the introduction of GenBank in the 1980s (Burks et al. 1985), the age of large data sets began in biological research. The development of automated sequencing technologies (Hood et al. 1987) and the initiation of the Human Genome Project (Watson 1990) led to exponential growth in the number and length of sequences stored in GenBank, a trend which continues to this day. The data volume quickly overwhelmed traditional analysis techniques, leading to the development of a heuristic algorithm capable of identifying similarities in sequences within large data sets (Altschul et al. 1990). The combination of large data resources and analysis tools capable of identifying patterns in those data revolutionized biological research.

M.F. Ochs (✉)
Division of Oncology Biostatistics and Bioinformatics, Johns Hopkins University, 550 North Broadway, Suite 1103, Baltimore, MD 21205, USA
e-mail: mfo@jhu.edu

M.F. Ochs et al. (eds.), *Biomedical Informatics for Cancer Research*,
DOI 10.1007/978-1-4419-5714-6_6, © Springer Science+Business Media, LLC 2010

A similar process in functional genomics began in the 1990s with the emergence of microarrays (Schena et al. 1995; Lockhart et al. 1996). The development of this technology relied on sequencing, as thousands of genes could be probed for expression, because their sequences were known and stored in GenBank. With the completion of the human genome project, arrays could be refined to match consensus sequences and not just expressed sequence tags (ESTs), and specific sequences could be designed to probe expression of each gene to minimize cross-hybridization with sequences from other genes. This has been followed by the development of exon-level arrays designed to determine alternative splicing in expressed genes as well.

The list of genome-wide, high-throughput data types continues to grow. Routine measurements are now made on single nucleotide polymorphisms (SNPs), methylation states of the DNA, miRNA levels, protein levels, and metabolite levels. While protein and metabolite levels remain limited to thousands of species, SNP-chips are now available that measure a million SNPs on a single chip.

Unfortunately, the analysis required to recover useful knowledge from functional data, such as from microarrays and proteomic measurements, and from SNP data is more complicated than for sequencing data. Not only is there the well-known "curse of dimensionality" due to millions of variables being measured on only thousands of individuals, but there are also issues that arise due to the need to preprocess the data to remove artifacts.

This chapter looks at the computational issues that arise in running analyses of new and emerging high-throughput molecular data. Such data will require the development of pipelines linking many algorithms that take raw measurements and produce meaningful information useful to clinicians and biologists. We focus on six issues: data volume, emerging data types, evolving algorithms, computational throughput, summarization and visualization, and interactive analysis. These issues drive the requirements for a data pipeline, which we describe in detail and then summarize in the conclusion.

This chapter is focused on the overall architectural issues and approaches for a data analysis pipeline. It does not address best practices in software development, but these should not be overlooked. It is critical that development of the pipeline include the standard testing and user feedback, so that the final pipeline fits the needs of the users, cancer researchers. It is equally essential that the design incorporate best practices of software design as well, since a useable system that does not match the computational, statistical, and maintenance needs is equally useless to a system that cannot be navigated by researchers and, potentially, more damaging to the overall cancer research effort.

The final pipeline will form part of the overall information systems for a cancer research center. As discussed in a recent review and summarized in Fig. 6.1, reproduced from that paper (Ochs and Casagrande 2008), a computational pipeline interacts with data systems that handle clinical trial information, medical records, high-throughput data systems, and potentially an internal data warehouse. In addition, it must interact with external data and annotation repositories and provide access to users, potentially through multiple technologies.

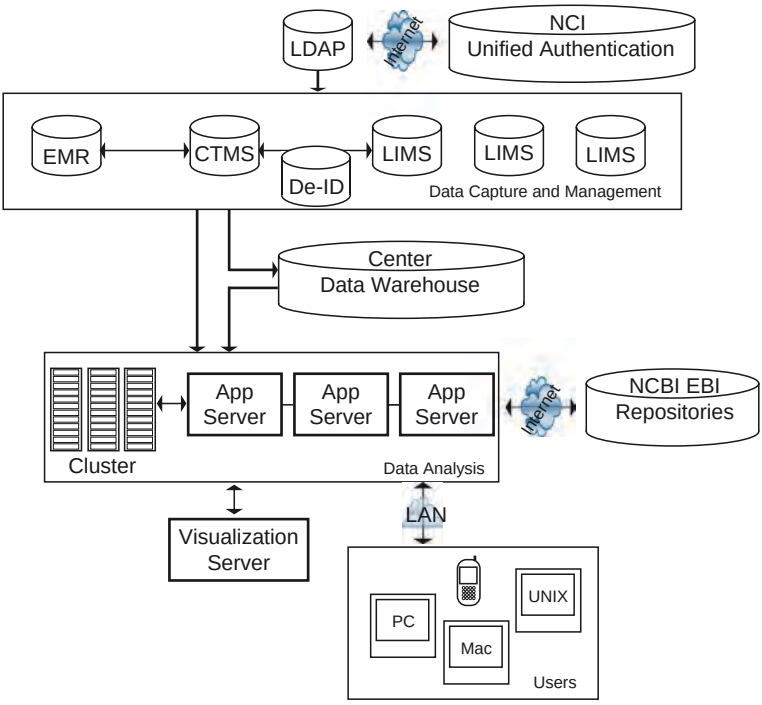


Fig. 6.1 One potential approach to implementing a data analysis pipeline within a research information system is shown, reproduced from (Ochs and Casagrande 2008). The pipeline would obtain data from a data warehouse or directly from research systems, it would obtain annotations and potentially other data from national resources, and it would analyze this data within a cluster. In this vision, visualization is handled by separate specialized computational resources, while in the text we assume closer integration in a pipeline. Both approaches have advantages

6.2 Data Volume

6.2.1 The Importance of Context

The first issue that arises with the new high-throughput biological data is that the volume of data far exceeds that seen even in the sequence databases. While the genome sequence is considered stable, even though in cancer this is true only for the germ line, the new data types are all strongly context dependent. The methylation status, mRNA transcript abundance, miRNA abundance, protein levels and states, and metabolite concentrations vary by cell type, time of measurement, environment, and other contexts even in perfectly healthy individuals. As such, the database must track context, and data for different contexts must be captured and stored, together with appropriate metadata. While this raises important issues on annotation (Ball and Brazma 2006; Whetzel et al. 2006), in this section we focus on the problem merely in data volume and corresponding data transfer for analysis.

There are different approaches for bringing together the data for analysis. A traditional data warehouse (see Chap. 3) can be created that brings together a subset of the total worldwide available data. Alternatively, data federation can be used to bring together data as needed for analysis, as in a grid architecture (see Chap. 4). In either case, a large database of either permanent or temporary nature must be established to gather the data and metadata. This database needs to be accessible by the data pipeline.

The need to bring together large data sets and to transfer this data to systems for analysis drives a need for very large data pipes. The connections to both internal systems and external systems must support high-speed and high-volume data transport, as even in cases of data warehouses, large amounts of information on annotations for the data will need to be routinely retrieved from national and international repositories. For data federation schemes, the data will need to flow consistently from remote systems into the analysis pipeline.

6.2.2 Data Standards

In order to bring the data into a single set for analysis, standards need to be utilized to allow search, retrieval, and integration of the data. Searches will require identification of the appropriate data sets that are associated with the topic under study and retrieval of the specific data elements in the sets that can be integrated together for the analysis. Identification of data sets will require ontologies and controlled vocabularies related to the phenotypes studied in the original experiment, including but certainly not limited to disease states, cell types, therapeutics, and protocols. Bringing together the different types of data will require establishing relationships between DNA locations (SNPs, methylation), RNA transcripts (microarrays, exon arrays, miRNA arrays), proteins and their isoforms (protein levels, proteins from alternatively spliced mRNAs, post-translationally modified proteins), metabolites (products of protein-driven enzymatic reactions), and therapeutics (small molecule therapeutics targeted at specific proteins, antibodies, etc.). The list of needed relations between data types continues to grow as well, and this will become significantly greater with modeling efforts.

Fortunately a number of efforts are underway to establish standards for both data elements and metadata. For clinical, diagnostic, phenotypic, and anatomic data, the Unified Medical Language System (UMLS) provides a single point of integration for numerous vocabularies that are now in use worldwide (Humphreys and Lindberg 1993). Work continues in development of text mining methods that can extract data structured with UMLS terminologies from the large medical literature, so the volume of annotated data continues to grow. Importantly, the UMLS also provides mappings between the vocabularies, so that data encoded with one vocabulary can be integrated with data encoded using a different vocabulary.

For molecular data, the ontologies are less mature but are being developed rapidly. The Open Biomedical Ontology (OBO) repository hosts well over a hundred

ontologies focused on molecular entities, cell types, model organism anatomy, pathways, phenotypes, and experimental protocols, to name but a few (Rubin et al. 2006). Among the most widely used is the Gene Ontology, which provides gene-based annotations for molecular function, biological process, and cellular location (Ashburner et al. 2000). This ontology has been widely used in predicting functions for unknown genes and for interpreting the results of high-throughput biological experiments. We discuss the encoding of data using ontologies in Sect. 6.3.3.

6.2.3 Pipeline Data Storage

Overall, the data volume is driven by both the primary data and the metadata describing it. Without this metadata, it is impossible to construct a meaningful analysis, making a data pipeline useless. The pipeline must be capable of connecting the data across these different data types in a meaningful and scalable way. The pipeline must handle the data volume flowing into it, perform analyses on this data based on the metadata, and present results to the user. Because most data requires extensive preprocessing to remove artifacts, it is also useful to include infrastructure to capture intermediate stages in the analysis, since the intermediate stage data are often reused in different analyses. This is an exchange of increased storage requirements against reduced computation time. Long-term storage of intermediate processed results may also be advantageous for publication and presentation of data to the community (see, for example, Chap. 8 on Reproducible Research).

The data storage needs for the analysis pipeline itself should, therefore, be comprised of large volume, short-term, maximum speed storage for data actively being processed; smaller volume, medium-term, fast storage for intermediate results that may need to be accessed for a new analysis or retrieved for archival storage; and potentially large, slower storage for results that users wish to retain indefinitely. This suggests a tiered architecture, with fiber-channel or similar drives connected to the pipeline computational core, RAID disks for medium-term storage, and archival disks or tapes for long-term storage. These tiered storage systems can be built in-house or purchased with associated retrieval and archiving software simplifying development of analysis tools.

6.3 Emerging Data Types

6.3.1 Data Modeling

The rapid advances in technologies for measuring molecular components of the cells have led to the explosion in data noted in Sect. 6.2. However, another consequence of this growth has been a rapid rate of change in the types of data that must

be captured, annotated, integrated, and stored. This has a direct impact on the development of statistical and data mining tools (see Sect. 6.4) and also on the data schema and metadata requirements for data management (see Chaps. 2 and 3). The traditional approach to data modeling, where all the data elements are identified, a schema constructed for a database to house these elements, and that schema normalized to improve stability and efficiency, faces a major difficulty when the data types change rapidly and when such changes cannot be predicted.

The rapid change in data types directly impacts the construction of data pipelines as well. A logical first step for any pipeline is data retrieval, typically by an SQL query of a database and population of data structures for analysis. However, the schema of the database will potentially be atypical, as it will need to be designed to address rapid changes in data types. The analysis steps built into the pipeline will face similar difficulties, as new data types will require both modification of variable structures in the code and development of novel algorithms. This places a requirement on the pipeline for great flexibility in the interface to the databases and in the establishment of the infrastructure to handle ongoing modifications of core code components. If a system is not designed initially with an understanding of this issue, maintenance can become overwhelming and the pipeline will quickly become either obsolete or highly costly to maintain. This raises a concern with the typical academic approach to data analysis tools, which aim to rapidly provide a simple tool to the community and then add to it. A spiral design approach such as this generally does not produce a tool that can be maintained in a cost-effective manner. However, the requirement in grant funding of an established user base encourages rapid deployment at the cost of careful design.

6.3.2 Object Oriented Design and Problems of Encapsulation

The constant need to update data types to reflect technical developments creates a significant problem for traditional object oriented design (OOD). An ideal OOD encapsulates the data together with the operations (i.e., methods) applied to this data. For example, one could imagine a microarray class that had matrix variables of mean values and error estimates on those values of the transcript levels of all genes across a series of measurements. Operations could include clustering of the data, statistical analysis, and output routines. This class might be inherited by classes for specific array technologies, such as Agilent, Illumina, and Affymetrix, to provide platform-dependent preprocessing of probe level data and conversion to gene level data. However, if we now add a data type that has emerged and affects gene expression, such as methylation, we may wish to define a gene as including upstream methylation status, since this could be needed by some microarray analysis algorithms. Thus, every class that requires this data must now include the new gene definition and data, requiring modification of many classes with the development of each new measurement technology. For microarrays, methylation and SNP data extend the “gene” beyond the transcript already.

One approach to handling this issue is to use design patterns that have been developed to address specific issues that arise repeatedly in software design (Gamma et al. 1995). One pattern, the Visitor pattern, is particularly applicable in this case, and it functions by splitting a typical OOD class into multiple classes, one comprising the data and the others comprising methods. The data is passed to the method classes by having a data class that contains only a single method (acceptVisitor), which passes the full data object to the Visitor (i.e., a class defining a specific method). Armed with the full data class, the Visitor is then able to use the necessary elements and update the class, including writing to a history file stored within the class. Updating classes with new data types now has no impact on the method classes, which are able to retrieve the data they need. This also simplifies addition of methods, which we discuss in Sect. 6.4.

Coupling the Visitor pattern to other features of OOD is particularly powerful. A single container class can be populated with many different data types through composition (inclusion of a class within another class). These data classes can all inherit from the root abstract data class, and thus guarantee implementation of the acceptVisitor() method. In addition, each data class can implement the same interface, which handles linking of different molecular types through annotations, providing a single point that defines all data integration. The container class is passed to the Visitor, and a method class can make use of any and all the data types as well as know how to integrate them. This structure was the basis of the Functional Genomics Data Pipeline prototype (Grant et al. 2004), and a portion of the class structure is shown in Fig. 6.2. Here, we expand on the class of proteomics, initially focusing only on mass spectrometry. The other classes in the first proposed version of the pipeline are also shown at the top right. The microarray data classes were fully implemented.

6.3.3 Ontologies and Controlled Vocabularies

One danger with this data structure is the possibility of "siloed data," with the silos here being data types instead of an individual research data system. The initial development of a pipeline may involve a single data type (e.g., microarrays), and all tools for analysis and visualization may assume a data structure useful for microarrays. Since it is very easy for a programmer working against a deadline to hardcode assumptions into the core of a system, it is important to design the structures early and enforce best practices against the design.

One best practice that can be implemented early and virtually eliminates the danger of data silos is the use of ontologies and controlled vocabularies for data elements. As noted in Sect. 6.2.2, ontologies provide a series of defined terms for describing data elements and relationships between these terms (Rubin et al. 2006). Inclusion of ontologies is an expensive prospect, but it has enormous rewards in terms of the ability to integrate large data sets and an ability to maintain the value of data in the future. Realistically there is a trade-off that is not supported well by

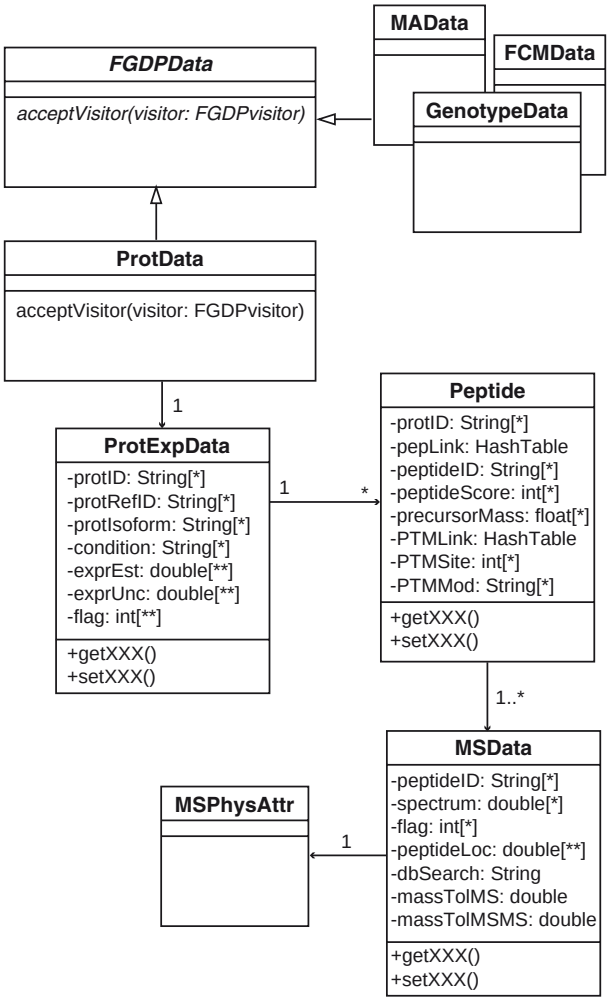


Fig. 6.2 Example class structures using object oriented programming and design patterns to introduce flexiblity into a data pipeline. Here a proteomics class is shown, which inherits from an abstract data class that insures implementation of the Visitor design pattern. An interface, FGDPLink (not shown), would be implemented by all data classes and would provide a single point for annotations and methods for integrating data during analysis. Examples of data integration approaches coded in the interface might include simple referencing back to a genomic location to combine SNP, transcriptomic, and proteomic data, or complex models linking cell signaling, gene expression, and metabolic flux

the present grant and rewards system, as use of an ontology does not generally improve the success of a small focused study, but it does add to its cost. In a scientific version of the "problem of the commons," there is little incentive for an individual researcher to make the effort to use ontologies or controlled vocabularies as

it benefits the whole research community (and likely also society) but also introduces added costs to the individual.

Assuming that the cost can be addressed, ontologies, or even merely controlled vocabularies, can enter the design process early and guide the architect by highlighting what data types are generally captured, how they relate to each other, and often what community is likely to generate the data types. This can aid in understanding and modeling the system dependencies during the design process, leading to improved efficiency and an improved likelihood that the system will avoid a costly redesign as the project moves forward.

6.4 Evolving Algorithms

6.4.1 Data Driving Algorithm Development

While data types being added create one issue, methods are created even more frequently. This is most obvious presently in microarray data analysis, where there are numerous clustering methods, statistical methods, and data mining methods already developed and others still being reported monthly (see Chap. 15 for a desktop tool that incorporates numerous methods). It is clear that there is great interest in improving analysis and that methods will continue to be developed, some of which are likely to be of interest to the users of a specific data analysis pipeline.

In addition, as new data types emerge with evolving technology, many algorithms for preprocessing and analyzing these data will be developed in the statistics and data mining communities. As these communities develop deeper understanding of issues such as correlated noise and biological complexity that underlie the data, the methods will improve and evolve. With a desire to analyze this data more globally, additional methods with potentially dramatic computational requirements are beginning to be developed.

6.4.2 Application Programming Interfaces

Even with a live system capable of accepting new analysis modules, the modules themselves must be carefully constructed for interchangeability. This is traditionally done through the use of well-defined application programming interfaces (APIs), which define the inputs and outputs of a method. This is a critical issue in software design. The need to reformat data to utilize a new method cannot be maintained with any scale, and it is often overlooked within the field of bioinformatics, where researchers routinely publish new methods that do not conform to any standards. One reason for the immense success of the R/Bioconductor system is the standardization of data structures and interfaces that permit thousands of separate programmers to create interoperable statistical tools (Gentleman et al. 2004).

As an example, consider again the analysis of microarray data that has now been preprocessed into matrices of mean values and error estimates on those values. An analysis module would accept the two matrices as inputs and return the results of the analysis, such as groups of genes that behave similarly based on their transcript levels across conditions in the experiment. Many such algorithms exist, and each would then be coded within a module. These modules would then be interchangeable within the analysis as desired by the user, if the inputs and outputs were forced to match (i.e., to have a standard API). This also makes clear why retaining intermediate steps in the analysis (such as data preprocessing) is so valuable, as without retention of this data the preprocessing would need to be redone whenever the researcher desires to repeat an analysis of the two matrices with a different algorithm.

In object oriented systems, inheritance can be used to establish relationships between analysis techniques and enforce a standard API. For instance, a parent, abstract clustering class could be created with a method "cluster" that takes as input the two matrices and returns a list of sets of genes. Each clustering method would then be coded in a concrete class whose parent is this abstract class. In addition, OOD allows for an additional feature through polymorphism, in that the method "cluster" could have two forms, one taking a single matrix for methods where no error measurements are used, and one taking two matrices for those where error measurements add value. The system can then handle both cases seamlessly at run-time.

6.4.3 Updating Live Analysis Pipelines

The rapid rate of development of new analysis programs raises an additional problem for maintaining a data pipeline. It is often desirable to add methods to a live system, since analyses could be ongoing for considerable lengths of time and initiated by different researchers with different schedules. Often a useful algorithm will be needed that is not yet implemented in a system, and taking the system off-line to add this may be impractical. One approach that has proven useful is to rely on reflection mechanisms, such as in Java and C#, where classes can be queried for their variables and methods at run-time. This allows a running pipeline to discover new method classes and deduce their proper place in the pipeline from their inheritance structure and methods. When properly implemented, new analysis choices can be presented to the user immediately following their inclusion in the pipeline. One example of this is the use of a Web interface with dynamically created menus.

Adding methods to traditional OOD approaches is easier than modifying data types, however using the Visitor pattern can simplify this further. A Visitor class serves as an abstract class defining the "doOperation" method, which is called automatically after a data class calls "acceptVisitor." The concrete Visitor is created for each method, and it implements the algorithm, retrieving the necessary data from the class, integrating it based on the interface that ideally has encoded the ontologies allowing complex relationships and controlled vocabularies to be used, and returns the results by modifying the data class appropriately.

6.4.4 Alternative Approaches

While this section and the last describe one potential approach to the creation and maintenance of a data analysis pipeline given the rapid data and algorithm changes, other approaches for solving these same problems are in use. The best known is a Web service or service-oriented architecture (Komatsoulis et al. 2007), which can be implemented within a workflow approach, such as Taverna (Oinn et al. 2004). In this case, each analysis can be hosted by a different system but with well-defined APIs. Servers publish their availability to perform analysis functions, and analysis pipelines interact with these servers, passing them the data and receiving the results, in order to complete a workflow. This reduces the need for computational throughput within the system hosting the analysis pipeline itself, but it increases the potential requirements on data flow, since data must move repeatedly to different systems. This approach underlies grid (and cloud) computing (see Chap. 4).

6.5 Computational Throughput

6.5.1 Computational Complexity

The integration in terms of phenotypic behaviors of the vast, diverse sets of molecular data emerging in cancer research promises to allow deeper questions to be asked about cancer etiology and focused treatment. However, since the data emerge from complex biological behaviors involving cellular systems, cell–cell interactions, responses to distal signals such as hormones, and complex interactions with treatment regimens, including chemotherapy and radiation, significant issues arise for both statistical models and computational modeling (see Chap. 7). The result is a high computational cost for many algorithms that seek to uncover meaningful relationships within the data that can provide answers to questions posed by prospective studies (e.g., treatment effectiveness, response of biological subsystems, etc.) and generate new hypotheses in discovery studies.

A data analysis pipeline must include a design that optimizes computational efficiency given a desire to both apply a series of algorithms in a linear fashion and permit parallel application of algorithms to explore the data by multiple methods. Some of these algorithms may be individually computationally expensive, so that the system must be designed to allow asynchronous operation and to potentially provide researchers with intermediate results while operations are still ongoing. The pipeline must handle both large data flows with associated algorithms and highly computational algorithms potentially running for many days on relatively small data sets. These diverse requirements suggest a need for different types and designs of systems that nevertheless work together to process data.

6.5.2 Beowulf Clusters and Grids

One of the standard approaches to such problems is a computer cluster that potentially has nodes devoted to different types of operations. Some of these nodes may be specialized to high-speed data transfer through optical connect backplanes and include large core memories to minimize the need to move data in and out of memory. Other nodes may include lower cost backplanes and smaller core memory, while maximizing processor speed and internal threading of code for multicore processors. Naturally, there will be a strong overlap in the capabilities and software design between these different nodes, and the choice reflects cost-benefit as much as hardware design. The traditional Beowulf Linux cluster, named for the ability of the eponymous hero of the Old English poem to slay the huge monster Grendel (i.e., mainframe computers), links a virtually unlimited number of computational nodes together. These nodes require some specialized coding to maximize their ability to work together for most problems, as the nodes must communicate status and exchange information for calculations. However, for data analysis pipelines, the code can sometimes be "embarrassingly" parallel, with the most obvious example being applying two clustering algorithms separately to a single data set. This process clearly can be run on separate nodes without any special coding. However, this is not typical, and coding will need to be specialized for parallel processing in many cases.

Embarrassingly parallel computing does open another approach to computational throughput, as the computers need not be colocated; so Web services or grid computing can be used. Perhaps the developers of the grid missed an opportunity not naming their new approach Dragon, after the noneponymous slayer of Beowulf. The grid approach divides the computational tasks into a sequence of steps with parallel branches, and sends the data to the registered computational engines for analysis. This permits highly specialized hardware tuned to the specific analytical approach, and this could be more efficient than general purpose nodes. This approach is not uncommon in physics and astronomy, where specialized hardware has long been designed to complete a single computationally expensive task (Rogers et al. 1983).

6.5.3 Data Persistence

The need to spread computations between different nodes and potentially different, geographically dispersed computers raises issues for data persistence. With many different processing steps being applied by the analysis pipeline to the data, intermediate processed results must not be discarded prior to their final use. Some intermediate steps, such as a fully preprocessed microarray data set, may be stored indefinitely, but others, such as the normalized values on a single array, may instead be recalculated later from the stored history of operations and their parameters.

Nevertheless, it may be necessary to retain these values for a relatively long time compared to typical computational time. Although this is not presently dealt with in most systems, it will become critical as data sizes and computational processing times increase. As with other issues of design, the failure to address this issue initially could lead to costly redevelopment needs later.

The approaches that could be used to address data persistence differ depending on the architecture of the pipeline. If the pipeline is maintained within a single computer cluster, then a local solution is the easiest to implement. In fact, with a tiered storage architecture, the persistence could be handled with a set of rules on data aging. For example, results from intermediate processing steps could be tagged as to whether they should be archived or not. Nonarchived data would have persistence rules guaranteeing its availability through the completion of an analysis, while archived data would persist indefinitely. For grid computing approaches or Web services, a number of approaches could be used. The simplest would be to require the pipeline to handle all data persistence requirements, accepting results from computational engines and tagging them appropriately. Obviously, a tiered architecture here would mirror that used for a cluster approach. However, an interesting alternative is to use a geographically distributed storage system that potentially could optimize the location of data storage based on the expected future use of the data. Intermediate results could then be piped to a storage server near the computational engine for the next operations, or even mirrored to be near systems for multiple next steps. A distributed storage system, TRANCHE, is already in use for proteomics data (Falkner et al. 2006).

6.6 Summarization and Visualization

6.6.1 Data Complexity

Upon completion of a pipeline run, the data would have been processed through a number of algorithms with the goal of providing insight into the biological and clinical behaviors of the cancer. However, unlike a well-proven clinical test, in which the results can be simply presented as a single number lying either within or outside a range of normal variation, the results of an analysis of high-throughput data is itself complex. It is generally not true that a single plot or set of values can summarize the information content of an analysis. Instead, a series of algorithms for summarization and visualization will be needed, and these may form part of the pipeline itself. At a minimum, the pipeline will need to be aware of the needs of these algorithms, so that the results can be presented with appropriate values.

The complexity of the data reflects both the unprecedented breadth in terms of measuring the molecular components of cells and the underlying complexity of the system giving rise to the data. The breadth is reflected in the overwhelming numbers of measurements, such as the ability to provide transcript levels for over 20,000 genes.

Obviously, just presenting 20,000 rows of data, be it as a figure such as a heatmap or as a table, overwhelms our standard methods of presentation. However, perhaps more difficult to handle is the fact that all measurements of the system are highly and complexly correlated. This reflects the fact that the biological processes underlying the data involve interactions between genes, proteins, metabolites, and all their derivatives. Our typical efficient statistical methods are almost always based on the concept of independence of measurement, and the graphical presentations of the data often derive from these methods. This impacts a data pipeline in two ways. First, the computational complexity increases as methods that can handle complex covariance are implemented. Second, the summarization and visualization modules must maintain the same flexibility as the rest of the pipeline, since methods are under active development for visualization.

6.6.2 Summary Statistics and Plots

The simplest presentation of results from the pipeline is likely to come in the form of standard statistical summaries. Five number summaries provide a quick description of the distribution of the data, the include the maximum, the minimum, the first (lower) quartile upper-bound, the third quartile upper-bound, and the median. Such numbers are particularly useful for looking at collections of raw data. For instance, the problems with microarray overall intensity varying strongly between different arrays independent of transcript levels is easily seen in five number summaries of replicated measurements. Other useful summary statistics can include mean and standard deviations from data, variance and covariance measures, and additional statistics that highlight unexpected correlations within the data, which can indicate artifacts (e.g., batch effects).

Simple statistical measures can be exploited for plots to visually verify normalization and other relatively simple procedures. These can be produced using open-source software as in Fig. 6.3, where R routines for boxplots are used to generate pre- and post-normalization graphs of the five number summaries on arrays for an Affymetrix experiment. The advantages of specialized graphical routines linked into the pipeline algorithms is that they can provide excellent feedback showing successful operation or helping to identify problems. Here it is clear that the arrays are not on the same scale prior to normalization, while following normalization using gcRMA (Irizarry et al. 2003) they now have the same medians and quartile ranges. Presenting such graphics provides a way to quickly summarize the results of pipeline operations and to allow the user to confirm that intermediate or final steps functioned as expected. An example of the power of these graphical summaries can be seen in the RReportGenerator application (Raffelsberger et al. 2008), which provides a simple interface to the R/Sweave reproducible research package (see Chap. 8).

The issue for pipeline construction is how to best integrate summaries and graphics within the pipeline. This will depend to a great extent on the architecture used for implementing the pipeline and providing a user interface. In this section

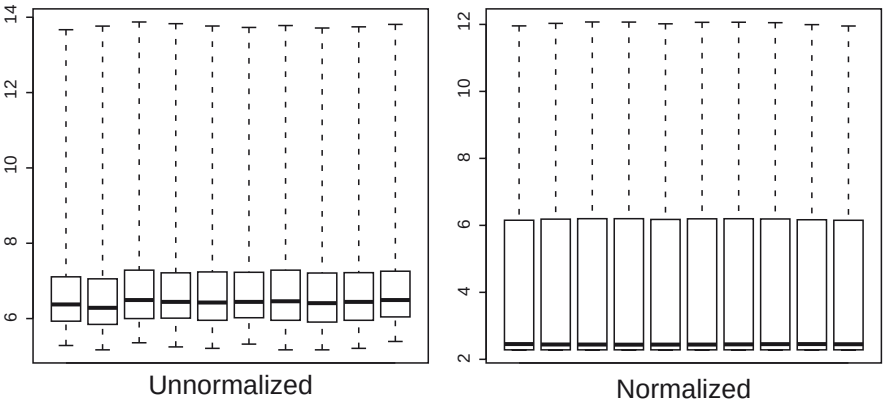


Fig. 6.3 A simple visualization that a pipeline could easily provide to give rapid feedback to investigators. In the example, microarray data is shown for a few Affymetrix GeneChips™ before and after normalization. Boxplots show both that no array showed outlandish variation before normalization and that all arrays have the same distribution following normalization. Such simple visualizations are easy to implement and provide valuable information to researchers

we focus on non-interactive reports and graphics, while in Sect. 6.7 we will address interactive graphics.

The typical user interface now has either a graphical user interface (GUI) such as provided by desktop software on Windows, Macintosh, or Linux (i.e., KDE). In a client-server environment this may be replaced by X, however the overall presentation is the same, with multiple windows presented to the user, some of which contain text and others which contain graphics. The windows are presented within a "desktop," which may have links to programs and data resources in it as well. The pipeline could then present results through text and graphics within these windows.

Presently most biological researchers obtain the results of numerically intensive computing through a Web interface. While a Web interface makes maintaining software easier, since upgrades occur only on a Web server and do not require upgrading all desktops, it can complicate the presentation of results to the user. Web interfaces remain more limited than desktop interfaces, and presentation must be done carefully to avoid problems with unexpected text sizes or browser window size. Results will generally be presented as text within a tabular format, or as image files (usually in JPEG format) displayed in the browser window. While programmers have exquisite control over a desktop window, its size, its shape, and its fonts, there is only limited control through a Web interface. The result can be poorly presented tables or graphics. Since browsers are typically geared to lower resolution tasks, it is also useful for the pipeline to be able to produce graphics for both Web presentation and publication. This minimizes data transfer requirements when presenting results in a browser at 72 or 100 dots per inch (dpi), while still supporting 600 or higher dpi graphics for publications. Typically, the high-resolution graphics would not be routinely produced, but would be created on demand when needed for publication.

6.6.3 *Biologically Motivated Visualization*

The difficulty of visualizing the overwhelming complexity of the data can be reduced by overlaying the data on biologically motivated images. For instance, changes in protein activity levels determined in an immunoblotting experiment can be visualized in terms of biological pathways, such as signaling pathways curated in the Kyoto Encyclopedia of Genes and Genomes (KEGG) (Kanehisa et al. 2002). In Fig. 6.4, constructed using the online KEGG-based visualization tool (Arakawa 2005), the BCR-ABL fusion protein, STAT5, and MYC are shown with higher activity (green on screen and shown with up arrows), while p53 is shown with decreased activity (red on screen and shown with a down arrow). This is a descriptive version of gene set analysis, which attempts to perform similar analyses through calculation of a statistical measure (Subramanian et al. 2005).

This simple approach will need to be expanded significantly within any pipeline that handles multiple data types, since the visualization should make clear the source (i.e., type of molecule) of the information. For instance, it is now clear that different types of molecular changes within signaling pathways can lead to similar cancer phenotypes (Parsons et al. 2008), and these differences should be reflected in the graphics. In addition, figures such as shown in Fig. 6.4 do not reflect biological structure nor standard modifications. The state of a protein in terms of post-translational modifications can play a crucial role, while its location (e.g., membrane-bound, nuclear, cytosolic) can change its function. Even for DNA, there can be multiple types of epigenetic modifications that could need to be visualized beyond simple methylation status. It should be clear that visualization of data is a complex problem, and it exists both as its own discipline and as a subfield within many areas. Cancer research has been slow to address this problem outside of medical imaging.

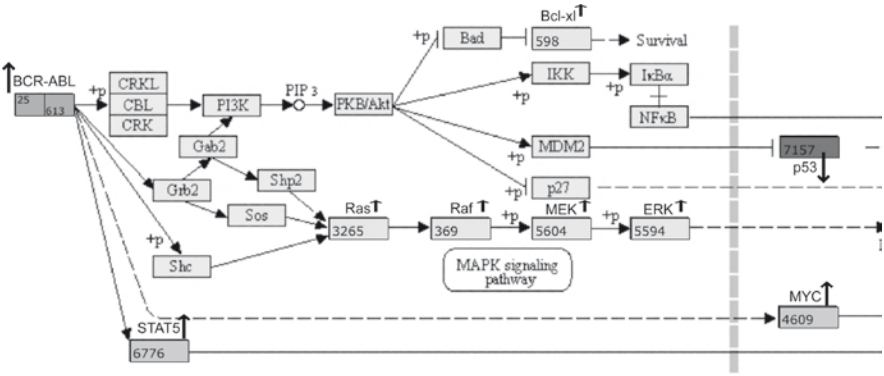


Fig. 6.4 A model based visualization such as provided by utilizing the Kyoto Encyclopedia of Genes and Genomes. The results of an analysis are visualized using color (here shown as *arrows*), with highly active proteins in *green* (*up arrow*), and highly repressed proteins in *red* (*down arrow*), as an example

Visualization of complex data has a significant history, both in terms of information transfer (Tufte 1991) and statistical analysis (Cleveland 1994). Many lessons critically important to cancer research and patient treatment have been learned. In a dramatic example of data presentation, a simple plot of the O-ring failure rate against temperature would clearly have led to a reassessment of the decision to launch the space shuttle Challenger. However, the data was not presented in a manner that highlighted the pattern prior to launch. It is often the case that extremely complex systems, whether space vehicles or cancer, have "directions" in the data that provide novel insight. Finding these directions is the purview of statistics, but presenting them to the researcher in a suitable form should be part of the pipeline.

The most logical approach for building visualization into a data pipeline is to leverage statistical insight into the important directions in the data and visualization research on how to present information. Realistically, it is hard to present more than three spatial dimensions graphically, with the potential for a fourth dimension, usually used for time, through animation. The results of analyses, even when looking across samples instead of genes, are not limited to so few dimensions. As such, methods are usually applied to either pick a few dimensions for visualization or to reduce dimensionality. Identifying the logical 'directions' for visualization is effectively a statistical problem in dimensionality reduction. While standard methods exist, this will likely require new methods for the complex data in cancer research, so the pipeline visualization routines will need to be as modular as the other elements.

An important point in the design of the visualization components is to insure decoupling between the analysis methods and the visualization methods. Although this is standard best practice in software design, it is extremely tempting to tie a specific visualization to an analysis approach arguing that visualization is closely tied to the statistical methodology. However, it is generally the case that a useful visualization will find new applications (e.g., boxplots, heatmaps), and, critically, making the code interdependent hinders code re-use. Code re-use is even more dramatically reduced if the code for the analysis and visualization routines is actually interspersed during coding. As with other best practices, there is a price in terms of initial development, so trade-offs exist.

6.7 Interactive Analysis

6.7.1 Exploratory Biological Research

The design of a data analysis pipeline as described above can support data retrieval, analysis of multiple parallel paths through a data set, combination of results of analysis, visualization of these results, and presentation to the researcher, including both on-screen plots and high-resolution figures. Coupled with tools developed for reproducible research (see Chap. 8), the pipeline can even produce associated text suitable for methods sections in publications. However, a much more difficult problem is to enable researchers to interact with their data.

Traditional biological research involves dynamically probing results, modifying protocols, and testing hypotheses. Researchers trained in this way often wish to modify visualizations and even underlying data filtering during analysis. While this can raise serious statistical issues due to hidden multiple testing of high-dimensionality data, it will be highly desirable for a data pipeline to provide interactive capabilities to encourage its adoption. Simple examples may include researchers who wish to explore standard pathways with components reduced or removed by experimental design, who wish to remove subsets of genes during clustering, or who wish to remove subsets of the overall data that they suspect is of low quality. One advantage of integrating such abilities in a pipeline is that the data classes can at least track how many potential significant modifications were made to the data and alert the researcher in final reports.

6.7.2 Interacting with Data

For researchers to interact closely with their data during analysis is straightforward for a desktop system. The complete control the developer has over all aspects of windows and their contents makes it relatively easy to allow researchers to rotate views, choose subsets of data, and perform operations on these subsets. While the overall development effort may be substantial, requiring careful work with users to guarantee proper interface implementation, there are no technical barriers to such a system.

However, realistic pipelines will not run interactively within a desktop system. Within client–server systems, the similarities between X and desktop systems will permit similar interactions. While X windows are somewhat more limited than desktop windows, they provide most of the needed functionality for interaction. The main additional requirement will be to maintain adequate network bandwidth and response to permit users to interactively modify their data. There will remain issues for the analysis of any modified data within the pipeline, and this is discussed in Sect. 6.7.3.

The major problem for interactive analysis arises with Web-based interfaces to pipelines. One workaround is to allow more demanding users access through a client–server system, with the bulk of users having no interactive data analysis capabilities. Alternatively, interactive Web technologies, such as Java WebStart and Microsoft .Net, can be used to provide limited functionality. These technologies allow applications to run on the client machine, but ease maintenance of the code by pushing updated versions to the desktop client on demand. These technologies also integrate the desktop environment with the server, either through Java or through Windows. One advantage to WebStart is that it will run on all widely used computer platforms, while .Net is limited to Windows. On the other hand, .Net is considered easier from a developer perspective.

Barring these technologies, Web-based interfaces to data analysis pipelines are very limited. There is some minor ability to isolate data, however it remains very

rudimentary. There is also a problem in communication of choices back to the pipeline, as the pipeline is essentially asynchronously linked to the Web browser and communication needs to be established with each update. Certain design patterns, such as Façade patterns, can simplify the communication aspects by enabling the browser to generate a single network link and provide all changes. Nevertheless, most users are likely to find browser-based interaction with the present Web technology overly limited.

6.7.3 Data Modifications and Pipeline Branching

Interacting with the data creates a problem for a pipeline that is not reflected in desktop systems – how to handle modifications to the data that affect the analysis. For example, should a user decide to eliminate a subset of the data, this will change the results of application of a clustering algorithm. On a desktop system, this is simply handled by generating an event to trigger a recalculation. With a pipeline however, there could be ongoing analyses using results involving the original data, which the user may wish to continue even while filtering the data to test other approaches. This presents two problems. First, new analyses must be launched and feedback presented back to the user. Second, care must be taken to maintain detailed histories within the results that inform the user of the specific steps taken and data used for the results being presented.

The introduction of new real-time computations that provide feedback to the user could be addressed by reserving portions of the pipeline hardware for interactive analysis, while the bulk of the pipeline remains focused on less interactive analyses. This could be accomplished equally well by giving such jobs higher priority within the full cluster. In either case, it may be desirable to enforce some balance between highly interactive use and more batch-like operations. Alternatively, using WebStart or .Net, certain operations that permitted interaction could initially be done on the desktop, limiting the interactive portions to algorithms of low computational complexity. This is probably realistic, as interactively changing data in computationally expensive algorithms is unlikely to be feasible in any case.

For tracking the data that is actually used as the input to an algorithm and the preprocessing and filtering of the data that was used is best done through the emerging use of tools from reproducible research efforts (see Chaps 8, 17, and 20). Since the potential exists for many edited versions of the data to be processed simultaneously, one useful approach is to rely on cloning of the data object at branch points, as with the Java clone() method. This will create a copy that includes the history of the object, so that a history of all manipulations prior to cloning are retained, and the two versions of the object can continue through the pipeline independently. However, if data sizes are intractable, it may be more feasible to maintain a history of operations in a way that permits the analysis to be exactly duplicated later. Sweave has indeed been used for this in a microarray pipeline (Rainer et al. 2006).

As this would involve only analyses that provided successful insight, it may not be overly burdensome to repeat them.

6.8 Conclusions

We have presented a summary of major issues that must be addressed by any future data pipeline for the analysis of genomic data. Overall, this entails handling data persistence, annotation, analysis, summarization, and visualization in an environment of emerging data types and novel algorithms. Numerous approaches have been developed within the computer science community to deal with these issues from different perspectives, including databases, software design, and workflow analysis. The statistical and computational learning communities continue to address problems arising from the noisy data and complex biological interactions underlying it during analysis. The developments in these fields lay the foundation for successful creation and maintenance of a high-throughput data pipeline.

References

Altschul SF, Gish W, Miller W et al (1990) Basic local alignment search tool. J Mol Biol 215:403–410

Arakawa K, Kono N, Yamada Y, Mori H, Tomita M (2005) KEGG-based pathway visualization tool for complex omics data. In Silico Biol 5:419–423

Ashburner M, Ball CA, Blake JA et al (2000) Gene ontology: tool for the unification of biology. The gene ontology consortium. Nat Genet 25:25–29

Ball CA, Brazma A (2006) MGED standards: work in progress. OMICS 10:138–144

Burks C, Fickett JW, Goad WB et al (1985) The genbank nucleic acid sequence database. Comput Appl Biosci 1:225–233

Cleveland WS (1994) The elements of graphing data. AT&T Bell Laboratories, Murray Hill, NJ

Falkner JA, Falkner JW, Andrews PC (2006) Proteomecommons.Org JAF: reference information and tools for proteomics. Bioinformatics 22:632–633

Gamma E, Helm R, Johnson R et al (1995) Design patterns: elements of reusable object-oriented software. Addison-Wesley, Reading, MA

Gentleman RC, Carey VJ, Bates DM et al (2004) Bioconductor: open software development for computational biology and bioinformatics. Genome Biol 5:R80

Grant JD, Somers LA, Zhang Y et al (2004) FGDP: Functional genomics data pipeline for automated, multiple microarray data analyses. Bioinformatics 20:282–283

Hood LE, Hunkapiller MW, Smith LM (1987) Automated DNA sequencing and analysis of the human genome. Genomics 1:201–212

Humphreys BL, Lindberg DA (1993) The UMLS project: making the conceptual connection between users and the information they need. Bull Med Libr Assoc 81:170–177

Irizarry RA, Bolstad BM, Collin F et al (2003) Summaries of affymetrix genechip probe level data. Nucleic Acids Res 31:e15

Kanehisa M, Goto S, Kawashima S et al (2002) The KEGG databases at genomenet. Nucleic Acids Res 30:42–46

Komatsoulis GA, Warzel DB, Hartel FW et al (2007) Cacore version 3: implementation of a model driven, service-oriented architecture for semantic interoperability. J Biomed Inform 41:106–123

Lockhart DJ, Dong H, Byrne MC et al (1996) Expression monitoring by hybridization to high-density oligonucleotide arrays. Nat Biotechnol 14:1675–1680

Ochs MF, Casagrande JT (2008) Information systems for cancer research. Cancer Invest 26:1060–1067

Oinn T, Addis M, Ferris J et al (2004) Taverna: a tool for the composition and enactment of bioinformatics workflows. Bioinformatics 20:3045–3054

Parsons DW, Jones S, Zhang X et al (2008) An integrated genomic analysis of human glioblastoma multiforme. Science 321:1807–1812

Raffelsberger W, Krause Y, Moulinier L et al (2008) Rreportgenerator: automatic reports from routine statistical analysis using R. Bioinformatics 24:276–278

Rainer J, Sanchez-Cabo F, Stocker G et al (2006) Carmaweb: Comprehensive r- and bioconductor-based web service for microarray data analysis. Nucleic Acids Res 34:W498–W503

Rogers AE, Cappallo RJ, Hinteregger HF et al (1983) Very-long-baseline radio interferometry: the mark III system for geodesy, astrometry, and aperture synthesis. Science 219:51–54

Rubin DL, Lewis SE, Mungall CJ et al (2006) National center for biomedical ontology: advancing biomedicine through structured organization of scientific knowledge. OMICS 10:185–198

Schena M, Shalon D, Davis RW et al (1995) Quantitative monitoring of gene expression patterns with a complementary DNA microarray. Science 270:467–470

Subramanian A, Tamayo P, Mootha VK et al (2005) Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles. Proc Natl Acad Sci U S A 102:15545–15550

Tufte ER (1991) Envisioning information. Graphics Press, Cheshire, CT

Watson JD (1990) The human genome project: Past, present, and future. Science 248:44–49

Whetzel PL, Parkinson H, Causton HC et al (2006) The MGED ontology: a resource for semantics-based description of microarray experiments. Bioinformatics 22:866–873

Chapter 7
Mathematical Modeling in Cancer

Robert A. Gatenby

Abstract Cancer is a complex, dynamical system that includes continuous interactions between evolving cancer cells and their spatially and temporally heterogeneous microenvironment. Despite intense research for decades, a comprehensive model of the cancer *system* remains elusive. Furthermore, failure to organize the vast extant data on cancer into a comprehensive theoretical framework has undoubtedly contributed to the steady mortality for cancer in the past 50 years compared to the marked decline observed in, for example, cardiovascular disease.

7.1 Introduction

In the latter half of the sixteenth century, Tyco Brahe laboriously documented the movement of the known planets in the solar system. His observations were remarkably accurate particularly since the telescope had not yet been invented. The movements of the planets, however, were difficult to understand as they often stopped their normal orbital motion and even seemed to briefly move backward as they traversed the night sky.

Astronomers of the era had two theoretical views of the universe. The ancient Ptolemaic model in which the earth stood at the center of the universe and planets revolved around it and the new Copernican theory in which the sun was at the center and the planets revolved around it in perfectly spherical orbits (Thoren 1990).

R.A. Gatenby (✉)
Departments of Radiology and Integrative Mathematical Oncology, Moffitt Cancer Center, 12902 Magnolia Drive, Tampa, FL 33612, USA
e-mail: robert.gatenby@moffitt.org

M.F. Ochs et al. (eds.), *Biomedical Informatics for Cancer Research*,
DOI 10.1007/978-1-4419-5714-6_7, © Springer Science+Business Media, LLC 2010

Brahe recognized that the stationary earth model did not fit his data. But, since theological considerations "required" a stationary earth, he proposed a compromise solution in which the planets orbited the sun and, in turn, revolved around the earth at a central point. However, Brahe also recognized he was in need of a mathematical assistant and, in 1599, hired Johannes Kepler. For 17 years, Kepler studied this vast data set to find patterns and connections. His arithmetic calculations involving just the orbit of Mars filled nearly 1,000 sheets of paper (Tiner 1977). The remarkable intensity of the effort is apparent in his description of the moment of insight:

> …and if you want the exact moment in time, it was conceived mentally on 8th March in this year one thousand six hundred and eighteen, but submitted to calculation in an unlucky way, and therefore rejected as false, and finally returning on the 15th of May and adopting a new line of attack, stormed the darkness of my mind. So strong was the support from the combination of my labor of seventeen years on the observations of Brahe and the present study, which conspired together, that at first I believed I was dreaming, and assuming my conclusion among my basic premises. But it is absolutely certain and exact that the proportion between the periodic times of any two planets is precisely the sesquialterate proportion of their mean distances… (Kepler 1619).

About 70 years later, Isaac Newton recognized the existence of gravity:

> … all matter attracts all other matter with a force proportional to the product of their masses and inversely proportional to the square of the distance between them. (Newton 1846)

Through the new mathematics of calculus, Newton derived Kepler's laws from interaction of gravitational and centripetal forces. The model of planetary motion derived from these fundamental principles both reproduced extant data and provided a deep understanding of the forces that govern the system. Furthermore, the modeling results led to predictions. For example, perturbation in the orbit of Uranus led to the conclusion that another massive object must be nearby. This, in turn, motivated new directed observations that led to discovery of Neptune.

In many ways, Kepler and Newton demonstrate the differences between bioinformatics and mathematical modeling. The former analyzed extant data using the most sophisticated computing tools available (his brain plus pen and paper) to find patterns and connections. The latter approached the problem of planetary motion by identifying first principles, modeling the system dynamics, and then demonstrating that these results matched experimental observations.

Since the time of Newton, the physical sciences have thrived on a research paradigm that deeply integrates mathematical modeling and empirical data. Richard Feynman expressed the relationship as follows:

> Mathematics is a deep way of describing nature, and any attempt to express nature in philosophical principles, or in seat-of-the-pants mechanical feelings, is not an efficient way. (Feynman 1965)

Despite centuries of successful experience in the use of mathematical models to understand complex systems in the physical sciences, this paradigm has been extended into cancer research only recently.

7.2 Mathematical Models in Cancer

The societal burden of cancer has stimulated decades of intense scientific effort that has resulted in many important new insights and therapies. Yet, despite these advances, the improvement in mortality rates for cancer patients still lags behind that of the other major causes of death such as cardiovascular and cerebrovascular diseases (Mortality data 1950 and 2001). Research in cancer biology has been greatly accelerated by new experimental technologies and the revolution in genomics and bioinformatics that have generated overwhelming amounts of biomolecular data. Lacking, however, are the conceptual frameworks necessary to organize these data in ways that guide more significant advances in understanding of the disease (Gatenby and Maini 2002, 2003). This state of affairs clearly suggests the need for interdisciplinary research that synthesizes experimental results with mathematical analysis and modeling to provide new insights into the underlying dynamics governing the disease and to help organize new experimental and treatment strategies.

Such an idea is not new. As noted above, since the days of Newton, the natural philosopher has used the tools of mathematics to quantify, with consistent success, the physical world around us – fully justifying the statement widely attributed to Galileo that "The book of Nature is written in the language of Mathematics." In contrast, the biological world, perhaps by virtue of its remarkable diversity, has been dominated by a tradition of observation, description, and classification. The potential role of mathematics in biological research has long been acknowledged. D'Arcy Thompson's monumental treatise "On Growth and Form" (Thompson 1992) opens with a number of quotations that includes one from Karl Pearson: "I believe the day must come when the biologist will – without being a mathematician – not hesitate to use mathematical analysis when he requires it." Over 100 years later an article in the Economist stated "If cancer is ever to be understood properly, mathematical models such as these will surely play a prominent role" (Thompson 2004).

However, for a scientific community steeped in the Aristotelian culture of empiricism, the introduction of mathematical methods is immensely challenging. Indeed, despite occasional bursts of enthusiasm, the role of mathematical and physical reasoning in the life sciences remains relatively limited. An occasional investigator such as Schrödinger in the lecture/book "What is Life" has used the principles (but not the mathematics) of physical sciences to generate insights into key biological questions (Schrödinger 1944).

The explosion of data generated by molecular biology *has* necessitated a widespread interest in the set of complex data mining tools and techniques generally described as bioinformatics. However, there remains little utilization of mathematical modeling in tumor biology and oncology to frame hypothesis, provide contextual frameworks for organizing data, and generate testable predictions. Modeling of this type in biological systems is very challenging, and the genuine successes of mathematics in biology, such as the Hodgkin–Huxley model in neurobiology and

knot theory in DNA conformations, are relatively rare. In part, this reflects the daunting intellectual demands of working, either collaboratively or individually, in such profoundly different disciplines. Biological and clinical investigators typically have little or no training in the applied mathematics necessary to write and analyze mathematical models. Similarly, applied mathematicians usually have little background in the complex, multiscalar (molecular, cellular, tissue, and populations) dynamics in the life sciences. As a result, their models are often of little relevance or interest to real biological or clinical problems.

Nevertheless, in a Nature article in 2003, Gatenby and Maini (2003) proposed that future advances in cancer biology and treatment demand development of a field of study they termed mathematical oncology. They pointed out that cancer is a complex, multiscalar disease dominated by nonlinear dynamics. Such systems, while difficult to model using a wide range of mathematical methods, are impossible to understand through the intuitive, "seat of the pants" approach that is currently employed by virtually the entire field tumor biology and oncology. They conclude that "In the absence of consistent application of rigorous mathematical models, theoretical medicine will largely remain empirical, phenomenological, and anecdotal, successful only in linear systems that can be defined by a single experiment or a few experiments."

Achievement of an integrative cancer research paradigm combining modeling and empirical research, such as that of the physical sciences, will need to overcome many historical, philosophical, and methodological barriers. The core component of cancer research must always remain biomolecular research including in vitro and in vivo laboratory and clinical observations. Clearly, mathematical models without data are useless. At the same time, tumor biology must recognize that data are not science. Hypothesis-driven, biologically informed mathematical models are necessary to provide theoretical frameworks to organize and understand data and to guide new experiments.

Critical to development of realistic mathematical models is the integration of the statistical methods used to analyze the torrent of data generated by modern molecular methods (see Chap. 6) – an approach described as "integrative mathematical oncology" (Anderson and Quaranta 2008). Although the dialog between bioinformatics and biomolecular experimentation is often systematic, it is not always strategic. In particular, these methodologies often provide only limited insight into tumor dynamics. That is, molecular biology typically requires tissue removal and homogenization, so that it generates large amounts of "average data" but limited information on spatial and temporal heterogeneity – critical properties in understanding the *dynamics* of cancer progression and treatment. Thus, continuous interactions of evolving phenotypes and chaotic microenvironment of the biological processes at the molecular, cellular, and tissue levels in cancer demand appropriate, dynamical mathematical models that can transform extensive, and occasionally haphazard, experimental programs into an integrated conceptual approach.

Thus, it seems reasonable to propose that a crucial missing methodological component in cancer research is mathematical modeling (Gatenby and Maini

2003; Thompson 1992), which provides a conceptual framework for and predictive value to the informatics data. The process of formulating a model invites the development and incorporation of first principles, clarifies assumptions, demands rigorous statement of hypotheses, and identifies key variables and parameters. Analyzing the failure of a model can often be as valuable as developing a successful one. By virtue of its predictive power, a good model can help plan experiments by identifying parameter regimes of interesting behavior – regimes that might otherwise be time consuming and costly to discover by systematic experimentation. Furthermore, a model can also be used to estimate important parameters by fitting data. In these ways, mathematical modeling completes the circle of discovery: experiments provide data that, in turn, informs the construction of new experimental designs.

7.3 An Example

The potential value of modeling is apparent in a sequence of studies investigating carcinogenesis. This stepwise transition of normal cells to the cancer phenotype through a number of premalignant intermediates is often described as "somatic evolution." The classic conceptual model depicted this evolutionary process as a series of genetic mutations typically in oncogenes and tumor suppressor genes (Fearon and Vogelstein 1990). Gatenby and Vincent (Gatenby and Vincent 2003; Vincent and Gatenby 2008) subjected this concept to rigorous analysis by applying evolutionary game theory. The results of the models demonstrated that mutations in oncogenes and tumor suppressor genes alone did not result in formation of a malignant cancer. In fact, these changes led only to self-limited growth because as the tumor population increased in size, proliferation was limited by substrate limitation (Fig. 7.1).

Thus, as a result of the modeling studies, the authors concluded that there is a previously unknown era of carcinogenesis in which environmental selection forces are dominated by competition for limited substrate (Gatenby and Vincent 2003).

These results demanded a reexamination of the adaptive landscape of somatic evolution and recognition of the critical role of the anatomy and physiology of epithelial surfaces (Gatenby and Gillies 2004). As shown in Fig. 7.2, evolving in situ cancer cells proliferate on the surface of the basement membrane that maintains a separation from the epithelial cells and the underlying stroma including blood vessels. As the tumor cells proliferate into the lumen, their distance from the blood vessels increases and the resulting diffusion reaction kinetics results in regional hypoxia, low glucose concentrations, and acidosis.

The authors developed a theoretical model (Gatenby and Gillies 2004) (Fig. 7.3) of carcinogenesis proposing that, due to this anatomy and physiology of tumor growth on epithelial surfaces, some regions of in situ cancer cells will be subject to cyclical hypoxia. This will promote adaptation by upregulating glycolysis.

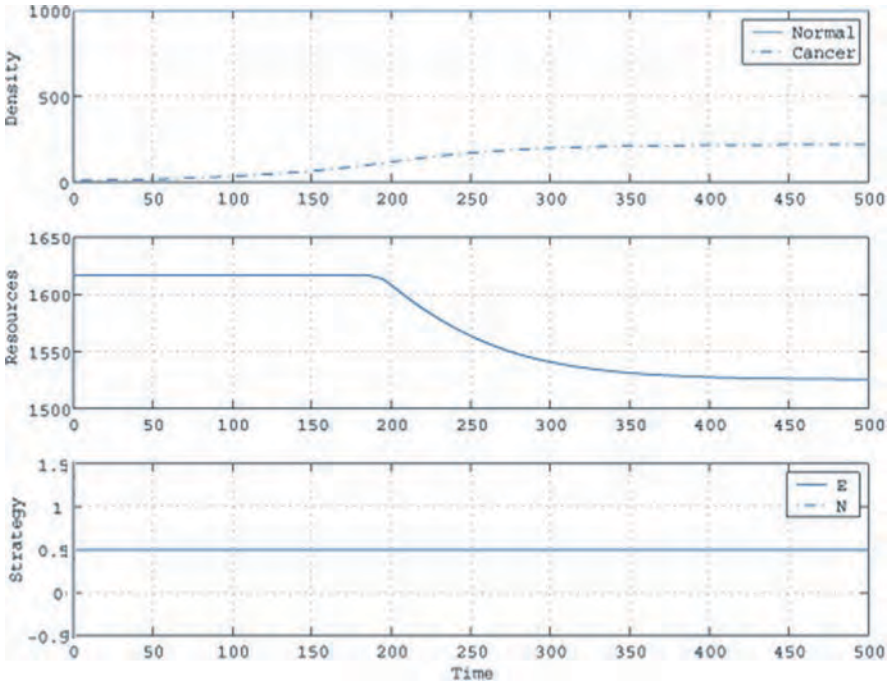


Fig. 7.1 Simulations of tumor progression from evolutionary mathematical models of carcinogenesis following a series of mutations in oncogenes and tumor suppressor genes. As shown in the *top row*, these genetic changes resulted in only self-limited tumor growth and not formation of an invasive cancer. The reason is shown in the *lower panel* which demonstrates that as the tumor population increased, local resource concentrations decreased to the point that additional growth could not be supported. This result led to the proposal of a previously unknown stage of carcinogenesis dominated by competition for substrate (from Gatenby and Vincent 2003)

However, due to the resulting increased production of lactic acid, this adaptive landscape is then replaced by one that is dominated by the toxic effects of acidosis. This requires a second evolutionary step that allows the tumor cells to adapt to unusually acidic environments. The final outcome of this sequence produces a cellular phenotype with a profound proliferative advantage because it can produce an acidic environment (through upregulated glycolysis) that is toxic to its competitors but not to itself. As a result of this study, it was hypothesized that adaptation to regional hypoxia and acidosis was a critical component in late carcinogenesis that promoted transition from in situ to invasive growth.

This theoretical result was examined (Gatenby et al. 2007) using in vitro experimental methods with tumor spheroids and clinical observations. As shown in Fig. 7.4, this work confirmed the presence of adaptation to hypoxia in the central

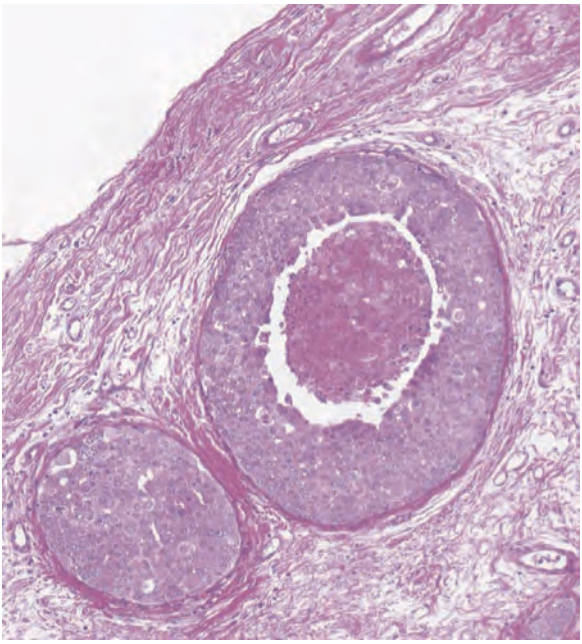


Fig. 7.2 A micrograph of ductal carcinoma in situ. Following the prediction of a substrate-limited era of tumor growth, it was recognized that the tumor cell growth into the lumen and away from the basement membrane will result in increasing diffusion distance from the blood vessels (which remain on the opposite side of the basement membrane) and development of regional hypoxia and acidosis (from Gatenby and Gillies 2004)

regions of DCIS and close association of upregulated glycolysis with regions of transition from in situ to invasive tumor growth.

7.4 Conclusion

Developing mathematical models of the complex, nonlinear dynamics of cancer biology is difficult. However, understanding these dynamics through intuitive reasoning is impossible. This will require development of a cadre of investigators conversant in both tumor biology and mathematical methods to develop quantitative theoretical models that provide conceptual frameworks to organize extant data, integrate new information, and guide future experiments. Accomplishing this will require surmounting many intellectual, social, and philosophical barriers. However, it seems likely that future progress in understanding and controlling cancer will depend upon it.

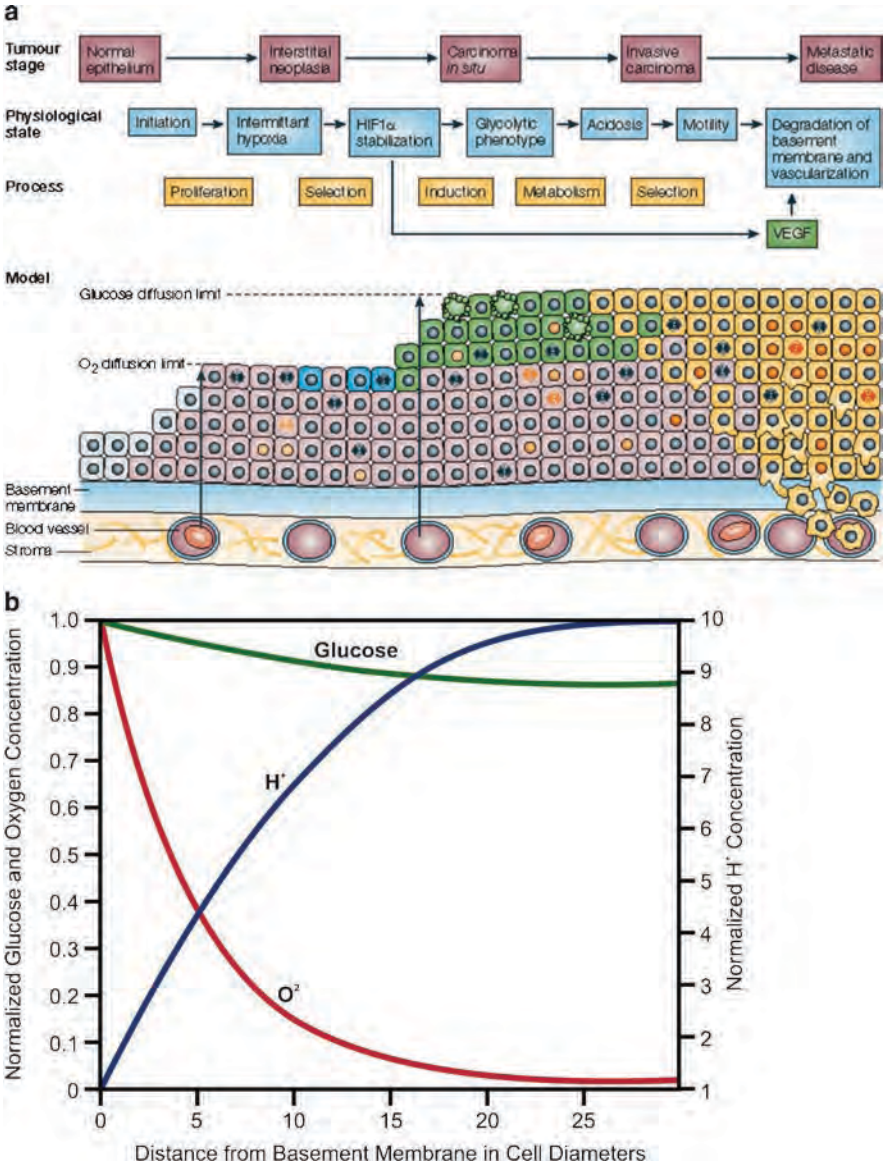


Fig. 7.3 (**a**) Theoretical model of carcinogenesis incorporating the diffusion-reaction kinetics associated with tumor growth on an epithelial surface. Note that the intact basement membrane enforces separation of the tumor cells from the blood supply. As a result, growth factors, substrate, and metabolites must diffuse over increasing distances to travel from the blood vessels to the proliferating layer of cells as the tumor grows into the lumen. (**b**) demonstrates the calculated variations in oxygen, glucose, and H^+ as a function of cell diameters from the basement membrane. Note that oxygen concentrations rapidly decline and H^+ concentrations rapidly increase with 4 or 5 cell layers from the basement membrane. This results in regional variations in adaptive parameters that govern local phenotypic evolution from Gatenby and Gilles (2004)

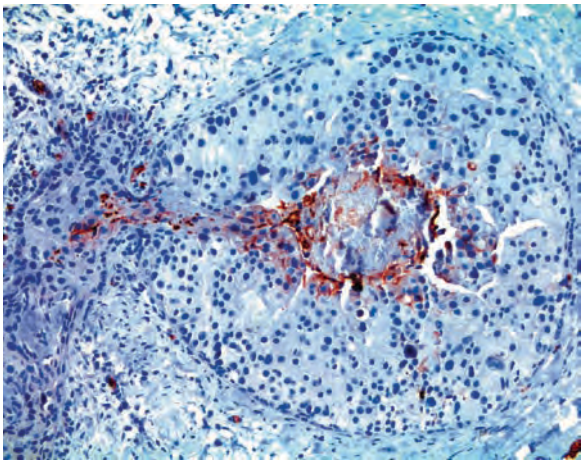


Fig. 7.4 Immunohistochemical stain of DCIS for glucose transporter 1 (GLUT-1). GLUT-1 upregulation serves as a biomarker for increased glycolysis. Note that GLUT-1 is upregulated in the central region of the DCIS presumably as a result of hypoxia. However, it is also upregulated in a population of cells that invades into the normoxic region of the DCIS and breeches the basement membrane forming a microinvasive cancer (from Gatenby et al. 2007)

References

Anderson AR, Quaranta V (2008) Integrative mathematical oncology. Nat Rev Cancer 8(3):227–234
Arias E, Anderson R, Kung H-C, Murphy S, Kochanek K (2003) Deaths: Final Data for 2001 National Vital Statistics Report 52(3):1–116
Fearon ER, Vogelstein B (1990) A genetic model for colorectal tumorigenesis. Cell 61:759–767
Feynman RP (1965) The character of physical law. MIT, Cambridge, MA
Gatenby RA, Gillies RJ (2004) Why do cancers have high aerobic glycolysis? Nat Rev Cancer 4(11):891–899
Gatenby RA, Maini P (2002) Modelling a new angle on understanding cancer. Nature 420:462
Gatenby RA, Maini P (2003) Mathematical oncology – cancer summed up. Nature 421:321
Gatenby RA, Vincent TL (2003) An evolutionary model of carcinogenesis. Cancer Res 63:6212–6220
Gatenby RA, Smallbone K, Maini PK, Rose F, Averill J, Nagle RB, Worrall L, Gillies RJ (2007) Cellular adaptations to hypoxia and acidosis during somatic evolution of breast cancer. Br J Cancer 97(5):646–653
Kepler J (1619) Harmonice mundi (Linz, 1619). English edition: Harmonies of the world, Book 5, Chap. 3 (trans: Aiton, Duncan and Field), p 411
Newton I (1846) Mathematical principles of natural philosophy (trans: Andrew Motte), First American edn. New York (London: Benjamin Motte, 1729)
Schrödinger E (1944) What is life? Cambridge University Press, London
Thompson DW (1992) On growth and form. Dover reprint of 1942, 2nd edn. (1st edn., 1917). Dover, New York
Thompson DW (2004) Malignant maths. The Economist 370(8359):77–78
Thoren V (1990) The lord of Uraniborg: a biography of Tycho Brahe. Cambridge University Press, Cambridge
Tiner JH (1977) Johannes Kepler: giant of faith and science. Mott Media, Milford, MI
Vincent TL, Gatenby RA (2008) An evolutionary model for initiation, promotion, and progression in carcinogenesis. Int J Oncol 32(4):729–737

Chapter 8
Reproducible Research Concepts and Tools for Cancer Bioinformatics

Vincent J. Carey and Victoria Stodden

Abstract "Reproducible research" refers to a publishing discipline, originating in the geosciences, in which journal articles are accompanied by publication of data resources and software sufficient to allow independent reproduction of all tables and figures presented in articles. This paper reviews concepts of reproducible research in connection with cancer bioinformatics. The importance of reproducible discipline in the face of analytic complexity of microarray studies is documented with two case studies, and the role of portable self-documenting data and software archives in securing reproducibility is described. Legal protections for those engaged in reproducible research are discussed in the context of current US copyright law; a reproducible research standard that formalizes rights and obligations of those engaged in reproducible research is detailed. There is every indication that reproducible discipline is feasible for microarray studies, and reliability of inferences in cancer bioinformatics will be enhanced if commitments to concrete reproducibility are broadly accepted in the research community.

8.1 Introduction

A scientific publishing discipline called "reproducible research" originated in the geosciences in the 1980s and has since garnered attention in a number of fields. Important references include Donoho et al. (2009) in computational harmonic analysis, Vandewalle et al. (2009) in digital signal processing, Peng et al. (2006) in epidemiology, Laine et al. (2007) in internal medicine, Gentleman (2005) in cancer bioinformatics, and Gentleman and Lang (2004) in general statistical computing. The paper of Donoho et al. is among the first to treat questions concerning costs of

V.J. Carey (✉)
Channing Laboratory, Brigham and Women's Hospital, Harvard Medical School,
181 Longwood Avenue, Boston, MA 02115, USA
e-mail: stvjc@channing.harvard.edu

M.F. Ochs et al. (eds.), *Biomedical Informatics for Cancer Research*,
DOI 10.1007/978-1-4419-5714-6_8, © Springer Science+Business Media, LLC 2010

implementing the discipline of reproducible research and the effects of data and software publication on competitiveness. Tradeoffs exist and these issues will not be considered here. This paper will focus on reproducibility issues for cancer bioinformatics.

To date, the primary approaches to genome-scale investigation in cancer bioinformatics involve transcript profiling using DNA microarrays. In a recent multiteam contribution to *Nature Genetics* Ioannidis and colleagues (2009) commented:

> Microarray-based research is a prolific scientific field where extensive data are generated and published. The field has been sensitized to the need for transparent design and public data deposition and public databases have been designed for this purpose. Issues surrounding the ability to reproduce published results with publicly available data have drawn attention in microarray-related research and beyond. The reproducibility of scientific results has been a concern of the scientific community for decades and in every scientific discipline. (p. 149)

Ioannidis' paper should be consulted for detailed references and illustrations for their survey of the reproducibility of 18 recently published microarray studies.

The primary concerns of this paper are to describe why reproducible research discipline is important for cancer bioinformatics and to show how the discipline can be implemented to improve reliability of work in the field. Two aspects of implementation are considered. First, there is discussion and illustration of relevant programming and data archiving techniques. Second, details are provided on protection of intellectual property through licensing of reproducible research products.

8.2 Case Studies

8.2.1 Case Study 1: Baggerly, Coombes, and Neeley, J Clin Oncol (2008)

In this section, we consider the direct criticism of reproducibility of a primary manuscript in oncology. The main objectives of this section are to state concisely the offerings of the primary manuscript, to describe the shortcomings identified by Baggerly et al. in their letter (Baggerly et al. 2008), and to analyze the rebuttal of the original authors in the context of some independent computations.

8.2.1.1 Context

Dressman et al. (2007) presents an integrative analysis of 119 microarrays of ovarian cancer tumors. Tumors are classified as responsive or nonresponsive to platinum treatment. Affymetrix U133A chips are used to profile expression patterns on all tumors. These chips are preprocessed using RMA (Irizarry et al. 2003) and then further corrected using a procedure called sparse factor regression (Carvalho et al. 2008).

A procedure called shotgun stochastic search (Hans et al. 2007) is used to identify a set of genes (probe sets) predictive of platinum responsiveness and to quantify probability of platinum responsiveness on the basis of gene expression values. Pathway activation scores for tumors were computed from experiments based on cell lines (HMECs) created by Bild et al. (2006) and modified to exhibit pathway deregulation. Dysregulation of Src and E2F3 pathways was independently found to be significantly associated with differences in survival distributions among platinum nonresponsive patients. A supplementary Web site provided clinical information on samples, CEL files, and sparse factor regression-corrected RMA quantifications of expression for all samples. The latter file of quantifications is provided in an Excel spread sheet, so we will refer to this resource as XLSQ.

8.2.1.2 Three Challenges: Nonreconstructibility and a Failed Sanity Check

In their letter to J Clin Oncol (2008), Baggerly et al. indicate a number of problems arising in their attempt to reconcile results in the published paper with the information in the online archive.

- *Problem 1*. Samples in XLSQ were incorrectly labeled. Proper labeling could be established for 116 of 119 samples through comparisons with pure RMA quantifications obtained using the CEL files. (Note added September 2009: The file of expression quantifications posted at the Duke Web site was revised in August 2009 to establish the correct sample labeling. Analyses in this chapter used the proper labeling established through comparison with CEL images. Assertions of this chapter are not affected by the authors' relabeling of the quantifications on the Web.)
- *Problem 2*. Using gene-specific *t* tests, no evidence of differential expression between platinum-responsive and nonresponsive tumors could be found for genes in the published signature.
- *Problem 3*. Upon creation of pathway activation scoring coefficients on the basis of singular value decompositions to expression data matrices derived from Bild's cell line array archive, asserted associations between E2F3 activation and survival among platinum-nonresponsive patients (Figs. 2C and 2E of the 2007 Dressman paper) could not be reconstructed.

Problems 1 and 3 will be referred to as conditions of *nonreconstructibility*. Data resources putatively employed in developing the results of the paper cannot be used by independent researchers to reconstruct the results.

Problem 1 seems to be purely clerical in nature – somehow sample labels were scrambled. Problem 3 is more intricate and will be discussed in more detail later.

Problem 2 does not directly concern reconstructibility, and its mention in Baggerly's letter led to a heated rebuttal. Baggerly et al. used two-sample *t* tests to assess the performance of the asserted signature for platinum responsiveness. Dressman et al. used stochastic shotgun search to identify the signature. It is an open question whether genes, *all* of whose expression distributions in samples of platinum-responsive and nonresponsive tumors have mean expression levels

that are indistinguishable using the *t* test, might nevertheless still be associated with different probabilities of responsiveness as identified by stochastic shotgun search. Had Baggerly et al. employed stochastic shotgun search and failed to generate the signature gene list published in Dressman et al. (2007), a nonreconstructibility problem would be present. It could be useful to have a concise term for the conflict between Baggerly's analysis and the published result. We call this a "failed sanity check." Sanity checks of scientific assertions involve simple computations that have readily predicted outcomes if the assertions are correct and are correctly understood. The individual *t* tests of differential expression of genes in a putative signature constitute a sanity check of the complex multivariate analysis conducted with stochastic shotgun search. The *t* tests are a weak sanity check; we will consider a different multivariate sanity check below (Sect. 8.3.1).

8.2.1.3 Rebuttals to the Basic Challenges

In their reply to Baggerly et al., Dressman, Potti, and Nevins apologize for the identifier scrambling noted in Problem 1 and indicate that it arose only in the preparation of the online archive. Thus, the paper is deemed to be unaffected by the scrambling, but any attempt to reconstruct the paper will be doomed if it takes literally the labeling of array quantifications for association with the clinical status table.

Problems 2 and 3 are addressed by Dressman et al. in very severe terms. The response is worthy of full quotation. After making four enumerated points in response to findings of clerical error and batch effects, they continue,

> (5) [Baggerly et al.] conclude that the genes identified in our model do not provide separation with respect to clinical response. Unfortunately, this is based on their methods, not ours, and reflects a serious flaw in the nature of this commentary – it is wholly inappropriate to make claims that a given method does not work if the precise methods used in the study being criticized were not followed. The failure of their methods to categorize clinical response only says that their methods do not work; it says neither anything about the procedure we utilized nor the reproducibility of the results we reported in JCO.
>
> (6) Using their own methods of analysis, Baggerly et al. conclude that batch effects confound our prediction of pathway activation in the ovarian tumor data set. This is one further example of a conclusion without basis, since it is clear they did not use our methods of analysis in either the development of the pathway signatures or the application of pathway signatures to tumor samples.

Response (5) indicates that Dressman et al. are not in favor of sanity checks based on *t* tests. Response (6) has the same tenor, but involves a much more complicated concern. The rhetoric – that an investigation of reproducibility must employ "the precise methods used in the study being criticized" – is strong and introduces important obligations for primary authors. Specifically, if checks on reproducibility are to be scientifically feasible, authors must make it possible for independent scientists to somehow execute "the precise methods used" to generate the primary conclusions.

This condition is partly met for signature identification based on the stochastic shotgun search method, which is distributed at Duke University. The supplementary Web site for the Dressman paper does include a "parameter file" indicating use with 119 samples and 6,088 genes. But because the 119 samples in the archive are mislabeled, and the identities of the 6,088 genes in use are not provided (gene filtering involved unstated criteria of "trivial variation" and "low median expression levels"), no independent scientist is in a position to meet Dressman's criteria for reproducing the signature of the primary paper. Undoubtedly the required information could be obtained on request from the authors, so reproducibility of the signature could be independently checked. This has not yet occurred.

We will now return to Problem 3 and some elaborations of it. Problem 3 is not directly addressed in Dressman et al.'s rebuttal. Problem 3 involves reconstructibility of the analysis relating pathway activation to survival time in subgroups defined by clinical definition of platinum responsiveness. Problem 3 asserts that the published curves cannot be reconstructed on their own terms, but response (6) discusses confounding and the definition of pathway signatures. We will show in Sect. 8.2.1.4 that Baggerly et al. did meet Dressman et al.'s conditions for checking reproducibility and discovered failures of reconstructibility and a deeper and scientifically more complex challenge to reproducibility.

8.2.1.4 More Challenges: Batches, Confounding, and Lifetime Data

- *Problem 4*. Through inspection of CEL file headers, Baggerly et al. obtained the dates of hybridization for all arrays used in Dressman's paper. They show that survival rates and expression values for many genes vary systematically with run date.
- *Problem 5*. By elaborating models for the relationship between pathway activation and survival time to include adjustments for batch effects, Baggerly et al. show that the asserted relationship between Src pathway activation and survival among platinum nonresponsive patients is confounded. The association's estimated magnitude and degree of significance are both substantially affected by adjusting for run date, and in fact, conventional statistical significance is lost.
- *Problem 6*. By comparing the clinical data on lifetimes published in the Dressman supplementary archive with other clinical data published on the same arrays in other papers, Baggerly et al. find that declarations of event status in the Dressman supplemental archive are unreliable. The clinical data archive was revised after Baggerly's letter was published, but it is not clear whether the most reliable survival status indicators were used in the published analyses.

Figure 8.1 provides concrete illustrations of Problems 3, 4, and 5. Evidence of confounding of batch, survival, and expression is provided in panels (a) and (b). In panel (a), the *x*-axis is calendar date of hybridization, and the *y*-axis measures expression of RPS11, a member of the Src pathway signature. The expression measures

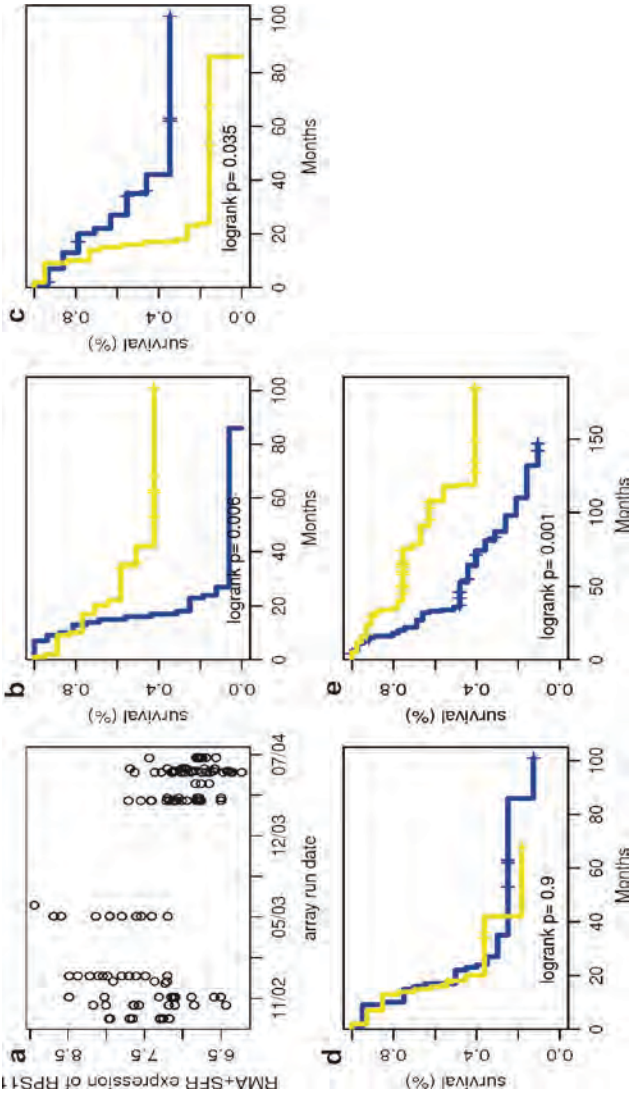


Fig. 8.1 (**a**) Variation in expression of RPS11 over array preparation dates; quantifications are those used by Dressman et al. based on RMA followed by sparse factor regression. (**b**) Survival distributions for early (*blue*) and later (*yellow*) array batches among platinum non-responders. (**c**) Association between Src pathway activation and survival among platinum non-responders. (**d**) Association between E2F3 pathway activation and survival among platinum non-responders. (**e**) As (**d**) but for platinum responders. For Kaplan–Meier graphs (**c**–**e**), *blue line* is for low pathway activation score, *yellow line* for high

are as posted on the supplementary Web site and thus result from RMA preprocessing followed by sparse factor regression corrections to remove artifacts. There is evident variation in mean RPS11 expression by date. Panel (b) consists of two Kaplan–Meier estimates of survival functions for platinum-nonresponsive subject. The blue curve gives survival for individuals whose tumor samples were hybridized up to and including 6/26/2003 ($N = 54$) and yellow for those hybridized after this date ($N = 62$). There is clear *potential* for confounding of the inferences of interest, and there is no reason, a priori or otherwise, to expect sparse factor regression to have removed *all* extrabiologic variation related to technical processing. Dressman et al. clearly do not appreciate this, remarking that "batch variations are not an issue at all in the interpretation of our results since our method of analysis, which Baggerly chose not to use, corrects for differences due to batch."

Confounding is a well-recognized problem in conventional epidemiological research, but has been less commonly addressed in microarray studies. Effective corrections for batch effects are a target of methodological research (e.g., Johnson et al. 2007), but the problem has to be recognized as a very challenging one, as it is possible for batch-related variation to have different manifestations in different genes. Caution and humility are called for when analyzing microarray data from multiple batches.

Before we conclude this case study, let us return to Problem 3. This is never directly rebutted by Dressman, Potti, and Nevins. Evidently the problem of total nonreconstructibility of Dressman's figures 2C and 2E has been submerged in the vehemence of denial of batch effects and outrage at criticism that does not employ "exact method of analysis" of the primary document. Figure 8.1c–e is the basis for our final arguments.

Figure 8.1c is a very close approximation to Dressman's original figure 2B. This is another sanity check. To construct Fig. 8.1c, we need only survival times and event indicators, stratification of samples into platinum responsive and nonresponsive states, and stratification of samples into those that exhibit Src pathway activation or quiescence. All this information is provided on the supplementary Web site, with the exception of the pathway activation classification. In their supplementary archive document ovca06.pdf, Baggerly et al. show how Bild's cell line arrays can be analyzed using singular value decomposition to create pathway activation scoring coefficients. This leads to stratification and survival contrasts highly consistent with Dressman's figure 2B. This seems to be a "passed" sanity check. The patterns seen in Fig. 8.1d, e, however, are completely inconsistent with Dressman's figures 2C and 2E. Figures 2C and 2E and the associated inferences are therefore nonreconstructible on the basis of the supplementary archive for Dressman's original paper. These findings are obtained in a way that is almost completely consistent with Dressman, Potti, and Nevins' strictures on acceptable criticisms of reproducibility. The original data and methods of analysis are used when available; pathway scoring coefficients were not made available, but the data used to estimate them was identical to that used by Dressman in the original study, and the methods seem to work in that they allow reconstruction of Fig. 2B. Nonreconstructibility of Figs. 2C and 2E require explanation on the part of the original authors.

To conclude this review of the reproducibility of Dressman et al. (2007), we summarize as follows:

- Several basic findings are demonstrably *nonreconstructible*, using the same methods and data employed by Dressman et al.
- The basic inferences may suffer from incompletely controlled confounding. If this is so, the inferences are likely nonreproducible, in that an experiment identical to the one published, but possessing a different structure of relationships between array batch and sample survival time, would likely yield different inferences.

The distinction between nonreconstructible and nonreproducible findings is worth making. Reconstructibility of an analysis is a condition that can be checked computationally, concerning data resources and availability of algorithms, tuning parameter settings, random number generator states, and suitable computing environments. Reproducibility of an analysis is a more complex and scientifically more compelling condition that is only met when scientific assertions derived from the analysis are found to be at least approximately correct when checked under independently established conditions.

8.2.2 Case Study 2: Michiels et al. and Reassessment of Predictive Accuracy in Cancer Transcriptomics

In this section, we investigate a methods paper (Michiels et al. 2005) that criticizes common practices in microarray studies in cancer. The main objectives of this section are to describe Michiels' arguments and findings and to discuss conceptual and computational issues surrounding the reproducibility of these findings.

8.2.2.1 Michiels' Gloss on Standard Approaches

Michiels et al. examined seven papers using microarrays to define prognostic molecular signatures in cancers of various types. They begin with a sketch of common approaches in applications.

> The standard strategy is to identify a molecular signature (i.e., the subset of genes most differentially expressed in patients with different outcomes) in a training set of patients and to estimate the proportion of misclassifications with this signature on an independent validation set of patients.

This characterization of standard strategy is probably not completely accurate, because considerations outside of differential expression often factor into the definition of microarray-based gene signatures in practice. Nevertheless, the "standard strategy" does get implemented in some rough sense in many papers, and it is worth understanding its advantages and disadvantages.

Formally, the process is equivalent to sample splitting (Picard and Berk 1990), even though in practice training and test sets may be acquired separately. In sample

splitting, N observations are on hand, all providing measurements on the same $G \gg N$ features. A set s of predictive features is identified through the use of aggressive data analysis techniques such as machine learning on $T < N$ training samples and then the predictive utility of feature set s is estimated using the $N–T$ remaining test samples. An advantage of this approach is that the estimate of predictive utility has intuitive appeal – it employs data not used to build the predictive model and thus constitutes a direct estimate of generalizability of the predictive procedure based on (a) the sampling procedure that yielded the N samples and (b) the signature s. Disadvantages of this approach include the fact that neither s nor the prediction procedure is uniquely defined (both depend on the specific partition of the N samples into training and test set), and the fact that the power of the prediction procedure is compromised by sacrifice of $N–T$ test samples, which might have been used in a complete model-building process.

8.2.2.2 A "Random Validation" Procedure and its Results

Michiels and colleagues define a procedure for randomly validating proposed signatures. For a given study, the dataset of size N is divided into a training set of size T with equal numbers of patients with favorable and unfavorable outcomes. Values of T range from 10 to a maximum number for which the validation set has representation of individuals of each outcome type. Five hundred such divisions were created for various choices of T for each study under analysis. The molecular signature for each training set is defined as the 50 genes with expression pattern most highly correlated with outcome as measured by Pearson's correlation. For any validation set, patients are classified to the nearest centroid (coordinatewise multivariate mean) of favorable or unfavorable signatures in the training set.

Michiels and colleagues deployed this procedure against datasets obtained in connection with seven major microarray studies. There were no supplementary data files described in the paper, so any reconstruction of the work would depend on independent acquisition of the underlying data resources. We accomplished this for three of the seven studies used by Michiels et al.: van 't Veer et al. (2002), 97 breast cancer samples classified by metastasis status; Yeoh et al. (2002), 233 pediatric leukemia samples classified by remission status; and Pomeroy et al. (2002), 60 medulloblastoma samples classified by survival after treatment. Key assertions of Michiels and colleagues resulting from this analysis are as follows:

- Signature identity is unstable and depends on specific composition of the training set.
- Predictive accuracy of signatures depends on training set size.
- "Five of the seven largest studies in microarray-based cancer prognosis did not classify patients better than chance" (abstract of Michiels' paper).

The publication of Michiels and colleagues indicates that cancer bioinformatics has made progress toward goals of reproducible research discipline.

Data were available for reanalysis through authors' Web sites or institutional distribution. The algorithms by which signatures were defined and predictions made in the original studies are not of specific interest to Michiels et al.; only the product of each paper, the prognostic signature, needed to be identified for their work to proceed. In all cases, this signature was available, at least as a list of gene symbols.

8.2.2.3 A Concrete Framework for Signature Assessment

In a simple formalism for modeling microarray study data, there are G genomic reporters on an array platform applied as uniformly as possible to N samples. Each sample is characterized through evaluation of R features, often constituting information on phenotype or experimental condition. The expression measures are preprocessed and normalized into a $G \times N$ matrix of expression values with elements x_{gi}, with g enumerating reporters and i enumerating samples. An $N \times R$ array of sample condition values has elements p_{ij}, with j enumerating conditions assessed and i enumerating samples.

A transcriptomic signature for phenotype P can be defined as any subset S of $\{1,\ldots, G\}$ for which the joint distribution of x_{si}, $s \in S$ depends on whether or not samples i have phenotype P. Preferences among signatures can arise from considerations of predictive accuracy, parsimony (keeping S as small as possible), or biological interpretability.

We will use the R programming language to express the prediction procedure underlying Michiels et al.'s random validation method. Text formatted in monospace font can be regarded as valid programming content for R. Assume that the expression data are available in a matrix X and that the index vector S indexes the elements of the array platform belonging to a putative signature of interest. Finally, suppose P1 is an index vector identifying only samples with dichotomous phenotype P_1. Then the centroids of the cohorts with and without phenotype P_1 are obtained, respectively, as

```
C1 = apply(X[S,P1],1,mean) and C2 = apply(X[S,-P1],1,mean).
```

For any array x obtained from an individual whose phenotype is unknown, Michiels (p. 489) proposes predicting phenotype P_1 if and only if `cor(x[P,1],C1) > cor(x[P,1],C2)`.

We now provide an excerpt of runnable code implementing the procedure (suppressing details of call and return interfaces). XCA and XCO are submatrices of the $G \times N$ normalized and filtered expression data matrix corresponding to cases and controls, respectively. s( ) is a function returning a signature (gene list) on the basis of T training columns from expression data matrix X and the associated values from the dichotomous phenotype class vector P taking values "+", "−". `p()` is a function predicting a response vector of N–T "+" and "−" on the basis of an expression

matrix representing *N–T* test (or validation) samples. `Tseq` is a sequence of training set sizes *T*, assumed increasing from a minimum value of 10. NRUN is the number of signatures to be obtained through randomly sampling the training set for each value of *T*.

```
for (i in 1:length(Tseq)) {
   ALLS[[i]] = list()  # storage for signatures
   MC[[i]] = rep(NA,NRUN)  # storage for misclass.rates
   T = Tseq[i]
   FTH = floor(T/2)
   if (FTH > min(c(NCA,NCO))-1) break # leave at least 1 of each in
#test set
   for (j in 1:NRUN) {
     TR1 = sample(1:NCA, size=FTH, replace=FALSE)
     TR2 = sample(1:NCO, size=FTH, replace=FALSE)
     XTR = cbind(XCA[,TR1], XCO[,TR2])  # balance cases and controls
in # train
     XTE = cbind(XCA[,-TR1],XCO[,-TR2])
     PTR = rep(c("+", "-"), c(FTH, FTH))
     PTE = rep(c("+", "-"), c(ncol(XCA)-FTH, ncol(XCO)-FTH))
     S = ALLS[[i]][[j]] = s(XTR, PTR)
     MC[[i]][j] = mean(p(XTE,S,XTR,PTR) != PTE)
  } }
```

For `s()`, Michiels et al. propose selecting the 50 genes possessing high Pearson correlations with the phenotype vector over all training samples. For `p()`, Michiels et al. propose the centroid correlation comparison procedure noted above. The quantities S, XTR and PTR can be used to construct the phenotype-specific centroids for comparison with test signature values.

Michiels' assertion about instability of signature identity arises through tabulating frequencies of genes selected among the top 50 differentially expressed in each of the 500 iterations of the loop given above (i.e., by inspecting the frequency table of the contents of ALLS). The assertion about the dependence of misclassification rate on training set size arises through inspection of the trajectory of the mean of MC in neighborhoods defined by ALLT. The assertion that a signature does not classify patients better than chance arises through comparison of the trajectory of MC and the line MC = 0.5.

Some aspects of the procedure seem arbitrary (number of genes in signature and criterion for selection), and Michiels et al. remarked:

> We did a sensitivity study using other strategies to identify signature genes: selection of the 20 or 100 most discriminating genes (instead of 50) or selection of all genes with a significant correlation ($p < 0.01$) between expression and outcome.

These variations in analytic strategy were asserted to have no qualitative impact.

8.2.2.4 Reconstructing Michiels' Results

Michiels et al. do not employ explicitly reproducible research methods. Reconstruction involves three main efforts as follows.

- *Data acquisition*. Institutional distribution allowed recovery of data underlying van't Veer (http://www.rii.com) and Pomeroy (Broad Institute). Previous work on packaging Yeoh's dataset can be obtained from an R package at http://packages.sgdi.org (Carey et al. 2008).
- *Algorithm implementation*. The *michPack* R package includes source code and data images relevant to this part of this paper. This package will become an element of the experimental data archive distributed at http://www.Bioconductor.org.
- *Representation of previous results to support comparison*. Their primary quantitative results of Michiels et al. are presented graphically. To compare our reconstruction with the published results, graphs were extracted from a PDF rendering of Fig. 2 of Michiels' original paper to TIFF using MacOSX *grab* and imported to R using Bioconductor's *EBImage*. Manual landmarking was conducted to map from the image space to data space.

Figure 8.2 gives the results of our attempt to reconstruct the results of Michiels et al. The top left panel confirms the instability assertion for the van't Veer data. Only a fraction of genes in the 70 gene signature are persistently present in the random validation strategy signatures. The top right panel depicts reproducibility for the van't Veer analysis. The Fig. 2 panel from Michiels' original paper is overlaid with data generated in the reconstruction based in part on the code given above. Michiels' estimated relationship between misclassification rate and training set size is plotted as an orange curve; the lowess scatterplot smooth for the reconstruction data tracks Michiels' result nicely. Similar reconstruction holds for the Yeoh data set (bottom left panel), but not for the Pomeroy data set (bottom right panel).

8.2.2.5 Summary of Reproducibility of Michiels et al.

Reconstruction of Michiels' procedure from the prose description is not too difficult and appears to have been successful on the basis of the van't Veer and Yeoh results. Failure to reproduce with Pomeroy is significant, as that dataset was one of the studies that Michiels et al. regard as providing a signature that does not classify patients "better than chance."

Since we now have reasonably well-designed software implementing this procedure, we can explore its implications and limitations. For example, we can compare its performance in simulation to more common approaches such as V-fold cross-validation. We can substitute alternate classification algorithms. Ideally, we could use the data resources and algorithms assembled in the research to make constructive proposals about signature identification and reliability. This will be explored below.

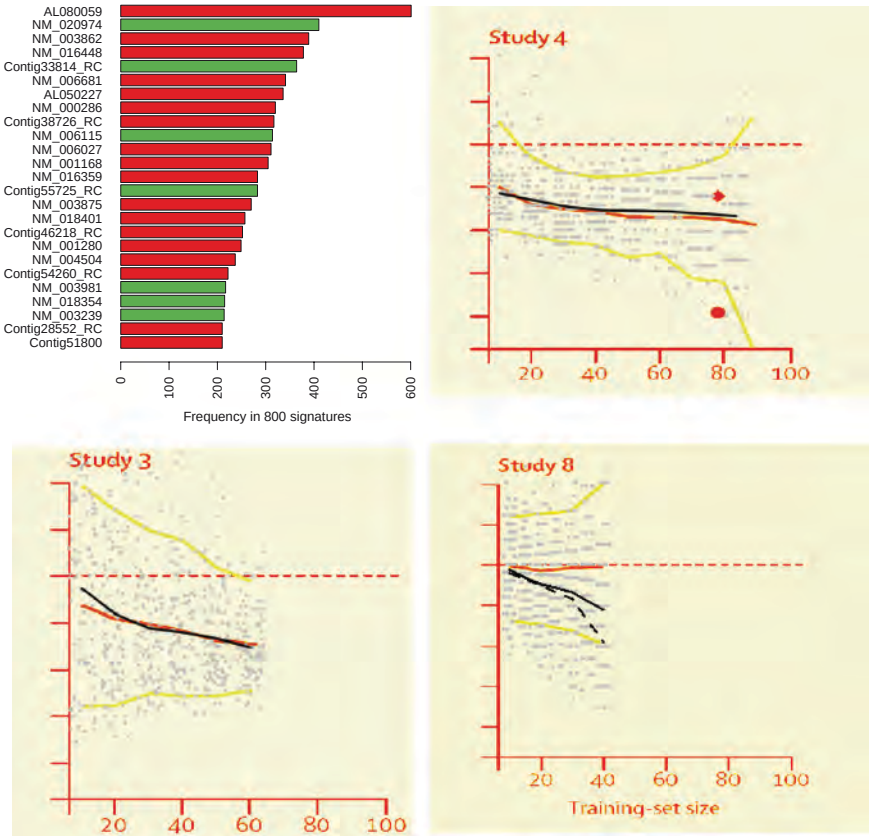


Fig. 8.2 *Upper left*: Top 25 frequencies of probes appearing in 800 signatures (8 training set sizes, 100 signatures per size) in Michiels' random validation strategy applied to van't Veer's data. *Bars* colored *green* are probes appearing in the 70 gene signatures published by van't Veer. *Upper right*: Michiels' misclassification trajectory display overlaid with data generated in this reproducibility exercise. *y*-axis is estimated misclassification rate with dashed line at 50%; *x*-axis is training set size *T*. The *black curve* is R's default lowess scatterplot smooth estimate of MC(*T*). *Lower left*: misclassification trajectory display for Yeoh's 2002 leukemia study. *Lower right*: display for Pomeroy's 2002 medulloblastoma dataset. The *solid line* is the estimate of MC(*T*) for randomly trained signature size 50; the *dashed line* is the estimate for randomly trained signature size 8

8.3 Illustrations of Reproducible Research Discipline

In this section, we describe specific implementations and impacts of reproducible research discipline in application to the case studies. We focus on data and algorithm availability. This section is somewhat technical and assumes familiarity with the R programming language. Readers who are interested primarily in substantive and procedural issues related to reproducibility can skip to Sect. 8.4, where a standard for reproducible research and its legal underpinnings are described.

8.3.1 Reproducible Discipline for Primary Research Papers

An R package is an attractive alternative to a "supplemental Web site" for promoting reconstruction and exploration of primary research. In connection with Dressman's paper, we have created the *dressCheck* package that provides concrete and self-describing representations of the key elements for analysis. When we issue the command `data (package="dressCheck")`, we see

```
Data sets in package 'dressCheck':

DrAsGiven      Quantifications of 119 ovarian cancer samples
          as distributed at Duke's platinum.php
E2F3sig.probes Affymetrix probe identifiers for the E2F3
                pathway signature of Bild et al.
Src.probes     Affymetrix probe identifiers for the Src
                pathway signature of Bild et al.
corrp116       Dressman's RMA+SFR corrected quantifications
                for 116 ovarian cancer samples
e2f3Wts        Coefficients to weight components of E2F3
                pathway to score for activity, derived from SVD
                applied to Bild's data
platsigprobes affymetrix probe identifiers for the platinum
                responsiveness signature
pwLines        A representation of Bild's HMEC lines for
                pathway signature identification
srcWts         Coefficients to weight components of Src
          pathway to score for activity, derived from SVC
          applied to Bild's data
```

These objects, whose names are completely arbitrary, encode the following components of the study.

1. *The primary and secondary quantitative data.* In this case, we use the Bioconductor `ExpressionSet` container, that binds together expression array results, sample level information, and experiment level metadata (Gentleman et al. 2004). The name `corrp116` is used to indicate which image of the ovarian cancer data we are working with among the many possible in this particular instance. Bild's array data on HMEC lines that have been perturbed to induce pathway activation are also contained here in ExpressionSet `pwLines`. In the following, we use a stylized formatting approach to distinguish commands entered to R (*`slant monospace`* font, preceded by >) and information returned by the R interpreter (`normal monospace` font). Here is how we get access to the corrected expression data:

```
> library(dressCheck)
> data(corrp116)
> corrp116

ExpressionSet (storageMode: lockedEnvironment)
assayData: 22115 features, 116 samples
 element names: exprs
phenoData
 sampleNames: 1665, 2505, ..., 860 (116 total)
 varLabels and varMetadata description:
 fixedid: NA
 OVC.TumorID: NA
 ...: ...
 rundate: NA
 (14 total)
featureData
 featureNames: 1007_s_at, 1053_at, ..., 222384_at (22115 total)
 fvarLabels and fvarMetadata description: none
experimentData: use 'experimentData(object)'
```

2. *Identities of signature elements.* The study involved discovery or knowledge of signatures of platinum responsiveness (platsigprobes) or pathway activation (e.g., Src.probes). These can be specified unambiguously as character vectors; if more metadata is desired, the *GSEABase* GeneSet container could be employed.
3. *Coefficients for sample scoring.* Named numeric vectors can store in a self-describing fashion the coefficients (e.g., `e2f3Wts`) used to determine whether or not a sample has evidence of pathway activation. This facility could also be provided in the form of a function.

To illustrate the benefit of such an integrated container, we show the code to qualitatively compare sample discrimination using transcriptomic signatures in two contexts: Dressman's ovarian cancer application and van't Veer's breast cancer application. These constitute multivariate "sanity checks" of the concept.

First we acquire the necessary resources. We will load data images for Dressman's samples as `corrp116` and for van't Veer's as `vv97`. Concise reports on these data structures are obtained upon mentioning them to the R interpreter.

```
>   library(dressCheck)
>   data(corrp116) # N=116 samples with good labels
>   library(vantveerSubset)
>   data(vv97) # N=97 samples assembled from rii.com
>   vv97

ExpressionSet (storageMode: lockedEnvironment)
assayData: 25592 features, 97 samples
   element names: exprs
phenoData
   sampleNames: Sample.111, Sample.112, ..., Sample.79 (97 total)
   varLabels and varMetadata description:
Sample..: NA
age: NA
...: ...
Brca1.mutation: NA
(11 total)
featureData
   featureNames: Contig45645_RC, Contig44916_RC, ..., Contig15167_RC
(25592 total)
   fvarLabels and fvarMetadata description: none
experimentData: use 'experimentData(object)'
```

We will use the principal components re-expression of the expression data for the predictive signatures published by each of these authors. The signatures are saved in different objects in each package:

```
>   data(platsigprobes)
>   platsigprobes[1:3]

       NEU1        TPR1        TPR2
"208926_at" "1557227_s_at"  "201730_s_at"

>   length(platsigprobes)

[1] 246

>   data(vvsig70)
>   length(vvsig70)

[1] 70

>   vvsig70[1:5]

[1] "NM_020974"   "NM_003882"   "Contig48328_RC"
"Contig46223_RC"
[5] "NM_003239"
```

We can use the X[G,S] idiom of Bioconductor to subset the basic containers to probe sets in the signatures. In the case of Dressman's quantifications, not all signature elements are available.

```
>   okdp = intersect(featureNames(corrp116), platsigprobes)
>   drSigEx = exprs(corrp116)[okdp, ]
>   vvSigEx = exprs(vv97)[vvsig70, ]
```

The van't Veer data includes missing values. We will assume that these correspond to unit ratios (log value 0):

```
>   vvSigEx[is.na(vvSigEx)] = 0
```

Finally we will get vectors of colors for sample states:

```
>   drCol = ifelse(corrp116$CR == 1, "green", "red")
>   vvCol = ifelse(vv97$metast == 0, "green", "red")
```

We compute the principal components analyses:

```
> drpc = prcomp(t(drSigEx))
> vvpc = prcomp(t(vvSigEx))
```

Figure 8.3 shows the results. In the scatterplot of the first two principal components, the separation of sample states for van't Veer's data is not perfect, but clear regions of predominance by sample type are apparent. The configuration is much more complicated for Dressman's data, for which we display the first three principal components.

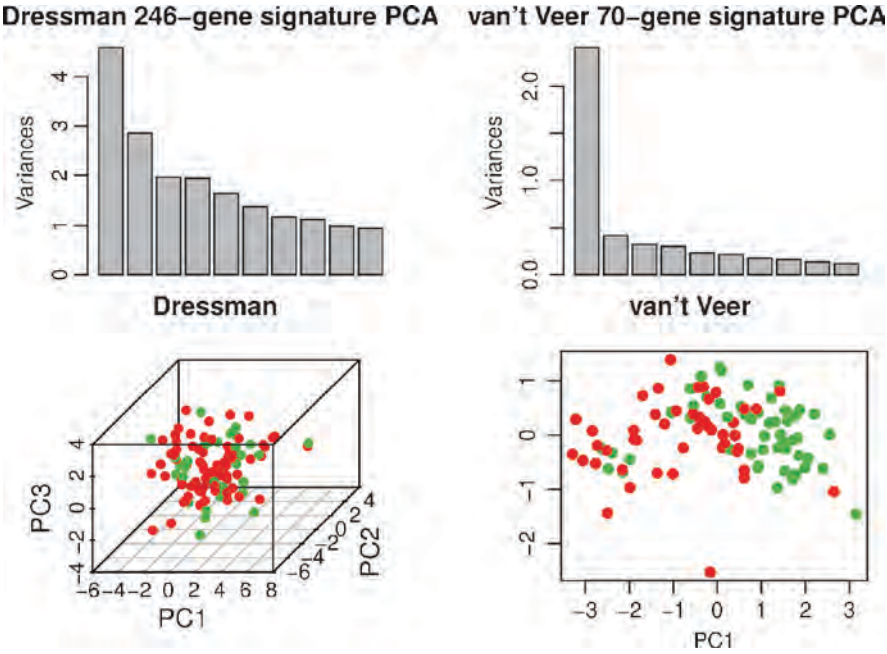


Fig. 8.3 Multivariate sanity check of transcriptomic signatures. *Left*: Principal components orientation of 116 ovarian tumor samples employing 246 genes constituting Dressman's platinum responsiveness signature. *Right*: Principal components orientation of 97 breast tumor samples employing genes constituting van't Veer's metastasis signature. *Green dots* are favorable outcomes, *red* unfavorable

8.3.2 Reproducible Discipline for Methods and Criticism

As Baggerly, Coombes, and Neeley have illustrated, careful criticism of complex analyses is itself a highly complex undertaking and risks of misinterpretation of primary work must be minimized. In fact, undertaking a reproducible investigation of an apparently nonreproducible primary work may involve more effort than the original study.

For Michiels' analysis of Pomeroy's data, it would be helpful to have direct access to the data used by Michiels. We have created packages to manage Michiels' algorithm components (*michPack*) and Pomeroy's published data (*pomeroy*).

As we have seen some discrepancy between our reconstructive work and the results of Michiels on the Pomeroy data, let us push further on this dataset. We load the packages and check the basic assertion of Pomeroy et al. that a misclassification rate of 13/60 was achieved with an eight-gene signature using leave-one-out (LOO) cross-validation with *k*-nearest neighbors (NN) classification. In their "Supervised learning" methods section, Pomeroy et al. indicate that they use $k=5$ with a

weighted nearest-neighbor algorithm. We do not have access to this weighted procedure, so use the unweighted one provided in R's *class* package.

```
>   library(pomeroy)
>   library(MLInterfaces)
>   data(pomes) # the 60 samples
>   data(failsig) # two 4-gene signatures
>   data(survsig)
>   survsig

[1] "L06419_at" "J02611_at" "D86974_at" "U37673_at"

>   pd = pomes[ c(failsig, survsig),]
>   k1 = MLearn( Current.status ~ .,   pd,   # use the 8 genes in
LOO
+    knnI(k=5, l=0), xvalSpec("LOO"))
>   confuMat(k1)

      predicted
given    D     A
 A       4     35
 D      10     11
```

We see 15/60 misclassifications, probably close enough given the unweighted approach. This is likely an underestimate of the true misclassification rate because the role of the given data in driving the choice of the signature is not factored in.

There are three features of Michiels' algorithm, which attempts explicitly to address the uncertainty connected with data-driven signature selection, that depart from the analysis just shown. First, Michiels uses centroid correlation as opposed to *k*-nearest neighbors. Second, Michiels forces a signature size of 50 genes. Third, Michiels requires that the training set be balanced on classes to be predicted and that the test set include at least one element of each class. These constraints are incompatible with leave-one-out cross-validation.

It is easy to show that centroid correlation combined with LOO cross-validation yields a misclassification rate for the eight-gene signature that is considerably below 50%:

```
>   library(michPack)
>   pdstat = ifelse(pd$Current.status == "D", "+", "-")
>   table(pdstat)

pdstat
   +   -
  21  39

>   pdex = exprs(pd)
>   micxv = sapply(1:60, function(i) p( pdex[,i,drop=FALSE], 1:8,
+    pdex[,-i], pdstat[-i]))
>   table (pdstat,micxv)

    micxv
pdstat       +    -
  +         15   6
  -          9  30
```

Thus the discrepancy between classifier types used by Michiels and Pomeroy is probably not very important. The dashed line in the lower right panel of Fig. 8.2 shows the impact of reducing the signature size of Michiels' procedure to 8 from 50.

When employed with a signature of size 50, Michiels' procedure seems to force overfitting in the Pomeroy application, and the sensitivity analysis described in Sect. 8.2.2.3 was inadequate to expose this.

In summary, Michiels' results on Pomeroy are not reconstructible, and we cannot explain why, because neither the data nor the software used to generate the figure is available. The suggestion that a valid estimate of the misclassification rate of Pomeroy's signature is close to 50% seems to be a misrepresentation. When reproducible discipline is followed so that data and algorithms are available for dissection and reinterpretation of unusual results, it becomes harder for such misrepresentations to remain in the literature without correction. Note that Michiels paper has, as of April 29, 2009, been cited 270 times (ISI Web of Science), with citations repeating concerns about "well-documented" signature instability, divergent results, and, in one case, indicating that microarray-based findings are "not robust to the mildest of perturbations" (Ramasamy et al. 2008).

8.3.3 Summary of Illustrations

We have focused on the value of employing R packages of various types of data and software for the implementation of reproducible research discipline.

In the case of primary bioinformatic research in cancer, investment of time in the assembly of one or a small family of R packages or their equivalents nicely localizes resources that are processed into statements, tables, and graphs for publication. When a question arises about a given finding, the *resources* for reconstruction are ready at hand. The actual *path* to computational reconstruction can take various forms – a script that may also be stored in the package or an integrated document that includes narration, code, and graphics. The use of Bioconductor containers helps ensure that interrogation and filtering processes have conventional implementations and allows use of established interfaces for generic processes like linear modeling or machine learning. These containers are also designed to minimize the risk of sample label scrambling or class confusion that were established or appear to be present in the Dressman analyses.

For methodologic research in cancer bioinformatics, adoption of reproducible discipline is also quite valuable. Questions about properties of procedures that have been established through simulation or applications to published data should have definite answers. For this to be the case, the actual data (or simulated data streams) need to be available, and the specific software tools employed need to be identified precisely.

Reproducible research discipline methods noted here address explicitly the problems of fallibility and necessity for revision that constantly crop up in applied science. Nontrivial experiments are subject to performance errors. Nontrivial software contains bugs. When these are found, results need to be regenerated with the revised resources. When the underlying resources are properly

stamped with version numbers, the evolutionary state of a research project can be identified and managed.

8.4 Legal Frameworks for Promoting Reproducible Research

In this section, we shift the focus away from motivations for reproducible research discipline and technical details of implementation. Our focus now is exploration of how researchers' control over use of scientific research results can be managed in the context of reproducible research discipline. Serious concerns can arise regarding loss of authorship accreditation and competitiveness through redistribution and reuse of data and software that have been made available through reproducible discipline. Copyright and licensing procedures are now reviewed to address these concerns.

Under US law, original expressions of ideas are accorded copyright protection by default. As discussed subsequently, the protection that copyright provides is not suitable for scientific research, and there are steps that scientists can take not only to realign the legal protection of their work with scientific norms, but also to extend attribution protection to their work.

8.4.1 What Protection Falls on Scholarship by Default?

Copyright is designed to protect original expressions of ideas and "follows the author's pen across the page" (von Hippel 2006). There is no action that an author must take to secure copyright over his or her work. It is possible to view US copyright law as a barrier to the sharing of scientific scholarship since it establishes exclusive rights for creators over their work, thereby limiting the ability of others to copy, use, build upon, or alter the research. This is precisely opposite to prevailing scientific norms, which provide both that results be replicated before accepted as knowledge and that scientific understanding be built upon previous discoveries for which authorship recognition is given. Copyright protection is essentially indefinite, lasting the length of the author's lifetime plus 70 years. As Larry Lessig states with respect to sharing creative works, "[t]he essence of copyright law is a simple default: No. For many creators, the essence of the creativity is: Of course." (Lessig 2008, p. 277.)

There are some exceptions to the rights ascribed to creators under copyright. In an academic setting, the most important one is the doctrine of "fair use." Fair use is a safety valve that permits use of copyrighted material in certain settings when previous permission has not been obtained from the copyright holder. In the statute fair use, claims can be made "for purposes such as criticism, comment, news reporting, teaching (including multiple copies for classroom use), scholarship, or research" and are not infringements of copyright (17 U.S.C. §107). At first glance, this seems an ideal workaround for the sharing of academic research, but in fact

what comprises “fair use” is not at all clear a priori. Courts must determine whether the use is fair on a case-by-case basis, driven by the facts in each case. There is not a clear application of the doctrine to scholarship and it is difficult for users to know when their use of a work would be called fair by the courts.

Lessig founded Creative Commons in 2001 to give creators of artistic works the ability to allow others to freely use and reuse their creation under terms they set. Creative Commons provides a suite of licenses that give terms of use for work that differ from, and are usually more permissive than, the default copyright. The licenses express the relevant freedoms in three separate layers: A “commons deed” that describes the associated freedoms in a form easily understood by people; “Legal Code” that forms the actual copyright license; and the meta-data that surrounds the content to express the freedoms contained in the copyright license in a machine readable way.

To bring clarity to the terms of use of scientific work, a similar idea can be applied.

8.4.2 The Reproducible Research Standard

The Reproducible Research Standard (RRS) is a licensing framework that allows scientists to communicate terms under which their scholarship can be used and to do so in a uniform way that reflects our common scientific norms (Stodden 2009). The RRS releases all aspects of scholarship, with the exception of raw facts, so that they can be copied, verified, and built upon when attribution is given. Building on the ideas behind Creative Commons, and using existing open-source licenses for code, if work satisfies the following four conditions, it can be marked as reproducible, under the RRS.

1. The full compendium is available on the Internet
2. The media components, possibly including any original selection and arrangement of the data, are licensed under the Creative Commons attribution license (CC BY; http://creativecommons.org/licenses/by/3.0/) or released to the public domain under Creative Commons CC0 standard (http://creativecommons.org/license/zero/)
3. The code components are licensed under one of Apache 2.0, the MIT License, or the Modified BSD license, or released to the public domain under CC0.
4. The data have been released into the public domain according to the CC0 license.

In joint work with Science Commons, one of the authors (Stodden) is developing a mechanism that would allow scientists to assert their compliance with these conditions, and publicly certify their work as reproducible. Since it may not be feasible for a scientist to satisfy all these conditions because of reasons beyond his or her control (such as privacy considerations for data), we propose different levels of compliance. If the work is publicly released and capable of being reproduced, it is marked as *Verifiable*. If the work has been verified by someone working independently, it can be marked as *Verified*. When the full compendium is only

partially released, the work can be marked as *Semi-Verifiable*, and if the code has been verified or the results are achieved in a different dataset (depending on which components were not released), the work is marked as *Semi-Verified*.

The RRS provides a way for scientists to signal the ability of others to legally download their work, copy it, rerun code, and manipulate data, with the requirement that attribution is given in any publication that depends on the original work. Without such a licensing structure, a scientist wishing to build on an author's work would need to obtain permission because of the default attachment of copyright protection to the work.

8.4.2.1 Data Under the RRS

Copyright law in the USA does not permit the copyrighting of *raw facts* but original products derived from those facts do fall under copyright, qualifying as original expression. In the US Supreme Court case *Feist Publications, Inc. v. Rural Telephone Service*, the Court found that the white pages from telephone directories are not themselves directly copyrightable, since copyrightable works must have creative originality, but "original selection and arrangement" of the data does fall under copyright automatically [499 U.S. 340 (1991)]. Attaching an attribution license, such as the Creative Commons BY license, to the original "selection and arrangement" of a database can encourage scientists to release the datasets they have created by providing a legal framework for attribution and reuse of the original selection and arrangement aspect of their work. Since the raw facts themselves are not copyrightable, it does not make sense to apply such a license to the data themselves. The selection and arrangement may be implemented in code or described in a text file accompanying the dataset, either of which can be appropriately licensed as suggested for code in the RRS. The RRS recommends that raw facts be released to the public domain using the Creative Commons CC0 certification. For details on the CC0 protocol, see http://creativecommons.org/press-releases/entry/7919.

8.4.3 RRS and Cancer Bioinformatics Research

In Sect. 8.2.2 above, we reviewed Michiels et al.'s reanalysis of seven microarray studies. Michiels chose the seven studies for inclusion in his work because of the public availability of the data. Michiels is not trying to reproduce results in the seven selected papers, but trying to establish a benchmark procedure to verify results obtained in correlation-based microarray studies. If the previous seven studies had used one of the licenses recommended by the Reproducible Research Standard for their code, Michiels could have used this to verify their results directly. The Michiels paper itself could be more readily subject to scrutiny if he had released the code used in the approach to testing the robustness of microarray findings in a compendium adherent to the RRS.

This example also illustrates ways in which the RRS affects interactions between publishers and scientists. Michiels published this paper in Lancet, and presumably the paper is under copyright (it is only accessible via subscription). If a copyright agreement has been signed with a publisher, this makes it impossible to apply a CC BY license to the final paper unless Lancet, the presumptive copyright holder, chooses to do so. In contrast, the code used by Michiels can be released under the RRS and so can any original selection and arrangement of the data. Microarray studies are poised to be on the forefront of reproducible research because of the established standards for data release upon publication. As Michiels notes, "Leading scientific journals require investigators of DNA microarray research to deposit their data in an appropriate international database, following a set of guidelines (Minimum Information About a Microarray Experiment)." With the release of *code* associated with published papers, the field could become the first truly reproducible discipline.

8.4.3.1 Benefits of the RRS

The RRS gives scholarship a consistent licensing structure while providing legal protection to scientists who wish to build upon others' code, text, and original selections and arrangements of data. The RRS communicates to other researchers not only that they are free to use the work, but also that they must attribute the work if they use it. But it also communicates something deeper: that the reproducible researcher ascribes to the scientific norms of sharing and to the replicability of results.

Reproducibility is essential for the verification of computational results, particularly as computation becomes a pervasive way to conduct scientific research. Inspection of code can communicate details of analysis and data processing that might not be readily expressed in a published paper and can allow an interested reader to vary parameter settings and explore the research more deeply. Without reproducibility, the credibility of results and of research in the field is jeopardized (Donoho et al. 2009).

The immediate benefits for the individual scientist from applying the legal framework of the RRS to scientific work are threefold. First, attribution is explicitly stated as a condition of use. Openly released work is apt to be cited more frequently than work that is not released, and affirming this scientific norm can help an author gain citation for their work. Attribution for data use can be communicated through the RRS, since an attribution license can be applied to the original selection and arrangement of the data. As will be discussed in more detail in Sect. 8.4.3.2, attribution can become machine readable and hopefully provide not only a further incentive to attribute correctly, but also create a mechanism to track scientific contributions more accurately. Collaboration in dataset creation, for example, can be tracked more accurately.

Second, the RRS can offset costs to full releasing of scholarship by making it public that your work ascribes to the principle of reproducible research. Using a logo, a scientist will be able to mark his or her work as reproducible and signal

adherence to this way of doing research. Consistent licensing practices also permit the mixing of code from different sources, obviating possible clashes in licensing terms that can arise from the use of licenses not specified by the RRS.

Third, working reproducibly encourages better science from the inception of the project. In the words of one proponent, he committed to principles of reproducibility because he "didn't trust himself to turn out good work unless his results were subject to the open criticism of other researchers" (Donoho et al. 2009). Being able to recreate your own results creates a check on quality of work being carried out. Open code and data also serve to promote your work to potential coauthors and collaborators, future employers and graduate students, and anonymous referees and make it easier for postdoctoral fellows to hit the ground running in a new lab environment.

8.4.3.2 Costs of the RRS

The RRS requires action on the part of the scientist to provide terms of use other than automatically assigned by copyright. The scientist needs to certify his or her work as reproducible.

The licensing structure of the RRS enforces attribution in a legal sense, as required by the licenses chosen. For the media component, it appears possible that this is substitutable with academic citation, but the code licenses typically require listing contributors in a file accompanying the work. Attribution tracking through this file, or through tags on the .html files where the code is located, does create extra work for the scientist working within the RRS framework.

Recent research has indicated that the more a scientist shares his or her work, the more citations the work garners. This is an indirect way to recoup the costs of working reproducibly. With wider use of the RRS, a framework is created for the discussion of reproducible research as an entity deserving of academic reward in itself. Although it requires extra work to be reproducible, promotion and hiring committees can discern work that was released in accordance with the standard of the RRS and through a machine-readable attribution system, those determining academic reward can search to discover the contributions a particular scientist has made, not only through publication but also through code and data and adherence to norms of replicability.

8.5 Conclusions

When concrete reproducibility of research results is established, questions of validity or robustness of research claims can be investigated immediately. Scientific curiosity regarding limitations and possible extensions of published results can also be addressed rapidly. This is because all primary data and software that are used to interpret the data are all published and documented in the reproducible research discipline.

In the case of Dressman et al. (2007), questions of validity were carefully addressed with a very thorough and extensive collection of datasets and documents as described in the letter of Baggerly et al. (2008). A more compact and self-describing approach to investigation is feasible in the form of the *dressCheck* R package described in Sect. 8.3.1. Had Dressman et al. published a comparable archive of quantifications, portable algorithms, and scripts yielding their primary findings, much effort could have been saved and validity of the main findings could be straightforwardly checked.

In the case of Michiels et al. (2005), several claims seem reproducible in the absence of adoption of concrete reproducible discipline by these authors. However, one strong claim about bias in estimation of the misclassification rate of the medulloblastoma signature of Pomeroy et al. is not reproducible.

These observations indicate that current approaches to "supplemental documentation" of complex research undertakings in conventional publication are insufficient to ensure reliability and extensibility of published results. Probably the strongest incentive to adoption of concrete reproducibility of quantitative research lies with the authors of primary investigations (such as Dressman or Pomeroy and colleagues) who naturally want to minimize the risk that their analyses will be found to be erroneous. When concrete reproducibility is secured by the primary authors, independent auditors can readily verify results prior to publication, and both the authors and the reading public benefit from such verification. Secondary researchers also benefit because the effort of "retooling" a complex data archive to facilitate reproduction of known findings can be reduced to finding and rerunning software that was known to yield the findings.

Scientists who work in areas requiring reproducible discipline must be alert to two challenges related to intellectual property control. First, reuse of published results may fall afoul of restrictions on republication engendered by copyright law. Second, adoption of reproducible research discipline may allow other investigators to benefit from reanalysis of primary data without attribution to the original creator. The emerging reproducible research standard described above gives scientists guidance and clarity concerning rights and obligations of those participating in reproducible research.

References

Baggerly KA, Coombes KR, Neeley ES (2008) Run batch effects potentially compromise the usefulness of genomic signatures for ovarian cancer. J Clin Oncol 26(7):1186–1187. doi:10.1200/JCO.2007.15.1951. URL http://www.hubmed.org/display.cgi? uids=18309960

Bild AH, Yao G, Chang JT, Wang Q, Potti A, Chasse D, Joshi MB, Harpole D, Lancaster JM, Berchuck A, Olson JA, Marks JR, Dressman HK, West M, Nevins JR (2006) Oncogenic pathway signatures in human cancers as a guide to targeted therapies. Nature 439(7074):353–357. doi: 10.1038/nature04296. URL http://www.hubmed.org/display.cgi? uids=16273092

Carey VJ, Gentry J, Sarkar R, Gentleman D, Ramaswamy S (2008) SGDI: system for genomic data integration. Pac Symp Biocomput 141–152. URL http://www.hubmed.org/display.cgi? uids=18229682

Carvalho CM, Chang J, Lucas JE, Nevins JR, Wang Q, West M (2008) High-dimensional sparse factor modeling: applications in gene expression genomics. J Am Stat Assoc 103(484):1438–1456

Donoho DL, Maleki A, Ur Rahman I, Shahram M, Stodden V (2009) Reproducible research in computational harmonic analysis. IEEE Comput Sci Eng 11(1):8–18

Dressman HK, Berchuck A, Chan G, Zhai J, Bild A, Sayer R, Cragun J, Clarke J, Whitaker RS, Li L, Gray J, Marks J, Ginsburg GS, Potti A, West M, Nevins JR, and Lancaster JM (2007). An integrated genomic-based approach to individualized treatment of patients with advanced-stage ovarian cancer. J Clin Oncol 25(5):517–525. doi:10.1200/JCO.2006.06.3743. URL http://www.hubmed.org/display.cgi? uids=17290060

Gentleman R (2005) Reproducible research: a bioinformatics case study. Stat Appl Genet Mol Biol 4. doi:10.2202/1544-6115.1034. URL http://www.hubmed.org/display.cgi? uids=16646837

Gentleman R, Lang DT (2004) Statistical analyses and reproducible research. Bioconductor project working papers 2, May 2004. URL http://www.bepress.com/bioconductor/paper2

Gentleman RC, Carey VJ, Bates DM, Bolstad B, Dettling M, Dudoit S, Ellis B, Gautier L, Ge Y, Gentry J, Hornik K, Hothorn T, Huber W, Iacus S, Irizarry R, Leisch F, Li C, Maechler M, Rossini AJ, Sawitzki G, Smith C, Smyth G, Tierney L, Yang JY, Zhang J (2004) Bioconductor: open software development for computational biology and bioinformatics. Genome Biol 5(10):R80.doi: 10.1186/gb-2004-5-10-r80. URL http://www.hubmed.org/display.cgi? uids=15461798

Hans C, Dobra A, West M (2007) Shotgun stochastic search for regression with many candidate predictors. J Am Stat Assoc 102:507–516

Ioannidis JP, Allison DB, Ball CA, Coulibaly I, Cui X, Culhane AC, Falchi M, Furlanello C, Game L, Jurman G, Mangion J, Mehta T, Nitzberg M, Page GP, Petretto E, van Noort V (2009) Repeatability of published microarray gene expression analyses. Nat Genet 41(2):149–155. doi:10.1038/ng.295. URL http://www.hubmed.org/display.cgi? uids=19174838

Irizarry RA, Bolstad BM, Collin F, Cope LM, Hobbs B, Speed TP (2003) Summaries of affymetrix genechip probe level data. Nucleic Acids Res 31(4):e15. URL http://www.hubmed.org/display.cgi? uids=12582260

Johnson WE, Li C, Rabinovic A (2007) Adjusting batch effects in microarray expression data using empirical Bayes methods. Biostatistics 8(1):118–127. doi:10.1093/biostatistics/kxj037. URL http://www.hubmed.org/display.cgi? uids=16632515

Laine C, Goodman SN, Griswold ME, Sox HC (2007) Reproducible research: moving toward research the public can really trust. Ann Intern Med 146(6):450–453. URL http://www.hubmed.org/display.cgi? uids=17339612

Lessig L (2008) Remix: making art and commerce thrive in the hybrid economy. The Penguin Press, New York, NY

Michiels S, Koscielny S, Hill C (2005) Prediction of cancer outcome with microarrays: a multiple random validation strategy. Lancet 365(9458):488–492. doi:10.1016/S0140-6736(05)17866-0. URL http://www.hubmed.org/display.cgi? uids=15705458

Peng RD, Dominici F, Zeger SL (2006) Reproducible epidemiologic research. Am J Epidemiol 163(9):783–789. doi:10.1093/aje/kwj093. URL http://www.hubmed.org/display.cgi? uids=16510544

Picard RR, Berk KN (1990) Data splitting. Am Stat 44:140–147

Pomeroy SL, Tamayo P, Gaasenbeek M, Sturla LM, Angelo M, McLaughlin ME, Kim JY, Goumnerova LC, Black PM, Lau C, Allen JC, Zagzag D, Olson JM, Curran T, Wetmore C, Biegel JA, Poggio T, Mukherjee S, Rifkin R, Califano A, Stolovitzky G, Louis DN, Mesirov JP, Lander ES, Golub TR (2002) Prediction of central nervous system embryonal tumour outcome based on gene expression. Nature 415(6870):436–442. doi:10.1038/415436a. URL http://www.hubmed.org/display.cgi? uids=11807556

Ramasamy A, Mondry A, Holmes CC, Altman DG (2008) Key issues in conducting a meta-analysis of gene expression microarray datasets. PLoS Med 5(9):e184. doi:10.1371/journal.pmed.0050184. URL http://www.hubmed.org/display.cgi? uids=18767902

Stodden V (2009) Enabling reproducible research: licensing for scientific innovation. Int J Commun Law Policy 13(1):1–25

van 't Veer LJ, Dai H, van de Vijver MJ, He YD, Hart AA, Mao M, Peterse HL, van der Kooy K, Marton MJ, Witteveen AT, Schreiber GJ, Kerkhoven RM, Roberts C, Linsley PS, Bernards R, Friend SH (2002) Gene expression profiling predicts clinical outcome of breast cancer. Nature 415(6871):530–536. doi:10.1038/415530a. URL http://www.hubmed.org/display.cgi?uids=11823860

Vandewalle P, Kovacevic J, Vetterli M (2009) Reproducible research in signal processing – what, why, and how. IEEE Signal Process Mag 26(3):37–47. URL http://rr.epfl.ch/17/

von Hippel E (2006) Democratizing innovation. MIT, Cambridge, MA

Yeoh EJ, Ross ME, Shurtleff SA, Williams WK, Patel D, Mahfouz R, Behm FG, Raimondi SC, Relling MV, Patel A, Cheng C, Campana D, Wilkins D, Zhou X, Li J, Liu H, Pui CH, Evans WE, Naeve C, Wong L, Downing JR (2002). Classification, subtype discovery, and prediction of outcome in pediatric acute lymphoblastic leukemia by gene expression profiling. Cancer Cell 1(2):133–143. URL http://www.hubmed.org/display.cgi

Chapter 9
The Cancer Biomedical Informatics Grid (caBIG®): An Evolving Community for Cancer Research

J. Robert Beck

Abstract This chapter describes the Cancer Biomedical Informatics Grid (caBIG®) project, a collaboration among the National Cancer Institute (NCI) and numerous discovery organizations, mostly NCI-designated cancer centers. The rationale for the caBIG® project opens the review, followed by description and analysis of the pilot phase (2004–2007). Changes in caBIG® as it transitions to an enterprise phase are evaluated. The chapter concludes with a review of important papers that either describe caBIG® components or are based on the initiative.

9.1 Introduction

Biomedical research, in particular cancer research, has exploded in the past half century, since the tenure of James Shannon as director of the National Institutes of Health. By any metric the breadth and depth of scientific information has expanded beyond any individual's ability to monitor. Valiant and largely successful attempts to cope with the information explosion have arisen in parallel. The *Index Medicus*, a comprehensive index of the published medical literature begun in 1879 by John Shaw Billings of the Surgeon General's Office, was automated as a time-sharing service in the 1960s as MEDLINE. The print *Index Medicus* ceased publication in 2004, as the PubMed search engine for MEDLINE and other online resources had rendered it obsolete.

Today the published biomedical literature is but the tip of an iceberg of usable data and information. Gene sequences, proteomic signatures, available clinical trials, and social networks are but a few of the information resources that are both more immediate and more difficult to navigate than the journals. Researchers, clinicians, and patients have recognized this knowledge explosion as both a boon and a bane to progress against disease in the twenty-first century. In cancer research, high-throughput

J.R. Beck (✉)
Fox Chase Cancer Center, 333 Cottman Avenue, Philadelphia, PA 19111, USA
e-mail: j.robert.beck@fccc.edu

M.F. Ochs et al. (eds.), *Biomedical Informatics for Cancer Research*,
DOI 10.1007/978-1-4419-5714-6_9, © Springer Science+Business Media, LLC 2010

technologies and systems biology have characterized “cancer” as a collection of complex diseases and syndromes with distinctive genetics and molecular signatures. While the challenges in controlling cancer are therefore increased, so too are the opportunities for targeted, individualized diagnostics and therapeutics.

In 2003 the Director of the United States’ National Cancer Institute (NCI), Andrew von Eschenbach, recognized the challenges associated with linking the cancer research community to promote interaction and collaboration necessary to continued progress in the field. In boldly envisioning a world without suffering and death due to cancer, he proposed an initiative to support collaborative research through informatics: a Cancer Biomedical Informatics Grid (caBIG®) built upon modern computational architectures and utilizing the talents of developers and scientists both within the NCI and throughout the academic research community (Kaiser 2004; Buetow 2005). The caBIG initiative, which quickly expanded to include industry, nonacademic cancer centers and the patient advocacy community, launched in early 2004. This chapter reviews caBIG®’s onset, pilot phase, and current enterprise implementations from the institutional perspective.

9.2 caBIG Prelaunch: 2003

In July 2003 the NCI announced its intention to create a cancer-based biomedical informatics network. The stated goal of this effort was to build a biomedical informatics network that would connect cancer research related elements of data, tools, individuals, and organizations, and that would leverage multiple foci of expertise. The overarching vision was that caBIG® would help redefine how research is conducted, care is provided, and patients and participants interact with the biomedical research enterprise. Thus from the outset of the initiative, caBIG® exhibited a commitment to patient care as well as research, and also embraced the cancer advocacy community.

The caBIG® leadership, vested in the Center for Bioinformatics at the NCI, envisioned the initiative as a sequence of phases. An initial pilot phase would enable the feasibility of the grid to be tested in a small scale before the caBIG® initiative would be made available to all members of the Cancer Center community. Conceptually it provided an opportunity to monitor the performance of the computational architecture and redirect efforts as required to ensure the needs of Cancer Centers were met and funds invested appropriately. The specific goals of the pilot phase, presented to NCI-designated Cancer Centers in August 2003, were to:

- Illustrate that a spectrum of Cancer Centers with varying needs and capabilities can be joined in a common grid of shared data, applications, and technologies
- Demonstrate that Cancer Centers, in collaboration with NCI, can develop new enabling tools and systems that could support multiple Cancer Centers
- Demonstrate that Cancer Centers will actively use the grid and realize greater value in their cancer research endeavors by using the grid

- Create an extensible infrastructure that will continue to be expanded and extended to members of the cancer research community

These goals were amended as the pilot phase took shape (see below).

The caBIG® pilot project was initiated by engaging the NCI-designated Cancer Center community in a dialog to establish a pilot network of potential participants. In July and August of 2003, the NCI Center for Bioinformatics (NCICB), with help from the NCI Cancer Centers Program, hosted informational seminars for NCI-designated Cancer Centers in Bethesda, MD and in San Francisco, CA. The objectives of the seminars were twofold: (1) to engage and update the Cancer Centers on the progress being made toward developing an integrated bioinformatics infrastructure platform supporting caBIG®, and (2) to provide a forum for disseminating information regarding the next steps for testing the feasibility of joining cancer center expertise and infrastructures into a common web of communications, data, and applications.

Following the informational seminars, onsite Cooperative Development Meetings were scheduled with Cancer Centers that volunteered to discuss their informatics-based strengths, needs, and potential contributions to the caBIG® pilot. From July to September 2003, caBIG® Project teams composed of scientific and information technology professionals from the NCI and a master contractor (Booz Allen Hamilton, BAH) met with key scientific and informatics personnel from 49 Cancer Centers. Each of the institutions submitted a pilot project summary outlining projects that they were potentially interested in pursuing in collaboration with NCICB to test the capabilities of the grid. Additionally, each institution prepared a list of capabilities not locally present that it would find valuable to obtain from the caBIG® project. The needs and possible projects were discussed at the Cooperative Development Meetings, which naturally took on the character of a project site visit from the NCI. The atmosphere at these meetings was intensified by the vagueness of the overall project plan, and by the initial information emanating from NCICB that approximately ten institutions would be selected as a pilot caBIG® development and adoption community.

Once all pilot project summaries were submitted, an intensive review process commenced to select a set of pilot projects and Cancer Center participants. During the process of selecting the pilot participants, a revised approach for the pilot structure arose. The initial vision for a Center-based approach incorporating only participants from ten Centers was replaced with an approach that "maximized Center participation and capitalized on synergies between Centers." Under this approach, common projects would be grouped together as areas called "Workspaces." Each Workspace represented an area of critical biomedical informatics need as identified by the Cancer Center Community involved in the earlier caBIG® meetings. It was envisaged that these Centers and their individual projects would become the primary mechanisms for testing the feasibility and capability of the data grid under development.

Figure 9.1 depicts the perceived needs of the Cancer Centers. All 49 visited Cancer Centers submitted pilot project summaries describing the potential projects that they wished to pursue in collaboration with NCICB. The majority of the projects submitted fell under the following categories:

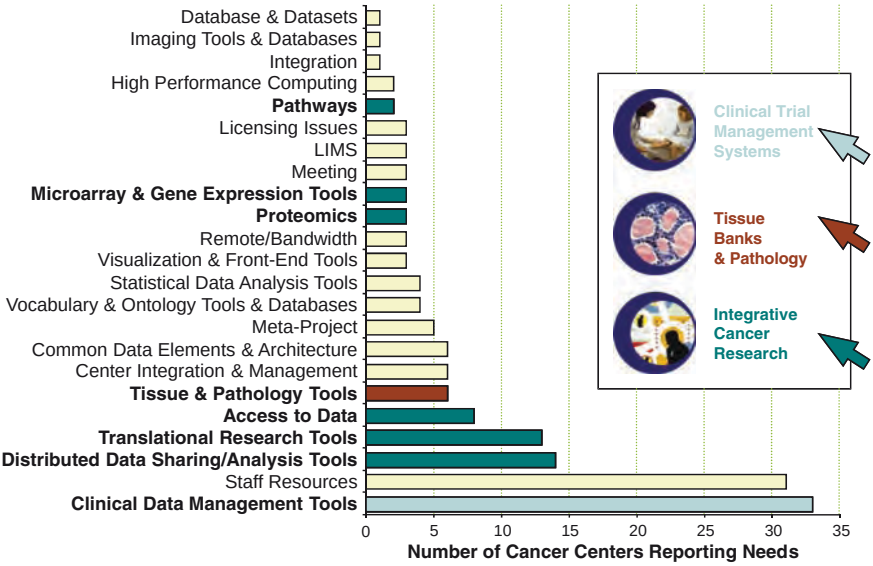


Fig. 9.1 Informatics Needs Reported by Cancer Centers, 2003. Highlighted bars indicate tools or solutions referent to Clinical Trials, Integrated Cancer Research, and Tissue/Biospecimen Banking

1. *Clinical Data Management*. Example areas included:

   - Adverse events
   - Patient recruitment
   - Patient management
   - Quality-of-life tools
   - Study protocol design
   - Study monitoring
   - Internal review board management

2. *Microarray and Expression Tools*. Example areas included:

   - Array statistical analyses tools
   - Microarray quality tools
   - Microarray comparison tools

3. *Translational Research Tools*. Example areas included:

   - Middleware for basic/clinical/genomic data integration
   - Clinical data profiling and integration

4. *Vocabularies and Ontologies*. Example areas included:

   - Ontology-based free-text data extraction
   - Distributed ontology extraction

5. *Distributed and General Data Sharing and Analysis*. Example areas included:
   - Middleware to create security layers around data for export
   - Web-based tools for sharing translational research data
6. *Tissue Banks and Pathology Tools*. Example areas included:
   - Open-source tissue database systems
   - Specimen tracking systems
   - Virtual specimen database-federating tools
   - Virtual tissue repositories

Cancer Center directors, who stressed the value to all of the cancer research enterprise by establishing a larger caBIG® community at the outset, successfully challenged the idea of only ten pilot sites. The directors also expressed concern that the project focus on critical needs (such as clinical trials management) over advanced informatics development. As the pilot selection process progressed, a revised approach for the pilot structure emerged driven by the idea of key project or topic areas rather than individual Centers. Under this revised approach, Centers were grouped together as part of "Workspaces" encompassing common areas of project activities. These Workspaces would drive the 3-year Pilot Phase from 2004 through early 2007.

9.3 The caBIG® Pilot Phase: 2004–2007

The caBIG® initiative was formally launched in February 2004, with the 49 Cancer Centers, NCICB, and BAH as the community. Quickly the community grew to incorporate patient advocates from NCI's Consumer Advocates in Research and Related Activities program, and over the course of the pilot additional Cancer Centers, research organizations and corporations joined the project. The pilot phase was characterized by Workspaces, Special Interest Groups, and Working Groups. From the perspective of the Strategic Planning Working Group, organized to assist NCICB and BAH with overall direction, the activities consisted of planning, tool development, and demonstration projects.

9.3.1 caBIG® Vision, Mission, and Principles

The vision of caBIG® is to be the information network that enables all constituencies in the cancer community to collaborate in the generation and application of new biomedical knowledge to cancer research, diagnosis, treatment, and prevention. Never envisioned as an overarching organization, caBIG® was seen as a community of communities, each contributing to a self-organizing web of knowledge, data, and people. In the pilot phase this required guidance, but overall the caBIG® initiative launched with a light administrative overhead.

The mission of the caBIG® project is to provide infrastructure for creating, sharing, and using biomedical informatics tools, data, and results. This mission is to be

accomplished using a computational network, caGrid, which exhibits aspects of a data grid and a community grid. The two components of this mission reflect its technological underpinnings: connection should be via a sharable, interoperable infrastructure, and a common language and toolset should be developed to facilitate information sharing.

At its first strategic planning meeting, the community endorsed four fundamental principles that would underlie caBIG®'s operations and decisions:

- Open Access: caBIG®'s products, its grid and community are open to all interested persons and groups. This never meant that the data would be open; rules of privacy and security would apply to biomedical and patient information, but the tools and resources would be available.
- Open Development: Software projects are assigned to teams (often from more than one institution), but regular interaction with Special Interest Groups (SIGs) and iterative opportunities for review and comment ensure that the projects are owned by the entire caBIG® community. Planning, validation, testing, and deployment materials similarly are open.
- Open Source: Software code developed with support from the NCI under the caBIG® project is available to developers for use and modification. The software is provided through nonviral open-source licenses to promote efficient and reusable code. Industrial partners can modify and commercialize derivative products.
- Federation: caBIG® software and standards enable Cancer Centers and other data producers to share resources with each other and the larger community. While data resides locally, aggregation to create multi-institutional virtual information sets is permissible within the frameworks of data sharing established by the caBIG® community. This requires a sophisticated authentication and authorization system residing on caGrid (see Chaps. 4 and 5).

9.3.2 Workspaces and Working Groups

Figure 9.2, taken from caBIG presentations from 2006 and 2007, illustrates the structure of the pilot phase. Initially three Domain Workspaces were established, based on the feedback from the Cancer Centers depicted in Fig. 9.1. Clinical Trials Management Systems (CTMS) was established to inventory Cancer Center needs for integrated clinical trials support, and originally was tasked to develop an open-source solution for this problem identified by over 60% of the Centers. Integrative Cancer Research was constituted to develop and adopt bioinformatics tools for translational cancer investigation. Tissue Banks and Pathology Tools were tasked to develop an open-source biobanking system. In 2004, NCI's imaging groups elected to join the caBIG community, and a fourth In Vivo Imaging Workspace was created to develop archives and a caGrid-compatible image distribution and analysis system.

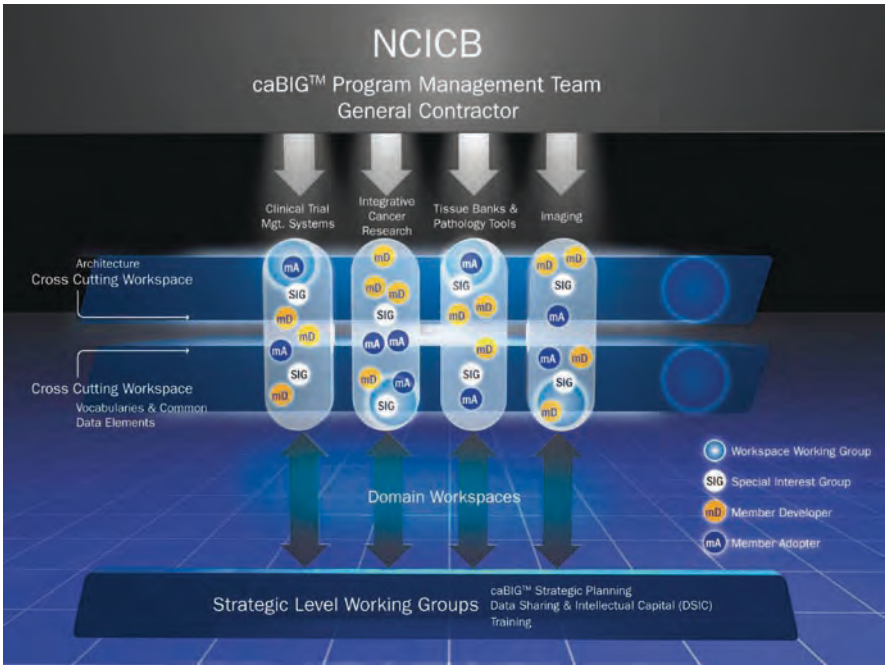


Fig. 9.2 Organization of the caBIG® Pilot Phase into Workspaces and Working Groups. This figure from 2005 shows four domain Workspaces, the two cross-cutting infrastructure Workspaces, and strategic working groups

Two Cross-Cutting Workspaces were established during the pilot, to address two components of the caBIG® Mission. An Architecture Workspace took responsibility for the computational infrastructure and the development of caGrid and for developing a set of guidelines and standards for developing and modifying biomedical informatics applications to be caGrid compliant. A Vocabularies and Common Data Elements (VCDE) Workspace provided for the development of the underlying data elements and vocabularies used by the project, and developed common mechanisms used throughout the caBIG® community via mentoring, white papers, and a structured software review process.

Early on these workspaces promulgated two desiderata. To bring systems online quickly, caBIG® committed to a "bias for action." This implies a commitment to making decisions and moving forward, even if perfection cannot be achieved. To allow long-term evolution and improvement of architectural design, caBIG® is committed to "designing for change." To turn these thoughts into action, the community also adopted a two-pronged practical approach: If requirements are well understood and good solutions are available, caBIG® initiates developmental activities within the architectural workspace. If requirements are less clear or if solutions are not yet available, caBIG commissioned analysis and assessment activities, followed by prototyping.

Three strategic level working groups supported these activities. The Strategic Planning Working Group supported NCICB and BAH in providing overall direction for the project, and developed a strategic plan which was presented in 2005 at the second caBIG® Annual Meeting. A Data Sharing and Intellectual Capital (DSIC) Working Group was organized to solicit, collect, and catalog specific examples (use cases) from each Domain Workspace that presented challenges to the caBIG® community with respect to study participant consent, Institutional Review Board issues, authorization, confidentiality, de-identification, data sharing, software licensing, biospecimen resources, and intellectual property. This was followed with white papers oriented toward making caGrid a usable resource for multi-institutional and clinical research. The Training Working Group created a Developer Boot Camp and published first editions of print and electronic training resources for the broader cancer and biomedical research community.

Also shown in Fig. 9.2 are the roles of the Cancer Centers and their representatives. Three roles were envisioned within the workspaces. Member Developers, comprising 20% of the Centers, were institutions that committed to software development, either of fundamental infrastructure or informatics tools. Member Adopters, another 20%, were institutions that agreed to test and adopt prerelease versions of tools, and provide feedback to the Workspaces and Developer groups. Individuals from the remaining 60% of the participating Cancer Centers could be members of strategic workspaces or Special Interest Groups, which were constituted to review workspace progress or recommend projects in specific areas within a Workspace.

9.3.3 Activities During the Pilot Phase

The kickoff Annual Meeting in February 2004 described the overall goals of the caBIG® project and provided opportunities for each workspace to begin to define their activities and for participants to meet each other. Existing projects at the Cancer Centers were presented, and the discussion centered on how a broad array of research and development might be integrated into the caBIG program. Information on the Pilot Phase of caBIG® is well summarized in the Pilot Phase Report (2007).

Within the first year several organizing principles emerged that guided the pilot phase. Foremost among them was the sense of *community*: informatics professionals and domain experts from the Cancer Centers were meeting each other, often for the first time. While some centers received more funding than others based on the maturity of their information technology (IT) platform and their role as developers, everyone started out with less resources for caBIG® activities than had been envisioned when the idea of 10–12 pilot institutions was conceived in 2003. There had, of course, been opportunities for Cancer Center informatics leaders to work together before. The American Association of Cancer Institutes had an informatics initiative early in the decade. The Pennsylvania Cancer Alliance Bioinformatics Consortium

had united the six NCI-designated cancer centers in that state to develop a virtual tissue bank system. A volunteer organization, the Biomedical Research Institutions Information Technology Exchange (or BRIITE) had been meeting for several years to discuss mutual areas of collaboration and knowledge sharing around research informatics. In fact, the membership of the Strategic Planning Working Group was drawn from these and other preexisting activities.

Most Cancer Centers had some degree of biomedical informatics technology at the time of caBIG® launch, and one of the early issues was whether caBIG® was envisioning a wholesale rewrite or planned to incorporate existing systems. The project recognized that there would need to be many paths to caBIG® adoption, and thus emphasized *interoperability and interfacing* working technology with the emerging caBIG® infrastructure. Interoperability would need to be syntactical and semantic. One of the early deliverables of the Architecture and VCDE Workspaces was a caBIG® Compatibility Guidelines document. This paper established four levels of caBIG® compatibility:

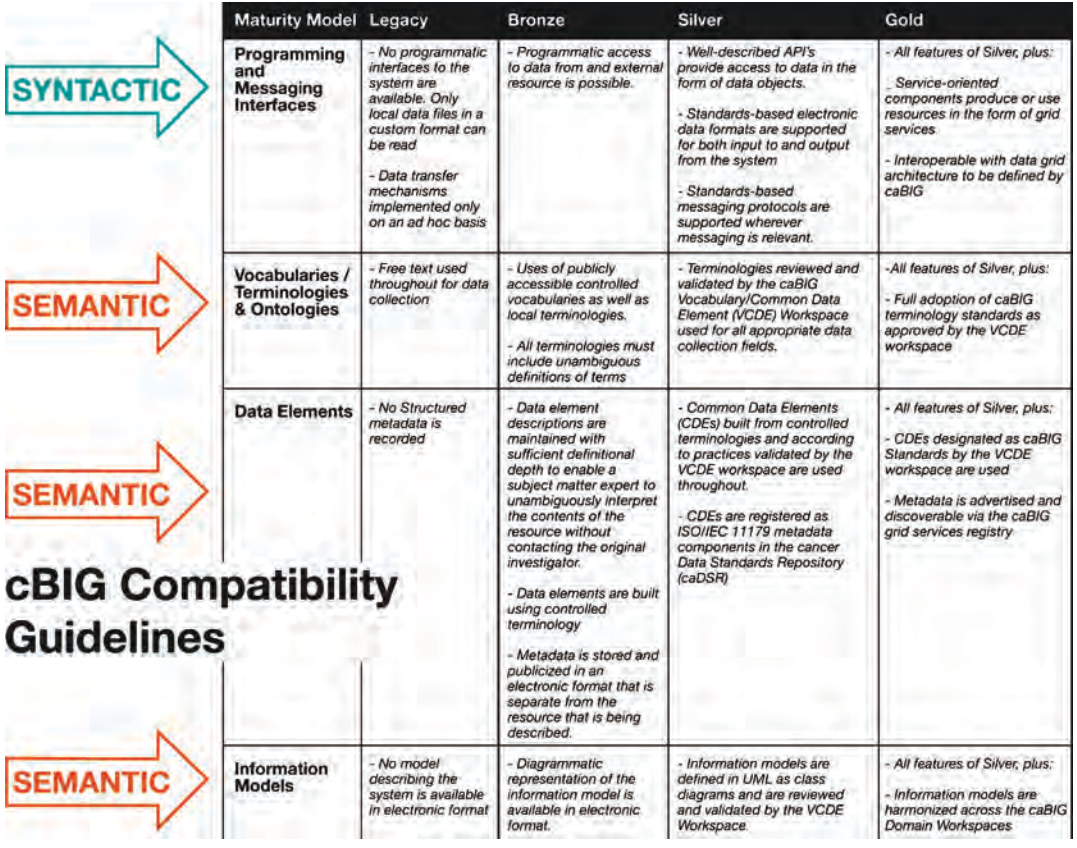
- Legacy: Existing systems without data models and rudimentary ability to interface with caBIG® component systems
- Bronze: Systems that included information models, controlled vocabularies and programmatic access to data
- Silver: Systems with a functioning Application Programming Interface, and information models and vocabularies vetted by the VCDE Workspace
- Gold: Silver systems that were fully compatible with caGrid

Syntactic interoperability was characterized by degree of maturity of programming and messaging interfaces. Semantic interoperability included components of ontology, information model, structured vocabularies, and data elements. A chart depicting specifics of the maturity model based on interoperability features is shown as Fig. 9.3.

Project management in caBIG® was diverse, complex, and a source of constant comment. An effort was made to balance top-down guidance from NCICB and innovation arising from the Cancer Center community. Regular SIG and Workspace teleconferences attempted to broker this dynamic tension. Software development was vested in the academic institutions in order to keep the tools close to the user base; however, the projects were funded as contracts rather than grants. The objective here was to establish timelines and specific deliverables, and it was based on an expectation that the development teams at the Cancer Centers could develop polished software tools.

A final principle was *leveraging*. Having multi-institutional development teams with existing academic and commercial software suggested that reinvention and redevelopment could be kept to a minimum. This principle assumed that the caBIG® projects would focus on connection to the grid, and that modularity and interoperability would arise as a matter of course.

During the second and third years of the Pilot Phase, efforts centered on generating results from the workspaces. SIGs took responsibility for shepherding development projects and met with regular teleconferences. Workspaces and some SIGs had face



Maturity Model	Legacy	Bronze	Silver	Gold
Programming and Messaging Interfaces	- No programmatic interfaces to the system are available. Only local data files in a custom format can be read - Data transfer mechanisms implemented only on an ad hoc basis	- Programmatic access to data from and external resource is possible.	- Well-described API's provide access to data in the form of data objects. - Standards-based electronic data formats are supported for both input to and output from the system - Standards-based messaging protocols are supported wherever messaging is relevant.	- All features of Silver, plus: _ Service-oriented components produce or use resources in the form of grid services - Interoperable with data grid architecture to be defined by caBIG
Vocabularies / Terminologies & Ontologies	- Free text used throughout for data collection	- Uses of publicly accessible controlled vocabularies as well as local terminologies. - All terminologies must include unambiguous definitions of terms	- Terminologies reviewed and validated by the caBIG Vocabulary/Common Data Element (VCDE) Workspace used for all appropriate data collection fields.	-All features of Silver, plus: - Full adoption of caBIG terminology standards as approved by the VCDE workspace
Data Elements	- No Structured metadata is recorded	- Data element descriptions are maintained with sufficient definitional depth to enable a subject matter expert to unambiguously interpret the contents of the resource without contacting the original investigator. - Data elements are built using controlled terminology - Metadata is stored and publicized in an electronic format that is separate from the resource that is being described.	- Common Data Elements (CDEs) built from controlled terminologies and according to practices validated by the VCDE workspace are used throughout. - CDEs are registered as ISO/IEC 11179 metadata components in the cancer Data Standards Repository (caDSR)	- All features of Silver, plus: - CDEs designated as caBIG Standards by the VCDE workspace are used - Metadata is advertised and discoverable via the caBIG grid services registry
Information Models	- No model describing the system is available in electronic format	- Diagrammatic representation of the information model is available in electronic format.	- Information models are defined in UML as class diagrams and are reviewed and validated by the VCDE Workspace.	- All features of Silver, plus: - Information models are harmonized across the caBIG Domain Workspaces

Fig. 9.3 The caBIG® Compatibility Guidelines. Semantic and syntactic interoperability were both stressed in caBIG® applications

to face meetings where progress was reviewed and projects modified to keep pace with the ever-changing biomedical research and informatics landscape. In 2004 nearly 200 teleconferences were held; this increased to nearly 500 in each of the second and third years. By the third year 0.5 versions and 1.0 versions of software tools were emerging from all of the domain workspaces, although the overall progress of the caBIG® community seemed less than had been envisioned at the outset.

9.3.4 Deliverables from the Pilot Phase

As summarized in the caBIG® Pilot Phase Report, deliverables included cultural, technical, managerial, and operational components. Figure 9.4 illustrates attendance at the caBIG® Annual Meeting from 2004–2008; this reflects the establishment of an ongoing caBIG® community. At the end of the pilot phase 54 Cancer Centers and over 900 people were working on the project. The cross-cutting workspaces had delivered standard data models, common data elements, and preliminary versions of caGrid. Over 80 services had been registered on the grid, although nearly all were intra-institutional test implementations.

By early 2007 approximately 40 tools had been released under the caBIG program. These are listed in Appendix. Some of these were de novo development; most were interoperable adaptations of software that was either released or in development during the period. By mid-2007 12 tools were certified as caBIG® Silver compliant, which required adherence to architectural and vocabulary standards. The lack of Gold compliance was more a reflection of the progress on caGrid; the vision of data and interoperable tools flowing across a computational and data grid had not yet materialized.

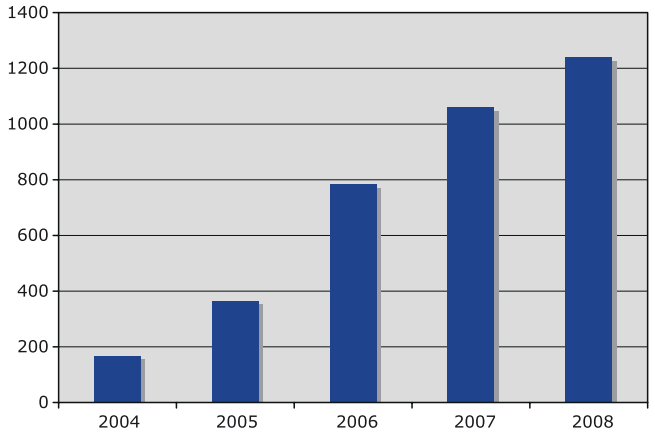


Fig. 9.4 Attendance at the caBIG® Annual Meeting, by year

However, the sheer size of caBIG® meant that it could not be dismissed as a force within an exploding biomedical informatics environment in cancer research. Several NCI initiatives, notably The Cancer Genome Atlas project, adopted tools and infrastructure from caBIG®. Reaching out to the Specialized Programs of Research Excellence (SPOREs), caBIG® provided the informatics platform for multi-institutional research in prostate and breast cancer. The NCI adopted caBIG® compatibility essentially wholesale in evaluating center grants and information intensive project proposals arising from 2005. The US Food and Drug Administration (FDA) partnered with the NCI on a Regulatory Data Exchange project based in part on caBIG® technologies. Very important for the future of clinical trials research, a consortium of the NCI, the FDA, the Clinical Data Interchange Standards Consortium (a standards group from the biopharmaceutical industry) and Health Level Seven (the health-care informatics standards organization) established the Biomedical Research Integrated Domain Group (BRIDG). This spurred development of the BRIDG Model, a comprehensive standard data model that captures metadata about clinical and preclinical trials.

9.3.5 Critical Evaluation of the Pilot Phase

As the Pilot Phase drew to a close, an internal and external evaluation of the caBIG® project identified several problems that would guide the evolution of the initiative. Although a large community had been engaged and stayed engaged, and a number of deliverables achieved, a number of common issues kept being advanced. The decision to involve over 50 Cancer Centers led to management and control difficulties, and all but ensured that progress, while broad, would not be as swift as with a focused group of pilot centers. Also, with so many institutions involved for relatively little participation level or financial incentive, the communication devolved onto the informatics professionals to a great extent. Although this community grew and became close over the pilot period, end users and cancer center directors felt largely out of the loop. A constant tension arose between the need to develop working tools that would satisfy users and directors, while maintaining progress on the strategic objective of caBIG®: developing a grid architecture, vocabularies, and standards that would support long-term development of interoperable tools and systems. This was particularly true in the CTMS Workspace, where the original idea to build an open-source interoperable clinical trials management system in 3 years was somewhat overambitious.

With such a large community working on the deliverables, the Firm Fixed Price contracting mechanism used in the Pilot Phase proved unworkable. Cancer centers had to assume that funds would eventually flow as work was completed; stringent task orders led to short time frames and limited dollars to complete major tasks. Almost all developer institutions used internal resources to augment caBIG® funds, leading to increased scrutiny of the project's value for the dollar.

The quality of caBIG® tools was uneven, in part due to the reliance on the academic community as developers. In hindsight one might have recommended commercial partnerships for development from the beginning, but in 2003 it was unclear how that could have worked. The major IT corporations stayed on the sidelines during the first few years of the project, and smaller tool-building firms participated in SIGs. The state of the art in bioinformatics tools was relatively primitive at the outset of the caBIG® initiative. Again, concentration on a few institutions with a few partners might have accelerated progress in a more limited scope project.

As the caBIG® Pilot Phase concluded most participating institutions acknowledged that the role of biomedical informatics had evolved and become more important. Connectivity and interoperability were valued, and the community was worth maintaining. However, software tool adoption at individual centers was problematic, and a number of centers were wondering if their commitment to caBIG® principles was sustainable. For the program to continue to develop, a new model embracing public and private components would need to supplant the communitarian approach of the pilot phase.

9.4 The caBIG® Enterprise Phase: 2007

In the spring of 2007 the NCI announced the goals of the postpilot, Enterprise Phase of the caBIG® Project. Three areas were stressed:

- A disciplined, systematic delivery of caBIG® infrastructure, tools and concepts to NCI-designated Cancer Centers
- Replacing the developer–adopter–participant model of the Pilot Phase largely with an Enterprise Support Network of institutions and firms vetted to provide assistance with caBIG® deliverables
- Expansion of the community to include government, academic, and especially private sector entities

9.4.1 caBIG® Adoption Program

The Cancer Center initiative is known as the caBIG® Adoption Program. In 2007, each institution was given the opportunity to perform a limited self-assessment and propose a staff member (who could possibly be a “to be named”) as its deployment lead. Upon acceptance by the NCICB (which was in the process of changing its title to the Center for Bioinformatics and Information Technology-CBIIT), half salary of the lead worker was added as a supplement to the institution’s Cancer Center Support Grant. This mechanism is still in place as of mid-2009.

Cancer Centers that opt in to the Adoption Program are required to explore the feasibility of working in four areas. First, they are expected to activate a caGrid node. For caBIG® tools there are three "bundles" or related sets of products and services that are to be evaluated and deployed if possible. These include

- Clinical Trials Compatibility Framework – components that support human clinical trials and clinical research. These are to be evaluated individually, and adopted into the institution's clinical trial management system architecture as appropriate.
- Life Sciences Distribution – a set of tools and applications that support various aspects of basic and translational research. These tools are evaluated individually, and added to the suite of products available at an institution.
- Data Sharing and Security Framework – a bundle of policies, standard operating procedures, and model documents that assist a Cancer Center in utilizing the Grid for collaborative research.

In the 2008 caBIG® Annual Report 50 Cancer Centers are identified as participating in the Adopter Program. Highlights of tool adoption include:

- Over 30 Cancer Centers deploying parts of the Clinical Trials Suite, the retitled Clinical Trials Compatibility Framework.
- Over 20 institutions deploying caTissue, the principal product of the Tissue Banking and Pathology Tools Workspace, and among the most mature new products developed during the Pilot Phase. Several institutions are conducting joint research using caTissue on the Grid.
- Nearly 20 institutions adopting caArray, a solid product for collection and management of microarray data. caArray is fully (Gold) caGrid compliant.
- About 40 organizations, including seven Cancer Centers, using some or all of the tools developed in the Imaging Workspace.

As this formal process evolved to provide tools and resources to the Cancer Centers, the Workspaces reduced in size and focused their scope. A number of SIGs are still communicating regularly, and applied research and development continues within the Workspaces. A review of the well-constructed caBIG® website's Integrative Cancer Research landing page indicates that one SIG (Population Science, which tried to become its own workspace but funding was unavailable), three workgroups, and over 20 supported tools form the bulk of the Workspace's operations (https://cabig.nci.nih.gov/workspaces/ICR).

9.4.2 The Enterprise Support Network

Figure 9.5, from the caBIG® website, depicts the Customer Support Map. In addition to a growing set of web-based resources and the Workspaces, the NCI offers two other services itself. Product Representatives are CBIIT staffers who offer initial guidance about specific tools and services. Six Knowledge Centers have been

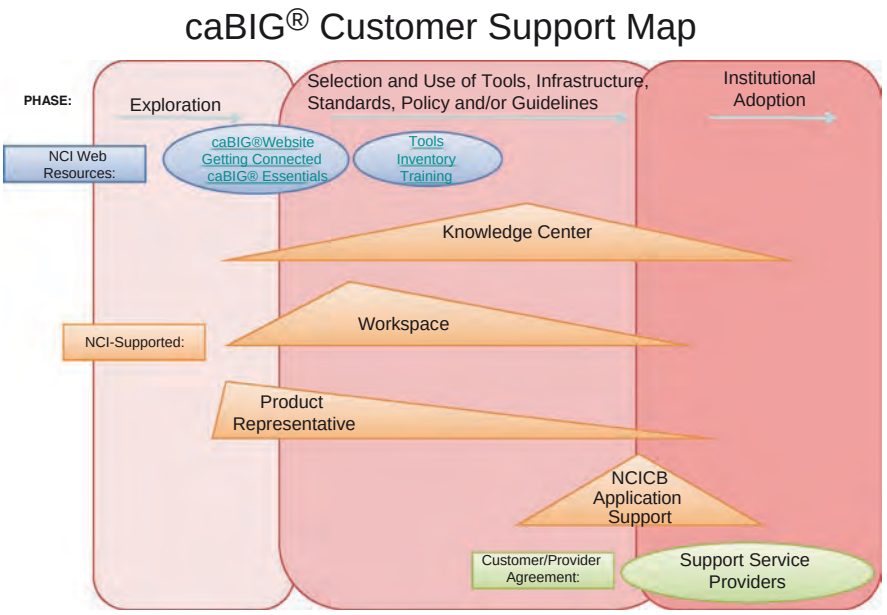


Fig. 9.5 The caBIG® Customer Support Map. Web resources are shown in blue, NCI support in pink, and commercial service providers in green

established through contracts with institutions that provide web-based support for education and outreach, domain expertise with the tools and the biomedical informatics problems they address, and a centralized resource for information about the products and services. The six Knowledge Centers as of mid-2009 are:

- Clinical Trials Management Systems, led by the Duke and Northwestern cancer centers, Cancer and Leukemia Group B (an NCI-supported cooperative clinical trials group), and SemanticBits, LLC, a commercial software engineering firm oriented toward the health sciences.
- Molecular Analysis Tools, led by the Columbia cancer center and the Broad Institute of MIT and Harvard.
- Tissue/Biospecimen Banking and Technology Tools, led by the Washington University cancer center.
- caGrid, led by the Ohio State University Medical Center.
- Data Sharing and Intellectual Capital, led by the University of Michigan.
- Vocabulary, led by the Mayo Clinic.

The direction of the Enterprise Support Network, however, is toward Support Service Providers, commercial firms (and one university group to date) who have been vetted through the caBIG® mechanism to provide help desk support, adaptation, and enhancement of caBIG® tools to individual environments, deployment support, and/or documentation/training materials and services. Fifteen organizations have qualified as Support Service Providers by mid-2009.

The NCI made good use of feedback and criticism arising from the caBIG Pilot Phase. With any biomedical informatics initiative the principal challenge is sustainability – how to support tools as the field matures, how to deal with the underlying architecture as it undergoes revision, and how to balance the users' need for systems that work today with the developers' and researchers' desire to advance the architecture and position for the future. By maintaining focused development in the Workspaces, utilizing Cancer Centers as Knowledge Centers to bridge development and implementation, and contracting out the deployment and support (and enhancement) of caBIG® tools, CBIIT has maximized the likelihood that the caBIG® project is sustainable.

9.4.3 caBIG® in the Literature

Beginning in 2004, 46 articles have appeared in PubMed through mid-2009. A review of these papers indicates that infrastructure, applications, and the community all have been reported. Hanash (2004), in a perspective, summarized the state of the art in integrating bioinformatics tools for cancer research. Hanash identified caBIG® as an emerging collaboration to accelerate this process, and pointed to similar activities underway at the United Kingdom's National Cancer Research Institute. The next 18 months saw several more introductory articles appear on the promise of caBIG®.

caBIG® tools began to appear in the peer-reviewed literature in 2006. Phillips and colleagues from NCICB described the rationale for a standard set of tools and modules to promote interoperability of caBIG® solutions (Phillips et al. 2006). In the same year the caGrid team published the core architecture paper (Saltz et al. 2006). As shown in Fig. 9.6, a widespread collection of services and resources was envisioned from the beginning. Investigator-maintained data sources were enclosed in caGrid Data Service interfaces; analytical applications were exposed to remote users through the caGrid Analytical Service. Brief reports of caBIG® applications also began to appear in 2006 (Tchuvatkina et al. 2006; Tobias et al. 2006; Zhu et al. 2006).

As the caBIG® Project transitioned to its Enterprise Phase, the number and variety of reports emanating from the community increased. A summary of the initiative appeared at an international biomedical informatics congress (Beck and Bondy, 2007). The Imaging Workspace released GridIMAGE (Gurcan et al. 2007) and an eXtensible Imaging Platform (XIP), an open-source application development environment for imaging projects (Prior et al. 2007). Image-based clinical trials are supported using these tools as a basis for a specific ontology-based application (Channin et al. 2009). The caGrid team introduced Grid Authentication and Authorization with Reliably Distributed Services, or GAARDS. GAARDS, based on the Dorian server model (Fig. 9.7), allows user identification, recognition, and service authorization to be fully distributed on caGrid (Langella et al. 2007 and Chap. 16).

VCDE, which had begun to certify applications and tools for level of compatibility with the caBIG® architecture, accelerated its processes. A compatibility evaluation system developed under the auspices of the Vocabulary Knowledge Center was

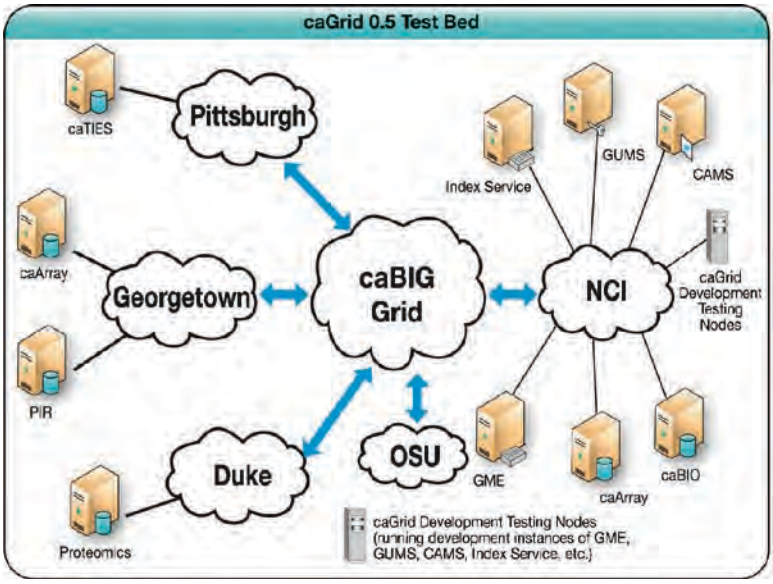


Fig. 9.6 A high level view of caGrid. A central cloud of grid services allows research groups and institutions to connect to a network that includes federal resources

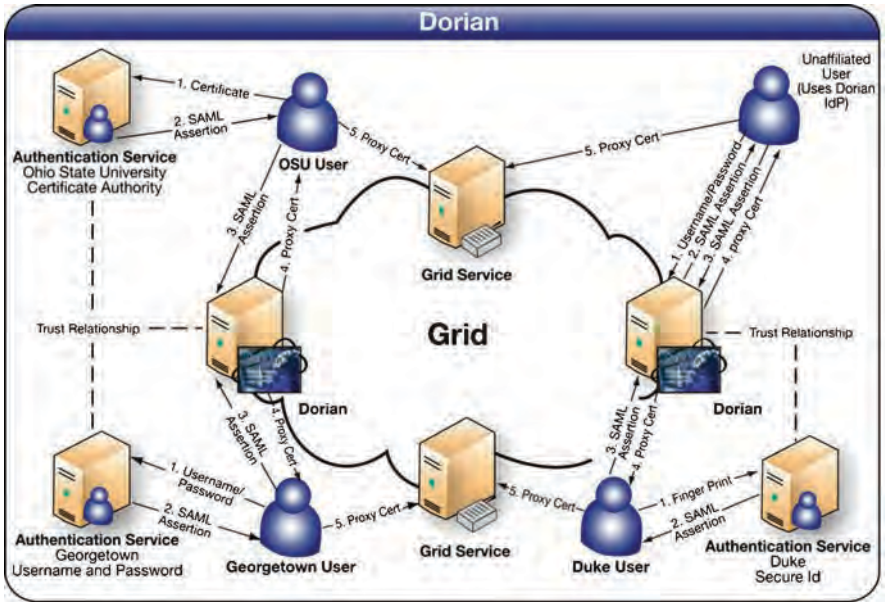


Fig. 9.7 Authentication and Authorization Scheme for GAARDS. Dorian servers manage the process internally, permitting institutions to establish their own approaches to identification

presented by Freimuth et al. (2008). This system standardizes the review process but also provides for some automated checking against core components such as the cancer Data Standards Repository (caDSR).

caBIG® tools such as the Cancer Common Ontologic Representation Environment (caCORE) Software Development Kit began to show results during the Enterprise phase. A group from Emory University reported on a Silver level compliant database for lymphomas, and demonstrated the ability to build caBIG® compliant data systems that can draw information from disparate sources (Huang et al. 2009). The Cancer Translational Research Integration Platform (caTRIP) was presented in 2008 as a sophisticated novel caBIG® tool for aggregating clinical and molecular data (McConnell et al. 2008). Designed originally to satisfy the needs of translational research groups at Duke, it represents an integration tool that has potential for use at any institution that is deploying caGrid-compliant software.

The private sector began to develop and publish tools and concepts during the Enterprise Phase. semCDI is a Small Business Innovation Research-funded query formulation for semantic data integration over multiple applications and multiple data sources (Shironoshita et al. 2008).

Informatics research based on caBIG® is also underway. Kunz et al. (2009) reported on metadata mapping and reuse, applied to the Common Data Elements in caBIG®. They demonstrated a similarity measure that could be used to find appropriate CDE's rapidly in support of new tools and datasets.

9.4.3.1 Collaboration

In 2008, examples of collaborative research based on caBIG® tools and architecture began to emerge. The products of the Tissue Banking and Pathology Tools pilot workspace were among the most mature examples of new development. The Clinical Annotation Engine was used to develop a National Mesothelioma Virtual Bank (NMVB), a collaboration with three contributing and four participating institutions (Amin et al. 2008). Not a caBIG® project per se, the NMVB is funded by the Centers for Disease Control and Prevention and the National Institute of Occupational Safety and Health. Other examples of collaboration that either uses caBIG® tools now or will do so in the near future include The Cancer Genome Atlas and the Repository of Molecular Brain Neoplasia Data (REMBRANDT) (Madhaven et al. 2009).

9.4.3.2 Continuing Community Papers

White papers and community development documents continue to appear when they add substantive information to the biomedical literature. The VCDE Workspace has developed a caBIG® terminology review process; Cimino and associates reported on the application of this process to four standard terminologies (Cimino et al. 2009). BRIDG released a technical report in 2008 (Fridsma et al. 2008). Clinical trials applications seem to have a long incubation period; one particular strength of BRIDG is that it keeps all of the relevant communities together while advances in software tools progress.

The security workgroup of the DSIC pilot workgroup conducted an interview-based study of representatives from a sample of US cancer centers, in order to determine how challenging it would be to comply with the Health Insurance Portability and Accountability Act (HIPAA) and Code of Federal Regulations privacy and security standards (Manion et al. 2009). Responders included Institutional Review Board directors, research administrators, counsel, and security officers. They identified nine specific criteria that must be fulfilled in order to place data on the Grid that would be regulated by HIPAA and 21CFR11. Although the operations of caBIG® are moving increasingly into the private sector, these community documents maintain an important role for the academic community in advancing the concepts that underlie caBIG® as well as its tools and applications.

9.5 Conclusion

The definition of usable information for biomedical research and treatment continues to expand beyond the literature to include databases, sequences, and metadata. Achieving the goals of a personalized approach to mid-twenty-first century cancer care will require interoperable information systems that ultimately collate data from worldwide sources. Information models, technology, services and policies arising from the caBIG® Program already enable collaborative research and clinical trials registration. Scholarship and applications continue to arise in the Enterprise phase of the project, and the community has broadened to include over 1,000 active participants. As a Big Science initiative caBIG® has had its issues to overcome, and progress in some application areas may not have been as significant as originally envisioned. In particular, the clinical trials applications have been challenged by marketplace solutions. However, the underlying architecture of caGrid and the development tools and resources are strong and generating useful results by an ever-expanding user base (Buetow 2009). A comprehensive website and clear avenues to support enhance the dozen or so currently valuable applications; researchers can experiment with nearly 50 tools and systems under development. Institutions interested in integrative and collaborative cancer research will need to stay abreast of the results likely to arise from caBIG® in its second 5 years.

9.6 Appendix: caBIG® Tools and Technologies at the End of the Pilot Phase

9.6.1 Clinical Trials Management

- Cancer Central Clinical Database (C3D): a clinical trials data management system based on Oracle Clinical, including protocol building, remote data capture, and review.

- Cancer Central Clinical Participant Registry (C3PR): a web-based application for patient trial registry.
- Clinical Data Exchange/Lab Integration Hub (caXchange): an open-source software tool used to manage laboratory data during a clinical trial.
- Clinical Trials Object Data System (CTODS): a virtual data warehouse for clinical trials data that can capture trial information and deliver it de-identified.
- Clinical Data System (CDS): a stand-alone data submission infrastructure for NCI-supported clinical trials.
- Patient Study Calendar (PSC): an open-source application that can create and edit study calendar templates and manage them during a study.
- Cancer Adverse Event Reporting System (caAERS): an open-source software tool that is used to collect process and report adverse events that occur during clinical trials.
- caTRIP: a meta-system that allows users to query across a number of caBIG data services, based on Common Data Elements (CDEs) and integrate results for viewing and further analysis.
- Federal Investigator Registry of Biomedical Information Research Data (FIREBIRD): a system that automates the Form 1572 registration process for clinical trials investigators.

9.6.2 Biospecimen Banking

- caTissue Core: a tissue bank repository tool for biospecimen inventory, tracking, and basic annotation.
- Cancer Text Information Extraction System (caTIES): a system to extract coded cancer information from free-text surgical pathology reports.
- caTissue Clinical Annotation Engine (CAE): a web-based user interface for standards-based manual annotation of biospecimens.

9.6.3 Image Analysis

- National Cancer Imaging Archive (NCIA): a searchable national repository of cancer images, integrated with clinical and genomic data.

9.6.4 Data Mapping

- caAdapter: an open-source tool set that facilitates data mapping, validation, and transformation among various data sources and standard formats.

9.6.5 General Research Tools

- Electronic Laboratory Management Information Resource (caELMIR): a Laboratory Information Management System (LIMS) for recording experimental data.
- geWorkbench: an open-source software platform for genomic data integration.
- GenePattern: another software platform for genomic data integration.
- caIntegrator: a translational informatics platform that can integrate clinical and molecular information across patients and trials.
- caBench-to-Bedside (caB2B): a caGrid client that permits users to search caGrid data services.
- caArray: an open-source microarray data management system, that supports other caBIG® tools like geWorkbench and GenePattern.
- Protein Information Resource (gridPIR): a data resource for genomic and proteomic resource that connects to relevant PIR databases.
- Cancer Models Database (caMOD): a resource for animal models for human cancer, involving submission and search capabilities.
- Bioconductor: an established open-source collection of software packages for high-throughput genome analysis.

9.6.6 Genome Analysis

- SEED: a tool for making and sharing genomic annotations, and for access and analysis of annotations.
- Transcript Annotation Prioritization and Screening System (TRAPSS): an analytical tool for screening gene sequences for mutations, including notable ones.
- Function Express (caFE): a tool for annotation of probes on microarrays.
- GoMiner: a tool for biological interpretation using the Gene Ontology.
- GeneConnect: a caBIG® mapping service that interlinks various genomic identifiers.

9.6.7 Protein Analysis

- RProteomics: a package for analyzing mass spectrometry proteomics data.
- Q5: a classification tool for expression-dependent proteomic data.
- Proteomics Laboratory Information Management System (protLIMS): a prototype LIMS for proteomics.
- Cancer Molecular Pages (CMP): an automated annotation system for cancer-related proteins.
- Computational Proteomics Analysis System (CPAS): an open-source tool for peptide scoring based on the Trans Proteomics Pipeline.

9.6.8 Pathway Analysis

- Quantitative Pathway Analysis in Cancer (QPACA): a tool that provides a set of routines for analysis of microarray data in the context of genetic pathways.
- Reactome: a service that establishes the Reactome system as a caGrid service.
- Pathway Interaction Database: a curated database of information about known biomolecular interactions and cellular signaling processes.
- Pathway Tools: a suite of tools that interact on an open-source pathway database.

9.6.9 Statistical Analysis

- Visual Statistical Data Analyzer (VISDA): an analytical tool for cluster modeling and visualization.
- Distance Weighted Discrimination (DWD): a tool that corrects microarray analyses by reducing systematic bias.

9.6.10 Core Infrastructure

- caGrid: the underlying network architecture, based on the Globus toolkit, that balances local access control and central services.
- BRIDG Model: a collaborative data model for clinical research.
- caCORE: an open source group of software products developed by NCICB that provide the tools for open source, interoperable applications that are caGrid compliant.
- Cancer Bioinformatic Infrastructure Objects (caBIO): a set of software objects that provide a programming interface to caCORE.
- NCI Enterprise Vocabulary Server (EVS): a service that produces metathesaurus entities to support standard controlled vocabularies for caBIG® projects and products.
- caDSR: a metadata registry in caCORE that stores and manages CDEs developed by participants and NCI-supported organizations.
- caCORE Software Development Kit (caCORE SDK): a set of tools designed to aid in a caBIG® compliant, semantically integrated system.

9.6.11 Vocabularies

- LexBIG: a set of software and services to load, publish, and access caBIG® compliant vocabulary.

- Mouse–Human Anatomy Mapping Ontology (MHAP): a mapping and harmonization of murine and human anatomical descriptors.
- Cancer Nutrition Ontology: a unified set of Nutrition vocabularies and mappings.

References

Amin W, Parwani AV, Schmandt L, Mohanty SK, Farhar G, Pople AK, Winters SB, Whelan NB, Schneider AM, Milnes JT, Valdivieso A, Feldman M, Pass HI, Dhir R, Melamed J, Becich MJ (2008) National Mesothelioma Virtual Bank: a standard based biospecimen and clinical data resource to enhance translational research. BMC Cancer 8:236

Beck JR, Bondy JC (2007) The cancer biomedical informatics grid (caBIG): infrastructure and applications for a worldwide research community. Stud Health Technol Inform 129:330–334

Buetow KH (2005) Cyberinfrastructure: empowering a "third way" in biomedical research. Science 308:821–824

Buetow KH (2009) An infrastructure for interconnecting research institutions. Drug Discov Today 14:605–610

Channin DS, Mongkolwat P, Kleper V, Sepukar K, Rubin DL (2009) The caBIG annotation and image markup project. J Digital Imag (in press)

Cimino JJ, Hayamizu TF, Bodenreider O, Davis B, Stafford GA, Ringwald M (2009) The caBIG terminology review process. J Biomed Inform. doi:10.1016/j.jbi.2008.12.003

Freimuth RR, Schauer MW, Lodha P, Govindrao P, Nagarajan R, Chute CG (2008) caBIG compatibility review system: software to support the evaluation of applications using defined interoperability criteria. AMIA Annu Symp Proc 197–201

Fridsma DB, Evans J, Hastak S, Mead CN (2008) The BRIDG Project: A technical report. J Am Med Inform Assoc 15:130–137

Gurcan MN, Pan T, Sharma A, Kurc T, Oster S, Langella S, Hastings S, Siddiqui KM, Siegel EL, Saltz J (2007) GridIMAGE: a novel use of grid computing to support interactive human and computer-assisted detection decision support. J Digit Imaging 20:160–171

Hanash S (2004) Integrated global profiling of cancer. Nat Rev Cancer 4:638–643

Huang T, Shenoy PJ, Sinha R, Graiser M, Bumpers KW, Flowers CR (2009) Development of the Lymphoma Enterprise Architecture Database: a caBIG silver level compliant system. Cancer Inform 8:45–64

Kaiser J (2004) Von Eschenbach revises the NCI agenda. Science 303:1952

Kunz I, Lin M-C, Frey L (2009) Metadata mapping and reuse in caBIG. BMC Bioinformatics 10(S2):S4

Langella S, Oster S, Hastings S, Siebenlist F, Phillips J, Ervin D, Permar J, Kurc T, Saltz J (2007) The cancer biomedical informatics grid (caBIG) security infrastructure. AMIA Annu Symp Proc 433–437

Madhaven S, Zenklusen JC, Kotliarov Y, Sahni H, Fine HA, Buetow K (2009) Rembrandt: helping personalized medicine become a reality through integrative translational research. Mol Cancer Res 7:157–167

Manion FJ, Robbins RJ, Weems WA, Crowley RS (2009) Security and privacy requirements for a multi-institutional cancer research data grid: an interview-based study. BMC Med Inform Decis Mak 9:31

McConnell P, Dash RC, Chilukuri R, Pietrobon R, Johnson K, Annechiarico R, Cuticchia AJ (2008) The cancer translational research informatics platform. BMC Med Inform Decis Mak 8:60

National Cancer Institute. The caBIG Pilot Phase: Report: 2003-2007. https://cabig.nci.nih.gov/overview/pilotreport/. Accessed 3 Jan 2009

Phillips J, Chilukuri R, Fragoso G, Warzel D, Covitz PA (2006) The caCORE software development kit: streamlining construction of biomedical information services. BMC Biomed Inform Decis Mak 6:2

Prior FW, Erickson BJ, Tarbox L (2007) Open source software projects of the caBIG in vivo imaging workspace software special interest group. J Digit Imaging 20(S1):94–100

Saltz J, Oster S, Hastings S, Langella S, Kurc T, Sanchez W, Kher M, Manisundaram A, Shanbhag K, Covitz P (2006) caGrid: design and implementation of the core architecture of the cancer biomedical informatics grid. Bioinformatics 22:1910–1916

Shironoshita EP, Jean-Mary YR, Bradley RM, Kabuka MR (2008) semCDI: a query formulation for semantic data integration in caBIG. J Am Med Inform Assoc 15:559–568

Tchuvatkina O, Shimoni L, Ochs MF, Moloshok T (2006) Proteomics LIMS: a caBIG project, year 1. AMIA Annu Symp Proc 1116

Tobias J, Chilukuri R, Komatsoulis GA, Mohanty S, Sioutos N, Warzel DB, Wright LW, Crowley RS (2006) The CAP cancer protocols – a case study of caCORE based data standards implementation to integrate with the cancer biomedical informatics grid. BMC Med Inform Decis Mak 6:25

Zhu Y, Wang Z, Feng Y, Xuan J, Miller DJ, Hoffman EP, Wang Y (2006) Phenotypic-specific gene module discovery using a diagnostic tree and caBIG VISDA. Conf Proc IEEE Eng Med Biol Soc 1:5767–5770

Section 2
Tools and Applications

Chapter 10
The caBIG® Clinical Trials Suite

John Speakman

Abstract As part of its Cancer Biomedical Informatics Grid (caBIG®) program and its overall commitment to drive the reengineering of the clinical research enterprise, the National Cancer Institute has developed the caBIG® Clinical Trials Suite. The Suite is a free, stable, supported collection of open-source software tools for clinical trials management, developed in response to the expressed need of the biomedical research community and deployable either as one or more standalone components or as an integrated Suite. It is also a reference implementation of the caBIG® enterprise architecture paradigm of modular, interoperable software components. This chapter gives an overview of the context within which the Suite was envisaged and developed, describes the functionality and availability of the Suite and gives an overview of the enterprise architecture of which the Suite is intended to serve as a reference implementation.

10.1 Background: Characteristics of, and Trends in, Information Handling in Clinical Research

Since the mid-1980s, biomedical science has undergone revolution after revolution. The advent of molecular medicine has ushered in the paradigms of translational research and personalized medicine, both marked by the growing role of biology in the clinical process. This dynamic environment is in marked contrast to the development process of therapies through clinical trials. The introduction in the 1950s of the controlled, randomized clinical trial, and its widespread acceptance as a best practice by investigators, sponsors, and regulatory bodies, marked a sea change in the data management process framework for the conduct of clinical research.

J. Speakman (✉)
Center for Biomedical Informatics and Information Technology, National Cancer Institute, 2115 East Jefferson Street, Suite #6000, Rockville, MD 20852, USA
e-mail: speakmaj@mail.nih.gov

M.F. Ochs et al. (eds.), *Biomedical Informatics for Cancer Research*,
DOI 10.1007/978-1-4419-5714-6_10, © Springer Science+Business Media, LLC 2010

Arguably, there have been no such changes of comparable magnitude in the clinical trials process since then.

Beginning in the mid–late 1990s, this dissonance, manifested specifically in the "pipeline problem" – a slowdown, instead of an expected acceleration, in the number of innovative therapies being made available to patients – was noted in multiple reports and articles analyzing the issue. The US Food and Drug Administration (FDA) noted in 2005 that *"the applied sciences needed for medical product development have not kept pace with the tremendous advances in the basic sciences."* In July 2005, the founder of Intel Corporation, Andrew S. Grove, Ph.D., wrote a widely cited editorial contrasting the rapid pace of innovation in microchip development with its slow pace in the development of new therapies for disease.

Nowhere has this dissonance between the pace of discovery and that of development been more keenly felt than in oncology. The National Cancer Institute's (NCI) 1997 report of the Clinical Trials Program Review Group noted that the complexity of the clinical trials infrastructure had "*eroded the ability of the system to generate new ideas to reduce the cancer burden.*" This was followed in 2005 by the report of the Clinical Trials Working Group to the National Cancer Advisory Board, which again identified the need for a new clinical trials infrastructure that could enhance coordination and communication, scientific quality and prioritization, standardization of tools and procedures and operational efficiency. Studies of the clinical trials process, such as those by Dilts and Sandler (2005), point to the unnecessary complexity and interorganizational variance of the clinical trials process as a key impediment to speeding new therapies to patients.

The development of specialized information technology to handle clinical research data began in earnest in the 1980s and 1990s. Increasing study volume, and increasing reporting burden from sponsors and regulatory bodies, began to drive study sponsors, both commercial and governmental, and major academic medical centers to automate their clinical research processes. Clinical research information systems tend to share characteristics with clinical care information systems, and more broadly with information systems in industrial and business environments characterized by near-real-time, transactional data processing where information is durable and the process must be highly regulated and standardized. These characteristics are distinct from those typically found within the field of bioinformatics, where the science is moving rapidly and information tends to be short lived, so the desire is merely to share the information as quickly and usefully as possible. In bioinformatics, post hoc data tend typically to be integrated into analysis workflows, which are constantly updated by the investigator according to the results of a previous "run."

Several large academic centers have spent many years developing tailored inhouse systems intended to support local clinical workflows which, as already indicated above, vary significantly between organizations. The most successful of these clinical research information systems are very highly tailored, effectively hand-crafted boutique systems that integrate tightly with the organization's clinical systems, workflows, and processes. This facilitates rapid acceptance within the organization but also makes the systems inherently ingrained, sclerotic, and resistant

to change – in short, legacy systems. Their operations and interfaces are brittle, i.e., almost any change in process or to another system that interacts with them will cause them to break down and require code modification to fix. Large amounts of staff time are required simply to maintain systems and their interfaces and to manage the enterprise-wide change control process, leaving software development staff struggling to keep up with demands for enhancements and changes to functionality. Architectural considerations and refactoring are at the bottom of the priority list, ensuring the system is further consigned to legacy status.

A second, related challenge stems from the monolithic nature of these systems. Building all the required functionality into one big system eliminates the need to standardize or coordinate interprocess communication. The downside is that this condemns the system's development team to be the sole source of new functionality or new interconnections, even if the requisite code is available, sometimes free of charge, elsewhere. Similar functionality and interconnections are developed over and over throughout the biomedical research enterprise, effectively reinventing the wheel again and again, with the attendant waste and possibility of error. This becomes more of a challenge as the importance of biological markers in clinical research grows. Clinical investigators and their teams are faced with a tsunami of new data from new sources, especially laboratory information management systems (LIMS). Data from LIMS and other sources must be rapidly integrated into clinical trials, in the transactional, near-real-time modus operandi of clinical systems, if the promise of translational medicine is to be realized. For instance, clinical trials increasingly assume the availability of molecular expression data for use in determining criteria for eligibility of potential study participants and for their randomization to study arms, and for imaging data as determinants of response to treatment. Human rekeying, always the bane of clinical trials data management, is especially undesirable in the case of these data. Rekeying of existing data from existing source systems is not only wasteful of effort but is also the cause of unacceptable delays and errors, doubly so if the data were "born digital," i.e., the measurements that generated them were performed by machines and instantly rendered digitally without any human intervention beyond the calibration and operation of the machines. Better than rekeying, but still labor-intensive and error-prone, is the development of complex semimanual ad hoc data synchronization procedures involving interim steps (e.g., export files in comma-delimited or spreadsheet format). In these cases, the investigators and their scientific teams typically act as data integration specialists and software engineers, at once limiting their own time for scientific work and increasing the risk of coding errors, transposition errors, data loss, and data corruption.

As the pace of science continues to accelerate, specific LIMS rapidly become obsolete. As they are replaced, demand begins for clinical trials, and thus for clinical trials management systems, to rapidly incorporate data from the new replacement LIMS. As already noted developers, already overburdened with maintenance and enhancement requests for these brittle legacy systems, must run to stand still. Academic medical centers may be forced to decide between expanding development staff to cope with the demand for new interfaces and allowing some degree of

manual process, with its attendant waste and errors, rather than requiring investigators to wait for systems staff to develop the interfaces.

10.2 Context: The Cancer Biomedical Informatics Grid®

NCI's Cancer Biomedical Informatics Grid (caBIG®) program was launched in 2004 as an information network enabling all constituencies in the cancer community – researchers, physicians, and patients – to share data and knowledge. The mission of caBIG® is to develop a truly collaborative information network that accelerates the discovery of new approaches for the detection, diagnosis, treatment, and prevention of cancer, ultimately improving patient outcomes. Specifically, the goals of caBIG® are to:

- Connect scientists and practitioners through a shareable and interoperable infrastructure.
- Develop standard rules and a common language to more easily share information.
- Build or adapt tools for collecting, analyzing, integrating, and disseminating information associated with cancer research and care.

In 2003, in preparation for the launch of the program, NCI visited nearly all of its Designated Cancer Centers, a network of over 60 of the most advanced centers in the United States conducting research into, and treatment of, cancer, to better understand the research informatics needs and capabilities of these organizations. Notably, far and away the greatest need expressed by Centers was that for tools to manage clinical data, i.e., in clinical trials (see Fig. 9.1). A full discussion of the caBIG® initiative is presented in Chap. 9.

10.3 Description of the caBIG® Clinical Trials Suite

The caBIG® Clinical Trials Suite (hereinafter referred to as the Suite) was developed by the caBIG® program to fulfill the expressed needs of NCI's cancer research community for tools to manage clinical trials. The Suite is a modular clinical trials management system designed primarily for use in trial sites, e.g., NCI-designated Cancer Centers and other academic medical centers, and comprises a collection of interoperable modules covering a broad range of key functionality in clinical trials management. The functionality of the modules, and of the Suite as a whole, was selected and prioritized by the caBIG® Clinical Trials Management Systems Workspace, an open community of practice comprising individuals and groups interested in the management of clinical trials and the development of informatics solutions to facilitate this activity. The caBIG® Clinical Trials Management Systems Workspace includes representatives from NCI-designated Cancer Centers, NCI

Cooperative Groups, NCI Specialized Programs of Research Excellence (SPOREs), NCI Community Clinical Oncology Program (CCOP) sites, NCI National Community Cancer Centers Program (NCCCP) sites, other groups within NCI and the broader National Institutes of Health, academic medical centers, patient advocacy groups, biopharmaceutical companies, standards bodies, regulatory organizations, and software vendors.

Functionality supported by the current version of the Suite includes the management of study participant registration (C3PR), study participant scheduling (PSC), management and reporting of adverse events (caAERS), import of data to clinical trials systems from clinical laboratory and other data source systems (caXchange), viewing and selection of imported laboratory data (Lab Viewer), and exchange of information with clinical data management systems (CDMS Connector). Implementation of the Suite is based upon the caGrid infrastructure, with Clinical Data Exchange (caXchange) providing reliable message routing.

10.3.1 Cancer Adverse Events Reporting System

The Cancer Adverse Events Reporting System (caAERS) is an open-source, standards-based, web application for documenting, managing, reporting, and analyzing adverse events (AEs). The system operates both as a repository for capturing and tracking routine and serious AEs and as a tool for preparing and submitting expedited AE reports to sponsors and regulatory agencies, supporting regulatory and protocol compliance for adverse event reporting.

caAERS uses accepted standards for classifying adverse events, such as NCI's Common Terminology Criteria for Adverse Events (CTCAEs) and MedDRA. A rules engine in caAERS allows automated assessment and disposition of AE reports – common rule sets are included "out of the box" and institution-specific rules can be added. Furthermore, caAERS features the unique ability to send automated AE reports to NCI's Adverse Event Expedited Reporting system (AdEERS), as well as the generation of populated MedWatch 3500A forms as required by the FDA.

10.3.2 Cancer Central Clinical Participant Registry

The Cancer Central Clinical Participant Registry (C3PR) is a web-based clinical trials information management system available for use by multiple cancer research centers for end-to-end registration of study participants (i.e., patients) to single-site and multisite clinical trials. C3PR can support large-scale, geographically dispersed studies and provides current enrollment statistics and a repository for participant information across studies, sites, systems, and organizations.

C3PR manages participant registrations to clinical trials (ensuring that the study is open, the participant eligible, consent received, etc.). It can stratify participants,

randomize them to trial arms and register them to companion protocols, then track them across sites. Registration workflow is streamlined through the use of role-based access to registration data, and users can be notified of registration events, including the reaching of accrual thresholds, via e-mail and through C3PR's dashboard. C3PR also reports data to facilitate generation of summary reports required by NCI of its designated Cancer Centers, and facilitates compliance with Federal regulations including 21 CFR Part 11, HIPAA, and Section 508.

Clinical workflows are enabled by both subject- and study-centric views into the registration process. C3PR can be run in a standalone mode where study definitions, investigators, study personnel, and sites are entered into the system, or in an integrated mode with the Suite. C3PR also enables multisite clinical trials where registration information is entered locally at affiliate sites and the registration is completed by call-out to the coordinating site.

10.3.3 Clinical Data Exchange

caXchange is a configurable service and messaging hub for exchanging clinical information between applications and systems. For instance, caXchange can be used to map and automatically transfer clinical data from point-of-care systems, such as clinical chemistry laboratory systems, into standard formats such as the Health Level Seven (HL7) Version 3 message format for periodic reporting of laboratory data in clinical trials. caXchange can then route this data to clinical trials databases, using the reliable transaction control of an Enterprise Service Bus (ESB) architecture to ensure it is successfully received. The configurability of caXchange enables this mapping and transfer no matter how nonstandard is the data format of the source system, or for that matter the destination system.

caXchange also provides a virtualizable data warehouse, the Clinical Trials Object Data System (CTODS), to collect laboratory data gathered during a clinical trial, and an application (Lab Viewer) that allows users to view and query laboratory data sent to the hub and select subsets of tests to be sent to a clinical trials database. Lab Viewer allows search by Medical Record Number (MRN) and date range, and automatically flags out-of-range laboratory result values that may indicate toxicities.

10.3.4 Participant Study Calendar

Participant Study Calendar (PSC) is an open-source, standards-compliant software application intended to manage study participants on clinical trials. At the highest level, it takes as input a study calendar template defining the schedule of tests, administrations of study agents, etc., required by the study protocol, and the on-study date of a study participant, and uses them to generate a personalized calendar

of events for the participant. PSC provides the ability to create, edit, import, and export study calendar templates, generate and view prospective calendars of study participant activities, track activities as they occur, manage participant schedules as they change during a study and share them with study participants, either on paper or via standard electronic calendar files. PSC accommodates all types of studies and facilitates management of the screening process, registration, active monitoring, and long-term follow-up. Templates can be exchanged and shared between instances of PSC. Reporting includes the ability to provide prospective and historical views of participant activities, and to track the history of changes to an activity as well as its ideal date. PSC can manage changes to templates from protocol amendments, reconsent of participants on a study and access control to participant calendars within a multisite environment. It can also receive adverse event notifications from a caBIG® compatible adverse event system, such as caAERS, and display them in the participant calendar.

10.3.5 Clinical Connector

The Clinical Connector is a connectivity utility. It provides a semantically integrated service that allows clinical data management systems (CDMSes), including commercial and open-source solutions, to integrate their functionality with the Suite. For instance, C3PR can use the Clinical Connector to enroll study participants onto clinical trials hosted by the CDMS.

10.4 Role of the caBIG® Clinical Trials Suite within the caBIG® Program

The Suite is not just a set of interoperable tools intended to fulfill specific research needs; it is also a reference implementation of NCI's enterprise architecture. Implicit in the notion of modularization of software is a need for true interoperability; when needed, the modular components must work together seamlessly, as though they were a single monolithic system. NCI's clinical research community expressed the need for the Suite's components to fulfill a number of specific real-world interoperability scenarios that require dependable, near-real-time coordination of actions between components; in short, computable semantic interoperability (CSI).

NCI's Enterprise Conformance and Compliance Framework distinguishes between three kinds of interoperability as follows:

Syntactic Interoperability guarantees the exchange of the structure of the data, but carries no assurance that the meaning will be interpreted identically by all parties. Web pages built with HTML and/or XML are good examples of machine-to-machine

syntactic interoperability since a properly structured page can be read by any machine with a web browser. However, the meaning of the page to a particular machine may vary substantially. This is not usually deemed to be a problem since the semantics of a page's contents are meant to be interpreted by human viewers. The ability of browsers to display HTML pages regardless of the implementation technology (i.e., the browser) or the content of the web page is an example of syntactic interoperability.

(Human) Semantic Interoperability guarantees that the meaning of a structure is unambiguously exchanged between humans. Documents such as progress notes, referrals, and consultant reports rely on the specificity of medical vocabularies and on common community practice to guarantee semantic interoperability at a clinician-to-clinician level. The ability of a human being to read a clinical discharge summary formatted in multiple ways in multiple contexts and still extract the "true meaning" irrespective of its presentation is an example of human semantic interoperability.

Computable Semantic Interoperability (CSI) requires that the meaning of data be unambiguously exchanged from machine to machine. Note that this does not necessarily mean that all machines need to process the received data the same way, but rather that each machine will make its processing decisions based on the same meaning.

Thus, for instance, in the caBIG® Clinical Trials Suite, the action of registering a study participant to a study in C3PR needs to trigger different actions in other components: caXchange will attempt to retrieve the study participant's baseline laboratory values, PSC will generate a personalized trial calendar for the study participant, and so on. Computable semantic interoperability is required because the data transmitted – in this case the data point that indicates the registration of the study participant – must cause other component applications receiving it to perform predictable actions without human intervention.

Furthermore, because new component applications will periodically be added to the Suite, and newly required real-world interoperability scenarios may be identified at any time, it is important that additions and changes to the way the components interoperate be accommodated with minimal, if any, changes required in the way in which the information is expressed. This requirement is especially challenging because the components of the Suite have been developed by different, indeed competing, teams of software engineers, so the opportunities for "hands-on" coordination during software development are constrained. This inherent "coopertition" between teams is by design; as already noted, implicit in the caBIG® paradigm is an expectation that disparate groups, including government, for-profit and not-for-profit entities can contribute components that, if they conform to the architecture, will interoperate without the need for complex negotiations.

Key to the ability of the caBIG® Clinical Trials Suite to achieve this interoperability is the requirement that all development teams harmonize the semantics of the information that they process with a single Domain Analysis Model, *i.e.*, a shared view of the dynamic and static semantics that collectively define a shared domain-of-interest. The accepted model of the semantics of the domain of clinical and preclinical research is called the Biomedical Research Integrated Domain

Group (BRIDG) model, which has been developed jointly by caBIG®, the Clinical Data Interchange Standards Consortium (CDISC), HL7 and the FDA. Harmonizing an application's data model with BRIDG not only ensures that the semantics of that application are aligned to those of another application similarly harmonized, it also enables domain experts (in this case, practitioners of clinical research) to understand, and thus to review for correctness and completeness, the semantics of the application. Furthermore, as BRIDG is itself mapped to HL7's Reference Information Model (RIM), mapping an application to BRIDG ensures that the application's semantics are HL7-compliant. In the case of the caBIG® Clinical Trials Suite, introducing the requirement for BRIDG harmonization in the inception and elaboration phases of application development meant that post hoc, it was possible to add new, unplanned, real-world interoperability scenarios and deliver a working demonstration of those scenarios within 2 months.

The second component of the Suite that facilitates the required interoperability is caXchange. In addition to being an application component that delivers integration of clinical care data into clinical research, caXchange also acts as middleware, using an ESB to ensure reliable transaction control, i.e., that component applications can know whether a signal to another application to perform some action was not only received, but also acted upon in the appropriate way within a set time interval.

10.5 Support

NCI has established an Enterprise Support Network in order to provide users of the Suite with options for obtaining support. NCI has designated a series of Knowledge Centers to provide free domain-focused support and to serve as a venue for a peer support community of practice via wikis, bulletin boards, bug notification/feature request tracking and a source code repository. The Suite is supported by the Clinical Trials Management Systems Knowledge Center. In addition, NCI has licensed a series of Support Service Providers, commercial organizations qualified to provide full-service support for organizations using the software on a contract basis.

10.6 User Base

Users of the Suite and its applications fall into three groups. First, each application has between one and three designated Adopter organizations from the NCI clinical research community – these organizations work with NCI and the application developers to install and use the applications, and deliver feedback on features and functionality. In addition, NCI has established a Center Deployment Program for the NCI-designated Cancer Center community. At the time of writing, 49 out of the 65 NCI-designated Cancer Centers were participating in the Center Deployment Program, and 31 out of these 49 were either adopting, or evaluating the adoption

of, the Suite or one or more of its component applications. Finally, as caBIG® software is freely available for download, NCI often does not know about users of software until they contact NCI for support. The University of Arkansas for Medical Sciences is a notable example of an organization that, despite being neither a designated Adopter not a Center Deployment Program participant, has taken full advantage of the open-source nature of the software to seamlessly integrate its local applications with those of the Suite.

10.7 Enterprise Architecture

In parallel with the development of the first generation of the Suite, caBIG®'s enterprise architecture was strengthened to incorporate NCI Enterprise Services – common, shareable sources of record for areas of information that they have in common. As an example, all applications in the Suite have a concept of a clinical trial protocol. Not only does allowing each application to represent and store protocols locally mean that the applications have to synchronize with each other when one application adds, removes or, in some cases, edits a protocol, the way in which each application represents a protocol will almost certainly be different. Even after all applications have harmonized with BRIDG, typically each application will store only the elements it requires for a protocol. The first step in refactoring the Suite applications to use common services is for a common functional specification to be developed, in this case for representation of a protocol. This representation is then implemented by all applications, turning redundant, inconsistent representations into redundant, consistent ones. The next step is to eliminate the redundancy by implementing the functional specification as a common service which all applications access. This not only eliminates the need for synchronization, it also reduces the amount of code in each application, making them more agile and less error-prone. Furthermore, the services can be federated, allowing for some protocols to be stored in a local instance of the service while others can be stored and curated in a central instance. The first four NCI Enterprise Services – for protocols, people, organizations, and correlations between these three, were released into production in January 2009, and at the time of writing, all applications in the Suite were being refactored to leverage NCI Enterprise Services, with a planned release date of November 2009. In addition, many other NCI Enterprise Services were in development, including those for other key clinical trials concepts such as Adverse Event, Schedule, and Registration.

10.8 Availability

The caBIG® Clinical Trials Suite is available freely for download and use under a nonviral open-source license that permits redistribution, modification, and incorporation of the code in other products. Free domain-specific and peer support is available

from the caBIG® Clinical Trials Management Systems Knowledge Center. Access is available at the following links:

Main page for the Suite: https://cabig.nci.nih.gov/tools/toolsuite_view#CCTS

Documentation page for the Suite: https://cabig-kc.nci.nih.gov/CTMS/KC/index.php/CCTS

Documentation page for all Suite applications: https://cabig-kc.nci.nih.gov/CTMS/KC/index.php/Documentation_Library

caAERS overview: https://cabig-kc.nci.nih.gov/CTMS/KC/index.php/CaAERS

C3PR overview: https://cabig-kc.nci.nih.gov/CTMS/KC/index.php/C3PR

PSC overview: https://cabig-kc.nci.nih.gov/CTMS/KC/index.php/PSC

caXchange overview: https://cabig-kc.nci.nih.gov/CTMS/KC/index.php/CaXchange

Download Center for the Suite: http://ncicb.nci.nih.gov/download/cctslicenseagreement.jsp

Knowledge Center (Support): https://cabig-kc.nci.nih.gov/CTMS/KC/index.php/Main_Page

Support for the Download Center, or if any of the above pages are unavailable, can be obtained via e-mail at ncicb@pop.nci.nih.gov, or by telephone at +1 301-451-4384 (toll free from within the United States: 888-478-4423).

Reference

Dilts DM, Sandler AB (2005) Invisible Barriers to Clinical Trials: The Impact of Structural, Infrastructural, and Procedural Barriers to Opening Oncology Clinical Trials. J Clin Oncol 24:4545–4552. DOI: 10.1200/JCO.2005.05.0104

Chapter 11
The CAISIS Research Data System

Paul Fearn and Frank Sculli

Abstract CAISIS is a research data system that was started to support predictive model development in urologic oncology at Memorial Sloan-Kettering Cancer Center (MSKCC) and has become widely adopted across different departments and institutions to manage biomedical research data for a variety of diseases, mostly cancers. It was developed using ASP.NET/C# and Microsoft SQL Server, and is freely distributed under the GPL open-source license. This system complements both clinical systems and clinical data repositories, and its functionality has been extended recently to manage biorepositories and prospective clinical trials. The database structure is organized temporally and around patients rather than around protocols or individual projects, which allows it to be extended to manage data for multiple diseases and medical specialties.

11.1 Introduction

CAISIS (pronounced "keisis") is an open-source, web-based system that was initiated at Memorial Sloan-Kettering Cancer Center (MSKCC) as a Microsoft Access database for retrospective prostate cancer outcomes research in the late 1990s. With contributions from many groups and individuals, it has grown over 10 years and multiple iterations to manage a wide range of clinical research activities across many oncology disease groups, departments, and cancer centers (Fearn et al. 2003; Potters et al. 2003; Fearn et al. 2004; Fearn et al. 2007b). Although CAISIS would probably not be classified as a clinical system (e.g.,

P. Fearn (✉)
Department of Medical Education and Biomedical Informatics,
Division of Biomedical and Health Informatics, University of Washington,
1959 NE Pacific St, HSC I-264 Box 357240 Seattle, WA 98195-7240
e-mail: fearnp@uw.edu

M.F. Ochs et al. (eds.), *Biomedical Informatics for Cancer Research*,
DOI 10.1007/978-1-4419-5714-6_11, © Springer Science+Business Media, LLC 2010

Electronic Medical Record or Electronic Health Record, see Chap. 2) by most IT or informatics professionals, the current version (v4.5) has a number of features that enable integration of research data collection with clinical practice, including a framework for browsing clinical histories, creating and managing structured templates for clinical data capture, and generating documentation for the medical record. Development of CAISIS from 2006 to 2009 has been partly supported by a grant from the National Cancer Institute (R01-CA119947). This funding was used to make the application more modular and extensible to handle a number of new diseases, to make the interface dynamically driven from metadata and XML configuration files, and to make the application easier to set up, configure and use for a rapidly growing user community. The CAISIS community is active and growing across centers in the U.S. and internationally, and the primary development teams are at MSKCC and BioDigital Systems. Since 2007, a number of groups in the CAISIS community have started to fund the development of enhancements and new functionality (e.g., patient quality of life survey management, biospecimen management, clinical trials management, LDAP authentication). At most sites, data feeds are established to pull demographic, scheduling and laboratory data into CAISIS from clinical systems to reduce costly manual data abstraction and data entry. Most of these data feeds are extract-transform-load (ETL) procedures that pull structured data from an enterprise data warehouse (see Chap. 3) or clinical data repository, and these interfaces are most often implemented using Microsoft SQL Server Integration Services (SSIS). The CAISIS data model and vocabulary are well documented within the system metadata tables, and there are no proprietary or obscure formats or APIs required to interface with the system. As caBIG® and other data exchange stacks continue to emerge, MSKCC and BioDigital are working to make CAISIS interoperable with these standards (National Cancer Institute 2009).

11.2 Contact Information

CAISIS public web site: http://CAISIS.org
CAISIS LinkedIn group: http://www.linkedin.com/groups?gid=1778711
Email for CAISIS team at MSKCC: CAISISAdmin@mskcc.org
BioDigital Systems, LLC public web site: http://www.biodigital.com

11.3 Availability

All of the CAISIS system requirements, installation packages, upgrade scripts, source code, and documentation are available on http://CAISIS.org or through SourceForge (http://sourceforge.net/projects/caisis/). It is freely distributed under version 2.0 of the GPL open-source license (Free Software Foundation 1991).

The development source code is managed using Subversion, and access is granted upon request to any interested individual or group (Tigris 2009).

11.4 Links for Documentation, Installation Guides, Notes

http://caisis.org/wiki/index.php?title=Installation. *Installation test examples:* http://caisis.org/demo/login.aspx. *User guides:* http://caisis.org/wiki/index.php?title=Data_Entry_Guide.

11.5 Description of Tool

Many of the ideas behind CAISIS originated at Baylor College of Medicine in the 1990s out of the need to assemble large datasets in a reusable and sustainable format to produce predictive models (Kattan et al. 2000; Kattan et al. 2001; Cuzick et al. 2006; Stephenson et al. 2006; Kattan et al. 2008). Previously built retrospective research databases and project-specific databases had become difficult to scale or impossible to sustain as the number of patients followed grew from hundreds to thousands and as the number of research projects drawing from these databases increased. Most research groups were not able to scale up their productivity because the per-project cost of data collection and dataset generation is too great.

Duplication of data entry activities is common. Many groups abstract clinical data from medical records and enter them into disparate systems for different purposes (e.g., retrospective research projects, prospective outcomes protocols, clinical trials, clinical practice operations, quality assurance, outcomes reporting, tumor registry reporting.) The intent of CAISIS is to provide a single repository to centralize and coordinate multiple streams of data entry, data processing, and production of information.

Staff entrenchment and inability to easily deploy staff resources across a variety of disease- or project-specific databases to cover changing research data management needs is also a common issue that CAISIS was designed to address. In many research groups, individual research project databases for single diseases are managed by individual staff members. Much of the metadata for these systems is undocumented, and researchers are completely dependent upon the individuals who understand how to enter and retrieve data from single-purpose databases. The CAISIS data model was designed to organize data temporally and provide a granular structure that facilitates the use of the similar data across multiple diseases and medical specialties. With this approach, it is possible to train and allocate staff resources more efficiently, allowing resources to shift according to demand rather than being locked to a single purpose and dataset. A person who can enter data into or query data from CAISIS for one disease has most of the knowledge needed to manage data for other diseases. Although there is definitely a learning curve associated

with this approach, the startup and training costs are offset by downstream economies of scale, scope, and experience.

Because the patient-centric CAISIS database has a higher degree of normalization than many study centric databases, and a patient's medical history is stored in highly structured, temporally organized fields, there is less risk of introducing investigator bias and subjective data interpretation during data collection or analysis. Before statistical analysis, data from CAISIS is generally queried and processed into denormalized research datasets or reports, but the methods for generating these datasets are explicit and reproducible, often taking the form of configurable data processing algorithms.

From its early days as a Microsoft Access database to the current web-based ASP.NET and SQL Server application, CAISIS was made freely available under the GPL open-source license to promote collaborative research. There is generally a high per-project cost for interinstitutional collaboration, which often requires painstaking mapping, quality assurance and reabstraction of data when combining datasets across multiple sites. By running CAISIS as an open-source effort and helping many other sites set up the system, organize their data management activities around this common data structure, and interface with other systems in their local IT ecosystem, the CAISIS developers at MSKCC aim to reduce the per-project transaction costs of interinstitutional research, as well as to establish working relationships across cancer centers that are conducive to future research collaboration.

The aims of the CAISIS initiative are to create a reusable patient history structured in a temporal format, to increase reproducibility of study results by separating the interpretation of raw data for analysis from its collection and storage in a research database, to promote interinstitutional collaboration, to assemble large, high quality, and minimally biased datasets for predictive modeling, to improve productivity of investigators and research staff, and to achieve economies of scale and scope in the overall research data supply chain. In order to achieve these aims, CAISIS was built around a single, patient-centric database model that would scale to manage clinical and research data for all diseases and medical specialties. The database and web application have been specially architected to support these end goals.

11.5.1 Details on Tool Function

Unlike more monolithic systems, version 4.5 of CAISIS is a tiered framework built to handle the rapid addition of new modules and diseases. Individual modules are loosely coupled ASP.NET applications that run within distinct tabs and share a common platform for navigation, security, database access, transferring data objects and cross-browser presentation. Similar to common libraries like Hibernate (Red Hat Middleware, Inc. 2009), the CAISIS object/relational mapper (ORM) liberates the developer from programming repetitive data persistence tasks. With the creation of a CAISIS Business Object, the complexities of reading, writing, and

transferring data across tiers is abstracted from the developer. Use of attributes within these objects further hides implementation details. For example, classes attributed with “Exportable” and “Breast” are automatically included in breast cancer data exports. Members within that class attributed with “Deidentify” are automatically removed from the export. At the user interface (UI) level, database-driven metadata allows customization without programming. Presentation properties such as labels, input control types, validation, field widths, visibility, drop-down options and form styles can be modified by system administrators with little technical expertise. The ease of local customization is one of the primary factors in widespread adoption of CAISIS and remains an integral design principal for all new development.

The Patient Data tab of CAISIS (Fig. 11.1) is completely dynamic, driven from metadata about tables, fields and vocabulary for drop-down lists, as well as XML configuration files. The left side of Fig. 11.1 shows the chronological or “Date/Variable/Value” list, which is essentially a stacked and sorted subset of key fields from data tables directly connected to the patient table (i.e., medications, lab tests, procedures, pathology, image studies, comorbidities, etc.). This chronological list of patient data allows clinicians, researchers, and data management staff to quickly view or navigate information across a patient’s entire medical history.

As the user clicks on items in the chronological list, the details branch off to the right frame of the web application, which has a number of noteworthy features: (1) all dates in CAISIS are stored both as text and as datetime fields, allowing for entry of incomplete dates; (2) all web controls, drop-down lists and field attributes (i.e., required labels, help bubbles) are drawn from metadata tables in the database; (3)

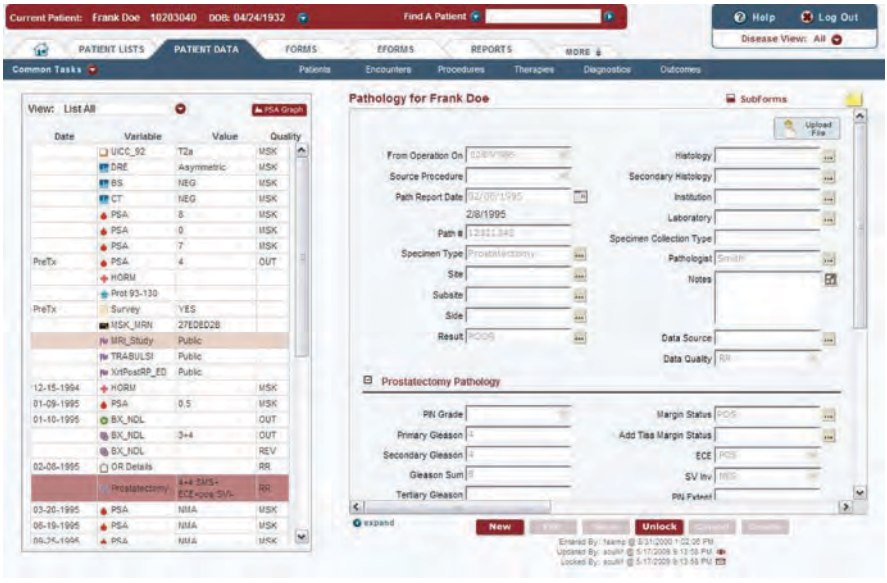


Fig. 11.1 CAISIS Patient Data section

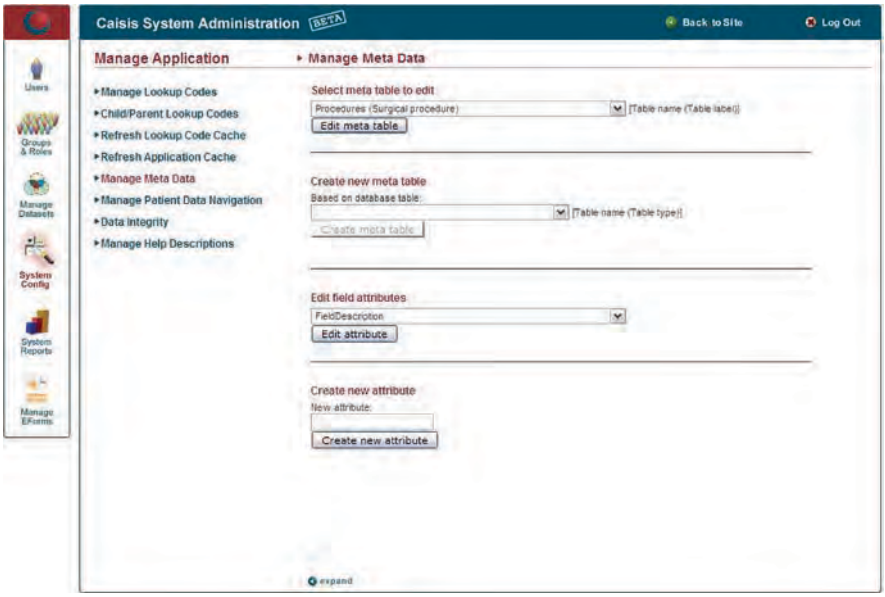


Fig. 11.2 CAISIS System Administration: configuring the UI through Metadata

all record updates and deletes are audited so that entire change logs for each record can be viewed from the UI; (4) the amount of variation allowed for data entry is configurable from metadata (Fig. 11.2) so that an instance of CAISIS can use any combination of open text boxes, select boxes, combo boxes (input controls that allow either drop-down or type-in of desired value); (5) the File Upload Utility Plug-in can be included on any patient data form to allow users to attach source document Word, Excel, PDF or image files to individual records; (6) most forms allow users to annotate records with a specific data source, data quality and notes or comments. The Patient Data section reflects the underlying data model of CAISIS, which is patient-centric and designed to handle multiple diseases. For disease- or treatment-specific differences, the system has subtables and associated subforms as well as the ability to add virtual entity-attribute-value (EAV) fields without programming (Nadkarni et al. 2000).

The Patient Lists tab allows the user to list groups of patients in CAISIS by clinic schedule, last name, protocol, physician, user-defined categories, and other options.

The eForms tab is one of the key features of CAISIS and an active area of development. It offers a way to provide users with custom interfaces that follow specific data entry workflows and hide the underlying database structure. Entered data is stored temporarily in XML until approval by a user in the appropriate role. Upon approval, the system parses the XML, inserts the data into the database, and generates a report through an XSL transform. This functionality is analogous to

clinical documentation in an EMR. It allows users to capture data through structured templates and to generate notes that replace dictation. The Paper Forms tab is a predecessor to the eForms tab. It has been used to prototype structured and pre-populated templates to capture data for research during clinical practice encounters.

The Reports and Data Export sections of CAISIS allow users to drop canned SQL reports into the framework and to export data into an Access database to run operational reports. Access to any tab can be restricted to certain groups under the role-based security framework.

The CAISIS Project Tracking module is the most recent addition to the application. It is based on a Microsoft Access prototype that has been used at MSKCC to track all departmental research projects from idea through publication. The functionality has been extended to track all files associated with projects, manuscripts, citations, as well as protocol development for clinical trials.

The Protocol Manager module is a tool for developing a protocol schema and then automatically generating patient study calendars. It also guides data entry for protocols. This feature will be extended as a key part of CAISIS' multisite clinical trials management functionality.

The Specimen Manager is a simple biospecimen tracking system designed to link with the clinical annotation in CAISIS' Patient Data section. The Specimen Manager development was originally funded by Westmead Millenium Institute in Sydney, Australia (Carpenter et al. 2007; Fearn et al. 2007a).

The Patient Education tab allows clinical staff to create and deliver customized patient education materials for individual patients based on their unique history. It is currently a content management system with an interface for navigating and selecting portions of content and printing them for patients, but it could be extended to automatically select appropriate content by applying rules or queries to the underlying patient data.

The System Admin tab allows designated users to configure everything about the application, including setting up datasets, configuring and auditing security, configuring the UI or drop-down lists, browsing and modifying data.

Overall, the unified look and feel of the application UI, including the images and icons, color palette, fonts, screen layout and navigation were designed by a single graphic/web designer. Visual and interaction design is often an underappreciated but important aspect of medical systems. Initial reactions and user satisfaction seem to be significantly impacted by the look and feel of the UI.

11.5.2 Description of Input/Output

At most sites, CAISIS is implemented as departmental research data repository, and users input data into the system from a variety of sources. Abstracting data from medical records and manually entering data into the patient data section is generally a fundamental activity to support CAISIS. There is a steep but short learning curve

that most users go through when shifting from a field-oriented system to the relational and temporal structure of CAISIS. For the first couple cases entered, the UI seems slow and complex; however, most users are do not notice the complexity as much after entering 10 cases, and after 50–100 cases they tend to prefer the CAISIS format over other systems. Most sites start implementation with a prospective data entry process, and then migrate existing retrospective datasets into the system through a one-time data mapping and batch import process. High volume data that can be easily retrieved from an enterprise data warehouse or clinical data repository such as lab tests, demographics, clinic visits, and surgical procedures are commonly pulled into CAISIS from regularly scheduled SSIS/ETL data feeds. Most sites have not found it necessary to build real-time HL7 interfaces or use service oriented methods to exchange data with CAISIS; however, each individual site has different needs and circumstances, and in some cases real-time application interfaces are desired. Many cancer centers are in the process of implementing EPIC, Eclipsys or another leading EMR system, and several are planning to pull structured data from those systems into CAISIS for research.

There are several methods to output data from CAISIS. The Reports tab is a simple reporting engine that allows users to drop a SQL query into the framework, configure it with an XML file, and then allow users to run the report through the UI. From the UI, users can export the report to Excel. The Data Export tab allows authorized users to export data from CAISIS to XML. An import of the XML into Microsoft Access or a statistical package allows users to write queries and analyze the data themselves. In larger installations such as at MSKCC, which has over 120,000 patient histories in the system, the production instance of CAISIS is replicated into a warehouse copy, disease and dataset specific views are created, and authorized users are allowed to connect to and query the warehouse using Microsoft Access, SQL Server, or their own query and reporting tool of choice. A number of utilities such as longitudinal follow-up tools to assist with mailing letters to patients, algorithms for generating research datasets, and database browsers are available in an “Export Analysis Utility” which is a separate Microsoft Access application that can be downloaded from the website and linked to CAISIS.

11.5.3 Technical Details

CAISIS was originally a prostate cancer database prototyped in Microsoft Access in the late 1990s, and the current database structure reflects some of the early database structure and application design. With the help of BioDigital Systems, it was scaled up to a web-based ColdFusion and SQL Server platform in 2002, and in 2004 the web application was completely rewritten using ASP.NET and C#. BioDigital Systems has developed the majority of the web application, and most of the SQL Server database was developed by MSKCC staff in the Department of Surgery.

- *Language(s) used*: ASP.NET/C#, JavaScript, HTML, XML.
- *Ancillary tools needed for compilation, application hosting, etc.*: Microsoft Visual Studio 2005/2008, Microsoft SQL Server 2005 or greater, IIS 5+, and Windows Server. See System Requirements on CAISIS.org for details.
- *Version/Suite dependencies*: CAISIS v4.5 requires SQL Server 2005 or greater and .NET framework version 2.0. Cross-browser compatible with MSIE v6+, Safari, and Firefox 2+.
- Documentation is available on the wiki (http://caisis.org/wiki/index.php). There are also links on the public website for the Issue Tracker and Forums. There is a CAISIS mailing list and bi-weekly web conference call which users can sign up for by emailing CaisisAdmin@mskcc.org. The CAISIS group on LinkedIn.com is also a useful way of interacting with the broader CAISIS user community.

CAISIS is freely distributed under the open-source GNU Public License, version 2.0 (Free Software Foundation 1991).

11.6 Suggested Best Practices for Use of Tool in Research Setting

First, there are no magic bullet solutions. Implementing CAISIS or any other clinical research system will not cure operational problems. However, implementing a new information system like CAISIS can be an opportunity or catalyst for changing roles, processes, and practices in ways that will pay off in the long-term. The best practice for using CAISIS is to explore and learn how to use it most effectively through local pilot projects, and to leverage the larger user and developer community for ideas and assistance.

Second, many of the technical stumbling blocks encountered in implementation of CAISIS seem to be due to inadequate knowledge and skills in the underlying technologies: Microsoft SQL Server, ASP.NET and C#. Sites interested in using CAISIS should have support from IT professionals with up-to-date Microsoft training and certifications for these technologies (Microsoft 2009). In most cases, running CAISIS in a professionally managed enterprise data center is preferable and in the long run less expensive than "under the desk" implementations with inadequate support.

Third, CAISIS should be considered part of an overall IT ecosystem and research data supply chain rather than just a standalone research application for a particular project (Fearn et al. 2007c). Manual data entry can be expensive and time consuming, so identifying and implementing a variety of opportunities for data capture from different staff, systems and methods, routing multiple streams of data into CAISIS, and producing useful output in return for data entry (e.g., clinical notes, operational and outcomes reports, research project datasets) can go a long way toward reaping value from the system, gaining stakeholder buy-in, and building a sustainable data supply chain.

Fourth, there are learning curves and economies of scale, scope, and experience. Many groups choose apparently "simple" database solutions that work well in the

short-term, but that are ultimately not scalable or sustainable. Any solution seems to work for hundreds of cases, but managing thousands of cases requires a different approach. At MSKCC, CAISIS is currently being used to manage over 120,000 cases in over ten cancer types. When a new group starts using the system, they learn a new, more generalizable method for entering and retrieving data, which at first seems overly complex and unfamiliar. There is a tendency among new users to attempt to overcustomize the application in the early stages of implementation rather than wrestle their way up the learning curve. This is probably a mistake with any system implementation. Groups that take the time to learn the system and explore options for redesigning their operations before customization achieve payoffs in research productivity and start to experience economies of scale and scope as they learn to use it effectively, integrate it with other systems, and reorganize their data supply chain.

11.7 Future Development and Enhancement Plans, Extensions, or Novel Uses

At this point, with increasing modularity, metadata-driven dynamic interfaces, custom controls, and virtual fields, CAISIS is becoming more like a framework for adapting to local clinical research needs than a specific application. As existing sites expand their usage of this system and new sites start using it, the development team at MSKCC and BioDigital plan to further generalize the system so that it can be configured to serve many functions in both patient care and research.

Most sites that are using CAISIS are also in the process of implementing enterprise clinical systems and EDW/CDRs, and building integrated data supply chains so that data captured in a variety of source systems can be piped to downstream uses in other systems. CAISIS will need to become more interoperable and standards-based (i.e., using or mapping to standard terminologies like SNOMED and LOINC, providing APIs for caBIG® and CDISC, and integrating with HL7 and service-based feeds).

CAISIS is currently expanding rapidly to manage data for multiple services within surgical oncology, and the clinical trials management functionality will enable further extension across medical oncology groups. In terms of functionality, the clinical trials management features in CAISIS will be significantly enhanced and integrated over the next couple years to enable management of clinical trials across a network of participating sites. To effectively broker the secure exchange of data between these sites, a robust middle tier API will be built using industry standard messaging protocols.

Biorepositories are the resource and data link that enables translational research. The CAISIS specimen manager will likely continue to be enhanced rapidly, and will need to interface with caTissue or the caBIG® Common Biorepository Module to provide interoperability with this maturing infrastructure.

References

Carpenter JE, Miller JA, Sculli F et al (2007) The caisis system for biorepository data requirements--breast cancer tissue bank, Australia. In: Abstracts of the ISBER Annual Meeting, Singapore, 30 May – 2 June 2007

Cuzick J, Fisher G, Kattan MW et al (2006) Long-term outcome among men with conservatively treated localised prostate cancer. Br J Cancer 95:1186–1194

Fearn PA and Carpenter J (2007a) Collaborative development of caisis systems to support clinical and biospecimen data and workflows. In: Abstracts of the Australasian Biospecimen Network (ABN) 2007 conference, Melbourne, Australia, 16–17 October 2007

Fearn PA, Regan K, Sculli F et al (2003) A chronological database as backbone for clinical practice and research data management. In: Proceedings of 16th annual IEEE symposium on computer-based medical systems, New York, NY

Fearn PA, Lafferty HJ, Bauer MJ et al (2004) A clinical research database solution for HIPAA privacy and security requirements. In: Abstracts of MedInfo, San Francisco, CA

Fearn PA, Regan K, Sculli F et al. (2007b) Lessons learned from caisis: An open source, web-based system for integrating clinical practice and research. In: Proceedings of 20th annual IEEE symposium on computer based medical systems, Maribor, Slovenia

Fearn PA, Regan K, Sculli F et al (2007c) Caisis 4.0: re-designing the data supply chain. In: Proceedings of advancing practice, instruction and innovation through informatics (APIII), Pittsburgh, PA, 10 September 2007

Free Software Foundation, Inc (1991). GNU general public license (GPL). http://www.opensource.org/licenses/gpl-2.0.php. Accessed 17 May 2009

Kattan MW, Fearn PA, Leibel S et al (2000) The definition of biochemical failure in patients treated with definitive radiotherapy. Int J Radiat Oncol Biol Phys 48:1469–1474

Kattan MW, Potters L, Blasko JC et al (2001) Pretreatment nomogram for predicting freedom from recurrence after permanent prostate brachytherapy in prostate cancer. Urology 58:393–399

Kattan MW, Cuzick J, Fisher G et al (2008) Nomogram incorporating PSA level to predict cancer-specific survival for men with clinically localized prostate cancer managed without curative intent. Cancer 112:69–74

Microsoft. (2009) Microsoft certifications overview. http://www.microsoft.com/learning/mcp/default.mspx. Accessed 17 May 2009

Nadkarni PM, Brandt CM, Marenco L (2000) WebEAV: Automatic metadata-driven generation of web interfaces to entity-attribute-value databases. J Am Med Inform Assoc 7:343–356

National Cancer Institute (2009) Cancer Biomedical Informatics Grid. https://cabig.nci.nih.gov. Accessed 18 May 2009

Potters L, Kattan MW, Fearn P (2003) A chronological database to support outcomes research in prostate cancer. Int J Radiat Oncol Biol Phys 56:1252–1258

Red Hat Middleware, Inc (2009) https://www.hibernate.org. Accessed 17 May 2009

Stephenson AJ, Scardino PT, Eastham JA et al (2006) Preoperative nomogram predicting the 10-year probability of prostate cancer recurrence after radical prostatectomy. J Natl Cancer Inst 98:715–717

Tigris (2009) Subversion. http://subversion.tigris.org. Accessed 17 May 2009

Chapter 12
A Common Application Framework that is Extensible: CAF-É

Richard Evans, Mark DeTomaso, Reed Comire, Vaibhav Bora, Jeet Poonater, Aarti Vaishnav, Scott Catherall, and John T. Casagrande

Abstract Data collection and management is an essential activity for most scientific investigations. To be able to "quickly" develop and support the data collection needs for the various types of investigations undertaken in a comprehensive cancer center is a challenge. CAF-É is an object-oriented development environment that combines common objects with study-specific data entry form(s) libraries with common and study-specific metadata has been used successfully to address this task. It is based on a Windows forms front-end with a SQL Server database as the back-end.

12.1 Introduction

Data collection and management is an essential activity for most scientific investigations. To be able to "quickly" develop and support the data collection needs for the various types of investigations undertaken in a comprehensive cancer center is a challenge. Investigations and/or data collection needs run the gamut from "administrative" type activities such as specimen inventory and dispersal to tissue microarray data capture thru translational research projects involving the integration of clinical/disease status with tissue and/or "omics" or lab information from a variety of sources.

Investigators have historically tried to address these needs using a variety of strategies and tools. Many options are available for an investigator ranging from purchasing commercial systems tailored for a specific purpose through designing and building a custom study-specific application. Every alternative has its own set of advantages, challenges, and costs that must be considered and no single solution may work for all situations. We present here a framework developed to support the data

J.T. Casagrande (✉)
University of Southern California, Harlyne Norris Cancer Research Tower MC 9601, Los Angeles, CA 90089-9601, USA
e-mail: Casagrande_j@ccnt.usc.edu

M.F. Ochs et al. (eds.), *Biomedical Informatics for Cancer Research*,
DOI 10.1007/978-1-4419-5714-6_12, © Springer Science+Business Media, LLC 2010

collection activities at the University of Southern California's Norris Comprehensive Cancer Center.

CAF-É is an object-oriented development environment that combines common objects with study-specific data entry form(s) libraries with common and study-specific metadata. It was initially developed for protocol management, patient enrollment, and electronic data capture (EDC) to manage clinical trials (CTMS). It has now been generalized and has been used for tissue microarray data capture, epidemiologic studies, prevention trials, laboratory management, tissue repositories, and administrative systems such as the Center's membership database.

12.2 Description of CAF-É

CAF-É utilizes a Microsoft SQL server back-end database that is accessed using a Windows forms based front end to access the databases and present data entry forms and reports to the user. The CTMS application developed in CAF-É includes many of the essential features typical of commercial clinical trials EDC systems (e.g., capability for complex business rules, audit trails, role-based security model), but in contrast to most commercial EDC systems, it uses a hierarchical organization model that nests individual studies within a more global data model for patient and study data, thus making it well suited for cancer center environments where studies must be tightly managed and where a single patient can participate sequentially or concurrently in multiple studies or trials.

CAF-É applications rely upon several databases (CAFÉ, Look Up, App, Study) detailed below and containing all user and application-specific research data and metadata. CAF-É provides a dynamically configurable user interface via a menu tree built from a database and handles user access and role-based security. For data storage/retrieval, CAF-É utilizes a common data access layer that simplifies development of audit trails and the binding of form controls to database fields. A key support feature utilizes built-in .NET functionality that provides the deployment of a fully featured Windows application from a Web server, so that a user can download a Windows executable simply by accessing a URL. When updates are needed, the existing components on the Web server are updated, and these are transparently moved to the client desktops by .NET when the application is started on the user's machine, so that users always run the most current version of the application without the need for intervention from IT staff. Figure 12.1 shows a pictorial view of the various components of the CAF-É framework.

12.2.1 Databases Used

The initial implementation of CAF-É was based on Microsoft SQL Server 2000, but as new versions were released CAF-É was updated accordingly. Currently CAF-É is using the Microsoft SQL Server 2008 Database engine but is downward compatible to SQL 2000. The data and metadata used by the framework was

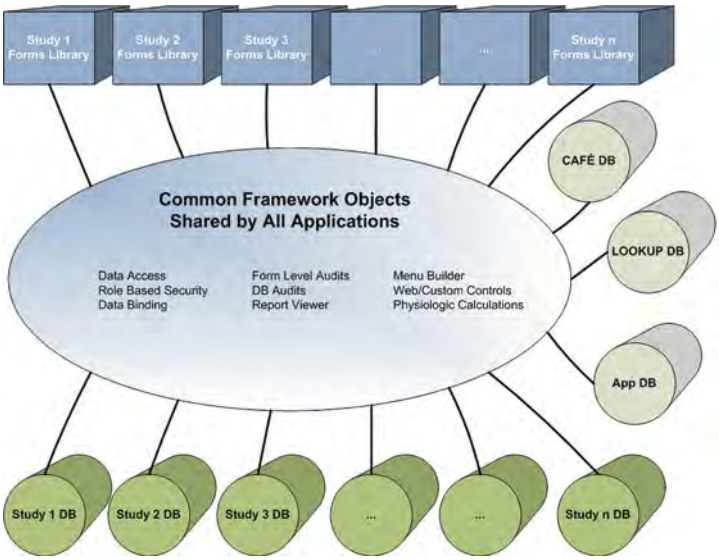


Fig. 12.1 Schematic overview of CAF-É framework components

distributed across several databases for organizational reasons. These databases include: *APP*, *CAFÉ*, *Lookup*, and *Study Specific*. *APP*, *CAFÉ*, and *Lookup* contain control and metadata used by the framework and the *Study Specific* database houses the unique data specific for each investigation or application.

The *APP* database contains several tables that contain information regarding each individual application, the users, application roles, and connection string information for the databases used in the framework. For each application, its name, id, URL, data source, connection string, application roles, and Active Directory security group name are captured. For users, their username, hashed password value, last name, first name and e-mail address, and other contact information are recorded. All user logins and normal logouts are recorded here.

The *CAFÉ* database contains several tables that contain deployment and configuration information for each individual application. A table defining the menu items for each application and the roles allowed for the application is contained here. For each application the path for the application's forms library is specified, parameters controlling whether the same form can be opened simultaneously for multiple individuals, parameters defining whether the context window or menu windows are displayed, parameters controlling form and item level auditing as well as the menu ID for the application's main "demographic" entry form are all housed here.

The *Lookup* database contains all the code lists used by the various applications for drop down lists, combo boxes, tree views, and other controls. This forces a certain level of data consistency and standardization for data capture. A standardized nomenclature and formatting has been implemented to make the access of these

lookups consistent and straightforward. When needed, study specific lookups can be used in addition to many "common" lookups that are used across one or more applications. Many standard definitions such as ICD-9 (International classification of Diseases, 2002), CPT (Current Procedutal Terminology, 2003), and Common Toxicity Category Versions 2 and 3 (Cancer Theraphy Evaluation Program, 1998, 2003) have been included here to expedite data entry.

The *Study Specific* database contains all the data that is needed by an investigator for his/her specific study. During the initial design of an application, the study workflow, the enumeration of required study data items and the content and relationships between the various tables are defined in collaboration with the user.

12.2.2 Source Code Projects

Visual Basic (VB) .Net is the programming language used for the CAF-É framework; however, objects created in any CLR-compliant language can be incorporated. Version control is done using Microsoft's Visual Source Safe. The rationale for utilizing these technologies was based on the large number of existing research applications that were already in use at our center using either Microsoft Access or Excel. We also wanted to take advantage of other Microsoft Office products such as Word or Excel in our framework, so we thought it is best to utilize Microsoft-centric tools. Microsoft's Visual Studio interactive development environment is used for the creation of all forms and program modules.

A CAF-É framework application consists of seven VB projects; six of these are "common" for all applications developed in CAF-É and only one is application specific, and it is referred to as the forms library project, since it contains the application specific data collection forms and program code. A description of each project is below.

AppExplorer is the startup project for all applications. It contains the *Login* form and the *CAF-Elite* form, which is the start-up form for all applications. The *CAF-Elite* form has three screen display areas for each application: the main menu pane, the study context pane, and the pane to display the forms, reports, or Web pages that are selected by the user via the main menu option. This latter pane is best thought of as a nested tab page. Each application may have a slightly different look and feel; however, all CAF-É applications essentially look and work the same way. A user first enters his/her username and password in a Login screen and then based on the user's roles, a drop-down list of applications that the user has been granted access to is displayed. The user then selects the desired application and the application starts.

BaseFormsLibrary is the heart of CAF-É. This project contains several base forms and user controls that are used to develop every application. The key class in this project is *BaseFormKid*. *BaseFormKid* is used to create "forms," more correctly User Controls that will display in the "forms" window and will collect and/or store the data needed by the application. Another useful base form is *BaseRowPopup*. This is a base Windows' form used to display data in a data row contained in an

Ultragrid. It provides a "pop-up" form that the programmer can use to request all data items populating the grid row. This makes the entry of a large number of data columns easier for the user since all items needed are displayed on the pop-up and eliminates scrolling through the grid columns which might be hidden from the user. Pop-up forms are windows modal forms (user must close before proceeding) which are opened separately to perform some specific function within CAF-É. They are used to select a study subject, a protocol, etc., and, since they are usually used for "picking" something, they are sometimes also referred to as "pickers."

BaseObjects contains classes used in CAF-É that are used for developing an application. Some classes that are very useful are *BaseLookup* and *FakeIdCollection*. *BaseLookup* provides a number of useful tools to work with lookup tables for drop downs and combo boxes. *FakeIdCollection* is very useful when dealing with database tables with parent–child relationships where it might be necessary to insert a foreign key in a table. This project may also contain user-developed user controls that have dependencies on other projects which do not allow them to be placed in *NCCGUIControls* discussed below.

DataAccess contains all the data layer code to connect to a SQL Server database, retrieve data from the database, and store data to the database. It also provides all the audit tracking for inserts, updates, and deletes for all transactions. In some cases Federal, Health Insurance Portability, and Accountability Act of 1996 (HIPAA) (21CFR Part 11, 1997), State or local law mandates that audit trails are kept for research databases, so if this is a requirement for an application, all database connections, and modifications must use *DataAccess* code for auditing to be done. In addition, form auditing can also be used to record user activity at the form level. The developer controls these auditing features via an *App* database parameter and there are no other specific coding issues to be concerned with. Commonly used functions in *DataAccess* are *GetDataTable* (to retrieve data) and *TableUpdateExecQuery4* (to store data).

NCCGUIControls contains many graphical user interface (GUI) controls that are used by a CAF-É application. Many of these controls are just wrappers for existing Windows controls. However, there are also some unique controls here that are useful, for example, the *MadTabContext* control that one of us (MD) has developed. The *MadTabContext* control is like a tab control and simplifies using the .Net Binding Context construct. Binding context allows the developer to bind a database column with a property of a control; for example, the text property of a textbox control, the value property of a date/time control or the selected index or value property of a drop down list control. For convenience, some user controls that need to reference resources that are not available in *NCCGUIControls* have been placed in *BaseObjects* instead of here.

SharedCode is kind of the basement of the CAF-É framework. It (figuratively speaking) contains lots of the pipes and wires that make CAF-É work. It also has a module called *Utilities* which contains common utility functions used by the CAF-É framework and by CAF-É applications. Another important module housed here is *MyConstants*. *MyConstants* allows the developer to change CAF-É framework variables for running an application in Debug mode or switching between the production and development SQL Server databases.

The *AppNameFormsLibrary* contains the data collection forms and program code unique to each application. As mentioned above, a CAF-É application inherits lots of common code from the CAF-É framework via the common projects. CAF-É does not do any auto-code generation so a programmer knowledgeable about the use of CAF-É, Visual Basic, and SQL programming and relational databases concepts is still needed to build a working application. However, since CAF-É does many of the "lower" level functions using reflection and other abstractions, the developer can spend more time on the design and the form specifications for the investigation. Although CAF-É does most of the rudimentary functions the developer needs, there are occasions when it is necessary to add new things to the base code. To service these features an inheritance and override strategy is used. All forms in an application are inherited from a base equivalent. This base form provides functions for binding controls to data table fields, database querying, and updates etc. Thus, when an enhancement is made to these base forms or in common code all future and existing applications can also utilize them, if needed. Training documents and tips on using CAF-É for new developers have been created and are available from the authors.

CAF-É's *user interface* consists of three windows. The menu window (upper left in the Fig. 12.2 showing the CTMS application) presents a hierarchically

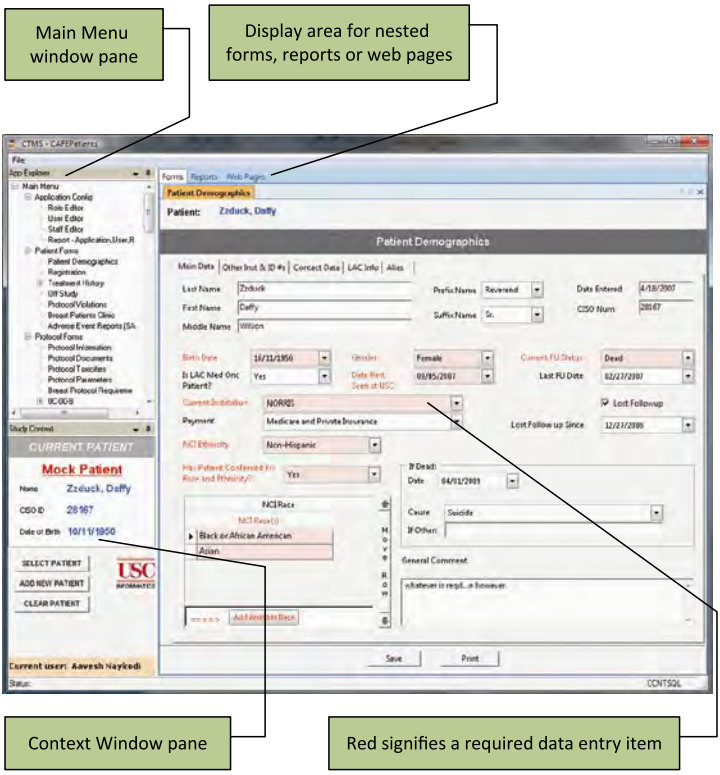


Fig. 12.2 Description of CAF-É user interface

structured set of user tasks. Each task in the menu has a corresponding form, report, or Web page that is displayed in the user's work window (right pane) when a task is chosen by the user. The user accomplishes this selection by a simple click of their mouse. Each form that is opened maintains its own context. This enables users to look at similar data for two or three patients and/or protocols at the same time, or if necessary dissimilar forms can also be simultaneously displayed. The work window is a nested tab page so forms, reports, and Web pages can be left open and the user can move between these various nested tab pages as needed. The form context window (pane on bottom left) is intended to display the patient/protocol displayed in the currently active work window and also which patient/protocol will be retrieved for any new form the user chooses to select from the menu window. The user can customize the look and feel of the application in many ways. The menu and form context windows can be undocked or minimized, or resized via scroll bars thus freeing up more screen space as needed. If not needed, the menu and/or context windows can be excluded from an application via a database parameter. Multiple forms, reports, and Web pages can be opened simultaneously, re-sized and/or closed. They can also be arranged horizontally or vertically at the user's discretion.

12.2.3 Other Dependencies

In addition to the two control libraries mentioned below, CAF-É has several other "external" dependencies for it to function. The first is a stored procedure that is utilized by the *DataAccess* layer and the second consists of several Web services used for encryption and passing of database connection strings and e-mail messages. The implementation guide available from the authors has VB code examples for these Web services but any language code be used as long as the service signature is identical to the one expected by CAF-É.

The *Data Access* layer utilizes a modified SQL System Stored Procedure to determine the data types for all the data columns in a data table. The name of the original stored procedure is *SP_Help* and the modified version is *SP_Help_JTC*. The SQL user ID associated with all application connection strings for an application needs to given execute rights for *SP_Help_JTC*. *SP_Help_JTC* should be created in the SQL Master database.

A *Send Mail* Web service is used within CAF-É to forward alerts to application users or to notify the development and/or support team of error messages or warnings the users' might encounter. The e-mail addresses used are stored in the *App* database in the Applications table for each application. The valid e-mail address of the person to monitor all error e-mail messages is entered here. This Web service requires a "validation" phrase that is needed for the service to function. If an incorrect validation phrase is used in the call to the Web service a message is not sent. For the message service to work the environment hosting CAF-É must have an SMTP mail server available.

To allow secure processing of database connection string user names and passwords, a public/private key encryption strategy is utilized. A public and private pair encryption key is generated. The public key is used in a Web method to fetch the encrypted username and password for the connection string. As above a "validation" phrase is used by this Web service to insure only appropriate users can communicate with it. These validation phrases are environment specific and usernames and passwords are database specific. Independently, another set of user specific credentials are maintained for each application, which is associated with the roles a user may have for an application.

In addition to using the *NCCGUIControls* project, CAF-É uses two Third Party purchased control "toolsets." The *AppExplorer* code makes extensive use of the Crownwood *DotNETMAGIC* library (Dot Net Magic, 2003) for the creation and manipulation of docking windows in the GUI. Most of the forms in the Application-specific Forms Library utilize one or more *Infragistics* windows controls (Net Advantage for .Net 2007, 2006). Common *Infragistics* controls used are – *UltraGrid, UltraCombo, and UltraDateTimeEditor*.

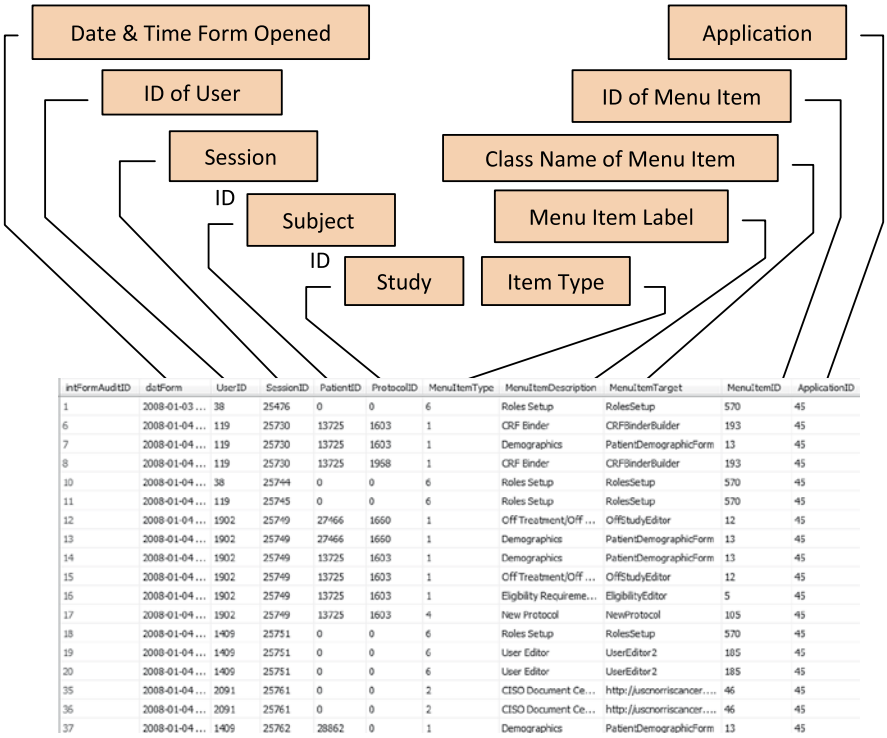
12.2.4 Unique Features of CAF-É

CAF-É includes a number of features that are provided in the base objects that assist in complying with regulatory requirements like 21 CFR Part 11 and HIPAA (21 CRF Part 11, 1997, 2003). CAF-É has three distinct types of audit trails that can be used: user login/logout, form access, and item level detail. These can be each turned on/off as needed by simple parameter changes in the database as mentioned previously.

Figure 12.3 shows the data items captured for each form a user accesses. This includes a unique ID, the date and time the form opened, the user's session ID, the subject/patient, and protocol/study, the type of menu item (form, report, web page, etc.), the description displayed for the menu item, the class name of the menu item, the unique ID of the menu item, and the application's ID. A similar database table records the following items each time a user logs in and successfully logs out: a unique ID, the date and time of the login, the user's unique ID, the IP address of the user's workstation, the date and time of successful logout, the application's unique ID, and several messages or comments relating to the type of access or error encountered by the user.

A third data table records all insert, update, and deletion to the individual data items on all forms. The items recorded include: a unique ID, the date and time of the action, the user ID, the session ID, the database table name, column name and value entered, and several other items relating to the type of action (insert, update, or deletion). A base form feature, Form History, utilizes the above information to present the user a pop-up that displays the entire audit trail for all items on a form – see Fig. 12.4.

Another unique feature is the "Status Review" feature that allows for a data entry, data review, approval, and form data locking cycle to be used. For example, after a data manager has completed data entry, they can set the status to "Needs review,"



intFormAuditID	datForm	UserID	SessionID	PatientID	ProtocolID	MenuItemType	MenuItemDescription	MenuItemTarget	MenuItemID	ApplicationID
1	2008-01-03 ...	38	25476	0	0	6	Roles Setup	RolesSetup	570	45
6	2008-01-04 ...	119	25730	13725	1603	1	CRF Binder	CRFBinderBuilder	193	45
7	2008-01-04 ...	119	25730	13725	1603	1	Demographics	PatientDemographicForm	13	45
8	2008-01-04 ...	119	25730	13725	1968	1	CRF Binder	CRFBinderBuilder	193	45
10	2008-01-04 ...	38	25744	0	0	6	Roles Setup	RolesSetup	570	45
11	2008-01-04 ...	119	25745	0	0	6	Roles Setup	RolesSetup	570	45
12	2008-01-04 ...	1902	25749	27466	1660	1	Off Treatment/Off ...	OffStudyEditor	12	45
13	2008-01-04 ...	1902	25749	27466	1660	1	Demographics	PatientDemographicForm	13	45
14	2008-01-04 ...	1902	25749	13725	1603	1	Demographics	PatientDemographicForm	13	45
15	2008-01-04 ...	1902	25749	13725	1603	1	Off Treatment/Off ...	OffStudyEditor	12	45
16	2008-01-04 ...	1902	25749	13725	1603	1	Eligibility Requireme...	EligibilityEditor	5	45
17	2008-01-04 ...	1902	25749	13725	1603	4	New Protocol	NewProtocol	105	45
18	2008-01-04 ...	1409	25751	0	0	6	Roles Setup	RolesSetup	570	45
19	2008-01-04 ...	1409	25751	0	0	6	User Editor	UserEditor2	185	45
20	2008-01-04 ...	1409	25751	0	0	6	User Editor	UserEditor2	185	45
35	2008-01-04 ...	2091	25761	0	0	2	CISO Document Ce...	http://uscnorriscancer....	46	45
36	2008-01-04 ...	2091	25761	0	0	2	CISO Document Ce...	http://uscnorriscancer....	46	45
37	2008-01-04 ...	1409	25762	28862	0	1	Demographics	PatientDemographicForm	13	45

Fig. 12.3 Description of user login audit data table

their supervisor or Q/A person, can then review the data and either accept it and/or lock it or set the status of items needing correction, at which point the data manager can alter as needed. Figure 12.5 is an example of this. In addition, to aid in the communication flow for these status changes, a Send Mail Web service is included to communicate the status changes between the data manager and Q/A person. This Web service is also used as a mechanism to report bugs or errors to development staff and to notify external systems when patients go on or off study; for example, for research billing compliance.

In most scientific investigations, those entering the data onto the computer forms and those performing the data analysis are very familiar with the screen displays of the forms but not the database tables and columns where the data is stored. This makes selection and retrieval of individual data item for analysis problematic. In the past, we have printed the screens and annotated them manually to assist in this process. This is obviously a very labor intensive and repetitive task as changes are made to the collection forms. We have leveraged a .Net feature called “reflection” to automate and simplify this process. Reflection is a feature in .Net, which enables one to retrieve information about an object at runtime. Since nearly all forms, controls, etc. are based on objects in .Net and all these objects are contained in class files, one can use reflection to get information about these objects.

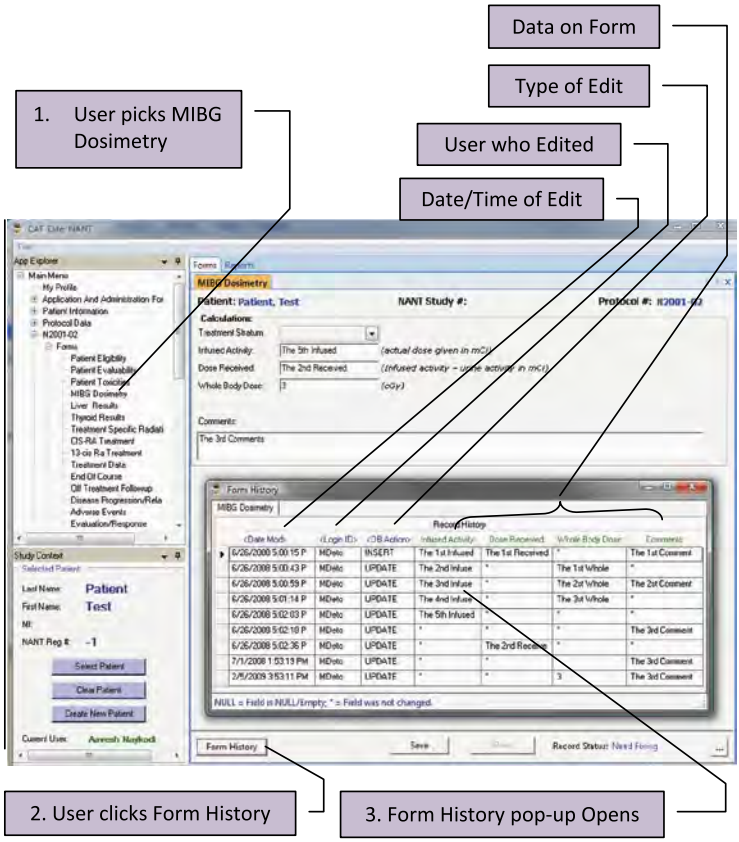


Fig. 12.4 Display of transaction audit history data

You can fetch a variety of information depending on the specific objects; for example, control names, database bindings, method names, and the constructors of objects. By using reflection, we programmatically generate the documentation needed for data retrieval and analysis via a base object called the Form Binding Viewer. Figure 12.6 depicts this in use; another feature not mentioned in the figure is that the columns in the data grid (5b) can be sorted by clicking on the column headers.

12.3 Future Development and Enhancement Plans

EDC in CAFÉ is currently implemented using Microsoft Windows forms, which are only supported on Microsoft Windows-based computers. This platform dependence is viewed as one of the main limitations of CAFÉ. In addition, the requirement

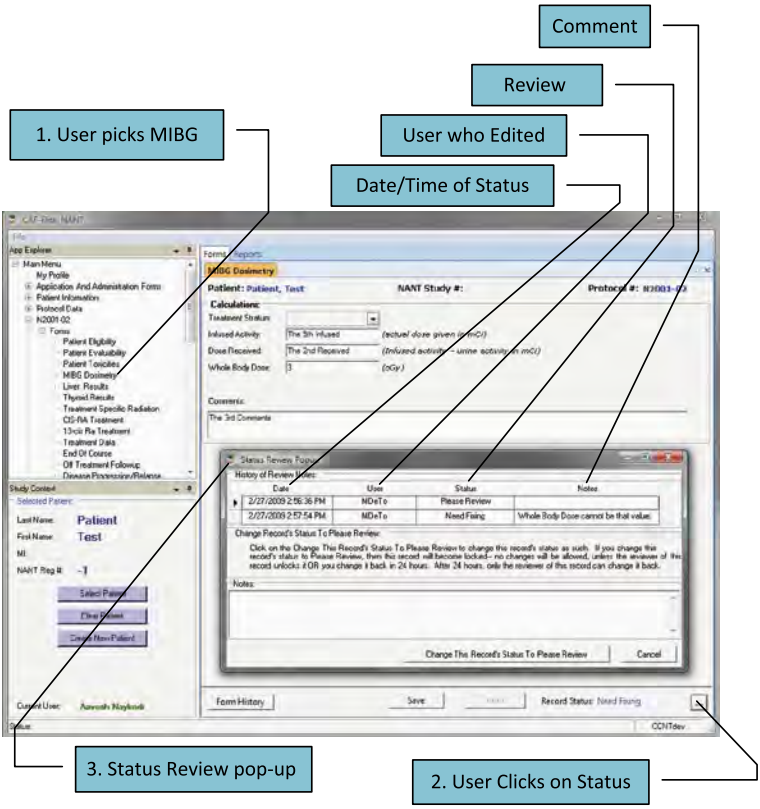


Fig. 12.5 Display of the form status review process

to install the application on the client machine(s) requires permission from or intervention by information technology departments at outside institutions. This complicates its use for management of national or international studies. Although we currently allow external collaborations via remote terminal access using remote desktop protocol we believe the best alternative is to add a web browser-based front-end to CAFÉ in the future.

In conjunction with the Web-based forms, we believe a formalized common data item repository and electronic screen generation component is essential to making CAFÉ more widely acceptable.

Finally, we are going to add a task queuing capability that will allow us to provide a “dashboard” type of form to allow users to get an overview of work requests needing attention for study/protocol or patient management.

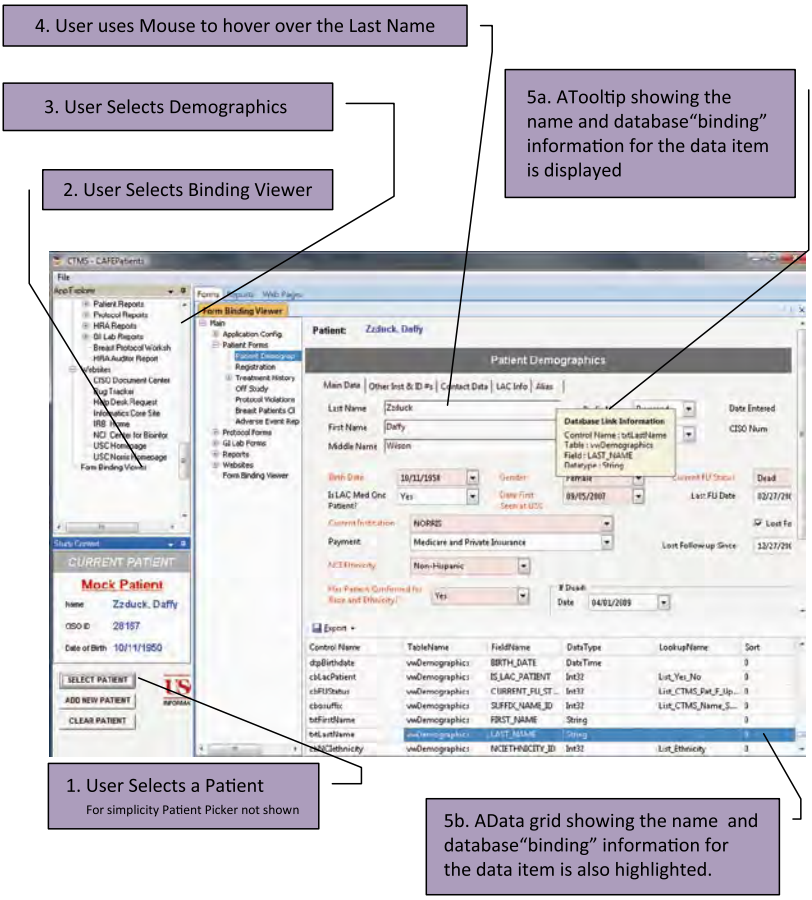


Fig. 12.6 Display of the form binding viewer data

References

American Medical Association, Current Procedural Terminology 4th Edition (2009), AMA, Chicago, IL

21 CFR, Part 11-The Final Rule, The Federal Register (March 20, 1997). United States Office of the Federal Register. http://www.fda.gov/ora/compliance_ref/part11/FRs/background/pt11finr.pdf, accessed 29 November 2007

21 CFR, Part 11-Industry Guidance, FDA (2003). http://www.fda.gov/Cder/guidance/5667fnl.pdf, accessed 29 November 2007

Cancer Therapy Evaluation Program Common Toxicity Criteria, Version 2.0 DCTD, NCI, NIH, DHHS (March 1998). http://ctep.cancer.gov/protocolDevelopment/electronic_applications/docs/ctcv20_4-30-992.pdf, accessed 30 October 2001

Cancer Therapy Evaluation Program, Definitions for CTCAE Version 3.0, DCTD, NCI, NIH, DHHS (June 10, 2003). http://ctep.cancer.gov/protocolDevelopment/electronic_applications/docs/ctcaev3.pdf, accessed 17 May 2004

Dot Net Magic, Crownwood Software Ltd, Australia (2003). http://www.dotnetmagic.com/index.html, accessed 9 February 2004

International Classification of Diseases, Ninth Revision, Clinical Modification (2002). Health Care Financing Administration, Washington, DC

Net Advantage for .Net 2007, Infragistics, (2006). http://www.infragistics.com/, accessed 10 December 2007

Chapter 13
Shared Resource Management

Matt Stine, Vicki Beal, Nilesh Dosooye, Yingliang Du, Rama Gundapaneni, Andrew Pappas, Srinivas Raghavan, Sundeep Shakya, Roshan Shrestha, Momodou Sanyang, and Clayton Naeve

Abstract Shared Resource Management (SRM) is a laboratory management system designed to support shared resource facility (or core laboratory) activities. It was originally designed to support the laboratories in the Hartwell Center for Bioinformatics and Biotechnology at St. Jude Children's Research Hospital, which includes High Throughput DNA Sequencing and Genotyping, Genome (Next Generation) Sequencing, Macromolecular Synthesis, Functional Genomics (spotted microarray), Affymetrix (commercial microarray), Molecular Interaction Analysis, and Proteomics facilities. In addition, it supports St. Jude's Protein Production Facility and Cell and Tissue Imaging Center. SRM was designed to be sufficiently modular and scalable to support other laboratory activities as needed. It could conceivably support all facilities on a campus or at a research organization and provide a single portal for investigators to access these resources, retrieve data, receive invoices for services, and generate reports.

In July 2007, SRM was released to the open-source community as STJUDE-SRM and is distributed under the GNU Lesser General Public License v3.0. A partial version of the system (minus invoicing and reporting capabilities) targeted at the MySQL database platform is available for download from http://stjude-srm.sourceforge.net. A complete platform targeted at Oracle 9i/10g as well as Postgres Plus® Advanced Server is forthcoming.

Basic installation and usage documentation is available via the STJUDE-SRM wiki at http://stjude-srm.wiki.sourceforge.net/. Practical experience with the JBoss Application Server (http://www.jboss.org), MySQL Database (http://www.mysql.com), and basic Unix/Linux administration, as well as a working knowledge of Java 2™ Enterprise Edition applications and the Apache Ant build tool (http://ant.apache.org) are required to install and administer SRM.

C. Naeve (✉)
St. Jude Children's Research Hospital, 262 Danny Thomas Place, Memphis, TN 38105, USA
e-mail: Clayton.Naeve@STJUDE.ORG

M.F. Ochs et al. (eds.), *Biomedical Informatics for Cancer Research*,
DOI 10.1007/978-1-4419-5714-6_13, © Springer Science+Business Media, LLC 2010

13.1 Description of Tool

13.1.1 Overview

SRM is a laboratory management system designed to support shared resource facility (or core laboratory) activities. It currently supports multiple laboratories at St. Jude Children's Research Hospital, offering the following services:

- High Throughput DNA Sequencing
- Genome (Next Generation) Sequencing
  - Illumina Platform

    ChIP-Seq
    Digital Gene Expression
    miRNA Analysis
    Resequencing
    Mate-Paired Sequencing
    De Novo Sequencing
    mRNA-Seq
  - Roche FLX Platform

    Amplicon Analysis
    ChIP-Seq
    De Novo Sequencing
    Digital Gene Expression
    Resequencing
- Genotyping
  - Minisatellite and Microsatellite Analysis
  - Insertion/Deletion Screening
  - Mouse Genotyping
  - SNP Screening
  - Multiplex Ligation-Dependent Probe Amplification (MLPA)
  - DNase Footprinting
  - PCR-Restriction Fragment Length Polymorphism (RFLP)
  - Surveyor mutation assay
  - Promega Powerplex system
- Macromolecular Synthesis
  - DNA (Oligo) Synthesis
  - Standard Peptide Synthesis
  - 96-Well Pin (Mimotope) Synthesis
  - HPLC Quality Assurance/Purification
  - Post-Synthesis Modifications

- Functional Genomics (Custom/Spotted Microarray Technology)
  - Expression Analysis
  - Comparative Genomic Hybridization (aCGH) Analysis
  - Clone Retrieval
- Proteomics
  - SDS-polyacrylamide Gel Electrophoresis
  - Large Format 2-D Gel Electrophoresis
  - Small Format 2-D Gel Electrophoresis
  - Electroblotting
  - Protein Identification – Tandem MS
  - Protein Peptide Chromatography
  - Protein Isoelectric Focusing
  - Protein Identification – LC/Tandem MS
  - Isotope-Coded Affinity Tag Analysis
  - Protein Mass Measurement
- Molecular Interaction Analysis
  - Surface Plasmon Resonance (Biacore) Analysis
  - Analytical Ultracentrifugation
  - Multiangle Light Scattering
- Clinical Applications Core Technology (Affymetrix)
  - RNA Gene Expression
  - Whole Genome Mapping and SNP Analysis
  - Genome Tiling, Exon Analysis, and Chromatin Immunoprecipitation (ChIP)
- Protein Production Facility
  - Bacterial Expression
  - Baculovirus Expression
  - Purification
  - Crystallization Trials
- Cell and Tissue Imaging Center
  - TEM Imaging
  - Negative Staining
  - 2D Analysis
  - SEM Services
  - Confocal Microscopy
  - Microinjection
  - Multiphoton Microscopy

This diversity of shared resources allowed the development team to take a broad look at shared resource management (SRM) activities and analyze the common

functions required to manage such facilities. SRM currently breaks these functions down into nine main subsystems:

- Online Ordering
- Online Scheduling
- Online Order/Sample Tracking
- Sample and Workflow Management Tracking (LIMS)
- Data Processing, Archival and Retrieval
- Messaging
- Client Management
- Billing
- Reporting

13.1.2 Online Ordering

The online ordering subsystem allows clients to place online orders for shared resource/core facility services (Fig. 13.1). Orders are ultimately linked to a principal investigator's group, or "PI Group," which is the first bit of information that must be selected by the client (clients can belong to multiple groups). This information allows SRM to link any resulting data produced by the service to the group, allowing any member of that group access to the data, which is deposited in a flat-file archive having the group's name as the parent folder. Online order forms are customized to fit each individual service's needs, with fields common to all services present throughout.

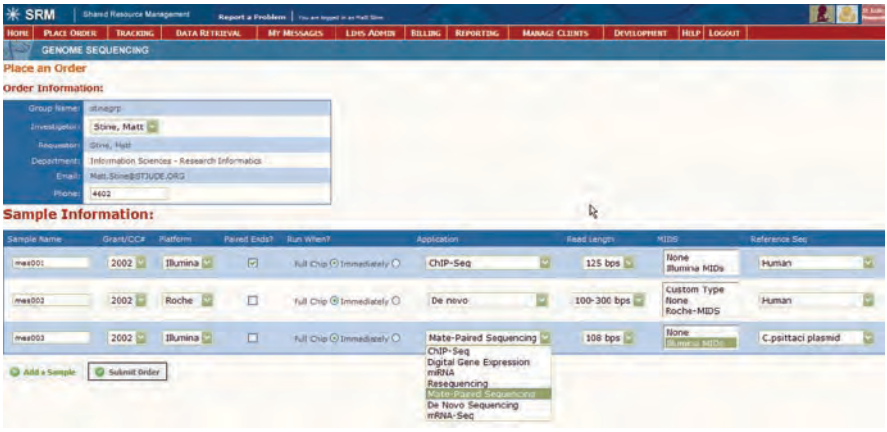


Fig. 13.1 Online ordering screen for St. Jude's Genome Sequencing Facility

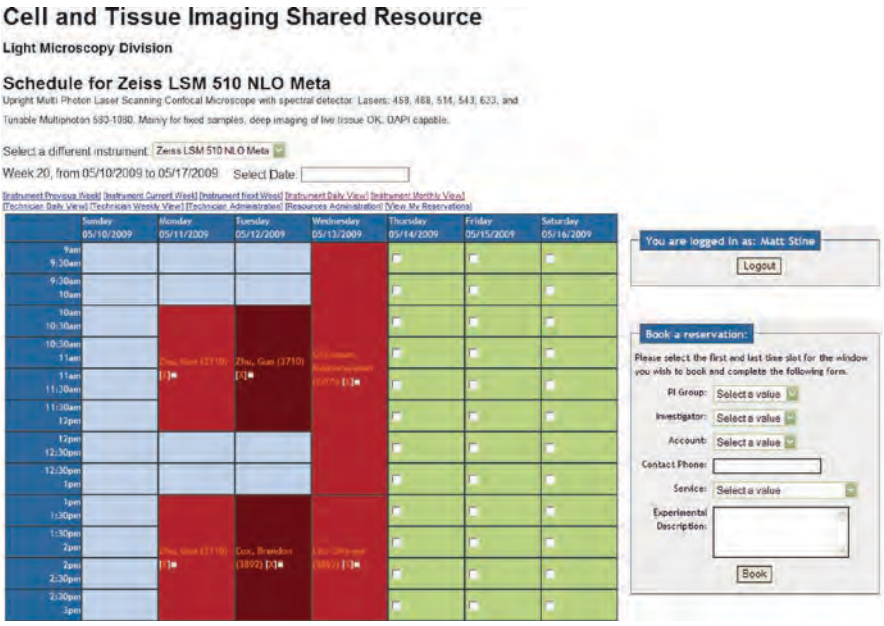


Fig. 13.2 Online scheduling screen for St. Jude's cell and tissue imaging facility

13.1.3 Online Scheduling

The online scheduling (Fig. 13.2) subsystem allows shared resources to schedule client appointments (through the ordering subsystem) for shared instrumentation usage. Instrument schedules are viewable as daily, weekly, or monthly calendars. Facility staff can create calendars for new instruments as necessary. The subsystem features the ability to request unassisted (for "power users") or technician-assisted usage of the instrument. Technicians are assigned to bookings via a ranking system, whereby the highest-ranked available technician is chosen. Staff can flag blocks of time as unavailable for technicians (such as when a technician is out for vacation) on an individual or recurring basis. Usage of the instrumentation is billed hourly, with configurable rates for each instrument as well as for unassisted vs. technician-assisted usage.

13.1.4 Online Order/Sample Tracking

The online order/sample tracking subsystem allows clients to track the progress of online orders they have placed for shared resource/core facility services. The Tracking subsystem is grouped into two portions, the first being a "My Pending Orders" view, which is a listing of all pending orders the logged-in client has placed in SRM, and the second being a search view, which allows clients to search for any pending orders placed by his/her PI Group.

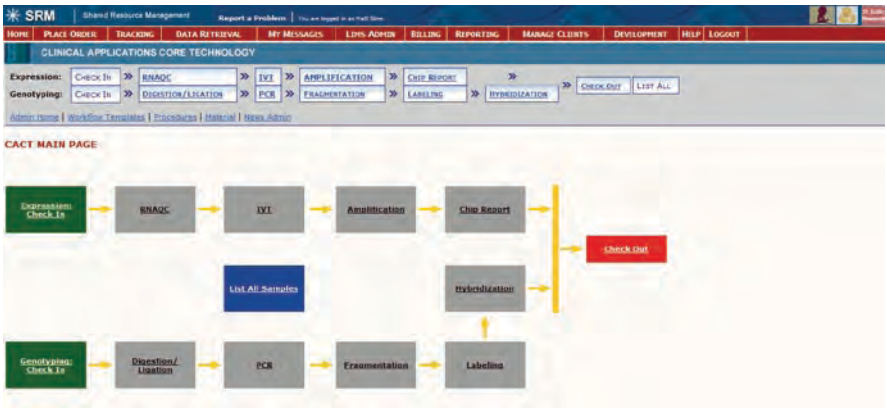


Fig. 13.3 LIMS screen for St. Jude's Clinical Applications Core Technology (Affymetrix™) Facility

13.1.5 Sample and Workflow Management Tracking (Laboratory Information Systems, LIMS)

The LIMS subsystem (Fig. 13.3) allows shared resource staff to track their activities as they process samples through their workflow. The LIMS system is broken down into individual pages representing the typical steps in the workflow. On each of these pages, staff can annotate samples currently at that step with various data items, as well as archive large data files for later retrieval (e.g., DNA sequencing data, quality control information, Affymetrix/Custom microarray data, etc.). Most LIMS systems also include sample check-in/check-out; a "List All Samples" page to allow a global perspective of the lab's queue, various procedure/material management pages; and an Edit Workflow/Workflow Template facility, where by staff can edit the typical workflows available to the lab, or make changes to the workflow for a particular sample.

13.1.6 Data Processing, Archival, and Retrieval

The data processing/archival subsystem (Fig. 13.4) allows developers to construct various data processing/archiving pipelines to manage the vast amounts of data generated by many shared resources. It allows for synchronous or asynchronous pipeline execution.

The data retrieval subsystem allows clients to retrieve data resulting from the experiments performed via various shared resources. Clients can search for samples based on various criteria, and then choose to look for data for all or a subset of the samples returned from the search. The subsystem will provide a list of all data archived by SRM for those samples, and clients can download all or a subset of those files to their local PC/Mac, or they may also copy them to an institutional

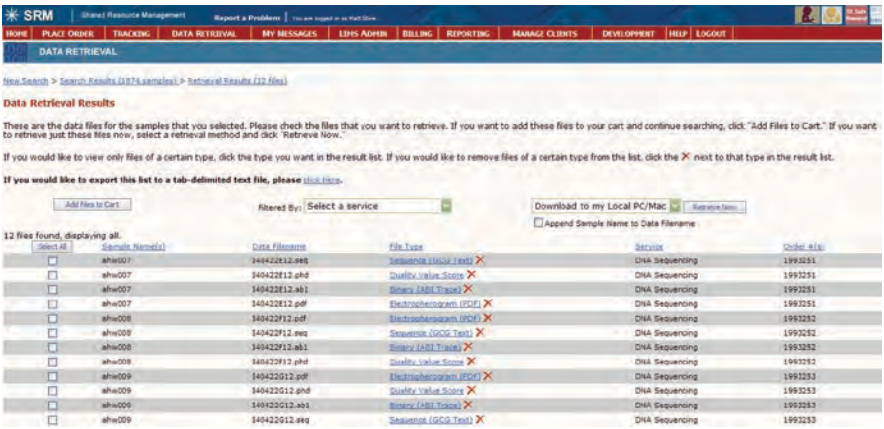


Fig. 13.4 Data retrieval screen

shared file system. Data Retrieval also includes a "shopping cart" style function, whereby clients may add files to their cart and then conduct additional searches. When finished they can download the entire contents of their cart.

13.1.7 Messaging

The messaging subsystem sends email to clients and shared resource staff when various events occur within the system. It also maintains an internal message queue for each user, so that users may view their messages in the event of an email system failure.

13.1.8 Client Management

The STJUDE-SRM Client Management subsystem allows administrators to update the information maintained for the principal investigators (PIs) served by the system, assign delegates for PIs, and update account information for PIs.

13.1.9 Billing

The billing subsystem (Fig. 13.5) allows shared resources to bill clients for services rendered. Bills can be generated according to a standard or custom billing cycle.

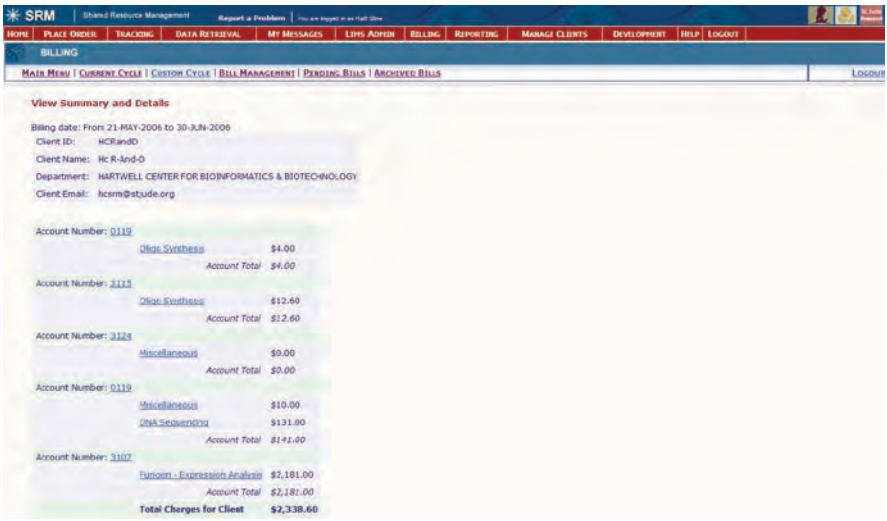


Fig. 13.5 Billing summary screen

Clients are notified via email of pending bills, and they (or their delegates) may login to SRM and approve the bills. If necessary, they may make changes to the distribution of charges amongst various accounts and grants. Approved charges may then be automatically sent to the institutional accounting system for balance updates.

13.1.10 Reporting

The reporting subsystem provides a valuable global view of the data contained in the system. It is broken down into two primary components. The first, custom reports, allows clients, administrators, and shared resource staff to perform ad-hoc queries for samples by various criteria. They may then drill down into the individual histories of samples returned from the search. The second component, prepared reports, allows quick-and-easy generation of various reports required by various users of SRM on a regular basis. One of the key prepared reports is the Cancer Center Support Grant report that allows an administrator to generate the required usage information for CCSG grant renewals.

13.2 Technical Details

The primary implementation technology for SRM is Java™. SRM is currently compiled using the Java Platform, Standard Edition 5.0 compiler. The primary supporting technology stack for SRM is the Java™ 2 Platform, Enterprise Edition 1.3

specification, including Enterprise JavaBeans 2.0, JavaServer Pages 1.2, and Java Servlet 2.3.

SRM is divided into multiple deployment units. The original platform was deployed as a monolithic enterprise archive (EAR) file containing both the business and data persistence services as well as the Web application archive (WAR) module. This EAR deployment supports all services provided by the original Hartwell Center laboratories.

As newer facilities were brought online, a shift to a more modular deployment scheme was made. The business and data persistence services are still provided by the EAR deployment, with the Web user interface (UI) for each new facility deployed in a separate WAR. This move was made for two reasons:

- As the core technologies supporting SRM continue to age and newer technologies come available, isolating the use of these newer technologies to the UI supporting a new facility reduced the risks and other barriers to introduction of these technologies. Thus, the introduction of the Spring Framework, Java Server Faces (JSF), and WebWork were enabled.
- Most coding specific to a new facility has historically been found within the UI layer of the system, so this enabled the isolation of defects introduced by new facilities (flesh out and rework).

The base environment required for an SRM deployment is the following:

- Java™ Platform, Standard Edition 5.0
- Any server OS platform supporting Java™ 5 (Windows/Unix/Linux/Mac OS X)
- JBoss Application Server 4.2.x
- MySQL 5.x
- Apache Ant 1.7.x

As was stated in the abstract, the currently available open-source version of SRM is incomplete and will run on the MySQL 5.x platform. St. Jude's current production deployment of SRM runs on Oracle 10g. A full migration of the database schema to MySQL was attempted and did not succeed for various technical/performance reasons. We have since completed a migration to Postgres Plus® Advanced Server, available from EnterpriseDB Corporation (http://www.enterprisedb.com). We will soon release a version of SRM that will run on either Oracle 10g or Postgres Plus® Advanced Server.

SRM must be built from its source for any new deployment. Many configuration options are compiled into the deployed code, so anyone installing SRM will need to be able to run the Ant build.

The SRM development team maintains a Google Group at http://groups.google.com/group/stjude-srm. So far this list has generated very little traffic, but SRM developers are always available via this channel to assist those looking to deploy the platform. This group is currently invite only, and potential members can request an invitation by providing the name of their institution and their interest in joining the group.

13.3 Examples of Usage

Usage of SRM is best illustrated by walking through the lifecycle of an order:

1. An investigator, desiring to sequence a 96-well plate of DNA templates, logs in to SRM. He clicks on the place order link, selects his group from the list provided, selects DNA sequencing from the service list, indicates that the order is grant funded, and submits the form. On the next page he selects plate order, the grant to which he will charge the work, and the number of plates he is submitting. On the next page he browses his computer to locate an Excel™ workbook containing his sample information and uploads it. On the final ordering screen he is shown the plate map generated from his workbook and clicks OK.
2. The investigator walks his 96-well plate down to the DNA Sequencing facility. A staff member logs in to SRM, locates the plate in the LIMS system, and checks in the plate.
3. A staff member uses the LIMS system to assign the plate to a sequencer and then generates a file used by the data collection software to analyze the plate. The staff member then proceeds with the sequencing laboratory work.
4. After the sequencing run is complete, the staff member imports a report file containing length-of-read (LOR) information into SRM. He then places the generated data into a shared network drive known by SRM and clicks "Archive." SRM picks up the data and moves it to a flat-file archive organized by investigator. The staff member then "checks out" the plate and the investigator receives an email indicating that his data is available to download.
5. The investigator clicks on a link in the notification email, which carries him to a page containing a list of files available for download. He selects the files he wishes to download and clicks "Retrieve Now." The files are transferred to his PC via FTP.

Another example of SRM usage is the billing workflow:

1. A facility's billing administrator logs into SRM and clicks "Billing." Once in billing, she clicks "Current cycle," and is shown a list of outstanding charges by investigator for the current billing cycle. She selects the investigators she wishes to bill and clicks "Bill." Emails containing the bills are sent to each investigator.
2. The investigator receives a billing notification and clicks on the link it contains. He is presented by a list of charges and the grants to which they are charged. When satisfied, he clicks "Approve."
3. The billing administrator accesses the "Bill management" screen to see the status of all outstanding bills. From this screen she can see the current status (pending vs. approved), send additional notifications, as well as see the number of times each investigator has been notified. She can also change the charged amount and the grants to which bills have been charged.
4. When ready, the billing administrator generates a spreadsheet containing the approved charges and submits it to the financial services department.

13.4 Future Development Plans

An original goal for SRM was for it to be sufficiently modular and scalable to support additional laboratories and potentially support all core facilities within an institution, providing a single portal for investigators to requisition services, retrieve data and invoices for services, and generate reports. The original implementation of SRM allowed modules for new facilities and/or services to be deployed within 3–6 months, assuming 2–3 developers concurrently dedicated to developing modules for a new facility. Little did we know that the rate of desired SRM adoption and introduction of new facilities to St. Jude would far outstrip our ability to deliver the required modules.

Clearly a different approach was necessary. Work is now underway to deliver SRM 2.0. SRM 2.0 will largely be driven by domain-specific metadata specified at runtime, allowing new facilities and/or services to be deployed (and existing facilities and/or services to be modified) without writing any additional code modules. According to our estimates, this will allow us to deliver at least 75–80% of the required functionality to support any one facility.

The remaining functionality will be delivered by utilizing an innovative plug-in system, whereby various predefined extension points throughout the system can be enhanced by developing domain specific plug-in modules that will be deployed independently of the core system. We believe these two fundamental concepts will allow us to shorten deployment time for new facilities and/or services to 3–6 weeks or possibly less. It is our vision that SRM 2.0 will represent the next generation of core facility management systems and be the best available software in the world for managing core facilities.

Chapter 14
The caBIG® Life Sciences Distribution

Juli Klemm, Anand Basu, Ian Fore, Aris Floratos, and George Komatsoulis

Abstract caBIG® is a virtual network of organizations developing and adopting interoperable databases and analytical tools to facilitate translational cancer research (von Eschenbach and Buetow 2007). It is an open-source, open-access program, and all the tools and resources are freely available to the research community. The National Cancer Institute is developing resources to assist enterprise-wide adoption of the caBIG® tools. To this end, we have bundled mature software tools together to facilitate easy adoption and installation. The Life Sciences Distribution (LSD) is comprised of tools to support the continuum of translational research: caArray, for the management and annotation of microarray data; caTissue, to support the collection, annotation, and distribution of biospecimens; the Clinical Trials Object Data System, for the sharing of clinical trials information; the National Biomedical Imaging Archive, for annotation, storage, and sharing of in vivo images; cancer Genome Wide Association Studies, for publishing and mining data from GWAS studies; and geWorkbench, supporting the integrated analysis and annotation of expression and sequence data. All the LSD tools are connected to caGrid (Saltz et al. 2006), which makes it possible for the databases at multiple institutions to be interconnected to support data sharing and integration.

More information on the LSD suite of products, including installation packages, user and installation guides, and links to exemplar installations can be found at http://ncicb.nci.nih.gov/NCICB/tools/lsd.

J. Klemm (✉)
Center for Biomedical Informatics and Information Technology, National Cancer Institute, 2115 East Jefferson Street, Suite #6000, Rockville, MD 20852 USA
e-mail: klemmj@mail.nih.gov

M.F. Ochs et al. (eds.), *Biomedical Informatics for Cancer Research*,

DOI 10.1007/978-1-4419-5714-6_14, © Springer Science+Business Media, LLC 2010

14.1 National Biomedical Imaging Archive

14.1.1 NBIA Overview

The National Biomedical Imaging Archive (NBIA) is a software package that allows for archiving and sharing of medical imaging data in a secure and federated fashion. The need for such an archive originated from NCI's goals to support the development and validation of analytical software for lesion detection and classification, accelerated diagnostic imaging decision making, and quantitative imaging assessment of drug response.

14.1.2 NBIA User Interface

NBIA is a Web-based application that allows researchers to query and retrieve imaging and any available related annotation data using multiple Digital Imaging and Communications in Medicine (DICOM) standard attributes (http://medical.nema.org/). The user is also able to browse JPEG versions of the stored images as a preview function before downloading an entire set. Images are loosely organized into "collections," a dynamic grouping that allows images to be grouped into relevant sets based on a clinical trial, a study, or any other common attribute across the set.

14.1.3 NBIA Inputs and Outputs

NBIA currently stores DICOM files submitted from any Picture Archiving and Communication Systems (PACS) using an open-source software called the Clinical Trials Processor (CTP, available at http://mircwiki.rsna.org/index.php?title=CTP-The_RSNA_Clinical_Trial_Processor). The CTP software allows submitters to anonymize and remotely submit DICOM files to configured instances of NBIA. The images are stored in a file system with pointers to their relative locations stored in a MySQL backend database. The DICOM tags are parsed and their values stored in the backend database.

Users are able to query and download the files supported by the archive to their local workstations. The query and retrieve functionality is available via an application programming interface (API) as well, allowing programmers to write queries to access any caGrid-enabled instance of an NBIA server.

14.1.4 NBIA Future Development and Enhancements

NBIA development continues to focus on supporting additional imaging modalities including pathology and optical modalities. Multiple service endpoints are also being developed for ease of access to the underlying data in a service-oriented fashion.

There are also plans underway to integrate with an image annotation service using the Annotation and Imaging Markup (AIM) standard (https://cabig.nci.nih.gov/tools/AIM). Multiple improvements are being made to the user interface to make it more intuitive and to allow for a more flexible query interface.

14.2 caTissue Suite

14.2.1 caTissue Suite Overview

Many biobanks manage their inventory in ad hoc tools, such as spreadsheets or paper-based systems. The caTissue project was initiated to develop an enterprise quality solution for managing biorepositories. caTissue Suite is a tissue bank repository tool for biospecimen inventory, tracking, and basic annotation. This tool permits repository staff to track the collection, storage, quality assurance, and distribution of specimens as well as the derivation and aliquotting of new specimens from an existing ones (e.g., for DNA analysis). It also allows end-user scientists to find and request specimens that may then be used in molecular correlative studies.

The latest version of caTissue Suite (version 1.1) is an integrated suite adding in clinical annotation of specimens and a tool (caTIES) for processing free text surgical pathology reports into a structured database that can be searched according to specific criteria such as diagnosis, surgical procedure, and anatomical site.

14.2.2 caTissue Suite User Interface

caTissue Suite provides customizable, role-based functions through its graphical user interface. Through the Administrator features, the sets of specimens required from each individual on a study can be defined. As the study proceeds, the accrual of specimens can be monitored. In addition, the Administrator also registers new users, containers (e.g., freezers, liquid nitrogen tanks), consent tiers, and distribution protocols into the system. A Supervisor adds participants to the system and registers them to collection protocols. Technicians add specimens to the system and capture participant responses for consent tiers. Users with the Scientist role may use the advanced query wizard to create and save complex, predefined, or parameterized searches to identify and order specimens of interest.

14.2.3 caTissue Suite Inputs and Outputs

In addition to the graphical user interface, an important part of caTissue Suite is its API. While principally intended for use by programmers, its value lies in enabling integration with other systems. In practice, the API has enabled many institutions to

load data into caTissue Suite from existing systems. A range of third party solutions have been developed using the API that allow loading of data from simple spreadsheets, XML files, or complex relational systems. The API has also been used to integrate caTissue with other systems at a number of institutions.

14.2.4 Future Development

Adaptations will be added to coordinate with services that enable full integration of research data by use of identifiers for specimens that are unique across the grid. This will also support integration with systems that manage the research results obtained on specimens, for example, caArray. Integration with Clinical Trials Management Systems (CTMSs), such as those in the companion caBIG® CTMS bundle (see Chap. 11), are planned and will enable integration of the workflows on such trials.

14.3 caArray Overview

caArray is a software application which supports the management and sharing of microarray data, including gene expression, SNP, and copy number data. The intended end users of caArray include lab scientists, principle investigators, and biostatisticians. caArray supports the annotation of microarray experiments in accordance with Minimal Information About A Microarray Experiment (MIAME) guidelines (Brazma et al. 2001) and provides a graphical user interface for the capture and display of annotations. The use of controlled vocabularies to describe the experimental details and biological samples is supported and encouraged through the caArray application; in particular, a number of data entry fields are limited to terms from the Microarray Gene Expression Data (MGED) Ontology (Whetzel et al. 2006).

A key feature of caArray is security management. When data is loaded to caArray, it is private to the data owner unless is it shared with a collaboration group or made public. Users create and manage their own collaboration groups and can give members of the group read-only or read/write access to the whole experiment or just selected samples in the experiment.

14.3.1 caArray User Interface

End users interact with caArray through the Web-based application user interface. A user who is not logged in is able to browse, search for, and download public data. Users can search for experiments of interest based on desired attributes, or they can search for individual samples of interest. From the search results, a user can choose to download the associated microarray data files for individual samples or for the entire experiment.

A logged-in user has the access to upload and annotate microarray data. When first logging in, the user is presented with "My Experiment Workspace" which contains a tab for Work Queue Experiments that are not yet public, as well as a tab for experiments that have been made public. When logged in, users have the option to add data to an existing experiment or create a new experiment, to manage collaboration groups, to load or edit array designs, and to load or edit reusable protocols and vocabulary terms.

14.3.2 caArray Inputs and Outputs

caArray currently parses data from Affymetrix, GenePix, and Illumina formats. Parsed values are available to analysis tools and services through the caArray remote Java API and grid service. In addition, caArray stores the native data files of these formats, as well as from the Agilent and Nimblegen platforms, among other formats. The full set of supported platforms is available in the caArray User Guide.

A popular feature of caArray is support for the import, update, and export of data in MAGE-TAB format (Rayner et al. 2006). MAGE-TAB is a spreadsheet-based format for capturing MIAME-compliant microarray experiment annotations. This easy-to-use format has rapidly become the approach of choice for loading data into caArray, largely due to the familiarity most scientists have with the use of spreadsheets and the readiness with which the format can be produced by other systems.

14.3.3 Future Development and Enhancement

caArray development continues to proceed with an open architecture and supportive documentation to allow for future enhancements. Areas of future planned emphasis include interfacing with additional analysis tools; integration with biospecimen annotation services, such as caTissue Suite, and seamless publishing to other array databases including GEO.

14.4 Clinical Trials Object Data System

14.4.1 CTODS Overview

The Clinical Trials Object Data System (CTODS) has been developed to enable the exchange of deidentified clinical trials data across multiple systems while supporting syntactic and semantic interoperability. As a reference implementation of the

Biomedical Research Integrated Domain Group (BRIDG) Model (Fridsma et al. 2008), CTODS provides a single, unified set of APIs that can access clinical data from multiple data sources.

14.4.2 CTODS User Interface

The CTODS user interface consists of a viewer that allows for querying and retrieving data from the underlying database based on four major categories of data – protocol, enrollment, treatments, and adverse events. The user interface allows for keyword searching using either exact or synonym-based searches for organs using NCI's Enterprise Vocabulary Services (EVS). Data generated from these queries can also be exported into comma separated values (CSV) or XML formats.

14.4.3 CTODS Inputs and Outputs

A separate module called the CTODS Loader is used to load data into the backend database. Data exported from a local CTMS using a specified format (as described in the CTMS Theradex version 3.12 available at https://gforge.nci.nih.gov/svnroot/cactus/ctods/docs/ctodsdataloader/CTMS_312.pdf) can be loaded into the CTODS system using this utility. Sample data files using this format are also available from the location described above. Output from CTODS is in the form of structured reports and the data generated from these reports can also be exported into CSV or XML formats.

14.4.4 Future Development and Enhancement

Future developments of CTODS include additional reports and API enhancements to harmonize it with the current version of the BRIDG model.

14.5 Cancer Genome Wide Association Studies

14.5.1 caGWAS Overview

Cancer Genome-Wide Association Studies (caGWAS) allows researchers to integrate, query, report, and analyze significant associations between genetic variations and disease, drug response, or other clinical outcomes. caGWAS supports the correlation of Single Nucleotide Polymorphism (SNP) genotype data, SNP association findings, population frequency data, and clinical phenotypes and provides tools for search and retrieval of GWAS findings in the context of genes or chromosomal regions of interest.

14.5.2 caGWAS User Interface

caGWAS provides an easy-to-use graphical user interface for performing three types of searches: SNP Association Findings, Population Frequencies, and Subjects Data. A user starts their investigation with caGWAS by selecting a study of interest and the version of that study to interrogate and then indicating which kind of search to perform. For SNP Association or Population Frequency searches, the user may designate a genomic location, one or more genes of interest, or one or more SNPs of interest. Association queries also include *p*-value or whole genome rank cutoff; Population Frequency queries allow specification of the Hardy–Weinberg *p*-value, Minor Allele Frequency, and Completion rate. The output includes these data elements in a sortable table. Subject data may also be searched by population or analysis group and by gender, age, case/control status, and/or family history. In the case of a subject search, returned data includes a subject identifier as well as the gender, age, affection status, family history, and population.

14.5.3 caGWAS Inputs and Outputs

caGWAS was built upon common but specific use cases and as such, caGWAS analysis results must first be precomputed. To load data into caGWAS, the source data must first be converted to the required file format and then formatted data is loaded to a database staging area. Data in the staging area is then transformed and transferred to the tables in the caGWAS warehouse schema. caGWAS also provides the ability to bulk download all data from a specified study.

14.5.4 Future Development and Enhancement

Planned enhancements include support for more scans with additional search capabilities, online analytic and visualization capabilities, and cross-study comparisons.

14.6 geWorkbench

14.6.1 geWorkbench Overview

geWorkbench is a platform for genomic data integration, bringing together analysis and visualization tools for gene expression, sequences, protein structures, pathways, and other biomedical data. It gives scientists access to a number of external caGrid-enabled data sources and algorithmic services, combining these with many built-in tools for analysis and visualization.

Over 50 geWorkbench plugin components are currently available, covering a wide range of genomics domains. Analyses include several popular clustering and classification methods (e.g., hierarchical clustering, self-organizing maps, support vector machines, *K* nearest neighbors, and principal components analysis), differential expression tools, as well as advanced systems biology algorithms for regulatory network reconstruction [ARACNe (Margolin et al. 2006), MINDy (Wang et al. 2009), and MatrixREDUCE (Ward and Bussemaker 2008)]. Sequence support includes BLAST (Altschul et al. 1990), pattern discovery, transcription factor mapping, as well as access to novel pipelines for protein structure prediction and structured-based functional annotation. A wide variety of visualization modules accompany these tools, enabling users to interact and interrogate data and analysis results in sophisticated ways. Further, integration with a number of external annotation sources [caBIO gene and pathway data (Komatsoulis et al. 2008), Gene Ontology (Ashburner et al. 2000), GeneWays (Rzhetsky et al. 2004), and the B-Cell Interactome (Lefebvre et al. 2007)] facilitates the incorporation of biological knowledge in evaluating the plausibility of analysis results. A listing of components as well as detailed project documentation and user tutorials is available at the project Web site (http://www.geworkbench.org).

14.6.2 geWorkbench User Interface

Effective interaction with high-throughput genomic data requires the use of sophisticated, fast, and flexible user interfaces. geWorkbench provides a wide range of advanced visualization tools. Some of these tools were developed de novo while others incorporate and leverage third party community technologies, such as JFreeChart (http://www.jfree.org/jfreechart/) (for displaying graphs), Cytoscape (Shannon et al. 2003) (for displaying molecular interaction networks), JMol (http://www.jmol.org/) (for displaying 3D structures), and Jalview (Waterhouse et al. 2009) (for displaying sequence alignments), among others. A small representative sample of screenshots can be found at the project Web site, under the "Screenshots" section: http://wiki.c2b2.columbia.edu/workbench/index.php/Screenshots.

geWorkbench utilizes the workspace-projects paradigm that is popular with many workbench-type applications. Users work in workspaces that contain projects. Each project is used to group and manage all data (input files and analysis results) involved in a single analysis workflow. Prior to exiting, the application users can save their entire workspace so that it can be reloaded at a subsequent session.

14.6.3 geWorkbench Inputs and Outputs

geWorkbench can parse data files for many popular gene expression, genomic sequence, and protein structure formats, including Affymetrix, txt, GenePix, RMAExpress (http://rmaexpress.bmbolstad.com/), GEO, FASTA, and PDB. Several

custom formats have also been developed (e.g., a file format for representing gene interaction networks). Data can be loaded from the local file system or from remote sources. A caArray connector component offers the ability to retrieve experiments from caArray instances. Three-dimensional protein structures can be directly downloaded from the PDB. The geWorkbench Sequence Retriever component allows the retrieval of DNA sequences from the Human Genome repository at the University of Santa Cruz as well as protein sequences from EBI.

14.6.4 Future Development and Enhancement

geWorkbench provides an integration and dissemination platform for the state-of-the-art systems biology and structure analysis tools developed by the MAGNet Center investigators (http://magnet.c2b2.columbia.edu/). As more such tools become available, their integration into the application will remain a key area of focus. Additional effort will be targeted on hardening the server side component of geWorkbench, so that it too can be made available for download and deployment to support users who would like to run the geWorkbench services on their own infrastructure. Finally, the plugin framework will be extended to support dynamic loading of components from Internet-based repositories.

14.7 LSD Bundle Technical Overview

All of the tools in the Life Sciences Distribution (LSD) are written with, and deployed on, entirely open-source software. The Web-based data services (caArray, caTissue, NBIA, caGWAS, and CTODS), are all run on the Java EE technology stack. The applications use Java Server Faces (JSF) technology for the front end Web pages and a JBoss server acts as the middle tier for the application. In addition, all of the Web-based tools use Hibernate for object-to-relationship mapping and MySQL for the database persistence layer. User provisioning, security, and administration are performed using NCI CBIIT's Common Security Module (CSM) and the User Provisioning Tool (UPT). Finally, the data services also include a Web service component that leverages the Globus grid computing (Foster 2005) platform and the caGrid informatics infrastructure (Saltz et al. 2006).

geWorkbench is a client application developed in Java and requires a compatible Java Run Time Environment (JRE). From a code design stand point, the overarching objective has been to create a plugin architecture that enables the addition of new tools with minimal effort. As a result, geWorkbench has been implemented around a data representation and exchange framework that provides basic communication services to the various plugin components. These components

are developed independent of one another (each on its own code directory) and leverage the framework in order to share information. In many ways, geWorkbench resembles an application server, with each plugin component behaving as a "deployed" service.

14.8 Licensing and Support for the LSD Bundle Applications

14.8.1 Licensing

All tools in the LSD Bundle are distributed under the caBIG® License Agreement and the model agreement can be found at https://cabig.nci.nih.gov/working_groups/DSIC_SLWG/Documents/caBIG_Model_Open_Source_Software_License_v2_20080107.doc. The license is nonviral, such that derivative works are not subject to the original open source terms.

14.8.2 Support

NCI CBIIT is committed to end user and technical support for tools that are part of the LSD (Table 14.1) and there are a variety of mechanisms through which users can gain assistance, depending on their needs:

- caBIG® Knowledge centers: Knowledge centers have been established at institutions with demonstrated expertise in a specific area of focus or domain of interest to caBIG®. Within the LSD, caTissue, CTODS, caArray, caGWAS, and geWorkbench are supported through knowledge center resources. The caBIG® knowledge center resources are available at https://cabig-kc.nci.nih.gov/MediaWiki/index.php/Main_Page. The knowledge centers provide primarily Web-based resources for these tools including moderated user and developer forums, a knowledge base of frequently asked questions, as well as training and outreach materials.
- Support service providers: Support service providers are organizations that provide client-specific caBIG® support under negotiated client-provider business arrangements. Support service providers are distinguished in that they hold a limited license to NCI's caBIG® program trademarks. A full listing of available support services providers can be found at https://cabig.nci.nih.gov/esn/service_providers.
- NCI CBIIT application support: The NCI CBIIT application support team provides general support for all LSD tools, aimed particularly at deployment and training needs. The application support team can be contacted at ncicb@pop.nci.nih.gov.

Table 14.1 The tools in the Life Sciences Distribution

Product	Description
National Biomedical Imaging Archive (NBIA)	Repository for DICOM images integrated with the associated image markup, annotation, and metadata. https://cabig.nci.nih.gov/tools/NCIA
caTissue Suite	Tissue banking tool for tracking the collection storage, annotation, quality assurance, and distribution of biospecimens. https://cabig-kc.nci.nih.gov/Biospecimen/KC/index.php/CaTissue_Suite_v1.1
caArray	Standards-based microarray data management system that connects to analysis tools in caBIG® and supports prepublication collaboration through owner-driven data access controls. https://cabig-kc.nci.nih.gov/Molecular/KC/index.php/CaArray
Clinical Trials Object Data System (CTODS)	Enables the exchange of identified and deidentified clinical trials data across multiple systems while supporting syntactic and semantic interoperability. https://cabig-kc.nci.nih.gov/CTMS/KC/index.php/LabViewer
Cancer Genome-Wide Association Studies (caGWAS)	Supports the correlation of SNP genotype data, SNP association findings, population frequency data, and clinical phenotypes and provides tools for search and retrieval of GWAS findings in the context of genes or chromosomal regions of interest. https://cabig.nci.nih.gov/tools/caGWAS
geWorkbench	Enables the integrated analysis of genomics data including gene expression, sequence, and pathway information. User-friendly interface provides access to over 50 analysis and visualization modules. https://cabig-kc.nci.nih.gov/Molecular/KC/index.php/GeWorkbench
caBench-to-Bedside (caB2B)*	Allows investigators to discover and query data repositories across multiples sites on caGrid, to refine queries based on results, and to export data locally. https://cabig.nci.nih.gov/tools/caB2B
caIntegrator2*	Enables the creation of custom Web portals that bring together heterogeneous clinical, microarray, and medical imaging data such that these data can be queried, analyzed, visualized, and securely shared with collaborators. https://cabig-kc.nci.nih.gov/Molecular/KC/index.php/CaIntegrator2

An asterisk indicates those that will be added to the LSD bundle in release 1.2. The URL listed for each tool is the primary location for supporting tool information

14.9 Best Practices for Implementing the Life Sciences Distribution

Since the first release of the LSD Bundle in 2008, tools in the bundle have been installed at many cancer centers across the country. The caGrid Portal, https://cagrid-portal.nci.nih.gov, provides a view of caBIG® applications which are exposing data

to caGrid, and as of September 2009, the largest numbers of grid nodes for LSD tools were caArray, with 25 grid nodes, and caTissue, with 16 grid nodes. Many of these organizations are participating in the caBIG® Cancer Center Deployment Initiative, a structured deployment process begun in 2007 to help participating centers deploy caBIG® tools and infrastructure. Deploying centers receive resources and support to achieve caBIG® interoperability by adopting caBIG® tools and infrastructure or by adapting existing systems to be caBIG® compatible. More detailed information about this initiative can be found at https://cabig.nci.nih.gov/center_deployment.

A number of cancer centers participating in the deployment activities have shared their experiences on rolling out elements of a new translational research informatics infrastructure at their institution. A common theme shared by these groups is that a successful implementation of new software in a research setting requires a "team science" approach involving not only IT-professionals but also the adopting scientists, trainers, and project managers. An example of a successful deployment of LSD tools is at the Jackson Laboratory. The team at Jackson Labs has installed a grid-enabled instance of caArray and has developed an Extract–Transform–Load (ETL) method that integrates their internal laboratory information management system for microarray data with caArray, thereby allowing their scientists to continue using a system they were familiar with for data entry while enabling the implementation of a new tool that will allow data sharing over the grid and connection to caBIG®-compatible analytical tools including geWorkbench. Other caBIG® adoption case studies can be found at http://cabig.cancer.gov.

14.10 Future Development and Enhancement of the Life Science Distribution

The vision for the LSD is a comprehensive informatics infrastructure to support the continuum of translational research. A central goal of translational research is the correlation of molecular findings with clinical and pathological observations. This type of integrative analysis requires that data be collected in such a way that the necessary associations are maintained – for example, between biospecimens and their molecular derivations and assays – and the tools to aggregate and analyze across the component data types. The strategic roadmap of the LSD is aimed toward this critical goal. An overview of how all of the components of the LSD fit together is shown in Fig. 14.1. Individual tools currently in the LSD are being enhanced to more seamlessly support the capture of data in a way that represents the association of information across systems. In addition, the next release of the LSD will include two additional tools – Cancer Bed-to-Bedside (caB2B) and caIntegrator2 – that support data aggregation and analysis of an institution's local data as well as data distributed across the Grid.

Cancer Bed-to-Bedside (caB2B) is a caGrid-aware tool to allow end-users to query local and remote LSD components through a keyword and templated search interface. caB2B supports search and retrieval of data across distributed systems

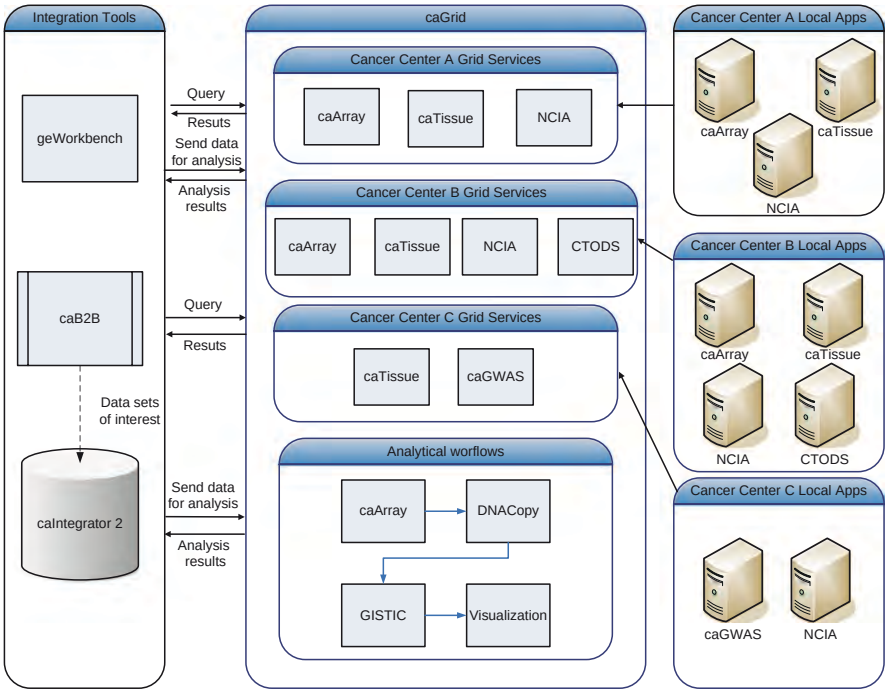


Fig. 14.1 Overall vision for the Life Science Distribution Bundle. On the *far right* of the figure are examples of individual cancer centers' deployment of one or more elements of the bundle. Represented in the *center* of the figure are the caGrid nodes for each of these tools. On *left* are the LSD tools that enable integrative browsing, aggregation, and analysis of grid data within and across institutions deploying components of the LSD bundle

and supports queries such as, "Are there any gene expression microarray data available from patients with Stage III lung cancer and are there corresponding in vivo images available for the affected patients?" Such a query would potentially span information federated across caArray, caTissue, and NBIA.

In order to share a logical set of data with collaborators – from a directed study or from in silico data mining with a federated grid search tool such as caB2B – and to perform in depth reasoning and analysis over that data, it is often necessary to bring the information into a common data warehouse. The caIntegrator2 platform has been developed to support this requirement. caIntegrator2 provides a graphical user interface to allow a study author to "point" to data of interest in systems on the grid and to then bring that data (or pointers to it, in the case of images) into the data warehouse. Once information is in the caIntegrator2 environment, end user scientists can then run advanced queries, perform correlative outcomes analysis using Kaplan–Meier plots, and access analysis and visualization tools on and off the grid. caIntegrator2 allows data from multiple studies to be stored in the same database, providing the same query and analysis functionality across all studies.

All tools in the LSD are open source and community members are encouraged to participate and contribute. The application code is made available through the code management repository at NCI. For production-level components (those included in official production releases), a formal software engineering life cycle is followed comprising development of functional requirements and design documentation, execution of formal system testing, and updating of end-user documentation (user guide, online help, and Web tutorials).

References

Altschul SF, Gish W et al (1990) Basic local alignment search tool. J Mol Biol 215(3):403–410

Ashburner M, Ball CA et al (2000) Gene ontology: tool for the unification of biology. The gene ontology consortium. Nat Genet 25(1):25–29

Brazma A, Hingamp P et al (2001) Minimum information about a microarray experiment (MIAME)-toward standards for microarray data. Nat Genet 29(4):365–371

Foster I (2005) Globus toolkit version 4: software for service-oriented systems. Netw Parallel Comput, Proc 3779:2–13

Fridsma DB, Evans J et al (2008) The BRIDG project: a technical report. J Am Med Inform Assoc 15(2):130–137

Komatsoulis GA, Warzel DB et al (2008) caCORE version 3: Implementation of a model driven, service-oriented architecture for semantic interoperability. J Biomed Inform 41(1):106–123

Lefebvre C, Lim WK, Basso K, Dalla Favera R, Califano A (2007) A context-specific network of protein-DNA and protein-protein interactions reveals new regulatory motifs in human B cells. Lect Notes Bioinform 4532:42–56

Margolin AA, Nemenman I et al (2006) ARACNE: an algorithm for the reconstruction of gene regulatory networks in a mammalian cellular context. BMC Bioinform 7(Suppl 1):S7

Rayner TF, Rocca-Serra P et al (2006) A simple spreadsheet-based, MIAME-supportive format for microarray data: MAGE-TAB. BMC Bioinform 7:489

Rzhetsky A, Iossifov I et al (2004) GeneWays: a system for extracting, analyzing, visualizing, and integrating molecular pathway data. J Biomed Inform 37(1):43–53

Saltz J, Oster S et al (2006) caGrid: design and implementation of the core architecture of the cancer biomedical informatics grid. Bioinformatics 22(15):1910–1916

Shannon P, Markiel A et al (2003) Cytoscape: a software environment for integrated models of biomolecular interaction networks. Genome Res 13(11):2498–2504

von Eschenbach AC, Buetow K (2007) Cancer Informatics Vision: caBIG. Cancer Inform 2:22–24

Wang K, Alvarez MJ et al (2009) Dissecting the interface between signaling and transcriptional regulation in human B cells. Pac Symp Biocomput 20:264–275

Ward LD, Bussemaker HJ (2008) Predicting functional transcription factor binding through alignment-free and affinity-based analysis of orthologous promoter sequences. Bioinformatics 24(13):i165–i171

Waterhouse AM, Procter JB et al (2009) Jalview Version 2 – a multiple sequence alignment editor and analysis workbench. Bioinformatics 25(9):1189–1191

Whetzel PL, Parkinson H et al (2006) The MGED ontology: a resource for semantics-based description of microarray experiments. Bioinformatics 22(7):866–873

Chapter 15
MeV: MultiExperiment Viewer

Eleanor Howe, Kristina Holton, Sarita Nair, Daniel Schlauch, Raktim Sinha, and John Quackenbush

Abstract MultiExperiment Viewer (MeV) is a freely available software application that puts modern bioinformatics tools for integrative data analysis in the hands of bench biologists. MeV is a versatile microarray data analysis tool, incorporating sophisticated algorithms for clustering, visualization, classification, statistical analysis, and biological theme discovery from single or multiple experiments. This chapter gives an overview of MeV technical details and its use in a real setting.

15.1 Introduction

MeV's simple interface provides easy access to an extensive library of bioinformatics algorithms and visualization tools normally only available through a complicated command-line interface. The program's graphical menu- and button-driven user interface makes the manipulation and visualization of large complex biological datasets possible for anyone possessing basic computer skills. MeV was originally built to analyze DNA microarray expression data; however, its functions have been expanded greatly such that it can now process data from Array Comparative Genomic Hybridization (aCGH) and protein–protein interaction (PPI) experiments. These data and many others can be loaded into the program and examined using a wide variety of functions: clustering, visualization, statistical testing, and annotation-based meta-analysis. The results of each analysis are presented in attractive graphical viewers and their numerical results can be saved as text files for later reference. If annotation is loaded along with the data, links to biologically relevant Web sites, such as NCBI, AmiGO, and others are also made, turning MeV into a launching pad for further exploration of the data on the Web.

J. Quackenbush (✉)
Dana-Farber Cancer Institute, 44 Binney Street, Smith 822A, Boston, MA 02115, USA
e-mail: johnq@jimmy.harvard.edu

M.F. Ochs et al. (eds.), *Biomedical Informatics for Cancer Research*,
DOI 10.1007/978-1-4419-5714-6_15, © Springer Science+Business Media, LLC 2010

15.2 Technical Details

This chapter is written in reference to MeV v4.4. MeV is released under the terms of the Artistic License (http://www.opensource.org/licenses/artistic-license-1.0.php) and as such is freely available to the commercial and scientific community. It is cross-platform compatible and has been tested on Windows 2000/XP/Vista, Mac OSX, and several distributions of Linux. MeV for Windows and Linux requires Java v1.6 or higher, while MeV for Mac OSX requires Java v1.5 or higher. Some modules of MeV that display 3D graphics also require the Java3D package be installed. MeV is one component of the TM4 Microarray Software Suite, and as such can be downloaded from http://www.tm4.org/. Users of the suite should cite (Saeed et al. 2003).

15.2.1 Getting Started with MeV

New MeV users should begin at the TM4 Web site, http://www.tm4.org/, where MeV and the other components of the TM4 Software Suite can be downloaded. The first stop should be MeV's quick-start guide, a short guided tour through downloading and installing the application and running a simple analysis. More detailed descriptions of the modules in MeV, the clustering interface, file formats, and many other topics are covered in the MeV user manual, a 200+ page pdf file. Both of these documents are also included in the MeV download package. Questions about MeV can be directed to the MeV forums, where the developers answer questions and provide support. There, users can report bugs, request new features and discuss their data analysis strategies. Users can also sign up for the TM4-announce mailing list that provides notification of updates to the program.

15.2.2 MeV In Use

Most will begin their work with MeV by loading genomic data that has been processed by a standard normalization tool, such as RMA, in the case of mRNA expression arrays. The most common format for loading data is a simple tab-delimited text file, containing both expression values and annotation data, such as Entrezgene IDs or Affymetrix probe IDs. In addition, MeV contains many specialized file loaders capable of reading files from a variety of sources, such as Agilent data files, GEO and ArrayExpress downloads, and many others.

Once the data is loaded, it will be displayed in a few basic viewers, including a heatmap, a multiline graph of expression values, and a tabular text-based view. These viewers can be customized to accommodate small monitors or the color preferences of the user. Groups of genes or samples can be labeled with color to

distinguish them from the other elements in the dataset. These groups are referred to as "clusters" and are an integral part of the MeV user interface. Using these clusters, the results of different analyses can be labeled with distinct colors and displayed alongside each other in the same view for comparison. These clusters can also be exported to text files and broadcasted via the Gaggle network to other programs.

The wide range of data-manipulation tools that are the core of MeV are available in a series of drop-down menus located at the top of the viewer window. Each item in these drop-down menus corresponds to an analysis module that takes expression or annotation input and produces groups of genes or samples and other data as output. On choosing a module the user is presented with a dialog, common in style across all modules in the program, where any parameters required by the module can be selected. Help text is available to explain the options and the statistical processes used by the module.

After a module has finished, its output appears as a node in the result tree on the left-hand side of the screen, under the heading "Analysis Results." The results there differ depending on the type of module selected, but most modules include a variation on the same common heatmap and tabular views used in the initial display. Some modules also produce custom output; for example, the Principle Components Analysis (PCA) module produces a three-dimensional interactive graph of the result data. Each new module that is run produces another addition to the result tree. These results can be stored as clusters in the cluster manager and used as input into other algorithms within MeV.

15.3 Highlights of MeV Analysis Tools

A complete description of all of the tools in MeV is outside the scope of this chapter. Instead we will describe a small selection of the more popular or more interesting modules. A complete list of all of the available modules in MeV can be found in Table 15.1.

15.3.1 Gene Selection Tools

A common goal of microarray analysis is the detection of genes that are differentially expressed under varying conditions. Numerous statistical techniques are available in MeV for the evaluation of the statistical significance of these expression changes. MeV provides several traditional methods, such as *t* test and analysis of variance as well more novel approaches such as Significance Analysis of Microarrays (SAM), Bayesian Estimation of Temporal Regulation (BETR), and Rank Products.

The widely used Significance Analysis of Microarrays (SAM) algorithm (Tusher et al. 2001) is readily available from the drop-down analysis menu. Like all other

Table 15.1 Complete list of modules included with MeV

Clustering	*Statistics*
Hierarchical clustering	Pavlidis template matching
Tree-EASE	*t*-Test
Support trees clustering	Bridge
Self-organizing tree algorithm	Significance analysis of microarrays (SAM)
k-means clustering	One-way ANOVA
k-means support clustering	Two-way ANOVA
Cluster affinity search technique	Nonparametric tests (includes Wilcoxon, Man-Whitney test, Kruskal–Wallace test, Mack–Skillings test, Fisher exact test)
Figure of merit	Bayesian estimation of temporal regulation
QT cluster	Rank products
Self-organizing map	*Data-reduction*
Classification	Relevance networks
Support vector machines	Principle components analysis
Uncorrected shrunken centroid classification	Correspondence analysis
K-nearest neighbors classification	Expression terrain map
Discriminant analysis classification	*Visualization*
Meta-analysis	Linear expression maps
Gene set enrichment analysis	Gene distance matrix
Expression analysis systematic explorer	
Miscellaneous	
Gene shaving	
Bayesian network	
Literature mining	

modules, running SAM presents the user an intuitively visual initialization dialog. Users have five options for choosing the analysis type: one-class, two-class paired and unpaired, multiclass, and censored survival. Their choice will depend on the details of their data. Assigning samples to groups is easy in SAM. Like other modules, users may employ either an individual radio button technique or a cluster assignment feature to place their samples into appropriate groups. Numerous additional parameters, such as permutation count and the use of hierarchical clustering (HCL) are available to be adjusted in the area below the group selection box.

After receiving the user-specified parameters and group assignments, MeV's SAM calculates gene-specific test statistics and plots them against unaffected scores. MeV then plots the graph and asks the user to select a delta value before assigning significance. By using the scroll bar in the SAM graph, one can choose a suitable number of significant genes with an acceptable false discovery rate to go with it.

The Rank Products module (Breitling et al. 2004) provides a similar function to SAM. It is a nonparametric method, which is based on ranks of fold-changes rather than the actual expression values. This allows for greater robustness against noisy data and can provide reproducible results with a smaller number of replicate experiments.

The BETR module is a novel technique specifically designed for time-course data. BETR (Aryee et al. 2009) is a flexible linear random-effects modeling framework that takes into account correlations between samples and the corresponding sampling times. It attempts to regain information from the data that is often lost by

many methods that treat each data measured at a different time-point as independent and ignore crucial clues hidden within the arrays.

The typical statistics analysis produces several standard viewers. Genes are divided into significant and nonsignificant clusters and output in the form of expression images, centroid graphs, expression graphs, and tables. The latter contain important information regarding the module's calculations, such as *p* values, *q* values, fold-change, or any other value pertinent to the analysis.

15.3.2 Clustering Tools

HCL is one of the cornerstones of unsupervised data analysis in MeV. It allows users to visualize their dataset's heatmap in a more organized manner, via a dendrogram, to look for emergent trends (Eisen et al. 1998). Constructing hierarchical trees from the results is also an option in many of the statistics modules.

The HCL module can be reached via the "Clustering" drop-down menu at the top of the Viewer window. The first section of user-defined input asks whether to create clusters by gene, sample, or both. The next section allows the user to select if they would like to optimize the leaf ordering. Optimized leaf ordering arranges the tree nodes such that samples/genes with shortest distance appear adjacent in the resulting dendrogram. This makes for a neater heatmap, but the process (and HCL in general) is memory-intensive and may require trimming down the dataset. HCL will prompt the user if this is the case. The HCL module also allows users to pick a distance metric for constructing hierarchical trees, and asks them to choose the linkage methods among average, complete, and single linkage.

Once these parameters have been selected, MeV will create a results viewer labeled "HCL" in the result tree on the left-hand side of the window. Opening up the result node reveals the results in the form of a tree diagram attached to a heatmap. Right-clicking on the nodes allows users to save genes/samples in the resultant subsection of the tree as a cluster. Right-clicking on the heatmap and selecting "sample/gene node properties" will bring up a dialog allowing users to set the minimum/maximum node height, and to reduce the complexity of the tree by imposing a distance threshold. By using the "Apply" button in this dialog, the HCL tree can be fine-tuned visually. An example of the display resulting from HCL can be found in Fig. 15.1.

15.3.3 Functional Classification

The Expression Analysis Systematic Explorer (EASE) algorithm was developed by the DAVID Bioinformatics group as a means of identifying trends in a subset of genes relative to the parent group (Dennis et al. 2003; Hosack et al. 2003). The implementation of this tool in MeV has the advantage that it is fully integrated with the other MeV functions as well as the MeV annotation model. Often, a set of differentially expressed genes identified by a SAM analysis is used as an input of this

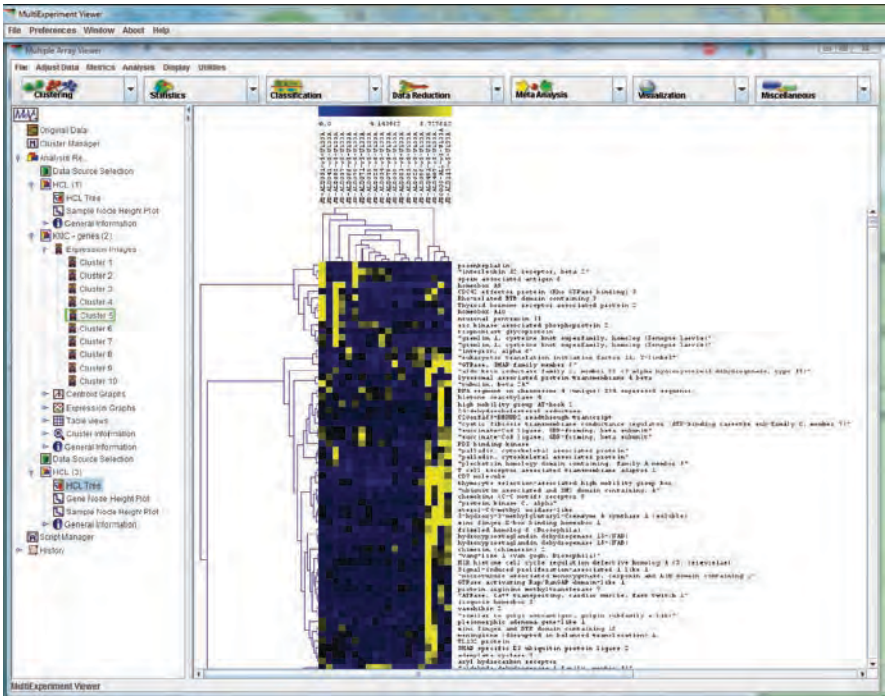


Fig. 15.1 The main graphical user interface containing output from the Hierarchical clustering module

algorithm. EASE will then return a list of functional categories, such as GO terms or KEGG pathways, that are more frequently associated with the genes in that group than with the rest of genes in the microarray. This list can help the user identify biological themes in their data.

EASE analysis requires support annotation information, and for that, MeV supports many microarray formats. The EASE module makes use of MeV's new annotation-downloading features to simplify this process. If a user has already loaded annotation using the automatic annotation loader, running EASE is as simple as selecting a cluster of interest and selecting the files containing the functional categories of interest. These files are given sensible names, such as "GO Biological Process.txt." Several statistical options are also available, such as various multiple-testing corrections and trimming parameters.

The basic results of the EASE include a list of terms (GO terms, KEGG pathways, etc.) that are overrepresented in the selected data subset, relative to the background population – in this case, the complete contents of the array. This list of resulting terms is presented in tabular form for easy viewing and can be saved as a text file. The standard MeV viewers are used to display the names and expression values of each gene, which is associated with these resultant terms. These groups can be stored as clusters or exported as text files for use with other applications.

Gene Set Enrichment Analysis (GSEA) is another functional classification tool, first introduced in 2003 (Subramanian et al. 2005). Several extensions and enhancements to this method have been proposed (Jiang and Gentleman 2007; Kim and Volsky 2005). GSEA provides a clear edge over classical DNA microarray analysis methods, by focusing on groups of functionally related genes rather than treating genes as independent entities. This design allows for detection of small but coordinated gene expression changes of genes within groups that are defined according to some significant biological property such as a metabolic pathway or biological process. GSEA has been implemented in various gene expression analysis programs including MeV, which has adopted the extensions proposed by Jiang and Gentleman (Jiang and Gentleman 2007).

The power of GSEA lies in being able to detect significant expression changes in groups of functionally related genes as opposed to testing genes individually. We utilize data from a clinical trial in Acute Lymphoblastic Leukemia (ALL) (Chiaretti et al. 2004) to demonstrate the insights gained from this method.

ALL has been associated with cytogenetic abnormalities. For the analysis of this data we define our gene sets of interest as those groups representing common chromosomal regions. The focus of the analysis of this data is to compare the expression changes within these groups for the BCR/ABL samples vs. those samples collected from patients with no chromosomal abnormalities. Previous analysis on the data set (Jiang and Gentleman 2007) suggests using the hyperdiploidy information and gender status of the patients as factors to infer phenotype effects. We, therefore, used a three-factor model with factors corresponding to phenotype, hyperdiploidy, and sex.

Gene annotations for the data set (Affymetrix HG-U95av2 chip) were obtained using the automatic annotation download utility in MeV. In mapping the signal from multiple Affymetrix probe sets to a single Entrez gene identifier, we used the probe with the maximum standard deviation (SD). Only probes with SD 0.6 or higher were included in the analysis.

Filtering the data set for samples not associated with either phenotype eliminated 49 samples, leaving us with 12,625 genes and 79 samples. In addition to this, Affymetrix quality control probes present in the dataset were also eliminated, leaving us with 12,558 genes and 79 samples. The gene sets were filtered to retain only those where at least five genes mapped to a chromosomal location. A basic tabular view of the significant gene sets from this analysis is presented in Table 15.2. Visualizations available for this analysis include all the standard MeV viewers and a graph of overenriched *p* values.

15.3.4 Network Analysis

Bayesian Network (BN) analysis attempts to learn biologically meaningful gene interaction networks (Directed Acyclical Graphs – DAGs) from mRNA expression data. The underlying assumption in this effort is that most popular bioinformatics

Table 15.2 GSEA result table

Gene sets (Chromosome)	*p* values
16p12.1	0.008
19p13.13	0.008
19q13.1	0.004
19q13.1-q13.2	0.004
1p34	0.008
1p34.1	0.008
1p36.1-p35	0
22q13.1	0.004
22q13.2-q13.31	0
Xq28	0.008

methods are good at identifying genes that discriminate among several subject/experiment groups or experimental conditions. However, they often fail to elucidate the underlying mechanism of gene interaction that captures a biological process.

This novel method (Djebbari and Quackenbush 2008) uses seeds learned from the biomedical literature, protein–protein interactions or KEGG interactions, or any combination thereof, to construct a starting "prior" network. The machine-learning algorithm then uses information obtained from the expression data to learn and refine the network, and predict a new, high-confidence network. The samples are bootstrapped to control overfitting of the network. The results of this process are then exported to Cytoscape, a network visualization and analysis tool, for viewing (Shannon et al. 2006).

As an extension to this already published method, a Cytoscape plug-in has been developed, which provides predictive modeling of the Bayes Network described above. This plugin is open-source and freely available, and designed to work closely with MeV. In a nutshell, it attempts to predict the state of a gene, given the state of its parent gene. The possible states that a gene can exist in are: up regulated, down regulated, or unchanged. Once a network has been learned, the method finds the conditional probability table (CPT) associated with each node (gene). The CPT of a node constitutes the individual probabilities of the gene being up, down, or unchanged given its parent(s) is/are in state up, down, unchanged (any combination of parent and state). Knowing the CPT of any node, one can then predict the exact probability of the node being in any state, given any combination of parents and their states. This method allows researchers to conditionally alter gene expression and predict the resulting changes in a biological process based on microarray data; these predictions can then be experimentally validated. Hopefully, this novel approach would enable researchers to (a) get a better understanding of the biological interactions that exists in an enriched set of genes and (b) to predict as-yet unknown processes and/or interactions that are important in a disease condition.

The BN module in MeV can be accessed from under the "Miscellaneous" category of algorithms. MeV provides annotations and support files for all major Affymetrix platforms required to run this module and constantly adds new arrays and platform to the annotation database.

15.4 MeV and Other Software

15.4.1 Further Analysis of MeV's Results

The results from MeV's many modules are easy to export to other programs for further analysis and display. Heatmaps can be saved as image files suitable for publication (e.g., bmp or jpg format). Lists of genes can be saved as text files. The results of these output groups can be labeled within MeV, allowing the user to identify them when they appear in other module's result viewers. This makes it easy to compare the results of differing analyses, or to select subsets of the loaded data for later analysis.

MeV implements the Gaggle framework, allowing it to share data with other programs that implement the interface (Shannon et al. 2006). These programs include the R programming environment (R_Development_Core_Team 2005) and Cytoscape (Killcoyne et al. 2009). The interface for doing this work is simple, and uses the same context-sensitive menus that allow the storing and manipulation of MeV's results. MeV can also receive data from other Gaggle-enabled programs, bypassing the need for text file intermediates.

MeV can be configured to launch directly from a Web site, preloaded with selected data and annotation. An example of this behavior in action is found on the Web site for the GeneChip Oncology Database or GCOD (http://compbio.dfci.harvard.edu/tgi/cgi-bin/tucan/tucan.pl). This site is a search interface to a large database of cancer-related DNA microarray studies. There, the results of any search can be launched directly into MeV simply by clicking on the appropriate link; MeV need not even be installed on the user's computer. Any data file that can be loaded by MeV's file loaders can be launched directly from a Web site in this way. Complete instructions for enabling this kind of display are available on the TM4 Web site.

15.5 The Future of MeV

15.5.1 Improvements and Maintenance of MeV

MeV is maintained and constantly improved by a group of programmers at the Dana-Farber Cancer Institute under the direction of Dr. John Quackenbush. These developers support the application through the SourceForge forums and regularly update the program with new features. A new version containing these latest new features is released approximately every 6 months, in June and December. This document refers to MeV v4.4, released in June of 2009.

This latest release included improvements to the GSEA module and the cluster-management tools, and the addition of the Cytoscape plugin that will allow network prediction as part of the Bayes Networks module. Within the next year the development

team expects to offer an upgraded version of the SAM algorithm, the Predictive Analysis for Microarrays (PAM) sample-classification algorithm (Tibshirani et al. 2002), and an implementation of the Linear Models for Microarray data (LIMMA) algorithm (Smyth 2004, 2005).

In addition to new numerical methods for expression analysis, the MeV development team plans to expand the application's support for nonnumerical support data: annotation. The constant updating of annotation data is one of the difficulties in high-throughput biological data analysis. Many of the algorithms implemented in MeV require annotation information to function. Furthermore, the results for any analysis are useless without up-to-date annotation to inform the investigator about the nature of the genes in their result list. MeV mitigates this problem for many array types by connecting to the internet and automatically downloading the appropriate annotation. The MeV team will be expanding the list of currently supported arrays from approximately 100 to many more. We plan to also support a variety of platforms, such as Affymetrix GeneChips, Illumina chips, Agilent chips, various popular 2-color spotted arrays, etc.

The MeV development team is committed to expanding the list of available modules and improving the user interface and display capabilities of the program. As research proceeds and new genomic analysis methods are developed, the team plans to add the most popular and most powerful of the new tools to MeV, so as to provide a complete resource for data analysis.

The team also hopes to continue a tradition of including new features contributed by collaborating developers. Collaborators working with other groups have added to MeV new file loaders, new user-interface features, and entirely new modules. We gladly accept such contributions and hope to encourage more in the future. To that end, we will be more thoroughly documenting the modular system that MeV is based on, in the hopes of making it very simple for an outside developer to add new functionality to the application.

15.6 Conclusions

MeV is not only an analysis tool useable by the average biologist but a development platform for building new software tools for microarray data analysis. It has a simple interface and a wide range of analysis tools, making it ideal for the bench biologist. Because it is free and open-source, it is easy to try out and modify to suit the specific needs of any given project. Small bioinformatics groups can easily write new modules that take advantage of all the visualization and data-manipulation infrastructure of MeV. These modifications, if submitted to the MeV team, can be distributed as part of the MeV package and made immediately available to thousands of end-users.

References

Aryee MJ, Gutiérrez-Pabello JA, Kramnik I, Maiti T, Quackenbush J (2009) An improved empirical bayes approach to estimating differential gene expression in microarray time-course data: BETR (Bayesian Estimation of Temporal Regulation). BMC Bioinformatics 10:409

Breitling R, Armengaud P, Amtmann A, Herzyk P (2004) Rank products: a simple, yet powerful, new method to detect differentially regulated genes in replicated microarray experiments. FEBS Lett 573(1–3):83–92

Chiaretti S, Li X, Gentleman R, Vitale A, Vignetti M, Mandelli F, Ritz J, Foa R (2004) Gene expression profile of adult T-cell acute lymphocytic leukemia identifies distinct subsets of patients with different response to therapy and survival. Blood 103(7):2771–2778

Dennis G Jr, Sherman BT, Hosack DA, Yang J, Gao W, Lane HC, Lempicki RA (2003) DAVID: Database for annotation, visualization, and integrated discovery. Genome Biol 4(5):P3

Djebbari A, Quackenbush J (2008) Seeded Bayesian networks: constructing genetic networks from microarray data. BMC Syst Biol 2:57

Eisen MB, Spellman PT, Brown PO, Botstein D (1998) Cluster analysis and display of genome-wide expression patterns. Proc Natl Acad Sci U S A 95(25):14863–14868

Hosack DA, Dennis G Jr, Sherman BT, Lane HC, Lempicki RA (2003) Identifying biological themes within lists of genes with EASE. Genome Biol 4(10):R70

Jiang Z, Gentleman R (2007) Extensions to gene set enrichment. Bioinformatics 23(3):306–313

Killcoyne S, Carter GW, Smith J, Boyle J (2009) Cytoscape: a community-based framework for network modeling. Methods Mol Biol 563:219–239

Kim SY, Volsky DJ (2005) PAGE: parametric analysis of gene set enrichment. BMC Bioinformatics 6:144

R_Development_Core_Team (2005) R: A language and environment for statistical computing. R Foundation for Statistical Computing. Vienna, Austria

Saeed AI, Sharov V, White J, Li J, Liang W, Bhagabati N, Braisted J, Klapa M, Currier T, Thiagarajan M, Sturn A, Snuffin M, Rezantsev A, Popov D, Ryltsov A, Kostukovich E, Borisovsky I, Liu Z, Vinsavich A, Trush V, Quackenbush J (2003) TM4: a free, open-source system for microarray data management and analysis. Biotechniques 34(2):374–8

Shannon PT, Reiss DJ, Bonneau R, Baliga NS (2006) The Gaggle: an open-source software system for integrating bioinformatics software and data sources. BMC Bioinformatics 7:176

Smyth GK (2004) Linear models and empirical bayes methods for assessing differential expression in microarray experiments. Stat Appl Genet Mol Biol 3:Article3

Smyth GK (2005) Limma: linear models for microarray data. In: Gentleman R, Carey V, Dudoit S, Irizarry R, Huber W (eds) Bioinformatics and computational biology solutions using R and bioconductor. Springer, New York, pp 397–420

Subramanian A, Tamayo P, Mootha VK, Mukherjee S, Ebert BL, Gillette MA, Paulovich A, Pomeroy SL, Golub TR, Lander ES, Mesirov JP (2005) Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles. Proc Natl Acad Sci U S A 102(43):15545–15550

Tibshirani R, Hastie T, Narasimhan B, Chu G (2002) Diagnosis of multiple cancer types by shrunken centroids of gene expression. Proc Natl Acad Sci U S A 99(10):6567–6572

Tusher VG, Tibshirani R, Chu G (2001) Significance analysis of microarrays applied to the ionizing radiation response. Proc Natl Acad Sci U S A 98(9):5116–5121

Chapter 16
Authentication and Authorization in Cancer Research Systems

Stephen Langella, Shannon Hastings, Scott Oster, Philip Payne, and Frank Siebenlist

Abstract Enforcing controlled access to resources in cancer research systems, while facilitating resource sharing among collaborators, is a challenging problem. This is especially challenging when resources are distributed across organizational boundaries and researchers from a distributed set of organizations require access to them. In this chapter, we will present motivating use cases, a software solution, and policies for enforcing access control in large distributed cancer research systems.

16.1 Introduction

The informatics requirements of multi-institutional translational research projects are characterized by the need to securely share and access data and analytical resources hosted at different sites. In a multi-institutional project, sites participating in the collaborative effort can be viewed as being part of a *virtual organization*. One of the major obstacles to forming virtual organizations in biomedical research has been the lack of interoperability among disparate data and analytical resources. Another major problem has been the limited availability of infrastructure to provide secure and efficient access to these resources. Without mechanisms that enable service providers to enforce access control policies to protect sensitive and proprietary information, data and analytical resources cannot be shared effectively. Traditionally, collaborative projects have created virtual organizations by employing a centralized system to host the databases and analysis tools at one of the institutions participating in the project. This approach, while alleviating some of the security and interoperability issues, is not scalable when the number of collaborating sites is large. It also is not efficient when it is desirable to rapidly and dynamically create, manage, and change virtual organizations.

S. Langella (✉)
Department of Biomedical Informatics, Ohio State University, 333 W 10th Avenue, Columbus, OH 43210, USA
e-mail: Stephen.Langella@osumc.edu

M.F. Ochs et al. (eds.), *Biomedical Informatics for Cancer Research*,
DOI 10.1007/978-1-4419-5714-6_16, © Springer Science+Business Media, LLC 2010

The National Cancer Institute's (NCI) Cancer Biomedical Informatics Grid (caBIG®) (caBIG® 2009a) is addressing the informatics issues that arise in multi-institutional studies in biomedical research (see Chap. 9). This effort is developing informatics standards, a suite of common tools and applications, shared data and analytical resources, and a Grid infrastructure, called caGrid (Oster et al. 2007; caGrid 2009; see Chap. 4) to dynamically link applications, clients, and community provided resources. Security is of paramount importance in the caBIG® program to ensure that any sensitive information such as patient-related information as well as the intellectual properties of researchers can be protected while promoting and facilitating collaborative projects.

Supporting authentication (i.e., determining whether or not a given user is who she/he claims to be) and authorization (i.e., controlling access to the functionality of a resource, for an authenticated user) in the caBIG® environment is difficult. User identities and credentials should be managed in a decentralized manner for scalability and manageability reasons, while allowing institutions to set up and enforce their access control policies locally for their resources. If there are many participants from different organizations, credentials should be managed in a federated environment. Tools are needed for system administrators to efficiently provision the credentials of users in their institutions in this federated environment. Another issue that becomes critically important in a dynamic and large-scale federated environment such as caBIG® is the management of a *trust fabric*. Because a given institution will have autonomous control over its policies for granting, managing, changing, and revoking user credentials for its users, it can be expected that other institutions will have varying levels of trust of its users wanting access to their resources. Moreover, there is a need to be able to efficiently propagate dynamic changes in policies and trust relationships, as well as any sensitive security-related events (e.g., a user's credentials are revoked, because they have been compromised) to other entities in the federated environment. Tools are needed to provide collaborating institutions a mechanism to define and manage these trust relationships, while infrastructure is needed to distribute this information to their participating services and users; this combination of policy and infrastructure defines the *trust fabric*.

The caGrid software provides a comprehensive security suite to address the security challenges in multi-institutional translational research. This suite is referred to as the "Grid Authentication and Authorization with Reliably Distributed Services" infrastructure (GAARDS) (Langella et al. 2007a, 2008). The salient features of the GAARDS infrastructure can be summarized as follows (1) it provides services to support (a) integration of institutional identity provider and authentication systems with the Grid environment, (b) efficient management and federation of user and service/server credentials, and (c) easy deployment of a Grid-enabled identity provider system; (2) it implements support for group (role) based access control such that a service provider can use both community accepted roles and locally defined roles to implement and enforce access control policies; and (3) it provides a service infrastructure for management of a trust fabric in the Grid environment, where institutions use different policies for provisioning of credentials for their local researchers and where credentials can be created, revoked, and reinstated

dynamically. While the requirements for GAARDS have been motivated mainly by use cases from the caBIG® program, the design and implementation of the infrastructure is generic and can be applied in other domains. The GAARDS infrastructure is available as both a stand-alone system and a component of the caGrid infrastructure, which is the Grid architecture of caBIG®.

In Sect. 16.2, we will discuss the security challenges faced in multi-institutional translational research. We will then present an overview of the infrastructure that addresses these challenges. Finally we will provide an overview of the policies caBIG® is investigating for governing this infrastructure, and will demonstrate how this infrastructure can be deployed in a production environment under these policies.

16.2 Security Challenges

The GAARDS infrastructure is designed to support authentication and authorization in a federated environment. This section presents the issues that have motivated the design and implementation of these components in GAARDS. We describe the issues in the context of the caBIG® environment, which is envisioned to span hundreds of institutions and thousands of researchers.

The objective of the caBIG® program is to help accelerate research toward curing cancer by implementing the enabling informatics technologies for researchers to more efficiently find, share, retrieve, integrate, and process clinical and research data from disparate sources. The caBIG® community consists of participants from cancer centers, research institutions, government organizations, and the informatics industry. Efforts underway in the caBIG® program include the development and deployment of (1) informatics standards, (2) guidelines and tools to improve semantic and syntactic interoperability among data and analytical resources, (3) open-source, common applications for data management and analysis, (4) guidelines and processes for data and tool sharing, and (5) an open-source, standards based Grid infrastructure that is designed to federate distributed resources. While the spirit of caBIG® is to promote and facilitate sharing of information and applications, not all information and tools can be made publicly available to everyone. Clinical information and the intellectual properties of researchers must be protected, and such information should be accessible only by those with appropriate privileges.

caBIG® is implemented as a federated environment where individuals, groups, and institutions manage and administer their resources locally. Resources are exposed to the environment and shared among institutions and researchers using the caGrid infrastructure, which provides a core suite of tools, services, and a runtime environment to enable secure federation of resources. Through caGrid, all data and analytical resources are implemented as Grid Services conforming to the Web Services Resource Framework standards. Interactions between the clients and the caGrid services are carried out using standard Grid Service protocols. The GAARDS infrastructure is designed to support security requirements in a service-oriented environment. We now discuss these requirements.

In order to access the caBIG® resources, users are required to authenticate with caGrid services that encapsulate those resources. For this purpose, users are issued *grid credentials through which they can* authenticate and prove their identity to the services. To support the mutual-authentication of users and services across organizational boundaries, a common type of credentials needs to be adopted. Furthermore, those credentials have to be issued by a trusted set of credential issuers (also known as certification authorities or CAs). The Grid uses X.509 digital certificates for the authentication and identification of users and services. By digitally signing the certificate, a CA asserts the binding of a name to a public key. An X.509 certificate with its corresponding private key forms a unique credential: the so-called *grid credential*.

Although this approach is very effective and secure, it is difficult to manage the certificate life-cycle in a multi-institutional environment. Using existing tools, the provisioning of grid credentials is a manual process, which is very complex for most users and system administrators and, therefore, error-prone. The overall process is further complicated if a user wishes to authenticate from multiple locations, because a copy of their grid-credentials, that is, their private key and certificate, has to be present at every location. Not only is this process complex, but securely distributing private keys poses an additional security risk as it is subject to easily made mistakes. Additionally, there are scalability and efficiency problems with vetting user identities. Often security policies that govern access to public health information require in-person identity vetting. Organizing the policies and human infrastructure required for in-person identity vetting in a multi-institutional environment that spans across countries is a very difficult process and resource intensive. In the majority of cases, organizations that wish to participate in multi-institutional studies have already invested a significant amount of resources into their existing identity management systems and already have processes in place for the vetting and issuing of some form of user-credentials. In such settings, it would be more efficient to leverage existing identity management systems to derive and provision Grid user accounts. Users would be able to use their existing credentials to "log on" to obtain Grid credentials, which in turn allow them access to the Grid services. This scenario requires a mechanism to allow users to obtain Grid credentials using their existing organization-provided credentials. In other words, in this scenario, the federation will operate a set of certification authorities that will issue certificates to users who present their existing organization-provided credentials. This mechanism should also remove the complications of using and managing Grid credentials, meaning from the user's perspective they will be using the locally provided credentials they use everyday; the Grid credentials should be abstracted from them and hidden.

The GAARDS infrastructure addresses these challenges using its Authentication Service (Langella 2009a) and Dorian (Langella et al. 2007; Langella 2009c) frameworks, each of which are discussed later in this chapter. To realize this scenario, organizations participating in the federation will need to trust one another's policies and procedures for local authentication in order to accept one another's credentials as an acceptable authentication mechanism. To accomplish this, a VO-wide policy must be developed to govern the procedures and operations of the local identity providers. This policy must be reasonable such that it gives the different organizations in the collaboration the confidence to trust one another but flexible enough that it

gives each organization the freedom to operate their identity provider based on local requirements. Each organization participating in the federation must agree to comply with this policy.

Besides leveraging traditional identity providers as described in the last scenario, organizations with existing CAs or those that employ third party CAs, should be able do so, provided those authorities can also comply with the VO's policies. Likewise in certain collaboration scenarios, credentials issued by the CAs from other Grids should be able to be leveraged. In such a setting, it is important to be able to verify and validate identities and privileges with a level of confidence. Because each CA will have different policies as to how it issues, controls, audits, and revokes its credentials, it can be expected that a service provider may want to use a tiered level of assurance (in terms of authentication) for the different users wishing to access the service. In addition, while institutions will want to collaborate, they will have services with different levels of security policy enforcement requirements. In the Grid (and in any public key infrastructure) services and users need to maintain a list of CAs they trust. The main challenge is that there may be dozens to hundreds of different CAs, each issuing certificates for potentially thousands of users. Properly maintaining this list of trusted CAs for all relying parties is critical to the security of the infrastructure, but is extremely difficult due to the distributed and large-scale nature of the environment. This problem is compounded by the fact that certificates will be issued and revoked continuously and CAs may be added to or deleted from the environment dynamically. A Grid-wide mechanism is needed to create and manage a *trust fabric* for the collaboration so that services and users can make authentication and authorization decisions based on the most up-to-date security configuration information. This challenge is addressed by the Grid Trust Service (GTS), which will be described later in this chapter.

Authorization is a challenging issue as well. It is desirable that access control policy be maintained and enforced locally, giving data providers the ability to determine who has access to their data. At the same time, it is important for scalability that access control policies be based on Grid-level information. To ease the burden of access control administration, many systems base their access control policies on abstractions like group membership and roles. As a result there is a clear requirement for a standardized mechanism to organize and manage groups and their membership spanning organizational boundaries. This challenge is addressed by an infrastructure service Grid Grouper (Langella 2009d), which will be described later in this chapter.

16.3 GAARDS

The GAARDS infrastructure provides services and tools for the administration and enforcement of security policy in an enterprise Grid. The GAARDS infrastructure has been developed as a suite of services and administrative tools on top of the Globus Toolkit (Globus 2009) and its Grid Security Infrastructure (GSI) component. It consists of the following core services/components:

Dorian (Langella et al. 2006; Langella 2009c): A Grid service for the provisioning and management of Grid credentials. Dorian provides an integration point between external security domains and the collaboration's Grid infrastructure. Dorian allows users to use their existing credentials (external to the Grid) to obtain Grid credentials that can be used for authentication to the Grid services. Dorian also allows service credentials to be issued, by binding them to an authorized user credential.

Authentication Service (Langella 2009a): A framework for issuing Security Assertion Markup Language (SAML) authentication-assertions for existing identity providers. It allows the identity provider to assert in a standardized digitally signed statement (SAML) that a user has already been authenticated, which is then presented to the Dorian service. Dorian validates the assertion and subsequently issues Grid credentials. The authentication service also provides an optional uniform log-in interface upon which applications can be built.

Grid Trust Service (GTS) (Langella et al. 2007b, Langella 2009e): A Grid-wide mechanism for maintaining a federated trust fabric of the VO's certification authorities. GTS also provides services for the provisioning of the associated trust configuration data, which allows Grid services and users to make authentication decisions against the most up-to-date security information.

Grid Grouper (Langella et al. 2008, Langella 2009d): A group-based authorization solution for the Grid, which enables services and applications to enforce authorization policy based on membership to VO-wide groups.

Credential Delegation Service (CDS) (Langella 2009b): A Grid service that enables users/services (delegator) to delegate their Grid credentials to other users/services (delegatee) such that the delegatee(s) may act on the delegator's behalf.

Web Single Sign On (WebSSO) (Garmilla 2009): This provides a comprehensive, Single Sign On (SSO) solution for web-browser clients and web-applications using GAARDS.

Common Security Module (CSM) (caBIG® 2009b): This provides a centralized approach to managing and enforcing access control policies.

GAARDS User Interface: Comprehensive graphical user interface to administer and to interact with the GAARDS security services.

In order for users/applications to communicate securely with services, they must be able to authenticate with an X.509 credential, that is, an X.509 public key certificate with its associated private key. Users with accounts with Dorian may request an X.509 credential from that service. Dorian provides two methods to register for a Grid user account: (1) the user can register directly with Dorian, or (2) they can register indirectly via their existing user account with the identity management system of their organization. In order to obtain X.509 credentials via an existing user account, a Dorian administrator must register the organization issuing the account as a Trusted Identity Provider. Users not affiliated with an existing identity provider, can register directly with Dorian. However, it is anticipated that most users will be able to use

their existing local credentials to obtain X.509 credentials. For Dorian to issue X.509 credentials, it requires proof that local authentication succeeded, which is asserted through a digitally signed SAML authentication statement. The GAARDS Authentication Service provides a framework for existing credential providers to issue SAML assertions to Dorian. The Authentication Service also provides a uniform authentication interface upon which applications can be built. Figure 16.1 illustrates the process to obtain Grid credentials. The user/application first authenticates with the appropriate local credential provider via the Authentication Service and obtains a SAML assertion as proof of successful authentication. With this proof the user can obtain X.509 credentials from Dorian. Assuming the local credential provider is registered as a trusted identity provider and that the user's account is in good standing, Dorian will issue X.509 credentials to the user. If a user is registered directly with Dorian, the user may contact Dorian directly to obtain Grid credentials.

After users have obtained X.509 credentials from Dorian, they are enabled to securely communicate with the Grid services. All secure communication is mutually authenticated and is established through the SSL/TLS protocol. As part of the authentication's validation process, it is checked that all X.509 credentials are issued by a trusted CA (e.g., Dorian). The GTS maintains the federated trust fabric

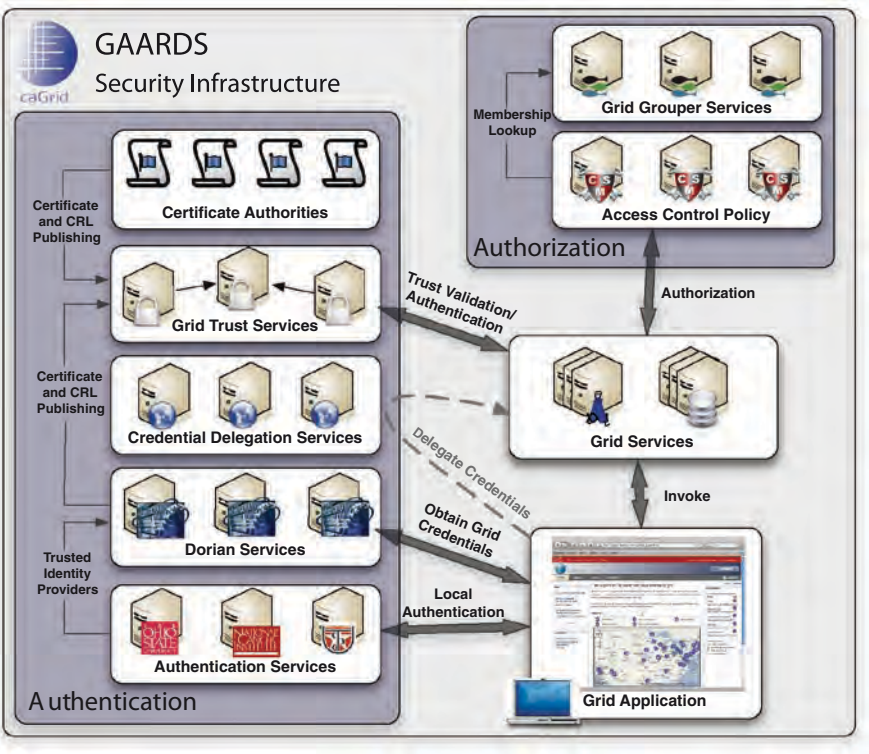


Fig. 16.1 GAARDS security infrastructure

of all the trusted digital signers in the collaboration. Credential providers such as Dorian and other Grid CAs are registered as trusted digital signers and updated trust information is regularly published to the GTS. As such Grid services validate credentials against the trusted digital signers registered with the GTS.

After a user has been authenticated to a service, that service determines if the user is authorized to perform the desired operation. The access control enforcement to resources that are made available through the Grid is controlled locally by those resources. The Grid services have many configurable options to perform authorization. GAARDS provides two approaches that can either be used independently or together. The first approach is group-based authorization provided by the Grid Grouper service. In this approach, Grid services and applications enforce authorization policy based on membership to Grid-level groups. Assuming the groups are provisioned by Grid Grouper, services can determine whether or not a caller is authorized based on certain group membership. The second approach is user-resource-operation authorization provided by the CSM. In this approach, Grid services ask CSM whether a user can perform a given operation on a specified resource. Based on the access control policy maintained in CSM, CSM decides whether or not a user is authorized. In Fig. 16.1, the Grid services defer the authorization to CSM. CSM can also enforce group-based access control policy by asking Grid Grouper whether the caller is a member of the groups specified in the policy. Note that, in addition to these two approaches, other authorization mechanisms (e.g., user-developed authorization or other community provided software) can be deployed in conjunction with the GAARDS authentication/trust infrastructure.

In addition to the components reviewed in the above use case, GAARDS provides three other components:

1. *The Credential Delegation Service* (CDS) allows users and services to delegate their credentials to other users or services such that they may operate on their behalf. Delegation plays an important role in being able to support things like workflow and federated queries, wherein infrastructure services invoke other services on behalf of a user.
2. *The Web Single Sign On Framework* integrates the GAARDS infrastructure and Grid security with web applications, enabling single sign on across web applications, allowing users to securely invoke Grid services through web applications.
3. *The GAARDS User Interface* provides a complete graphical user interfaces that allows administrators and users to manage and interact with any of the GAARDS security services.

16.4 Security Policy

In order to achieve and maintain confidence amongst the members of the federation, a commonly agreed, well-defined security policy is required. Given that a federated authentication model is being employed, a policy is required to govern the authentication

process. The National Institute of Standards and Technology (NIST) has developed the Electronic Authentication Guideline (Burr et al. 2006) which are intended for federal agencies implementing electronic authentication. The Electronic Authentication Guideline defines four levels of authentication, Levels 1–4, in terms of the consequences of the authentication errors and misuse of credentials. Level 1 is the lowest assurance and Level 4 is the highest; with each higher level of assurance the confidence in the authentication increases. This guideline defines the technical requirements for each of four levels of assurance in the areas of identity proofing, registration, tokens, authentication protocols, and related assertions:

Level 1. Authentication using username and password, enforcement of password security requirements, no identity vetting.
Level 2. Authentication using username and password, stricter password security requirements, in person identity vetting with government id or electronic identity vetting via government id number and financial records.
Level 3. Authentication via proving possession of cryptographic key or token, in person identity vetting with government id or electronic identity vetting via government id number and financial records.
Level 4. Authentication by proving possession of hard cryptographic key or token, in person identity vetting with government id and additional form of acceptable identification.

The Electronic Authentication Guideline is being adopted by federations such as InCommon (InCommon 2009) and caBIG®.

As mentioned previously, access control (or authorization) is locally enforced, giving service owners the right to determine who has access to their resources. Therefore authorization policy is evaluated locally and policy at the federation level is not needed (other than being soundly anchored on a common authentication mechanism). That being said, services such as Grid Grouper may optionally be operated and shared by the community, such that group membership can be defined on the VO-level and used by the local authorization policy at the discretion of the resource owner. In such cases, operational procedures are policies that are needed for governing the community operated services.

16.5 Deployment

In this section, we provide details on how the policy presented in the last section is enforced and how the GAARDS infrastructure is deployed in a production environment. In Grid environments users, clients, and services authenticate with one another using X.509 credentials. From the perspective of the Grid, certification authorities are credential providers for users and services. Dorian enables existing identity providers to be integrated into the Grid, by acting as a CA for those credential providers. Dorian enables existing users of the identity providers registered with it, to use their existing credentials to access the Grid. Figure 16.2 illustrates

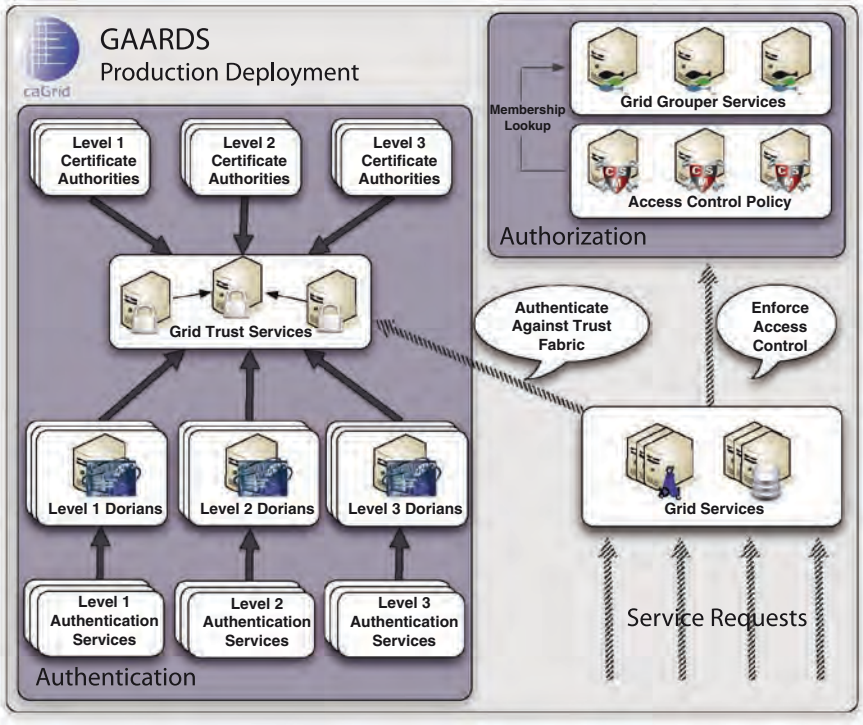


Fig. 16.2 GAARDS production deployment

how GAARDS can be deployed in a production environment in compliance with the Electronic Authentication Guideline. In this deployment we show Level 1, Level 2, and Level 3, which are the current levels targeted by the GAARDS Infrastructure. Certification authorities are registered with the GTS at the Electronic Authentication Guideline level of assurance that they comply with. In the diagram, we show two types of CAs: (1) Dorian and (2) Traditional CAs (VeriSign, Entrust, etc.). Authentication Services, representing organizational identity providers, are classified based on the level of assurance to which they comply. At a minimum, a Dorian service is deployed for each level of assurance, although multiple Dorians may be operated for a single level of assurance. Each Authentication Service is registered to the Dorian service that complies to the equal level of assurance. For example, Level 1 Authentication Services are registered to level 1 Dorian(s), level 2 Authentication Services are registered to level 2 Dorian(s), and level 3 Authentication Services are registered to level 3 Dorian(s). Each Dorian is registered with the GTS as a trusted CA at the level of assurance with which it complies.

Traditional CAs such as VeriSign and Entrust can also act as credential providers in the Grid. These certificate authorities are registered directly with the GTS at the level of assurance with which they comply.

When clients make requests to secure services, both sides must authenticate themselves using their X.509 credential. The authentication's validation process leverages the GTS to ensure that the presented certificates were issued and signed by a CA at the level of assurance that the service requires. For example, a service may specify that it will accept level 2 and level 3 credentials. Since all trusted CAs are registered with the GTS, the service needs to verify that a certificate presented is signed by a level 2 or level 3 CA that is registered to the GTS. If the certificate presented is signed by a CA that is not in the GTS or is signed by a CA not registered at level 2 or level 3 (i.e., level 1), the service request is rejected. Note in this scenario, no discussion of policy adherence is specified, and it is assumed that appropriate mechanism to verify and assure compliance to a particular level of assurance (e.g. service level agreements, memorandums of understanding, etc) is present; only the technological aspects of the deployment are discussed.

Once a client has authenticated, they have proven who they are to the service with the appropriate authentication credentials. The service must then determine whether or not to grant access to the requested resources based on the client's authenticated identity. As mentioned previously, access control policy is enforced locally; services may leverage GAARDS components such as Grid Grouper or the CSM to enforce access control or integrate their own mechanism. Note that Grid Grouper can be leveraged either locally where the local service provider operates their own Grid Grouper, or globally where the community or a subset of the community operates a Grid Grouper.

16.6 Conclusion

To provide informatics collaborations with a cross-organizational infrastructure that enables the secure sharing of data and results is a big challenge. It requires common protocols, services, and agreed upon policies. In this chapter, we presented a solution that meets those requirements: a set of implementations for infrastructure services and libraries, application programming interfaces (APIs) and tools to construct client and service applications that are tailored to work securely within that infrastructure. Furthermore, the solution includes the components that allow for a modular integration with different identity providers to leverage the organizations' existing identity management. Lastly, the infrastructure includes services and tools that allow one to define the trust fabric that pulls the different collaborating partners together into a single Virtual Organization with a well-defined and enforced trust-policy.

References

Burr W, Dodson D, Polk W (2006) Electronic authentication guideline, N.I.o.S.a. technology

caBIG® (2009b) Common security module (CSM) http://ncicb.nci.nih.gov/NCICB/infrastructure/cacore_overview/csm. Accessed 7 July 2009

caBIG® (2009a) Cancer biomedical informatics grid. https://cabig.nci.nih.gov/. Accessed 7 July 2009
caGrid (2009) caGrid community site. https://www.cagrid.org/. Accessed 7 July 2009
Garmilla S (2009) caGrid Web single sign on. http://www.cagrid.org/display/websso/Home. Accessed 7 July 2009
Globus (2009) The globus toolkit. http://www.globus.org. Accessed 7 July 2009
InCommon. (2009) InCommon. http://www.incommonfederation.org/. Accessed 7 July 2009
Langella S (2009a) Authentication service. http://www.cagrid.org/display/authenticationservice/Home. Accessed 7 July 2009
Langella S (2009b) Credential delegation service (CDS). http://www.cagrid.org/display/cds/Home. Accessed 7 July 2009
Langella S (2009c) Dorian. http://www.cagrid.org/display/dorian/Home. Accessed 7 July 2009
Langella S (2009d) Grid grouper. http://www.cagrid.org/display/gridgrouper/Home. Accessed 7 July 2009
Langella S (2009e) Grid trust service (GTS). http://www.cagrid.org/display/gts/Home. Accessed 7 July 2009
Langella S, Oster S, Hastings S, Siebenlist F, Kurc T, Saltz J (2006) Dorian: grid service infrastructure for identity management and federation**.** The 19th IEEE symposium on computer-based medical systems, pp 756–761
Langella S, Oster S, Hastings SL, Siebenlist F, Phillips J, Ervin DW, Permar JD, Kurc TM, Saltz JH (2007a) The cancer biomedical informatics grid (caBIG) security infrastructure. Proceedings of the 2007 AMIA annual symposium
Langella S, Oster S, Hastings S, Siebenlist F, Kurc T, Saltz J (2007b) Enabling the provisioning and management of a federated grid trust fabric. 6th annual PKI R&D workshop, Gaithersburg, MD
Langella S, Hastings S, Oster S, Pan T, Sharma A, Permar J, Ervin D, Cambazoglu B, Kurc T, Saltz J (2008) Sharing data and analytical resources securely in a biomedical research grid environment. J Am Med Inform Assoc 15(3):363–373
Oster S, Langella S, Hastings S, Ervin D, Madduri R, Phillips J, Kurc T, Siebenlist F, Covitz P, Shanbhag K, Foster I, Saltz J (2007) caGrid 1.0: an enterprise grid infrastructure for biomedical research. J Am Med Inform Assoc 15(2):138–149

Chapter 17
Caching and Visualizing Statistical Analyses

Roger D. Peng and Duncan Temple Lang

Abstract We present the cacher and CodeDepends packages for R, which provide tools for (1) caching and analyzing the code for statistical analyses and (2) distributing these analyses to others in an efficient manner over the Web. The cacher package takes objects created by evaluating R expressions and stores them in key-value databases. These databases of cached objects can subsequently be assembled into "cache packages" for distribution over the Web. The cacher package also provides tools to help readers examine the data and code in a statistical analysis and reproduce, modify, or improve upon the results. In addition, readers can easily conduct alternate analyses of the data. The CodeDepends package provides complementary tools for analyzing and visualizing the code for a statistical analysis and this functionality has been integrated into the cacher package. In this chapter, we describe the cacher and CodeDepends packages and provide examples of how they can be used for reproducible research.

17.1 Introduction

The replication of scientific findings using independent investigators, methods, data, equipment, and protocols is the standard by which scientific claims are evaluated. However, in many fields of study, there are examples of scientific investigations that cannot be fully replicated (see Chap. 8). Common reasons for a lack of replicability include a lack of time or resources. When scientific studies cannot be replicated, there is a need for a minimum standard that can fill the void between full replication and nothing. One candidate for this minimum standard is reproducibility, which requires that datasets and computer code implementing analyses be made available to others for verifying published results and conducting alternative analyses.

R.D. Peng (✉)
Department of Biostatistics, Johns Hopkins Bloomberg School of Public Health,
615 North Wolfe Street, Baltimore, MD 21205, USA
e-mail: rpeng@jhsph.edu

M.F. Ochs et al. (eds.), *Biomedical Informatics for Cancer Research*,
DOI 10.1007/978-1-4419-5714-6_17, © Springer Science+Business Media, LLC 2010

The need for publishing reproducible results is increasing for a number of reasons. Investigators are more frequently examining weak associations and complex interactions for which the data contain relatively little information. New technologies allow scientists in all areas to compile complex high-dimensional databases, and the ubiquity of powerful statistical and computing capabilities allows investigators to explore those databases and identify associations of potential interest. However, with the increase in data and computing power comes a greater potential for identifying spurious associations. In addition to these developments, recent reports of fraudulent research being published in the biomedical literature have highlighted the need for reproducibility in biomedical studies and have invited the attention of the major medical journals (e.g., Laine et al. 2007). Even without the presence of deliberate fraud, it should be noted that as analyses become more complicated, the possibility of inadvertent errors resulting in misleading findings looms large. In the example of Baggerly et al. (2005), the errors discovered were not necessarily simple or obvious and the examination of the problem itself required a sophisticated analysis. Misunderstandings about commonly used software can also lead to problems, particularly when such software is applied to situations not originally imagined.

While many might agree with the benefits of disseminating reproducible results, there is unfortunately a general lack of infrastructure for supporting such endeavors. Investigators who are willing to make their results reproducible are confronted with a number of barriers, one of which is the need to distribute, and make available for an indefinite amount of time, the supplementary materials required for reproducing the results. Readers who are interested in reproducing the results of others often need to expend substantial effort to gather the materials and study the statistical analysis code. There is currently a need and opportunity for the development of software to more efficiently connect authors and readers so that results can be reproduced and science can be advanced.

The distribution of reproducible results is a problem for which the solution varies depending on the complexity of the research. Small investigations involving moderately sized datasets and standard computational techniques can be archived and distributed in their entirety. Readers can subsequently study and rerun the entire analysis from start to finish to see if they can obtain the same results as the authors. Complex investigations involving large or multiple linked datasets and sophisticated statistical computations will be more difficult for readers to reproduce because of the resources and time required for running the analysis. In such a situation, a method is needed to give readers without equivalent resources the ability to conduct an initial examination of the details of the investigation and to partially reproduce or verify the results. In addition, complex statistical analyses will typically involve complex statistical code whose details may be difficult to understand upon first glance. Software that can help to visualize the analysis code itself can be useful to readers for understanding the flow of the analysis and for identifying potential points of interest.

A framework in which reproducible results can be distributed using cached computations is described in Peng and Eckel (2009). Cached computations are results that are stored in a database as an analysis is being conducted. These stored

results can be distributed via Web sites or central repositories so that others may explore the datasets and computer code for a given scientific investigation.

We have developed some tools for assisting authors and researchers in conducting reproducible research. In this chapter, we describe the cacher and CodeDepends packages for the R system. For authors, these packages provide tools for caching statistical analyses and for distributing these analyses to others in an efficient manner. For readers, these packages provide tools for visualizing the code and the results of a statistical analysis.

17.2 Description of Software

The cacher package is available from the Comprehensive R Archive Network (http://cran.r-project.org/). The CodeDepends package is available from the Omegahat project (http://www.omegahat.org/CodeDepends/). The Rgraphviz package (also available from CRAN) is required to use some of the functionality of the CodeDepends package described here. The cacher package is written primarily in R with a few components written in C. The CodeDepends package is written entirely in R. Both packages are licensed on the GNU GPL version 2 or higher.

The cacher package provides interfaces for two types of users. The first type consists of authors of statistical analyses who wish to cache their analyses in a database and distribute the cached analysis to others. The second type of user consists of readers who wish to obtain cached analyses over the Web and explore the data and code in those analyses in an efficient manner. In this chapter, we give a brief overview of the capabilities of both packages. Complete information about the design of the cacher package can be found in Peng (2008).

The primary function in the cacher package for authors of statistical analyses is the `cacher` function, which takes the name of an R source file as its first argument. This should be a standard source file containing R code to be evaluated and cached. The remaining two arguments to `cacher` specify the location of the cache directory (the default is `.cache`). Optionally, the log file can be specified where messages on the progress of `cacher` will be printed. The simplest invocation of `cacher` is

```
library(cacher)
cacher("myanalysis.R")
```

where "myanalysis.R" is the name of an R source file. The basic procedure of `cacher` is to read each R expression in the source file, evaluate it, cache the results to a key-value database, and then move to the next expression, until the end of the file is reached. More specifically, `cacher` will

1. Parse the R source file
2. Create the necessary cache directories and subdirectories

3. Set various configuration variables and hook functions for plotting
4. Copy the source file to the cache directory
5. Cycle through each expression in the source file:

   (a) If an expression has never been evaluated, evaluate it and store any resulting R objects in the cache database
   (b) If a cached result exists, lazy-load the results from the cache database and move to the next expression
   (c) If an expression does not create any R objects (i.e., there is nothing to cache), add the expression to the list of expressions where evaluation needs to be forced
   (d) Write out metadata for this expression to the metadata file

If a source file needs to be executed multiple times (e.g., because of revisions), cached results from previous runs can be used in place of actual evaluation in order to minimize the total time for evaluation. However, if the code changes, then some parts will need to be reevaluated. In order to assess changes to the R code, the cacher function creates a unique identifier for each expression in an R code file by taking the SHA-1 digest of the expression, the expression history (i.e., expressions preceding the current expression), and the name of the source file. For the first expression, the expression history is of length zero. Therefore, if the code in the source file changes, the digest of the expression will also change. Using the expression history to identify individual R expressions is a way to prevent expressions such as

```
x <- 1
y <- x^2
x <- 2
y <- x^2
```

from being inappropriately loaded from the cache. In this case, the expression `y <- x^2` appears twice, but the value of `x` changes in between. An expression such as this one may appear multiple times in a source file and we do not want to load the same value for `y` every time since the value of `x` may be changing. Using the expression history can uniquely identify each occurrence of a duplicate expression. The CodeDepends package has tools that can also keep track of the sequence of expressions and determine that the value of `x` has changed since `y` was last defined.

17.2.1 Distributing Cached Analyses

Authors who wish to distribute a cached statistical analysis over the Web and also have access to a local Webserver can post the cache directory on the Webserver so that others can download the materials using the `clonecache` function. All that is required is for the user to copy the directory to a location on the Webserver that is visible to outside users. We have placed a number of example analyses on the Web site of the Reproducible Research Archive (http://penguin.biostat.jhsph.edu), which is currently under development and is hosted and supported by the Department

of Biostatistics at Johns Hopkins University. While this Archive site is currently the only Web site hosting cache packages, our long-term goal is to develop multiple such sites to facilitate the distribution of packages, not unlike the network of mirrors that make up CRAN. Such a network of distribution sites would also mitigate the risk of any single site failing or no longer being supported. As an alternative to using a central repository, an author can use the `cachepackage()` function which creates a zip file of the entire cache. This zip file could then be distributed to others (e.g., via email or the Web) who can subsequently unzip the file and explore the contents of the cache using the functions described below.

The primary function for downloading a cached analysis is the `clonecache` function. The user can pass to `clonecache` the URL of the directory containing a cached analysis. Given a URL, `clonecache` creates a cache directory on the user's local machine and downloads the source files and metadata from the remote machine. The `clonecache` function also takes an ID string that can be used to retrieve analyses stored on the Archive Web site. By default, `clonecache` does not download any of the database files since these could be very large and the user may not be interested in every R object in the analysis. Rather, these database files are downloaded as needed when the users explore a cached analysis. In order to force the downloading of all database objects when initially cloning, the user needs to set the option `all.files = TRUE` when calling `clonecache`. Once an analysis is cloned, the functions described in the following section can be used to explore the code and data objects in the analysis.

17.2.2 Exploring and Visualizing Cached Analyses

The cacher package provides some basic tools to allow users to interact with the code and data provided in a cached analysis. The primary functions making up the user interface for readers wishing to explore a cached analysis written by someone else are:

- `showfiles`: Show what source files are available in the cache to be examined by the user. If the author of the package cached analyses from multiple source files, then this function can be used to determine which analysis should be examined. One can switch between different source files by calling the `sourcefile` function.
- `sourcefile`: Get or set the current source file for analysis.
- `code`: Show the expressions for a given source file. By default, code shows all expressions in a file in a one-line abbreviated form along with their expression sequence numbers. To see each expression in its entirety, the argument `full = TRUE` must be set.
- `showcode`: Show the original source file in the pager, which can be useful if one is interested in seeing any comments.
- `loadcache`: Lazy-load cached computation databases into an environment. This function takes a numeric vector of expression sequence numbers and loads objects associated with those expressions in the order that the expressions are

specified. Once a cache database is lazy loaded, the object names appear in the environment into which the database was loaded, but they do not occupy any memory until they are first accessed. If `loadcache` is used to load objects from a remote cache, then the corresponding database files will be downloaded on the object's first access.

- `runcode`: This function takes as input a numeric vector of expression sequence numbers that executes the code in those expressions. Each expression is evaluated in the order in which it appears in the input vector. By default, if a cached computation database is associated with an expression, then the database is lazy loaded via `loadcache` rather than executed. In order to force evaluation of code in an expression, one needs to set `forceAll = TRUE` when calling runcode. If an error occurs when executing the code in an expression, a message is printed to the console indicating the error and the expression is skipped. While the runcode function can be used to evaluate individual expressions, the results of such evaluation may not be correct if the dependent expressions have not previously been evaluated. At this point in development of the cacher package, reproducible results for a specific expression in an analysis can only be obtained by evaluating all of the expressions in order up to that expression. The CodeDepends package has functions for tracking the dependencies of R objects in an analysis and we will be working to integrate that functionality into the cacher package in a future release.
- `graphcode`: This function reads the source code for the cached analysis and creates a directed graph showing the relationships between the R objects created in the analysis and how they are used in defining each other. The `graphcode` function uses the capabilities implemented in the CodeDepends package to statically examine code and compute the various dependencies. For the creation of the graph itself, the Rgraphviz package is required.
- `objectcode`: This function takes the name of an R object and returns the sequence of R expressions that leads to the creation of that object. It returns the indices of the sequence of R expressions which could subsequently be passed to a function like `runcode`. This function can be useful for identifying the code for reconstructing an R object without having to run an entire analysis, which may contain many unrelated parts.

17.2.3 Example

As an example of how the cacher and CodeDepends packages can be used, we present a brief statistical analysis of particulate matter (PM) air pollution and mortality data. The data that we use come from the National Morbidity, Mortality, and Air Pollution Study (NMMAPS), and details about the data can be obtained from the Internet-based Health and Air Pollution Surveillance System Web site (http://www.ihapss.jhsph.edu/). Information about the original study is presented in Samet et al. (2000). The analysis presented here estimates the short-term association between daily PM levels and daily mortality. Briefly, a Poisson generalized linear time series

model is fit to daily mortality count and PM data from the largest 20 cities in the USA, adjusting for other factors like temperature, humidity, and seasonal trends. The primary target of inference is the log relative risk associated with short-term changes in ambient PM levels for each of the 20 cities. Further details about the methodology used in the analysis can be found in Peng and Dominici (2008).

We start by downloading the analysis from the Reproducible Research Archive Web site using the `clonecache` function. Each analysis on the Archive Web site is assigned a unique 40 character ID string that can be passed to the `clonecache` function. Any unique prefix of this ID string can also be used, and typically 4–8 characters are enough.

```
> library(cacher)
> clonecache(id = "092d")
created cache directory '.cache'
```

By default, `clonecache` downloads the source files and various metadata files about the analysis but does not download any data files. We can see what source files are available in this cache by using the `showfiles` function. In this case, there is only one file and so we deisgnate that file as the "active" file via the sourcefile function.

```
> showfiles()
[1] "top20.R"
> sourcefile("top20.R")
```

In this case, because there is only one source file available in this package, there is no need to call the `sourcefile` function explicitly because that file will be used by default. For analyses involving multiple files, the `sourcefile` function needs to be called to indicate the file to be examined. To obtain a listing of the code in the analysis we can use the code function. By default, the code function shows an abbreviated one-line representation of each expression.

```
> code()
source file: top20.R
1  cities <- readLines("citylist.txt")
2  classes <- readLines("colClasses.txt")
3  vars <- c("date", "dow", "death",
4  data <- lapply(cities, function(city) {
5  names(data) <- cities
6  estimates <- sapply(data, function(city) {
7  effect <- weighted.mean(estimates[1,
8  stderr <- sqrt(1/sum(1/estimates[2,
```

The original source file for this analysis was called "top20.R" (shown at the top of the listing), and there are eight expressions in the analysis. To the left of each expression is its expression sequence number. We can get a quick "sense" of the analysis by calling `graphcode` to see how the various objects relate to each other. The graph produced by `graphcode` is shown in Fig. 17.1. From the graph, we can see clearly that a number of elements come together to form the "data" which leads to an object called "estimates." From the estimates, we obtain an effect and a standard error. Given this visualization of the analysis code itself, we can decide on which objects we might

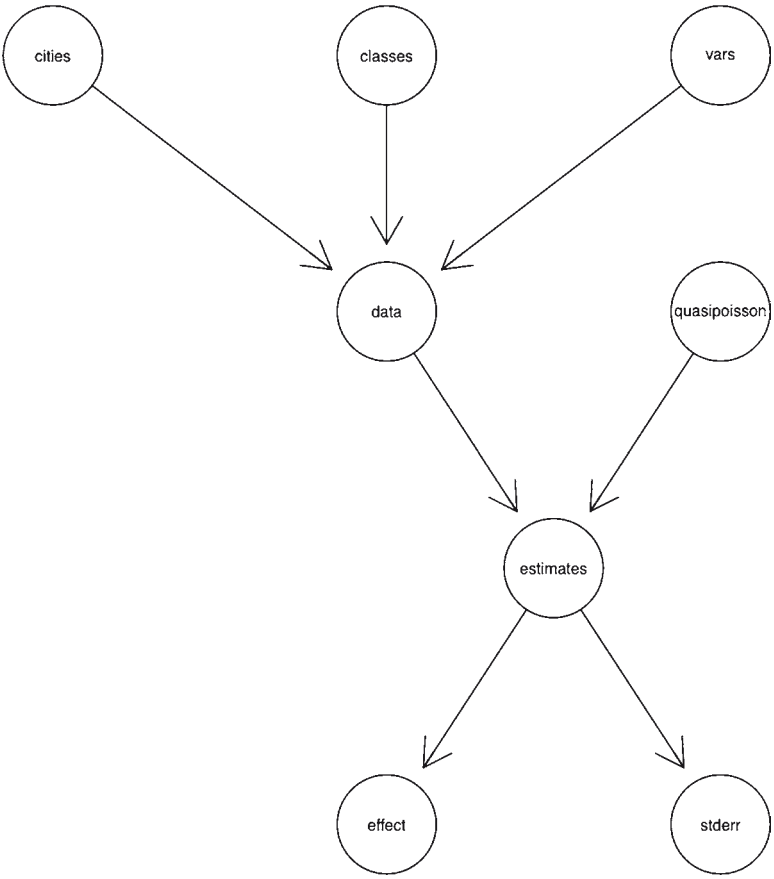


Fig. 17.1 Graph of statistical analysis code

wish to inspect more closely. For example, we can examine the code expressions that lead to the creation of the "data" object using the `objectcode` function.

```
> objectcode("data")
source file: top20.R
1  cities <- readLines("citylist.txt")
2  classes <- readLines("colClasses.txt")
3  vars <- c("date", "dow", "death", "tmpd", "rmtmpd", "dptp",
     "rmdptp", "l1pm10tmean")
4  data <- lapply(cities, function(city) {
          filename <- file.path("data", paste(city, "csv",
                                sep = "."))
          d0 <- read.csv(filename, colClasses = classes,
                         nrow = 5200)
          d0[, vars]
   })
5  names(data) <- cities
```

We can also inspect individual objects by printing them to the console. Here we examine the "cities" object which is simply a character vector that contains the

abbreviated names of the 20 cities used in the analysis. Before examining the object, we must load it from the cache with the `loadcache` function.

```
> loadcache()
> cities
/ transferring cache db file b8fd490bcf1d48cd06...
 [1] "la"   "ny"   "chic" "dlft" "hous" "phoe"
 [7] "staa" "sand" "miam" "det"  "seat" "sanb"
[13] "sanj" "minn" "rive" "phil" "atla" "oakl"
[19] "denv" "clev"
```

The `loadcache` function does not load the object directly, but rather "lazy loads" the object into the workspace. When the "cities" object is accessed for the first time, it is downloaded from the remote cache and then made available to the user. If the "verbose" option is set to `TRUE` for cacher (via the `setConfig` function), then the message "transferring cache db file" will be printed. This message indicates that an object needs to be downloaded from the remote cache. Once an object has been downloaded, it is available for future access and does not need to be downloaded again.

Finally, we can see the estimated effect pooled across the 20 cities.

```
> effect
/ transferring cache db file 584115c69e5e2a4ae5...
[1] 0.0002313219
```

This would translate into an approximately 0.23% increase in daily mortality associated with a ten-unit increase in ambient PM air pollution (ten units is a commonly used increment).

17.3 Summary

In this chapter, we have given a brief presentation of the capabilities of the cacher and CodeDepends packages. These packages provide functions for caching, distributing, exploring, and visualizing statistical analyses conducted in R. For the purposes of reproducible research, obtaining data and code as well as visualizing the flow of an analysis is critical. The cacher package comes with a vignette that contains more details of the functions in the package. Both packages are currently under active development, and the addition of features and capabilities is planned for future releases.

References

Baggerly K, Morris J, Edmonson S, Coombes K (2005) Signal in noise: evaluating reported reproducibility of serum proteomic tests for ovarian cancer. J Natl Cancer Inst 97:307–309

Laine C, Goodman SN, Griswold ME, Sox HC (2007) Reproducible research: moving toward research the public can really trust. Ann Intern Med 146:450–453

Peng RD (2008) Caching and distributing statistical analyses in R. J Stat Softw 26(7):1–24
Peng RD, Dominici F (2008) Statistical methods for environmental epidemiology in R: a case study in air pollution and health. Springer, New York
Peng RD, Eckel SP (2009) Distributed reproducible research using cached computations. IEEE Comput Sci Eng 11(1):28–34
Samet JM, Dominici F, Curriero F, Coursac I, Zeger SL (2000) Particulate air pollution and mortality: findings from 20 U.S. cities. N Engl J Med 343(24):1742–1757

Chapter 18
Familial Cancer Risk Assessment Using BayesMendel

Amanda Blackford and Giovanni Parmigiani

Abstract BayesMendel is an open-source software program created to provide individuals with personalized risk for carrying an inherited mutation of a cancer-causing gene and for developing cancer related to this mutation. BayesMendel is freely available as an R package (http://astor.som.jhmi.edu/BayesMendel/) or within the CancerGene software program (http://www4.utsouthwestern.edu/breasthealth/cagene/). Documentation for the R package is available on the web (http://astor.som.jhmi.edu/BayesMendel/BayesMendel.pdf) and questions can be directed to BayesMendel@jhu.edu.

18.1 Introduction

Cancer is caused by genetic alterations, many of which are changes in the DNA code (mutations). The discovery of genes mutated in cancers has provided key insights into the mechanisms underlying tumorigenesis and has proven useful for the design of targeted prevention and therapeutic approaches (Vogelstein and Kinzler 2004). Cancer-related mutations can be inherited (germline mutations) or can occur during one's lifetime (somatic mutations) (Vogelstein and Kinzler 1998). Identifying individuals at high risk of cancer because of inherited genetic susceptibility is critical in both prevention and treatment activities, through which probabilistic algorithms for risk evaluation are influencing the degree, quality, and cost of care received by millions of individuals in the United States. BayesMendel is a software package written in the statistical freeware language R (http://www.r-project.org) that provides an individual, referred to as the "counselee," an estimate of his or her probability of carrying an inherited mutation of a known cancer-related gene and the corresponding future risk of developing that cancer, based on the counselee's reported family history of cancer.

G. Parmigiani (✉)
Department of Biostatistics and Computational Biology, Dana-Farber Cancer Institute,
44 Binney Street, Boston, MA 02115, USA
e-mail: gp@jimmy.harvard.edu

M.F. Ochs et al. (eds.), *Biomedical Informatics for Cancer Research*,
DOI 10.1007/978-1-4419-5714-6_18, © Springer Science+Business Media, LLC 2010

Table 18.1 Summary of familial genes included in BayesMendel, by cancer site

Cancer	Model	Familial genes
Breast, Ovarian	BRCAPRO	BRCA1, BRCA2
Colorectal, Endometrial	MMRpro	MLH1, MSH2, MSH6
Pancreas	PancPRO	Hypothetical PANC gene
Melanoma	MelaPRO	CDKN2A/P16

18.1.1 Hereditary Cancer

Genetic research has identified a number of "susceptibility" genes, inherited mutations of which confer a significantly increased risk of one or more types of cancer (Foulkes and Hodgson 1998). These genetic effects are characterized by penetrance functions – probability distributions of developing cancer by age, conditional on a specific genetic variant. Operationally, we define a variant to be deleterious if the associated penetrance is increased compared to the penetrance of normal variants.

The models in BayesMendel currently focus on six major cancers: breast, ovarian, colorectal, endometrial, pancreatic, and melanoma. Each of these cancers is known to be more highly prevalent among individuals who carry one or more genetic susceptibility genes. The specific genes included in the BayesMendel models are highlighted in Table 18.1. More information about these cancers and their associated genes are described in Sect. 1.3.

18.1.2 Genetic Testing

The identification of a cancer gene allows for direct testing for deleterious mutations (Weber 1996; Ponder 1997). Myriad Genetics Laboratories provides testing for hereditary breast and colorectal cancers, among others, and by 2008 had tested over 250,000 individuals through 20,000 or more oncologists and obstetrician/gynecologists (Myriad Genetics 2008). Their two main products, BRACAnalysis and COLARIS, provide testing for the two main breast/ovarian cancer-causing genes, BRCA1 and BRCA2, and the hereditary nonpolyposis colorectal cancer genes, MLH1, MSH2 and MSH6. Many more patients are being counseled about whether to be genotyped, as cancer touches almost every family and it is not unusual to have more than one close relative who has cancer. Once a deleterious mutation is detected, steps can be undertaken to decrease the risk of mortality (Robson and Offit 2007), including increased screening (Syngal 1998) and, for BRCA genes, salpingo-oophorectomy, chemoprevention, and preventive mastectomy. It is now widely recognized that, when counseling individuals facing decisions about genetic testing, it is important to accurately evaluate the probabilities that he or she carries a deleterious mutation and that a mutation will be found if he or she is genotyped (Rimer and Glassman 1997; Petersen et al. 1999). Reliable strategies for individualized counseling enhance informed decision making, both about whether

or not to pursue testing, and about what to do with results (ASCO 2003). Currently, most cancer centers and many community hospitals have special clinics for families at high risk of cancer and staff genetic counselors. The demand for assessment of family histories has led to widespread use of statistical models to estimate mutation probabilities (Claus et al. 1991; Berry et al. 1997; Parmigiani et al. 1998). Model-based predictions are currently used in counseling and clinical activities, are included in materials distributed to patients considering genetic testing (Bluman et al. 1999), used for determining eligibility for screening and prevention studies (Hartman et al. 2004), and factored into coverage decisions by insurers (Domchek et al. 2003).

18.2 Model Development

18.2.1 Mendelian Modeling

Statistically, mutation prediction is inference on the genotype of an individual conditional on information about his/her disease history and his/her relatives' disease and genotype history (a pedigree). Two broad classes of modeling approaches have been used so far: "empirical" and "Mendelian." Empirical approaches model the conditional distribution of genotype given phenotype directly, by applying statistical or artificial intelligence techniques to collections of pedigrees from tested individuals. Candidate features are defined based on clinical and epidemiological expertise, and further selected for inclusion in models using variable selection techniques (Wijnen et al. 1998; Couch et al. 1997; Frank et al. 1998), classification trees (Hartge et al. 1999) or other techniques. By contrast, Mendelian models are built upon the conditional distributions of phenotypes given genotype (penetrance), and the marginal distributions of genotypes (prevalence). The probabilities required for counseling are then derived from these using Bayes' rule and Mendel's laws (Leal and Ott 1994; Szolovits and Pauker 1992). Validation studies indicate that Mendelian risk prediction models provide a well-founded approach to genetic counseling, and improved predictive performance compared to empirical approaches (Berry et al. 2002; Parmigiani et al. 2007).

This section outlines the general formalism by which the Mendelian prediction approach is implemented in all the models discussed in this chapter. Family phenotypes as a whole are referred to as family history F_{ij}, or pedigree data (following (Elston and Stewart 1971)), with the history of relative r denoted by H_{rij}. BayesMendel models have so far focused on one individual, who is termed the counselee. All probabilities refer to him/her. Let γ_0 be the vector of genotypes of the counselee at each of the genes considered. Each dimension represents a different locus. For example, a model considering BRCA1 and BRCA2 will have two dimensions. Let R be the family size, r be a relative in the family, and γ_r, for $r = 1, \ldots, R$ be the corresponding genotypes. Also, let $H_0, H_1, \ldots, H_R$ be the relevant phenotypes of the counselee and relatives. A mutation probability model provides the probability distribution of the counselee's genotypes given family history and pedigree structure, that is $p(\gamma_0 | H_0, H_1, \ldots, H_R)$.

Mendelian approaches can be formulated by combining Bayes' rule and Mendel's laws, along lines charted by the seminal work of Murphy and colleagues (Murphy and Mutalik 1969) and presented in more formal terms by Lange (Lange 1997). Formally, this involves an updating step and an integration step. The updating step is based on Bayes' rule:

$$p(\gamma_0|H_0,H_1,\ldots,H_R) = \frac{p(\gamma_0)p(H_0,H_1,\ldots,H_R \mid \gamma_0)}{\sum_{\text{all}\gamma_0\text{'s}} p(\gamma_0)p(H_0,H_1,\ldots,H_R|\gamma_0)}. \quad (1)$$

The prevalence $p(\gamma_0)$ can be taken as the carrier probability for an individual about whom nothing is known. This is then updated to incorporate information from the pedigree. The term $p(H_0, H_1, \ldots, H_R \mid \gamma_0)$ is the probability of the phenotypes for the whole pedigree given the genotype of the counselee. Because this is complex to evaluate directly, it is computed by conditioning on the entire set of family genotypes and then obtained using the law of total probability in the integration step:

$$p(H_0, H_1,..., H_R|\gamma_0) = \sum_{\text{all }\gamma_1,\ldots\gamma_R\text{'s}} p(H_0,\ldots, H_R|\gamma_0,\ldots,\gamma_R)p(\gamma_1,...,\gamma_R|\ \gamma_0). \quad (2)$$

The term $p(\gamma_1, \ldots, \gamma_R \mid \gamma_0)$ is known for all genotype configurations from Mendel's laws, as long as the mode of inheritance and the exact relationship of each relative to the counselee are known. In most models, the term $p(H_0, \ldots, H_R \mid \gamma_0, \ldots, \gamma_R)$ is further decomposed as in

$$p(H_0, H_1,\ldots, H_R|\gamma_0,\gamma_1,\ldots,\gamma_R) = \prod_i p(H_i \mid \gamma_i). \quad (3)$$

which assumes conditional independence of phenotypes given genotypes – this assumption may be relaxed in the future to account for shared environmental or behavioral risks. Together, these relationships connect the mutation probability with penetrance and prevalence information that can be abstracted from literature or estimated from family data, or both.

Given the array of interventions available for familial cancer syndromes, many counselees are reporting family members who have undergone medical interventions for prevention of cancer. Specifically, oophorectomy is increasingly common for women at high risk of breast and ovarian cancers. Approximately 50% of BRCA mutation carriers undergo oophorectomy (Katki 2007). Such medical interventions are accounted for by the Mendelian modeling approach because ignoring the intervention would incorrectly assume that penetrances are the same between those with and without the intervention.

18.2.2 Model Parameters

All of the models within the BayesMendel package follow the Mendelian approach outlined above. What distinguishes models from each other are the genes of interest and how mutations in these genes are associated with incidence of the cancer of

interest. The main inputs required for a BayesMendel model are the prevalance of deleterious mutations in the general population, and their penetrance. When we develop a new model to be included in the BayesMendel framework, we research and continually update the penetrance and prevalance estimates to be as current as possible (Chen and Parmigiani 2007; Chen et al. 2009), so that models constitute a comprehensive compendium of information relevant for genetic counseling. Other inputs, such as test results related to the cancer or the gene of interest, can be included and are specific to the model of interest. The open-source nature of the software allows users to customize the input and also create their own models for specific genes and diseases of interest. This is illustrated by Gonzalez et al., who adapted BayesMendel to create a model for identifying mutations on the DFNB1 locus for congenital cases of nonsyndromic autosomal recessive deafness (González et al. 2006).

18.2.3 Model Validation

When a new model is developed, it must be validated on a real set of data mimicking the conditions of clinical use. The data is generally from a set of individuals who have undergone germline testing for the genes of interest and have provided their relevant family history. The model being validated is applied to each individual, and the resulting predictions are compared to genetic test results. Standard evaluation criteria include concordance (area under the receiver-operating characteristic (ROC) curve) and calibration (ratio of observed number of mutations to the number of model-predicted mutations). In addition, sensitivity, specificity, and predictive values will be calculated to guide clinical use. The model's strengths and weaknesses will be assessed based on the outcomes of these validation measurements.

18.3 Existing Models

18.3.1 BRCAPRO

Deleterious mutations of BRCA1 and BRCA2, BRCA mutations for short, increase risk of breast and ovarian cancer (Struewing et al. 1997; Satagopan et al. 2001) and are associated with a large fraction of cases attributable to inherited susceptibility (Newman et al. 1997; Ford et al. 1998). Mutations in BRCA1 have been estimated to occur in 1 in 40 Ashkenazi Jewish, 1 in 400 non-Ashkenazi (Whittemore et al. 2004), and figures of the same general magnitude are thought to apply to BRCA2 (Risch et al. 2001; Antoniou et al. 2004). BRCA genes are polymorphic, and several hundreds of both deleterious and missense mutations have been identified (Biotechnology Information 2003; BIC 1997).

The probabilistic prediction algorithm BRCAPRO was developed for identifying individuals at high risk of breast cancer because of inherited genetic susceptibility. It calculates the probability that an individual carries a germline deleterious mutation of the BRCA1 and BRCA2 genes and, for individuals free of cancer, the probability that they will develop cancer in the future, in yearly or 5-year intervals.

During a counseling session, BRCAPRO receives information on the counselee's family history of breast and ovarian cancer in all relatives for whom it is available (Berry et al. 1997). History for members with a cancer diagnosis includes age at onset of breast and/or ovarian and/or contralateral breast cancers and molecular markers (ER, PR, CK5/6, CK14). For unaffected relatives, users input current age or age of death. For either unaffected or affected relatives, age at oophorectomy can be included. If any relatives have been previously tested for a BRCA1 or BRCA2 mutation, their test results can also be included in the model. The race/ethnicity and Ashkenazi Jewish status of the counselee is also considered (Chen et al. 2009).

The BRCAPRO model estimates the probabilities that an individual has a particular genotype profile based on the input family history. BRCAPRO considers three possible mutations (wild-type, heterozygous or homozygous) on two genes, BRCA1 and BRCA2, which yields nine genotype profiles. The model will return the probabilities of carrying each genotype, in addition to the four marginal probability estimates based on the input provided: probabilities of (1) not carrying a mutation in either BRCA gene, (2) being a BRCA1 mutation carrier, (3) being a BRCA2 mutation carrier, or (4) being a carrier of both BRCA genes. With these estimates, the model also provides future risks of developing breast and ovarian cancers in age intervals of the user's specification.

BRCAPRO has been extensively validated in both population-based and high-risk samples, and has been compared to other models that also provide risk estimates for carrying an inherited mutation of BRCA1 or BRCA2. One such study included 3,342 families, 1,668 population-based and 1,674 from high-risk counseling clinics. BRCAPRO was compared to six other models, and although all models performed similarly well, BRCAPRO was shown to discriminate carriers from noncarriers with the highest frequency (Parmigiani et al. 2007).

18.3.2 MMRpro

The Lynch Syndrome can be caused by a germline mutation of any one of five known DNA mismatch repair (MMR) genes: MSH2 (Leach et al. 1993), MLH1 (Papadopoulos et al. 1994; Bronner et al. 1994), and less frequently by PMS1, PMS2 (Nicolaides et al. 1994), and MSH6 (Miyaki et al. 1997). Inheritance is autosomal dominant. Carriers are characterized by early onset tumors that are more likely to develop in the proximal portion of the colon. For each of the genes, a large and growing number of deleterious mutations have been identified (Biotechnology Information 2003). Missense mutations are also common and their clinical significance is often uncertain.

The probabilistic prediction algorithm MMRpro was developed for identifying individuals at high risk of Lynch Syndrome because of inherited genetic susceptibility. It calculates the probability that an individual carries a germline deleterious mutation of the MMR genes, MLH1, MSH2, and MSH6. For individuals free of cancer, MMRpro also gives the probability that individuals will develop cancer in the future, in yearly or 5-year intervals.

During a counseling session, MMRpro receives information on the counselee's family history of colorectal and endometrial cancer in all relatives for whom it is available (Chen et al. 2006). History for members with a cancer diagnosis includes age at onset of colorectal and/or endometrial cancers, microsatellite instability (MSI) testing of the tumor and whether a colorectal tumor was found in the proximal or distal colon. For unaffected relatives, users input current age or age of death. If any relatives have been previously tested for an MMR gene mutation, their test results can also be included in the model.

The MMRpro model estimates the probabilities that an individual has a particular genotype profile based on the input family history. MMRpro considers three possible mutations (wild-type, heterozygous, or homozygous) on three genes, MLH1, MSH2, and MSH6, which yields 27 genotype profiles. The model will return the probabilities of carrying each genotype. These are typically summarized via the four marginal probability estimates: probabilities of (1) not carrying a mutation in any MMR gene, (2) being an MLH1 mutation carrier, (3) being an MSH2 mutation carrier, or (4) being an MSH6 carrier. With these estimates, the model also provides future risks of developing colorectal and endometrial cancers in age intervals of the user's specification.

MMRpro was validated on a set of 279 individuals from 226 families in three clinic-based groups. All individuals were tested for one of the MMR genes. Among the 279 individuals, there were 121 germline mutations found. The validation study compared MMRpro to two other prediction tools and demonstrated that MMRpro, both with and without MSI testing results included, discriminated carriers from noncarriers at the highest rate and was also the most well-calibrated (Chen et al. 2006).

18.3.3 PancPRO

Pancreatic cancer is the fourth leading cause of cancer death in the United Status, and 7–10% of patients have a family history of pancreatic cancer. Germline mutations in CDKN2A, PRSS1, BRCA2, and STK11 are known to increase this risk. However, when combined, these genetic factors account for less than 20% of the observed familiar aggregation, suggesting that additional susceptibility genes exist.

PancPRO is the first risk prediction model for familial pancreatic cancer and was developed to provide individuals with their probability of carrying a pancreatic cancer susceptibility gene and the absolute future risk of developing pancreatic cancer for user-specified age intervals (Wang et al. 2007). As input, PancPRO receives information on the counselee's family history of pancreatic cancer in all individuals

for whom it is available. History for affected family members includes the age of onset. For unaffected members, PancPRO takes in their current age or age of death. Because smoking is attributed to 25% of pancreatic cancers, future improvement to PancPRO includes adding smoking information on all family members.

PancPRO was validated on 6,134 individuals in 961 families who met the following criteria for inclusion in the validation study: alive and clinically free of pancreatic cancer at baseline; prospective follow-up data available; and not included in prior analyses used for model building (Wang et al. 2007). The study validated the absolute risk estimates from PancPRO and how well they discriminated between individuals who developed incident pancreatic caner during the follow-up period. PancPRO performed well in discriminating between those with and without incident pancreatic cancer and is useful in identifying high-risk individuals for ongoing and future early detection trials.

18.4 Software

18.4.1 BayesMendel R Package

The main framework of the BayesMendel models and all model development and improvements are housed in an add-on package of the statistical freeware R (http://www.r-project.org). BayesMendel is implemented in an object-oriented structure in the language R and distributed freely as an open-source library (Chen et al. 2004). In its first release, BayesMendel included the BRCAPRO and MMRpro models. PancPRO was added in 2007. R is a flexible environment that allows users to input their own genetic parameters, if desired, and run calculations for many families in an efficient way. The R package is regularly updated with improvements and additions to the models.

18.4.1.1 Functions and Data Sets

The BayesMendel R package library contains a series of functions and data sets. The functions can be grouped into two sets: (1) A core set of utilities and (2) A set of functions intended to be called by the typical user of BayesMendel. The core includes a set of functions (`peeling`, `calc.future.risk`, `CheckFamStructure`, `FamilyHistoryContributions`, `TestContributions`, `rescale`, `PostIntervention`, `MakePenetPostIntervention`, `findids`, `CensorAtIntervention`) that take the input family history information and perform the approach outlined in the Mendelian Modeling section above using the Elston–Stewart algorithm (Elston and Stewart 1971). These core functions are used by all of the specific models included in BayesMendel.

There are six main functions included in BayesMendel that are designed to be used directly. There are three main model evaluation functions: `brcapro`, `MMRpro`, and

`pancpro`, which take as input a family pedigree with all ages and diagnoses as discussed above, an indication of who the person being counseled is and other optional parameters. For each model, there is a function that allows users to set these optional input parameters, such as allele frequency, penetrance, sensitivity, and specificity of germline tests, and age intervals for computing risk estimates.

The data sets provided in BayesMendel include (1) penetrance values for each of the main models; (2) example breast, colorectal, and pancreas families; and (3) hazards of death excluding the cancer of interest for calculating future absolute risk of cancer. All of these data sets are in .RData format.

BayesMendel is implemented in an object-oriented fashion, which means that all data sets, numeric vectors or scalars, output from functions, models, plots, and lists are defined as objects and stored in the open workspace's memory. R objects can be easily saved, manipulated, and called. Two concepts are central to object-oriented programming: classes and methods (Team 2008). Classes define objects, and all objects in the BayesMendel library are members of the BayesMendel class. Methods are simply functions that are run on objects, such as the `brcapro` function in BayesMendel. In this case, `brcapro` is a method specific to the BayesMendel class. R contains generic functions, such as `plot`, which will look for the appropriate method to be applied to the input object based on its classes. In BayesMendel, objects output from the BayesMendel functions can be run through the function `plot` to produce a graphical display of the family pedigree and carrier probability results.

18.4.1.2 Input and Output

Each of the main BayesMendel R functions, `brcapro`, `MMRpro`, and `pancpro`, take three main objects as input: the family pedigree in data frame format, the ID of the counselee in scalar format, and the parameters to be used in the calculation in list format (output from functions `brcaparams`, `MMRparams` and `pancparams`). The functions return an object of class BayesMendel with the following components, referred to as slots: (1) `family`: the input family, (2) `posterior`: a matrix giving the joint probability of carrying mutations on the gene(s) of interest, (3) `probs`: a vector with the marginal probabilities of being a carrier or non-carrier of the gene(s) of interest, (4) `counselee.id`: The ID of the person for whom the calcualtion was performed, and (5) `loglik`: The total log-likelihood from the model. The functions also print the prospective absolute risks of developing the cancers of interest in age intervals of the user's choice.

18.4.2 CancerGene

CancerGene is a Windows-based, user-friendly, freely downloadable software program developed at the University of Texas Southwestern, Division of Surgical

Oncology (http://www4.utsouthwestern.edu/breasthealth/cagene/). It is run independently from R, but interfaces with R to calculate BayesMendel estimates. CancerGene provides genetic counselors, physicians and other health professionals with an easy-to-use framework for generating risk predictions for their patients. It allows users to input their family pedigrees in a point-and-click interface and then runs the BayesMendel R package in the background to compute the risk scores. There are currently over 3,000 registered users of CancerGene.

18.5 Examples

An example of a hypothetical family pedigree of a counselee who may present for genetic testing or counseling is shown in Fig. 18.1. The counselee is denoted by the arrow. Females are represented by circles and males by squares. Individuals unaffected by the cancers in question are denoted by empty symbols, and affected

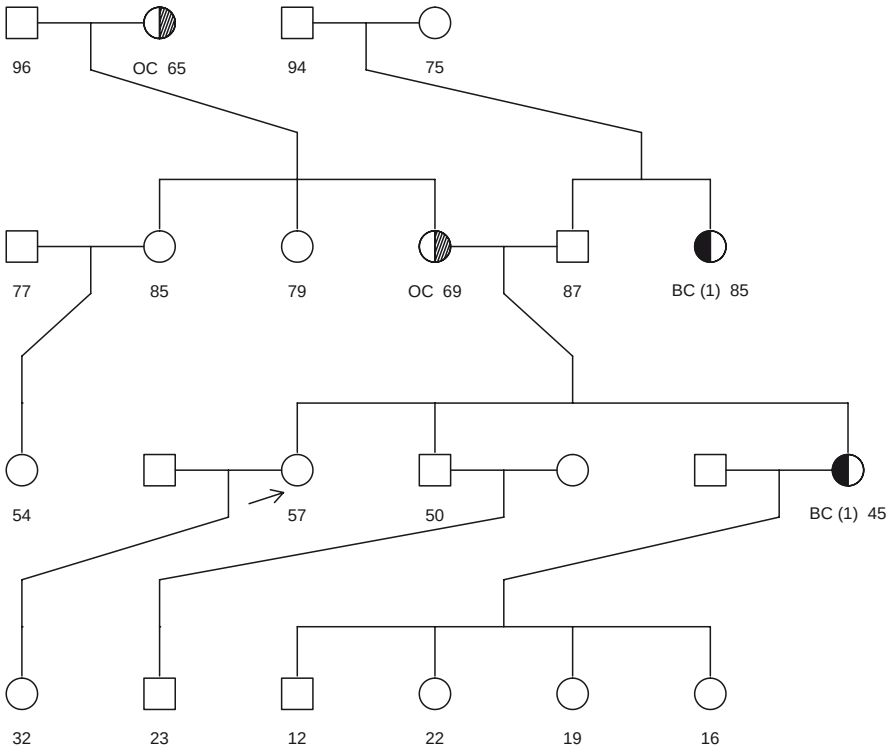


Fig. 18.1 Hypothetical family pedigree with a history of breast and ovarian cancers. *BC* breast cancer, *OC* ovarian cancer

members are either shaded black on the left for breast cancer and/or gray on the right for ovarian cancers. Ages of onset for affected and current age or age of death for unaffected members are listed below the symbols.

In this example, the counselee is a cancer-free, 57-year-old woman. Both her sister and maternal aunt had a unilateral breast cancer diagnosis at ages 45 and 85, respectively. Her mother had ovarian cancer at age 69 and maternal grandmother at age 65. Using the BRCAPRO program, the counselee's probability of carrying a BRCA mutation is 28% (14% risk of BRCA1 and 14% risk of BRCA2). Her risk of developing breast cancer by age 67 is 7.4%. Her risk of developing ovarian cancer by age 67 is 4.4%. This calculation assumes that the counselee is not of Ashkenazi Jewish descent.

If we change the counselee's mother to be a healthy 70-year-old woman, then the counselee's carrier probability decreases to 4%. Her risk of developing breast cancer by age 67 is now 3.7%. Because the mother is now healthy at age 70, this suggests that she is less likely to carry a BRCA mutation. If we switch the ovarian cancer diagnosis of the counselee's maternal grandmother to her paternal grandmother at age 60, then the counselee's carrier probability goes up to 8%. Her risk of breast cancer by age 67 is 4.4%. Although the cancer diagnosis was just swapped between grandmothers, the paternal aunt also has a breast cancer diagnosis, which suggests a lineage in the cancer diagnoses and that the paternal grandmother and aunt may have inherited mutations.

18.6 Discussion and Future Development

Family history is a highly efficient way of identifying individuals that are at high risk for cancer. Its role in cancer prevention is growing and will continue to grow with our ability to measure genomes inexpensively on a large number of individuals. The approaches we have developed for high-risk families can constitute the basis for a more extensive use of family information in personalized medicine.

A critical juncture concerns the beast approach for deploying prediction algorithms to the public. The approach described here is based on distributing tools to clinicians and counselors, who then use them directly. An alternative trend is to provide web-based services. From a computational standpoint, these have the advantage that the modelers retain full control over the calculations that are provided to the public. Upgrades are centralized and efficient. On the other hand it becomes more difficult to ensure that risk predictions are provided to the public in the context of a well-rounded interaction with a health-care professional. This is a concern motivated by ethical issues and by the challenges of communicating quantitative risk to the public in an effective way.

References

Antoniou AC, Pharoah PPD, Smith P, Easton DF (2004) The BOADICEA model of genetic susceptibility to breast and ovarian cancer. Br J Cancer 91(8):1580–1590

ASCO (2003) Policy statement update: Genetic testing for cancer susceptibility recommendations pertaining to clinical aspects of genetic testing for cancer susceptibility. J Clin Oncol 21:2397–2406

Berry DA, Parmigiani G, Sanchez J, Schildkraut J, Winer E (1997) Probability of carrying a mutation of breast-ovarian cancer gene BRCA1 based on family history. J Natl Cancer Inst 89:227–238

Berry DA, Iversen ES, Gudbjartsson DF, Hiller EH, Garber JE, Peshkin BN, Lerman C, Watson P, Lynch HT, Hilsenbeck SG, Rubinstein WS, Hughes KS, Parmigiani G (2002) BRCAPRO validation, sensitivity of genetic testing of BRCA1/BRCA2, and prevalence of other breast cancer susceptibility genes. J Clin Oncol 20(11):2701–2712

BIC (1997) National institutes of health, breast cancer information core: An open access on-line breast cancer mutation data base. http://www.nhgri.nih.gov/Intramural research/Lab transfer/Bic/

National Center for Biotechnology Information (2003) Online mendelian inheritance in man. http://www3.ncbi.nlm.nih.gov:80/Omim

Bluman LG, Rimer BK, Berry DA, Borstelmann N, Iglehart JD, Regan K, Schildkraut J, Winer EP (1999) Attitudes, knowledge, and risk perceptions of women with breast and/or ovarian cancer considering testing for BRCA1 and BRCA2. J Clin Oncol 17(3):1040–1046

Bronner CE, Baker SM, Morrison PT, Warren G, Smith LG, Lescoe MK, Kane M, Earabino C, Lipford J, Lindblom A et al (1994) Mutation in the DNA mismatch repair gene homologue hMLH1 is associated with hereditary non-polyposis colon cancer. Nature 368(6468):258–261

Chen S, Parmigiani G (2007) Meta-analysis of BRCA1 and BRCA2 penetrance. J Clin Oncol 25(11):1329–1333

Chen S, Wang W, Broman KW, Katki HA, Parmigiani G (2004) BayesMendel: an R environment for Mendelian risk prediction. Stat Appl Genet Mol Biol, 3(1):Article21

Chen S, Wang W, Lee S, Nafa K, Lee J, Romans K, Watson P, Gruber SB, Euhus D, Kinzler KW, Jass J, Gallinger S, Lindor NM, Casey G, Ellis N, Giardiello FM, Offit K, Parmigiani G, Colon Cancer Family Registry (2006) Prediction of germline mutations and cancer risk in the Lynch syndrome. JAMA 296(12):1479–1487

Chen S, Blackford A, Parmigiani G (2009) Tailoring BRCAPRO to Asian-Americans. J Clin Oncol 27:642–643

Claus EB, Risch N, Thompson WD (1991) Genetic analysis of breast cancer in the cancer and steroid hormone study. Am J Hum Genet 48:232–242

Couch FJ, DeShano ML, Blackwood MA, Calzone K, Stopfer J, Campeau L, Ganguly A, Rebbeck T, Weber BL, Jablon L, Cobleigh MA, Hoskins K, Garber JE (1997) BRCA1 mutations in women attending clinics that evaluate the risk of breast cancer. N Engl J Med 336:1409–15

Domchek SM, Eisen A, Calzone K, Stopfer J, Blackwood A, Weber BL (2003) Application of Breast Cancer Risk Prediction Models in Clinical Practice. J Clin Oncol 21(4):593–601

Elston RC, Stewart J (1971) A general model for the genetic analysis of pedigree data. Hum Hered 21:523–542

Ford D, Easton DF, Stratton M, Narod S, Goldgar D, Devilee P, Bishop DT et al (1998) Genetic heterogeneity and penetrance analysis of the BRCA1 and BRCA2 genes in breast cancer families. Am J Hum Genet 62:676–689

Foulkes WD, Hodgson SV (eds) (1998) Inherited susceptibility to cancer: clinical. Predictive and Ethical Perspectives. Cambridge University Press, Cambridge, UK

Frank T, Manley S, Olopade O, Cummings S, Garber J, Bernhardt B, Antman K et al (1998) Sequence analysis of BRCA1 and BRCA2: Correlation of mutations withfamily history and ovarian cancer risk. J Clin Oncol 16:2417–2425

González JR, Wang W, Ballana E, Estivill X (2006) A recessive mendelian model to predict carrier probabilities of DFNB1 for nonsyndromic deafness. Hum Mutat 27(11):1135–1142

Hartge P, Struewing JP, Wacholder S, Brody LC, Tucker MA (1999) The prevalence of common BRCA1 and BRCA2 mutations among Ashkenazi Jews. Am J Hum Genet 64:963–970

Hartman AR, Daniel BL, Kurian AW, Mills MA, Nowels KW, Dirbas FM, Kingham KE, Chun NM, Herfkens RJ, Ford JM, Plevritis SK (2004) Breast magnetic resonance image screening and ductal lavage in women at high genetic risk for breast carcinoma. Cancer 100(3):479–489

Katki HA (2007) Incorporating medical interventions into carrier probability estimation for genetic counseling. BMC Med Genet 8:13

Lange K (1997) Mathematical and statistical methods for genetic analysis. Springer, Berlin

Leach FS, Nicolaides NC, Papadopoulos N, Liu B, Jen J, Parsons R, Peltomaki P, Sistonen P, Aaltonen LA, Nystrom-Lahti M et al (1993) Mutations of a mutS homolog in hereditary non-polyposis colorectal cancer. Cell 75(6):1215–1225

Leal SM, Ott J (1994) A likelihood approach to calculating risk support interval. Am J Hum Genet 54(5):913–917

Miyaki M, Konishi M, Tanaka K, Kikuchi-Yanoshita R, Muraoka M, Yasuno M, Igari T, Koike M, Chiba M, Mori T (1997) Germline mutation of MSH6 as the cause of hereditary nonpolyposis colorectal cancer. Nat Genet 17(3):271–272

Murphy EA, Mutalik GS (1969) The application of Bayesian methods in genetic counseling. Hum Hered 19:126–151

Myriad Genetics Laboratories (2008) Myriad genetics 2008 annual report. Salt Lake City, UT

Newman B, Millikan RC, King M-C (1997) Genetic epidemiology of breast and ovarian cancers. Epidemiol Rev 19:69–79

Nicolaides NC, Papadopoulos N, Liu B, Wei YF, Carter KC, Ruben SM, Rosen CA, Haseltine WA, Fleischmann RD, Fraser CM et al (1994) Mutations of two PMS homologues in hereditary nonpolyposis colon cancer. Nature 371(6492):75–80

Papadopoulos N, Nicolaides NC, Wei YF, Ruben SM, Carter KC, Rosen CA, Haseltine WA, Fleischmann RD, Fraser CM, Adams MD et al (1994) Mutation of a mutL homolog in hereditary colon cancer. Science 263(5153):1625–1629

Parmigiani G, Berry D, Aguilar O (1998) Determining carrier probabilities for breast cancer-susceptibility genes BRCA1 and BRCA2. Am J Hum Genet 62(1):145–158

Parmigiani G, Chen S, Iversen ES, Friebel TM, Finkelstein DM, Anton-Culver H, Ziogas A, Weber BL, Eisen A, Malone KE, Daling JR, Hsu L, Ostrander EA, Peterson LE, Schildkraut JM, Isaacs C, Corio C, Leondaridis L, Tomlinson G, Amos CI, Strong LC, Berry DA, Weitzel JN, Sand S, Dutson D, Kerber R, Peshkin BN, Euhus DM (2007) Validity of models for predicting BRCA1 and BRCA2 mutations. Ann Intern Med 147(7):441–450

Petersen GM, Brensinger JD, Johnson KA, Giardiello FM (1999) Genetic testing and counseling for hereditary forms of colorectal cancer. Cancer 86(11S):2540–2550

Ponder B (1997) Genetic testing for cancer risk. Science 278(5340):1050–1054

Rimer BK, Glassman B (1997) Tailoring communications for primary care settings. Methods Inf Med 37(2):171–177

Risch HA, McLaughlin JR, Cole DE, Rosen B, Bradley L, Kwan E, Jack E, Vesprini DJ, Kuperstein G, Abrahamson JLA, Fan I, Wong B, Narod SA (2001) Prevalence and penetrance of germline brca1 and brca2 mutations in a population series of 649 women with ovarian cancer. Am J Hum Genet 68:700–710

Robson M, Offit K (2007) Clinical practice. Management of an inherited predisposition to breast cancer. N Engl J Med 357(2):154–162

Satagopan JM, Offit K, Foulkes W, Robson ME, Wacholder S, Eng CM, Karp SE, Begg CB (2001) The lifetime risks of breast cancer in ashkenazi jewish carriers of brca1 and brca2 mutations. Cancer Epidemiol Biomarkers Prev 10(5):467–473

Struewing JP, Hartge P, Wacholder S, Baker SM, Berlin M, McAdams M, Timmerman MM, Brody LC, Tucker MA (1997) The risk of cancer associated with specific mutations of BRCA1 and BRCA2 among Ashkenazi jews. N Engl J Med 336:1401

Syngal S (1998) Benefits of colonoscopic surveillance and prophylactic colectomy in patients with hereditary nonpolyposis colorectal cancer mutations. Ann Intern Med 129:787

Szolovits P, Pauker S (1992) Pedigree analysis for genetic counseling. In: Lun KC, Degoulet P, Piemme TE, Rienhoff O (eds) MEDINFO-92 proceedings of the seventh conference on medical informatics. Elsevier, New York, pp 679–683

RDC Team (2008) R language definition

Vogelstein B, Kinzler K (1998) The genetic basis of human cancer. McGraw-Hill, New York

Vogelstein B, Kinzler KW (2004) Cancer genes and the pathways they control. Nat Med 10(8):789–799

Wang K, Li M, Bucan M (2007). Pathway-based approaches for analysis of genomewide association studies. Am J Hum Genet 81(6) (Epub ahead of print)

Wang W, Niendorf K, Patel D, Blackford A, Marroni F, Sober A.J, Parmigiani G. and Tsao H. Estimating *CDKN2A* carrier probability and personalizing cancer risk assessments in hereditary melanoma using MelaPRO. Cancer Research; 70(2) January 15, 2010

Weber BL (1996) Genetic testing for breast cancer. Sci Med 3:12–21

Whittemore AS, Gong G, John EM, McGuire V, Li FP, Ostrow KL, Dicioccio R, Felberg A, West DW (2004) Prevalence of brca1 mutation carriers among U.S. non-Hispanic Whites. Cancer Epidemiol Biomarkers Prev 13(12):2078–2083

Wijnen JT, Vasen HFA, Khan PM, Zwinderman AH, van der Klift H, Mulder A, Tops C, Moller P, Fodde R (1998) Clinical findings with implications for genetic testing in families with clustering of colorectal cancer. N Engl J Med 339(8):511–518

Chapter 19
Interpreting and Comparing Clustering Experiments Through Graph Visualization and Ontology Statistical Enrichment with the ClutrFree Package

Ghislain Bidaut

Abstract In large-scale transcriptome analysis with DNA microarrays, experimentalists are typically using clustering or pattern recognition algorithms that group genes with similar expression profiles into clusters of biological significance, which aid in interpreting the data. The choice of clustering algorithm and their parameters is essential, since they have a large impact on the final results. However, no standards have been established with regard to these choices, leading to the development of bioinformatics methods to quantify their final impact on the analysis.

ClutrFree is a visualization software package that provides a solution to the problem of comparing multiple clustering experiments. It features a graphical user interface for cluster visualization and a method for organizing clusters into meaningful trees and highlighting stable features found in the data through a measure of persistence. It also allows for exploring gene lists within each cluster by measuring gene ontology enrichment and statistical significance.

Several data formats are supported (Multiple Experiment Viewer output and PattTools output), and a standard API is provided for developing plugin code for format extensions. This chapter presents ClutrFree features and shows examples of data analysis of phylogenomic profiles and of a microarray cancer dataset.

Availability: http://clutrfree.sourceforge.net
Links for documentation:

- User guides and documentation are available from http://clutrfree.sourceforge.net/
- ClutrFree test datasets are available from http://clutrfree.sourceforge.net/clutrfree-datasets/

G. Bidaut (✉)
Inserm, UMR891, CRCM, Integrative Bioinformatics, Marseille 13009, France
Institut Paoli-Calmettes, Marseille, 13009, France
e-mail: ghislain.bidaut@inserm.fr

M.F. Ochs et al. (eds.), *Biomedical Informatics for Cancer Research*,
DOI 10.1007/978-1-4419-5714-6_19, © Springer Science+Business Media, LLC 2010

19.1 ClutrFree Description and Typical Analysis Environment

19.1.1 Overview of ClutrFree

DNA microarrays have become a de facto standard for transcriptome activity exploration in several areas of biology. For instance, this technology has led to an established molecular classification of breast cancer (Sotiriou and Pusztai 2009) and has the potential of improving therapeutic strategies in personalized medicine (Sabatier et al. 2009). Public repositories such as the Gene Expression Omnibus (GEO, Barrett et al. 2007) have been widely adopted by researchers to host and share data in a standardized sharing format: the Minimum Information about a Microarray Experiment (MIAME) (Ball and Brazma 2006). This set of minimum information is aimed toward allowing repeated experimentation and meta-analysis on similar datasets – that is, datasets generated on similar biological questions. However, no standards have yet emerged in terms of analysis and, in particular, clustering, which is typically applied to extract patterns from DNA microarray datasets. While hierarchical clustering has been widely used, there exist many other choices in statistical and computer science literature [neural networks, support vector machines, Bayesian Decomposition – see Do and Choi (2008) for a review]. This has been underlined by software suites that natively implement many clustering/analysis techniques. Among them, the Multiple Experiment Viewer (MeV, Saeed et al. 2006) is a Java package that allows clustering DNA microarray data with multiple methods, including hierarchical clustering, self-organizing maps (SOM), neural networks, and others (see Chap. 15). A critical issue with this type of analysis is to make a rational choice for several parameters, which include the choice of clustering method and the choice of mathematical or statistical parameters for a given method [e.g., the number of patterns for Bayesian Decomposition (Bidaut et al. 2006) or *K*-means clustering (Quackenbush 2001)]. Typically, no guidance for a better algorithm choice or parameter choice regarding the data is provided. Nevertheless, the proper choice of such parameters is essential to extract the proper gene expression patterns and groups of genes from a large scale experiment, and techniques for providing such guidance for parameter optimization are needed.

Several strategies have been proposed to tackle this problem: A visualization approach to find the most appropriate clustering method to choose for an analysis has been proposed, based on detection of outliers and 3D visualization (Hibbs et al. 2005). The use of validation measures has also been proposed for choosing clustering strategies based on cluster consistency (Datta and Datta 2006).

The major focus of this chapter is to describe ClutrFree (Bidaut and Ochs 2004) software in order to aid in devising a clustering strategy for large datasets. The basic principle of ClutrFree is to run several clustering algorithms on a given dataset, or a given clustering algorithm with parameter variations, and compare the obtained clusters by creating a tree of consistently stable clusters. We will also review its intrinsic capabilities, including gene list visualization jointly with Gene Ontology (GO) enrichment analysis, and extraction of stable data (genes or experiments) that are consistently present in several clustering experiments, regardless of the parameters

used. Data analysis is demonstrated on two datasets, a bacterial phylogenomic profiles dataset analyzed with Bayesian Decomposition and a large-scale microarray dataset clustered with *K*-means.

19.1.2 Data Typically Analyzed

The data analyzed with ClutrFree are results of statistical clustering method(s) on DNA microarray data or phylogenomic profiles (Bidaut et al. 2005). Briefly, DNA microarray datasets are the results of measurement of RNA abundance across several experiments. For instance, the Affymetrix HGU133Plus2.0 platform contains 54,615 spotted probes, corresponding to ~47,000 gene transcripts in the human genome. Once the microarray has been scanned, expression data is quantified and normalized. Standard algorithms, such as GCRMA,[1] are typically applied here, leading to an $N \times M$ table describing the expression of N genes across M experiments. Data are then filtered (gene with low variance are removed) and clustered in suitable ways.

19.1.3 Clustering Methods

Many clustering methods exist, as widely described in the bioinformatics literature (for a review of feature selection techniques, see Boutros and Okey 2005). The most widely used clustering algorithm is hierarchical clustering with average linkage and Euclidian distance. Its basic principle is recursive profile aggregation to create a dendrogram grouping genes with similar expression profiles. Experiments can also be clustered the same way to group conditions with similar expression across genes (for instance patients within the same molecular subgroup). Other methods include other variations of hierarchical clustering (maximum linkage, variation of distance measure), SOM, *K*-means or *K*-medians, Self-Organizing Tree Algorithm (SOTA), Bayesian Decomposition, and others. In the following paragraphs, we briefly describe two clustering methods that are used in the examples section.

19.1.3.1 *K*-means

K-means (or *K*-medians when median is used instead of mean) is an algorithm in four steps:

(a) *K* centroids are chosen randomly in the data. *K* is the predetermined number of clusters to find in the data and the initial parameter of the algorithm.

[1] Wu J, Rafael IR. gcrma: Background adjustment using sequence information. R package version 2.16.0.

(b) For each data point in a cluster, the nearest centroid is computed.
(c) Centroids are updated (mean of all gene profiles that belong to the same centroid).
(d) Repeat (b) and (c) until convergence.

This is a heuristic that does not guarantee convergence and whose results depend strongly upon the initial conditions, that is, the choice of the parameter *K*, and the initial locations of centroids, which makes it a case to use with ClutrFree for optimizing the choice of parameters. It has been implemented in many packages such as the Multiple Experiment Viewer (MeV), Bioconductor (Gentleman et al. 2004), Matlab©, and others. *K*-means++ (Arthur and Vassilvitskii 2007) is a variation of the initial algorithm that chooses automatically initial centroids with a high distance from each other. In most practical implementation, the risk of falling into a local maximum is reduced through data point exchange between clusters during the search.

19.1.3.2 Bayesian Decomposition

Bayesian Decomposition (BD) is a matrix factorization algorithm that retrieves simultaneously two matrices **A** and **P** from the data **D**, so that $\mathbf{D} = \mathbf{A} \cdot \mathbf{P} + \boldsymbol{\varepsilon}$, **D** being the initial data; **P**, a set of vectors representing patterns devised by BD; and **A**, a set of coefficients that describe pattern distribution in the data. Mathematical details are available here (Moloshok et al. 2002). A Gibbs sampler samples the solution space and minimizes the chi-square distance between data and the model $\mathbf{A} \cdot \mathbf{P}$ and operates on two stages: (a) the burn-in period during which the Gibbs sampler reaches an area of high probability in the sample space and equilibrates and (b) the sampling stage, during which the sampler is taking samples to draw **A** and **P** distributions, with means and *p*-values for each matrix element.

This is a heuristic that does not guarantee convergence and whose results depend strongly upon the initial choice of parameters and the number of rows in **P**. It has been implemented in the PattTools package, available from the Ochs Lab.[2]

19.1.4 Issues with Typical Visualization/Analysis Methods

Using one of these algorithms, the user will typically explore several clustering results obtained from several runs of various algorithms or the same algorithm with parameters variation (for instance, variations and number in *K*-means). This will generate multiple clustering results that need to be compared. However, typical visualization methods do not include the capability of using Gene Ontology annotations for assessing biological functions of multiple clusters obtained from

[2] http://www.cancerbiostats.onc.jhmi.edu/ochs.cfm

separate sources. This is generally done separately by using other programs, such as ErmineJ (Lee et al. 2005). Also, there are no built-in capabilities for comparing clusters using standard visualization tools.

19.1.5 Details on Tool Function

ClutrFree allows for comparison of several clustering algorithms, or analysis of results from a unique clustering algorithm that runs with multiple parameter sets (for instance, *K*-means run with *K* ranging from $K=3$ to $K=15$). It uses a mixture of visualization approaches and statistical measures to integrate results from annotation databases such as Gene Ontology or the Munich Information center for Protein Sequence (MIPS). ClutrFree functionalities are as follows:

(a) Loading multiple clustering experiments from the MeV or PattTools
(b) Visualization of average gene profiles from these clusters
(c) Computation of persistence, a measurement of cluster stability
(d) Computation of statistical enrichment for Gene Ontology or any hierarchical-based annotations

19.1.6 Description of Input/Output: Data Organization

ClutFree input data is a mixture of clustering outputs from several experiments. For instance, it allows the analysis of output from *K*-means with the number of clusters spanning from 5 to 10 on a dataset using MeV. Several plugins have been developed to handle multiple data formats and the user can develop his or her own plugin to handle newer formats if needed – an example of code is given in the documentation.

Data is basically organized in two matrices **A** and **P**. The **P** matrix (pattern matrix) contains the set of patterns found in the data. In the case of *K*-means, these patterns are the average profile of the groups of genes that belong to the same centroid. In the case of BD or principal component analysis (PCA), these patterns are the vectors on which data is projected.

The **A** matrix (distribution matrix) describes which genes are present in each cluster. In the case of *K*-means, each row of the **A** matrix represents a gene profile, each column a pattern, and the values are descriptors of a gene being present ($v=1$) or absent ($v=0$) in a given cluster. In BD or PCA, the **A** matrix quantifies the distribution of patterns in the data.

File system organization is the following (see documentation for more details). For each clustering run, a directory is created containing calculation results (for instance, *K*-means-*X*, with *X* being replaced by the number of clusters). Additional files are created to describe the data annotations as follows:

- annot.txt: This is a tab-delimited gene annotation files. Annotations can be of multiple types: gene IDs (multiple gene IDs can be provided in a free format),

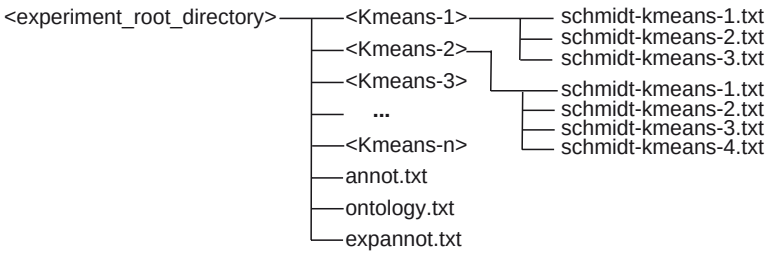


Fig. 19.1 An example of a typical ClutrFree directory structure. This example is taken from the Schmidt dataset analysis detailed in Sect. 19.2.2. The structure includes a set of subdirectory Kmeans*, each of them containing an individual clustering experiment. In this example, Kmeans-1 contains three clusters and Kmeans-2 contains four clusters. The set of flat files contains gene and experimental annotations

description (free format string), and gene ontology (multiple formats are supported, as seen in Sect. 19.1.10). The file must be in the same order as the **D** matrix.
- ontology.txt: This is an optional tab-delimited lookup table that contains gene ontology terms.
- expnames.txt: This file specifies conditions/experiments labels (listed on a single row).
- expannot.txt: This file describes annotations for experimental conditions. It has to be in the same order as the experiments in the **D** matrix. Only one of the files, expnames.txt or expannot.txt, can be used by ClutrFree.

A typical directory organization is shown Fig. 19.1. ClutrFree is capable of generating publication quality images (TIFF/PNG /JPEG and SVG formats). It also allows for export of textual data from statistical analysis.

19.1.7 Tree Algorithm

Two trees are constructed to allow data exploration and to underline cluster stability across parameter variation. The two trees are constructed using the same algorithm from rows of the **P** matrix or columns of the **A** matrix.

The different clustering runs are represented vertically, one run being represented by a set of horizontal nodes. The tree is started with the clustering results having the lowest number of clusters (Fig. 19.2a). Additional clustering runs are then added by connecting each node to the one having the highest correlation (Pearson correlation is used here). Remaining nodes are added the same way.

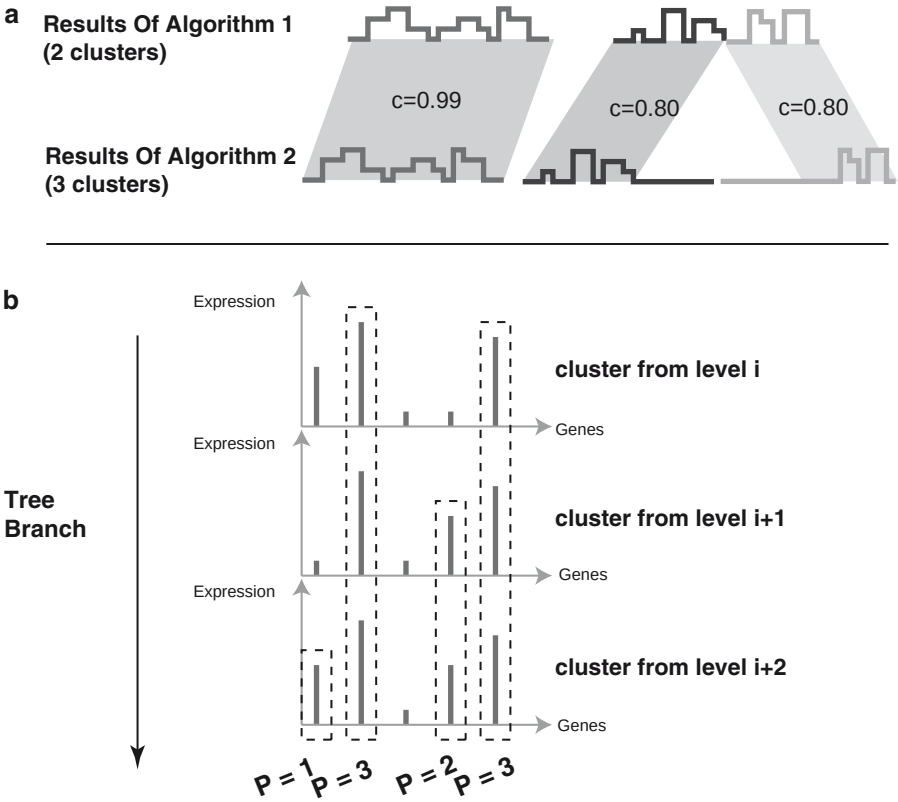


Fig. 19.2 ClutrFree algorithms: (**a**) the basic principle of the tree construction algorithm. The two clusters on top and the three clusters on bottom have been obtained from two separate clustering experiments. The lower level is connected to upper level through maximization of correlation. The branch splitting on the right may also represent biological function splitting. (**b**) The persistence calculation algorithm. Each row represents a cluster gene expression profile or a gene membership list from a tree branch. Data is first converted in binary form in "0" and "1" (see text for threshold calculation). Persistence is calculated as the number of consecutive times an experiment or a gene is set to "1" (meaning present) in the branch

19.1.8 Persistence Algorithm

Persistence can be measured at the gene level (**A** matrix) or the experimental condition level (**P** matrix). Data must first be transformed into binary according to the threshold $n \cdot \sigma$, n being a number set by the upper slider on the main window and σ the standard deviation on the gene expression profile.

Persistence at a given level is then calculated on values set to "1," that is, above the $n \cdot \sigma$ threshold, within a tree branch (see Fig. 19.2b), by counting the consecutive times a gene (or condition) has been set to "1" within the branch.

19.1.9 Description of User Interface

The ClutrFree graphical user interface (GUI) is composed of four windows to allow for multiple data type exploration and integration (Fig. 19.3):

- The Main window (Fig. 19.3a): This is the main ClutrFree window, which allows for loading the data and visualizing the cluster profiles (**P** matrix rows). It shows the average profile of a given cluster. An arrow pad is provided, which allows navigation throughout all the clusters organized in the pattern tree. Multiple graphical features help the interpretation of cluster profiles. For instance, it is possible to see whether the current cluster is conserved across multiple experiments by looking at the corresponding pattern tree window (current pattern is highlighted in yellow color on the tree). In the main window, on the graph itself, each data point is represented as a stem or as a continuous profile (useful to examine time-dependent data such as time series for instance). On every stem background, vertical blue box thickness (see persistence box Fig. 19.3a) is proportional to a measure of persistence (Sect. 19.1.8) that quantifies the stability of this data point across multiple clustering experiments. The blue value within the parenthesis (at the top of each stem) is the actual persistence value. The yellow background of the stem means that the corresponding data point is below the threshold $n.\sigma$. On top of the graph, experiment names are mentioned, and on the bottom, a color code is labeled. This color code is associated with hierarchical annotations provided in the "expannot.txt" file.

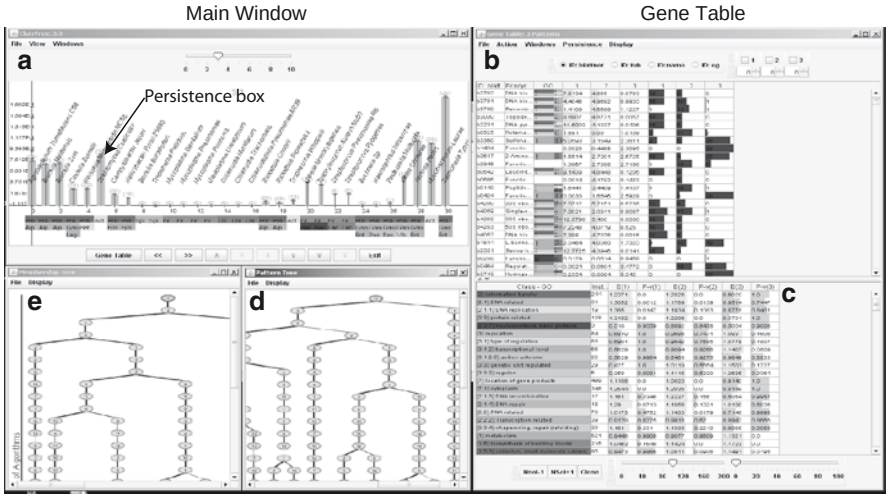


Fig. 19.3 The ClutrFree GUI. (**a**) The main window showing pattern found in the data. (**b**) The gene list window showing gene membership to each clusters and gene ontology enrichment statistics in (**c**). The pattern tree and membership tree windows are shown in (**d**) and (**e**), respectively

- The Gene Table (Fig. 19.3b): The table describes both gene annotation and gene contribution to each cluster and is divided into two parts horizontally, if files annot.txt and ontology.txt are correctly formatted and provided. The upper table shows the distribution of genes in the clusters (columns of **A** matrix), and the lower part gives enrichment statistics. In addition to the gene contribution to the cluster, the persistence value for each gene is also represented in dark blue. In addition to the values represented in the table, bar graphs are superimposed on these in order to improve general readability. Similarly, yellow bars are used for gene cluster contributions and blue for persistence. In order to filter genes that are considered more stable in the data, a persistence filter is on the window top – allowing the user to keep only genes that have a minimum persistence on one or several branches. Two columns show textual data: the first column is a gene ID (multiple ID systems can be provided to ClutrFree and chosen from a radio button), and the second is the Gene Ontology. Multiple ontology systems are supported, including the Gene Ontology Consortium (Ashburner et al. 2000), the MIPS database ontology, and the EcoCyc database ontology (see Sect. 19.1.10).
- In the lower part of the gene table (Fig. 19.3c), statistics are provided in two forms for gene ontology enrichment. Using the binary data for each annotated gene in a given cluster and the total number of genes in the current cluster and in the full dataset, two values are computed: the relative ratio enrichment and the *p*-value based on the hypergeometric distribution (see Sect. 19.1.11.7 for mathematical details). Each measurement is showed in the form of a bar graph in the table; yellow color is used for enrichment and orange for *p*-value. Two filters are present to highlight ontology in the table according to two criteria. The left slider controls the highlight threshold for enrichment values on the range [0–2.0] (represented on the UI by the 0–200 interval). The right slider controls the threshold to display ontology terms according to their instance numbers. This allows focus on either higher level ontology terms (more genes and less specific annotations) or lower level ontology terms (more specific annotations).
- Pattern tree (Fig. 19.3d): Two trees are generated by ClutrFree (see Sect. 19.1.7). The pattern tree windows display the tree generated with the patterns. Several parameters can be changed from the window menu. If ontology terms have been loaded by ClutrFree, they can be superimposed on the tree in the form of a pie (Menu [Display][Ontological Pies], see Fig. 19.4a). Since it is impossible to display all ontology terms on top of every node, only the most representative are displayed. These are parameterized by values set on the [Option] dialog (menu [Display][Options]). Tree options that can be parameterized include a *p*-value threshold, upon which ontology terms are not displayed (default = 0.1), a minimal level for ontology terms (default = 0 – all ontology terms are displayed), and a minimal enhancement value. Optionally, users can display correlations (Fig. 19.4b).
- Membership tree (Fig. 19.3e): This window is similar to the pattern tree window in features and functions and provides a view for the membership tree.

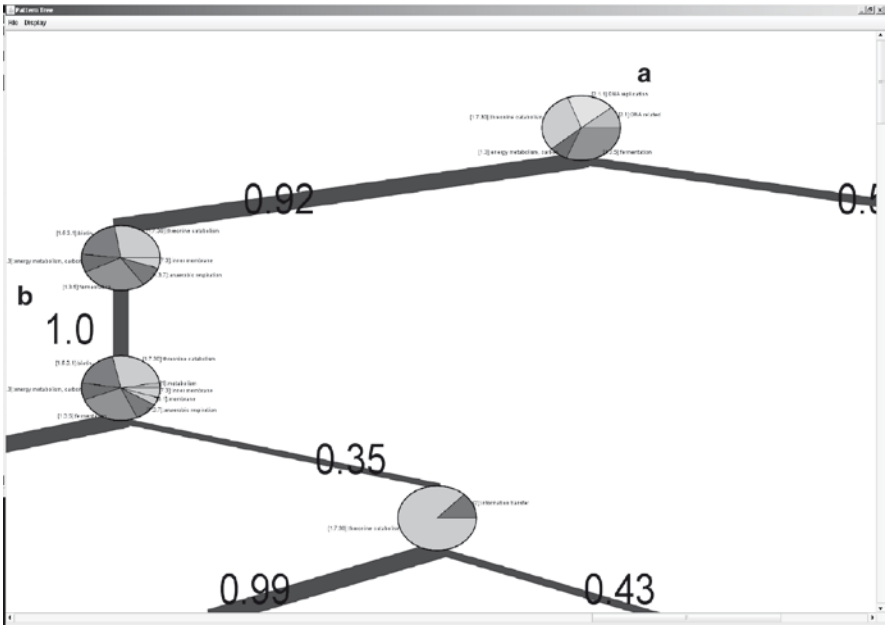


Fig. 19.4 Tree with superimposed ontology. If ontology is provided, it can be superimposed to the pattern tree, showing biological significance for all the experiments currently analyzed. Parameters for displayed ontological pies (threshold for *p*-values, ontology size) can be set from the options dialog

19.1.10 Ontology Types

Three Ontology formats can be used: the MIPS format, the Gene Ontology Consortium format, and the EcoCyc format. Three examples are shown here for the annot.txt or ontology.txt file format (only two rows are shown). The corresponding rendered display is shown Fig. 19.5:

- MIPS format (Fig. 19.5a):

Annot.txt File:

ID: names	Desc: Feature Type	Hipath:mips	Hipath: mips	Hipath: mips	Hipath: mips	Hipath: mips	Hipath: mips
YAL 067C	CDS	CELLULAR TRANSPORT, TRANSPORT FACILITATION AND TRANSPORT ROUTES BC-20	Transported compounds (substrates) BC-20.01	Ion transport BC-20.01.01	Anion transport (Cl, SO4, PO4, etc.) BC-20.01.01.07	CELLULAR TRANSPORT, TRANSPORT FACILITATION AND TRANSPORT ROUTES BC-20	Transport facilitation BC-20.03

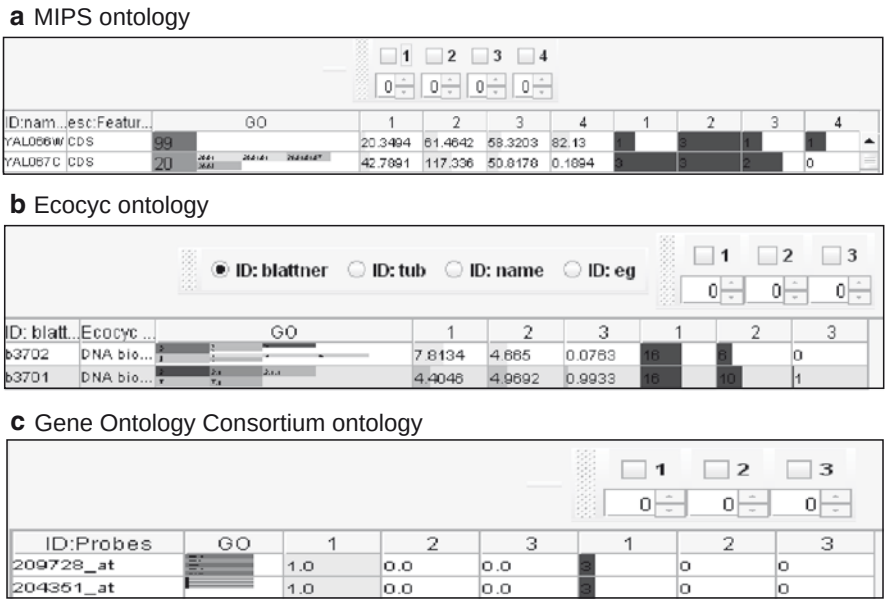


Fig. 19.5 This figure shows ClutrFree rendering for several gene ontology systems. ClutrFree currently supports: (**a**) The MIPS (Munich Information center for Protein Sequences) annotations, (**b**) the EcoCyc annotations, and (**c**) the Gene Ontology Consortium annotations

- EcoCyc format (Fig. 19.5b)

Annot.txt file

ID: blattner	ID: tub	ID: name	ID: eg	Desc: EcoCyc Description
b3702	Rv0001	dnaA	EG10235	DNA biosynthesis; initiation; binding protein
b3701	Rv0002	dnaN	EG10242	DNA biosynthesis; sliding clamp subunit; required for high processivity; DNA polymerase III beta-subunit

Ontology.txt file:

b3702	Regulation BC-3	Type of regulation BC-3.1	Transcriptional level BC-3.1.2	Action unknown BC-3.1.2.5	Information transfer BC-2
b3701	Information transfer BC-2	DNA related BC-2.1	DNA replication BC-2.1.1	Location of gene products BC-7	Cytoplasm BC-7.1

- Gene Ontology Consortium Format (Fig. 19.5c):

Annot.txt file:

ID:Probes
205916_at
205509_at

Ontology.txt file: (only the two first columns are represented – the file contains annotation for the whole Affymetrix HG-U133A platform).

209728_at	GO:0006955 – immune response	GO:0002376 – immune system process
209728_at	GO:0002504 – antigen processing and Presentation of peptide or polysaccharide antigen via MHC class II	GO:0019882 – antigen processing and presentation
209729_at	GO:0007050 – cell cycle arrest	GO:0022402 – cell cycle process

19.1.11 Technical Details

19.1.11.1 Language(s) used

ClutrFree is currently developed in Java under Eclipse Ganymede with JRE 6.0. The system can be compiled and used on any platform capable of running the Sun Java Virtual machine version 6.0.

19.1.11.2 Ancillary Tools Needed for Compilation and Application Hosting

ClutrFree is hosted on Sourceforge[3] (CVS and main Web site). Compiled code can be downloaded for direct use (File Clutrfree-1.40.zip), and source code can be checked out from CVS. Details are given in the documentation.

19.1.11.3 Version/Suite Dependencies

ClutrFree does not allow for any low-level analysis on microarray data (such as normalization and probe calculation) or clustering – this has to be done on third party software. For Affymetrix data normalization, we recommend the affy package from Bioconductor (Gautier et al. 2004). For clustering, we recommend the use of PattTools and MeV. The current version is ClutrFree 1.4, which can be used seamlessly with MeV 4.0.

19.1.11.4 Options for Assistance from Developers

The hosting Web site on Sourceforge hosts a Web page for feature requests[4] and a bug tracker.[5]

[3] http://clutrfree.sourceforge.net

[4] https://sourceforge.net/tracker/?group_id=182093&atid=899833

[5] https://sourceforge.net/tracker/?group_id=182093&atid=899830

19.1.11.5 References to Published Manuscripts

Bidaut G (2007) Gene function inference from gene expression of deletion mutants. Methods Mol Biol 408:1–18.

Bidaut G, Suhre K, Claverie JM, Ochs MF (2006) Determination of strongly overlapping signaling activity from microarray data. BMC Bioinform 7:99.

Bidaut G, Ochs MF (2004) ClutrFree: cluster tree visualization and interpretation. Bioinformatics 20(16):2869–2871.

19.1.11.6 Code Extension: Development of Plugins

Since the code is available under the General Public License, it can be modified to handle new calculation types, as allowed by its object architecture. Also, a plugin system allows for development of new data types handler. Details are found in the documentation.

19.1.11.7 Statistical Enrichment

- Relative ratio enrichment formula:

$$e(GO\_i, pattern\_j) = \frac{k/g}{K/G}$$

- Hypergeometric distribution

$$p - val(k, g, K, G) = 1 - \sum_{i=0}^{i=k} \frac{\binom{K}{i}\binom{G-K}{g-i}}{\binom{G}{g}}$$

- k being the number of occurrences of GO i in pattern j.
- g being the number of annotated gene(s) in pattern j.
- K being he number of occurrences of GO in whole experiment.
- G being the number of annotated genes in whole experiment.

Details of the practical implementation can be found in the ClutrFree source code (file GOData.java).

19.2 Representative Examples of the Tool in Use

Two representative examples of data analysis are provided, presenting analysis on two distinct data types. The first describes analysis of phylogenomic profiles based on Bayesian Decomposition (PattTools) and ClutrFree. The second analysis is

about clustering cancer type microarray data (Affymetrix platform U133A). Bioconductor and MeV are used for clustering, and those results are combined and compared in ClutrFree.

19.2.1 Analysis of Clusters from Bayesian Decomposition Analysis of Phylogenomic Profiles

19.2.1.1 Data Gathering

The data represents BLAST score alignments (*e*-values) for 31 bacterial genomes to a set of reference genes. Each gene profile quantifies sequence similarity with 31 bacteria. Data is generated with specific scripts which are not described here. The final dataset is provided in the supplemental data. The biological background has already been described (Bidaut et al. 2005).

19.2.1.2 Computing Environment

Data analysis is done on a Unix-like type of system (CentOS, Ubuntu, MacOSX, or Cygwin). Access to a high-end server or workstation is required, preferably a multicore processor with at least 2 GB of RAM and 100 MB of free disk space. The PattTools package has to be installed (http://www.cancerbiostats.onc.jhmi.edu/PattRun.cfm).

19.2.1.3 Clustering Analysis

The file “merged.txt” has to be loaded in PattTools. The algorithm used is Bayesian Decomposition, and several runs with variation of the number of patterns have to be done. For this dataset, we decomposed the data between 3 and 31 patterns. Description of the procedure is found in (Bidaut 2007). Annotations are from the EcoCyc database.

19.2.1.4 Visualization with ClutrFree

The resulting data is loaded in ClutrFree for visualization. The main window shows patterns with contributions of each bacterial genome. Each of these corresponds to clusters of genes that are conserved within groups of bacteria. Various patterns are described in (Bidaut et al. 2005).

19.2.2 Analysis of Clusters Generated from the Multiple Experiment Viewer on Publicly Available Cancer Data

This is an example of a complete data analysis of cancer microarray data. The data is from a large-scale breast cancer study using the Affymetrix U133A platform (Schmidt et al. 2008). Genomic prognostic motifs are studied on gene expression profiles from 200 tumor samples from breast cancer patients, who were not treated after surgery.

19.2.2.1 Computing Environment

Data analysis is done on a Unix-like system (CentOS, Ubuntu, MacOSX, or Cygwin). Access to a high-end server or workstation is required, preferably a multicore processor with at least 8 GB of RAM and 1 GB of free disk space. The following software has to be installed: R (Version 2.8) and Bioconductor (Version 2.3),[6] The MeV (Version 4.2.03),[7] ClutrFree (latest version 1.4),[8] and a spreadsheet program. Instructions and links for each program are available in the supplementary data Web site.

19.2.2.2 Data Gathering

We downloaded the data from the public repository GEO. The accession number for the dataset is GSE11121. Description of the dataset and links to various files are available at http://www.ncbi.nlm.nih.gov/projects/geo/query/acc.cgi?acc=GSE11121.

The data is available as a raw file and can be downloaded with the following commands:

```
$ mkdir schmidt-cel
$ cd schmidt-cel
$ wget http://www.ncbi.nlm.nih.gov/projects/geo/query/
acc.cgi  mode=raw&acc=GSE11121&db=GSE11121%5FRAW.tar&
is_ftp=true?
```

ClutrFree Annotation data for the platform (Affymetrix U133A) is available from the Automated Service Annotation Pipeline (ASAP) system v2.6.[9] Registration is needed at first use. Once logged, click on the "Query" tab and choose the "db/AffyAnnotation" plan. Then, choose the HG-U133A platform and select the "Additional output for ClutrFree" box. Than, start the job by clicking the "query" button.

[6] Installation instructions are here: http://www.bioconductor.org/docs/install/

[7] Available here: http://www.tm4.org/mev.html

[8] Available here : http://clutrfree.sourceforge.net/

[9] http://hammurabi.onc.jhmi.edu/cgi-bin/ASAP/login.pl

Once done, ASAP allows downloading ClutrFree files for the three main Gene Ontology Trees. Note that we keep only the "Biological Process" data for this analysis. Once downloaded, the file is saved under "NNN_ontology_process.txt," where NNN is an index number generated by ASAP.

19.2.2.3 Analysis

- Normalization:

  Normalization is performed with bioconductor. Detailed documentation is available from the affy vignette: (command `vignette("affy")` at the R prompt). Data has to be uncompressed with the following command:

  ```
  $ tar xf GSE11121_RAW.tar
  ```

  We then normalize it with the following R commands:

  ```
  > library(affy)
  > SchmidtData = ReadAffy()
  ```

  Data is normalized with the GCRMA function from the gcrma library.

  ```
  > library(gcrma)
  > expr = gcrma(SchmidtData)
  ```

  After normalization, the data is exported on the disk:

  ```
  > write.table(expr, 'NormalizedSchmidt150509.txt');
  > q()
  ```

- Variation filtering and *K*-means clustering:

  Clustering is performed within the `MeV` environment. On our system, `MeV` is invoked with the following command:

  ```
  $ /opt/MeV_4_3_02/tmev.sh
  ```

  Once loaded, click on the [file] menu, and select the "`NormalizedSchmidt 150509`.txt" data file.

– Adjust data, data filters, variance filter, and percentage of highest SD genes. We set the filter to 3%.
– Expand "data filter–variance filter."
– Save the resulting dataset as "schmidt-expression-file-668-200.txt."

  Edit the file within Excel© or OpenOffice™ spreadsheet to remove all annotation columns and to keep only the ProbeIDs. Restart the `MeV` session and load the "schmidt-expression-file-668.txt" file.

– Create a new Script, name it as "schmidt-kmeans-3-20"
– Right click Add Algorithm Node
– Select the "KMC" algorithm (*K*-Means/median Clustering)

– Parameters for *K*-means are "Distance Metric Selection": "Pearson Correlation" and "number of clusters": 03
– Restart the two last steps by varying the number of clusters from 04 to 20
– Right click on "Primary Data" on the "execute script" item

MeV is now processing the data.

- Data organization for ClutrFree:

  Every *K*-means output must be placed in an appropriate subdirectory. First, one creates a subdirectory containing the data to load in ClutrFree:

  ```
  $ mkdir Schmidt-clutrfree
  ```

  Then, one creates each subdirectory.

  ```
  $ mkdir KMeans-XX (with XX varying from 03 to 20)
  ```

  The ontology file generated with ASAP must be linked or renamed:

  ```
  $ ln –s 75012_ontology_process.txt ontology.txt
  ```

  The full directory tree is available from the supporting Web site as a tar archive.

  Back to MeV, results are found under the "Script Results" tab. Numerical data is under "Results." To open the results for the *K*-means run with three centroids, open the first "KMC – genes" tab, "Centroid Graphs." On the graphics pane, do a right click, to bring up the "Save all clusters" menu, and save the result under the "schmidt-clutrfree/Kmeans-03" subdirectory, under the name "Kmeans-03."

  The same operation has to be repeated for all *K*-Means runs until the directory structure has been filled.

- Data annotation:

  The first column of the "schmidt-expression-668-200.txt" file has to be copied into the "annot file." The column header has to be set to "ID:Probes." Additional information such as gene ID or gene symbol can also be specified, and the header must be specified as "ID:gene symbol" for instance. The file has to be stored under the schmidt-clutrfree directory.

  The file generated with ASAP has also to be copied in this directory and linked as "ontology.txt."

19.2.2.4 Visualization with ClutrFree

ClutrFree is launched with the following command:

```
$ java –Xmx1024M –jar clutrfree.jar
```

Data is loaded with the file menu. The experiment directory root must be selected for data loading ("schmidt-clutrfree"). Once data has been correctly parsed, ClutrFree displays the following message: "ClutrFree has successfully

loaded 18 experiments"; the current experiment has three clusters of length 18. The Tree windows show the current position within the tree.

Patterns found and corresponding clusters can then be explored and analyzed with GO enrichment and hypergeometric *p*-values measurements (Fig. 19.5).

19.3 Future Development and Enhancement Plans

Several development avenues are currently considered for ClutrFree. First, persistence measurements can be refined to include measures for stability of genes/conditions under threshold $n \cdot \sigma$ and set to "0." Also, the calculation of hypergeometric distribution could be done for depleted categories (as opposed to enriched categories) as an option.

To accommodate high-throughput studies on large dataset compendia, development of a command line version of Clutrfree for scripting is envisioned. It would be also possible to turn it into a Web application.

Hierarchical clustering is not supported at this time, even though this algorithm is widely employed in array data analysis. This is due to the inherent dendrogram structure that is more complex to handle than gene lists. Also, Gene Ontology display must be improved, and a color system to accommodate the GO hierarchy is envisioned.

On the longer term, the ASAP system could be integrated with ClutrFree, and data could be automatically loaded from it through the network.

Acknowledgments Ghislain Bidaut is funded by the Institut National de la Santé et de la Recherche Médicale, the Fondation pour la Recherche Médicale, and the Institut National du Cancer (Grant 08/3D1616/Inserm-03-01/NG-NC). Thanks to Wahiba Gherraby for proofreading the manuscript.

References

Arthur D, Vassilvitskii S (2007) *k*-Means++: the advantages of careful seeding. In: Proceedings of the eighteenth annual ACM-SIAM symposium on discrete algorithms, pp 1027–1035

Ashburner M, Ball CA, Blake JA, Botstein D, Butler H, Cherry JM, Davis AP, Dolinski K, Dwight SS, Eppig JT, Harris MA, Hill DP, Issel-Tarver L, Kasarskis A, Lewis S, Matese JC, Richardson JE, Ringwald M, Rubin GM, Sherlock G (The Gene Ontology Consortium) (2000) Gene ontology: tool for the unification of biology. Nat Genet 25(1):25–29

Ball CA, Brazma A (2006) MGED standards: work in progress. OMICS 10(2):138–144

Barrett T, Troup DB, Wilhite SE, Ledoux P, Rudnev D, Evangelista C, Kim IF, Soboleva A, Tomashevsky M, Edgar R (2007) NCBI GEO: mining tens of millions of expression profiles – database and tools update. Nucleic Acids Res 35(Database issue):D760–D765

Bidaut G (2007) Gene function inference from gene expression of deletion mutants. Methods Mol Biol 408:1–18

Bidaut G, Ochs MF (2004) ClutrFree: cluster tree visualization and interpretation. Bioinformatics 20(16):2869–2871

Bidaut G, Suhre K, Claverie JM, Ochs MF (2005) Bayesian decomposition analysis of bacterial phylogenomic profiles. Am J Pharmacogenomics 5(1):63–70

Bidaut G, Suhre K, Claverie JM, Ochs MF (2006) Determination of strongly overlapping signaling activity from microarray data. BMC Bioinform 7:99

Boutros PC, Okey AB (2005) Unsupervised pattern recognition: an introduction to the whys and wherefores of clustering microarray data. Brief Bioinform 6(4):331–343

Datta S, Datta S (2006) Evaluation of clustering algorithms for gene expression data. BMC Bioinform 7(Suppl 4):S17

Do JH, Choi DK (2008) Clustering approaches to identifying gene expression patterns from DNA microarray data. Mol Cells 30;25(2):279–288. Epub 2008 Mar 31

Gautier L, Cope L, Bolstad BM, Irizarry RA (2004) affy – analysis of Affymetrix GeneChip data at the probe level. Bioinformatics 20(3):307–315

Gentleman RC, Carey VJ, Bates DM, Bolstad B, Dettling M, Dudoit S, Ellis B, Gautier L, Ge Y, Gentry J, Hornik K, Hothorn T, Huber W, Iacus S, Irizarry R, Leisch F, Li C, Maechler M, Rossini AJ, Sawitzki G, Smith C, Smyth G, Tierney L, Yang JY, Zhang J (2004) Bioconductor: open software development for computational biology and bioinformatics. Genome Biol 5:R80

Hibbs MA, Dirksen NC, Li K, Troyanskaya OG (2005) Visualization methods for statistical analysis of microarray clusters. BMC Bioinform 6:115

Lee HK, Braynen W, Keshav K and Pavlidis P (2005) ErmineJ: tool for functional analysis of gene expression data sets. BMC Bioinform 6:269

Moloshok TD, Klevecz RR, Grant JD, Manion FJ, Speier WF 4th, Ochs MF (2002) Application of Bayesian decomposition for analysing microarray data. Bioinformatics 18(4):566–575

Quackenbush J (2001) Computational analysis of microarray data. Nat Rev Genet 2(6):418–427

Sabatier R, Finetti P, Cervera N, Birnbaum D, Bertucci F (2009) Gene expression profiling and prediction of clinical outcome in ovarian cancer. Crit Rev Oncol Hematol 72(2):98–109

Saeed AI, Bhagabati NK, Braisted JC, Liang W, Sharov V, Howe EA, Li J, Thiagarajan M, White JA, Quackenbush J (2006) TM4 microarray software suite. Methods Enzymol 411:134–193

Schmidt M, Böhm D, von Törne C, Steiner E, Puhl A, Pilch H, Lehr HA, Hengstler JG, Kölbl H, Gehrmann M (2008) The humoral immune system has a key prognostic impact in node-negative breast cancer. Cancer Res 68(13):5405–5413

Sotiriou C, Pusztai L (2009) Gene-expression signatures in breast cancer. N Engl J Med 360(8):790–800

Chapter 20
Enhanced Dynamic Documents for Reproducible Research

Deborah Nolan, Roger D. Peng, and Duncan Temple Lang

Abstract Dynamic documents that combine text and code, which is evaluated to dynamically create content when the document is "rendered," for example, Sweave, are a large step forward in reproducible data analysis and computation. However, to capture the research *process*, we need richer paradigms and infrastructure. The process includes all the investigations and computations, and not just the final reported ones, and the entirety represents reproducible research. In addition to richer paradigms for reproducability, we want to be able to capture more complex aspects of the computational process, such as the use of multiple languages, and also engage different communities using other programming languages so that reproducible computations and research become more widespread. We also need to integrate existing and future approaches with commonly used tools such as Microsoft Word and make the resulting documents richer for authors and readers. We present two approaches to structured, dynamic documents that use modern, ubiquitous standard technologies (XML) and provide extensible infrastructure for richer documents. The first integrates R and Microsoft Word for use by a broader audience and provides some innovations in this interface, and the second uses eXtensible Stylesheet Language (XSL) and R to provide a flexible and extensible infrastructure for richer, more accessible dynamic documents.

20.1 Introduction

Dynamic documents as a means to reproducing computational results are gaining prominence in statistics and science generally. Systems such as Sweave (Leisch 2002) and odfWeave (Max Kuhn 2008) provide ways to author documents containing

D.T. Lang (✉)
Department of Statistics, University of California, Davis, 4210 Mathematical Sciences Building, One Shield Avenue, Davis, CA 95616, USA
e-mail: duncan@wald.ucdavis.edu

M.F. Ochs et al. (eds.), *Biomedical Informatics for Cancer Research*,
DOI 10.1007/978-1-4419-5714-6_20, © Springer Science+Business Media, LLC 2010

code that is evaluated when the document is processed. This greatly enhances the standard of reproducibility in research. The computations that produce the reported results are available to both the author and other researchers.

Reproducible research is a broader, more ambitious goal than dynamic documents. While richer than regular documents, dynamic documents are still linear/sequential reports of what the author decides to present. They differ in that the results of the computations are "guaranteed" to come from the reported computations. Ideally, we would also be able to explore what the researcher actually did in totality. We would be able to see the different approaches that she pursued but did not report or that she considered but did not pursue and possibly why. This information is of great value to other researchers, reviewers, and students, but it is lost in publication and often to the researcher herself. To capture and archive the research process, we need to go much further than current dynamic documents.

Sweave and related systems are quite closely coupled to R (R Development Core Team 2008) and LaTeX (Mittelbach et al. 2006). The ideas are generalizable to other languages and formats and there are drivers for SAS and for Open Office (http://www.openoffice.org) and HTML output. However, the noweb-based (Ramsey 1994) syntax of Sweave is quite limited in supporting richer, more structured documents. Documenting analyses that use a mixture of programming languages such as the UNIX shell, Python, Perl, R, or C is not easily done. Furthermore, developing or extending drivers for different formats is complicated because the framework and associated tools are somewhat ad hoc and nonstandard in terms of word processing. Using more ubiquitous and standard technologies would facilitate new experiments and developments and disseminating the practice to other communities.

We envisage a system for dynamic documents that acts much like an electronic laboratory notebook. The researcher(s) would passively capture the computations they perform and be able to organize them as *tasks* and subtasks at different resolutions. Some code analysis tools would help her visualize and manage these tasks. She would be able to project the document into papers for different audiences, for example, a paper that describes the conclusions of her work for a journal, another that provides more extensive details about the work such as a technical report, and an interactive document which reviewers could explore at different levels of detail, that is, "drill down." Readers would be able to examine the tasks and the computations. They would be able to run "what-if" computations, bringing in new data sets or selecting alternative approaches to tasks, for example, using a different classification technique. Using (partially automated) metadata from the document and the code, interactive views of the document would allow readers to change parameters in the computations and explore, for example, sensitivity and robustness of the results and conclusions.

In order to develop a new system for reproducible research and rich documents capable of these facilities, we believe we need to use more extensible, ubiquitous, and standard technologies suited to both modern publishing and

programmatic manipulation. In this short overview, we describe two systems that provide infrastructure that we believe can grow to support this more ambitious style of dynamic and interactive documents. Both approaches exploit the eXtensible Markup Language (XML) and related technologies (XPath, XSL, XInclude) as the foundation. The first approach allows researchers to author dynamic documents using Microsoft Word. The second uses Docbook – an XML format for technical documents akin to LaTeX – and provides a highly extensible framework. XML is a natural choice as the model relies on being able to *markup* the different elements to provide structured documents. XML is very widely used in modern software, and the connection with XML technologies allows us to easily connect the documents with new Web formats and modern publishing tools. The structured nature of these documents and powerful tools for operating on them also allows us to build more automated document validation tools (e.g., the XDocTools package for R) that check cross references, synchronize and update documents and the software they reference, verify code, check table and figure captions, dynamically construct content, spell check diagrams, and so on.

The software we describe is available in the R packages RWordXML and XDynDocs with support from several additional packages (ROOXML, Rcompression, and XML). These packages are made available under the very permissive Berkeley Software Distribution (BSD) license. They have been designed with extensibility and customization by others as a primary goal. As a result, they offer a platform for us and others to experiment with richer forms of dynamic documents and reproducible computational-based research techniques. They also transfer to other programming environments, for example, MATLAB or Python, very naturally. Similarly, some of the new ideas from the R-Word interface for dynamic documents apply to odfWeave and Open Office.

In the rest of this chapter, we give a very high-level description of how one can use the software that we have developed for authoring and processing dynamic documents. We describe the high-level aspects of Microsoft Word in Sect. 20.2 and follow this with a discussion of the R-Docbook-XSL approach.

20.2 Using Microsoft Word and R for Dynamic Documents

Before we discuss details of how one authors or generates content in a dynamic document, it is useful to describe some terms that we will use below. The author/researcher creates a Word document (a .docx file) that contains both text and R code. This is the *source* or input document. To generate the paper or document for a reader, we take a copy of this source .docx file, evaluate the code, and insert the results into this newly created copy of the input .docx file. The original .docx file remains unaltered by the processing to generate the results. The RWordXML package does this dynamic processing and we will discuss this below. The important

thing to keep in mind is that the input and output documents are very similar but they are two separate documents. One can use Word to create a PDF or HTML version of the output document.

20.2.1 Authoring Dynamic Documents with Word

The aim is to allow users of Microsoft Word to conveniently create dynamic documents using a familiar interface. The author writes and formats text in the usual manner, adding new sections, titles, lists, tables, and regular text. To make the document "dynamic," she adds code by writing it directly or cutting and pasting it from an R session or a file. The key step is that she must identify the code as being dynamic (and not just text that happens to be code) so that it will be evaluated when we project/process the document. Styles are used to perform this *markup* and identify the content of the document as a block of R code, an inline R expression (the value of which is part of sentence), R code that produces a graphical display, an R function definition, or R language elements such as a reference to an R package or function or class, a function parameter, and so on. There are also styles for identifying code that is to be evaluated but not displayed in the output document and for code that is to be displayed for the reader, but not evaluated.

The author sets the style for a paragraph or segment of text either by selecting/highlighting the existing content and applying a style or by setting the style and then adding content. To apply a style, she chooses the particular style from the Styles section of the Formatting Palette. This is shown in Fig. 20.1. This example document contains code for the two R plots, each of which is indented and colored red via the "R lattice plot style" (a particular graphics system in R). The selected word "lattice" is the name of an R package and so will be given the "R package style."

Because the number of available styles can be overwhelming and difficult to work with, we provide an additional "R formatting" toolbar that presents the collection of R-specific styles and makes applying styles significantly more convenient. We also provide key bindings or shortcuts for applying these R-related styles for those of us who do like to avoid using the mouse.

In addition to using styles to identify R code, an author can also include R output from the computations and use the "R output" style to identify it. This helps the author to see the actual results in the document as she is authoring it. When the document is processed and the code is reevaluated, this output area will be replaced with the actual output. However, this manually inserted output serves a potentially valuable purpose. The author can format the output as she wants the results to appear in the final view/projection of the document. For example, she may change the margins, color or font of the text, and the width and color of columns in a table or specify a style for the output. Similarly, she may include an R plot and specify its format (PNG, JPG, PDF) and its dimensions. When we process the document

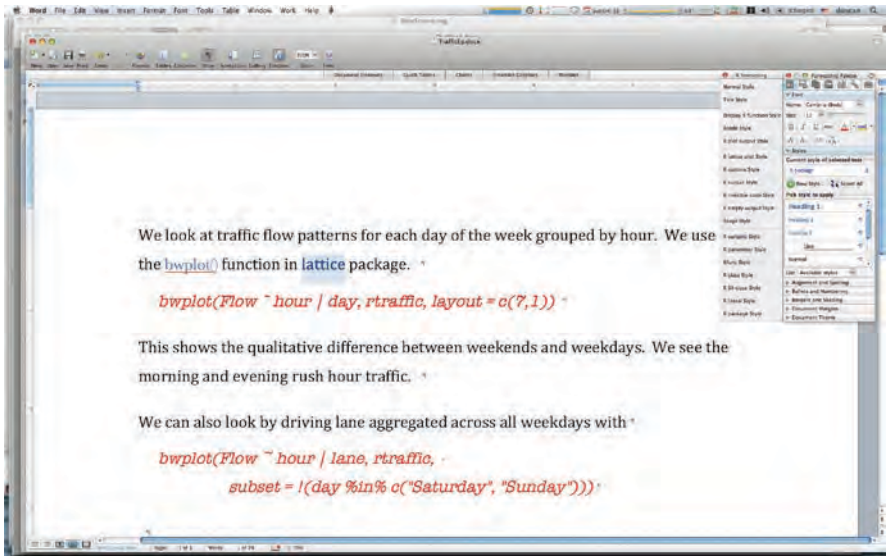


Fig. 20.1 Here, the author of a Word document uses styles to markup content such as R plot code, R function, and package references. The word "lattice" has been selected for marking up as an R package. Styles can be selected from the regular Formatting palette toolbar on the *far right*. For convenience, we provide an additional toolbar with just the R-related styles which is on the *left* of the Formatting palette

dynamically and insert the new results of the computations, we attempt to insert the R results into this format. Rather than specifying options via noweb syntax within the document, the author can use Word tools to specify the format of the results. This gives the author a great deal of control over the appearance of the final document in a familiar and natural manner for Word users.

Styles are pivotal for our software. They are also an underused but important part of rendering text in word processors, HTML documents [via Cascading Style Sheets (CSSs)], etc. They attempt to separate content and structure from appearance. They allow authors to control the appearance of all the content that share a particular style (both within and across documents). Centralizing the definition of a style makes it easy to update the characteristics of the style and immediately update the appearance of all corresponding text. In addition to appearance, we make use of styles as markup and structure so that we can identify the nature or purpose of particular text when processing the document. It is imperative that authors use these styles in order to identify the code. It is also useful to identify R concepts such as the names of R functions, packages, classes, and parameters so that we can programmatically not just manipulate the dynamic code, but also validate and synchronize the content with respect to the software being used in the code.

20.2.2 Processing the Dynamic Document

Having created the dynamic input document, the author turns to the separate step of creating the output document. To do this, she (or someone else) calls the R function `wordDynDoc()` in the RWordXML package. The function requires two arguments: the name of the input .docx file and the name of the output file. For most uses, this is all that is needed. The function reads the content of the Word document and finds all the content that has R code-related styles. It then evaluates the code in these blocks sequentially. It takes the values obtained by evaluating each block and uses the generic function `toWordprocessingML()` to create XML representations that are inserted into the output document. At the end of this process, the newly generated document is stored in the output .docx file.

The `wordDynDoc()` function has options that control how and where the code is evaluated and whether the code in the original document is displayed in the output document or not. The function also reads R-specific options stored in the input document's metadata as Word properties. This is a convenient way to specify characteristics such as the default number of digits, the type of graphics device, etc. without requiring the caller of `wordDynDoc()` to specify them each time.

The process is highly extensible. An R programmer can define methods for the `toWordprocessingML()` function to control how R objects of different types are converted and displayed in a Word document. The RWordXML package provides many utility functions for such programmers to leverage when creating WordprocessingML (Vugt 2007) content and querying and modifying Word documents, for example, to find all section titles or hyperlinks; insert data from R as a list or table. An R programmer can also pass their own function to `wordDynDoc()` that is used to process each code node within the Word document. This can use alternative techniques to evaluate the code and render the results. This mechanism allows us, for example, to integrate caching of results using a package such as cacher (Peng 2008). This permits us to avoid reprocessing lengthy computations each time the document is generated when the particular code nodes have not changed, only evaluating the code for modified content.

In addition to programmatic extensibility, the author of a Word document can introduce new styles. These can be new formatting of existing styles such as how the margins or color for R code appears, or they can be markup for new structured elements within the document. The new styles should be built or extended from existing ones using Word's "based on" property for styles. For example, by basing a new code style on the "R code" style, the authors are guaranteed that `wordDynDoc()` will recognize content that uses such a style as R code and include it in the processing. This gives a simple object-oriented flavor for structured styles.

While we have focused on the dynamic aspect of these documents, we should note that the author can easily extract just the code from the document or even

source the code directly into the R session using the `xmlSource()` function. This allows Word to be used as a literate programming environment.

There are many additional aspects to the RWordXML and ROOXML packages for working with Word (and Excel) documents, but these are not our focus here. We end this section by noting that the package is available for both Windows and Mac OS X operating systems. It can be installed with all its dependent packages using the R command:

```
install.packages ("RWordXML", repos = "http://www.omegahat.org/R",
                  dependencies = TRUE)
```

20.2.3 Drawbacks

The approach leverages the familiarity and strengths of Word and is attractive to those who write documents using this interface. This does limit the audience to Microsoft Windows and Mac OS X users. Some of the ideas are available via Open Office via new elements of odfWeave. Cutting and pasting code from R into a Word document can be somewhat tedious. We would prefer a mechanism that allows the code to be inserted directly from R into the current point of the Word document. This is not feasible in this approach as the Word document cannot be edited while it is being accessed from within R. On Windows, it is reasonably straightforward to use DCOM technology from within R to have synchronized access within both systems.

Word is a graphical user interface and it allows its users to perform word processing tasks for "linear" or sequential documents. It is somewhat difficult to enhance the interface and markup to conveniently allow for richer markup such as customizations for evaluating the code or alternative approaches/branches within an analysis. We turn our attention next to an approach that does allow this and removes us from the world of graphical word processors.

20.3 Dynamic Documents with XML Technologies

Microsoft Word and Open Office use XML to represent a document internally, but provide high-level graphical interfaces for the authors to format the document as they want it to appear for the reader (WYSWIG – what you see is what you get). Many authors prefer writing documents directly using a typesetting language such as LaTeX (or TeX) to specify how the content is to appear. This is more direct in some ways and also gives the author significantly more control. It also allows the author to programmatically manipulate the content of documents to some extent. (La)TeX, however, is not widely used outside of mathematically oriented communities. LaTeX source documents are also not amenable to robust structured query

and manipulation. Furthermore, they do not fully separate content from appearance. Also, LaTeX does not readily support more modern aspects of dynamic, interactive publishing used with Web technologies.

Instead of using a LaTeX-like language for formatting text, some authors use the XML-based vocabulary Docbook (Walsh and Muellner 1999) for writing structured text documents such as books, articles, and software documentation. Once the author has used this markup language to describe the content, we use XML technologies such as XSL to transform the document into any of several different formats (e.g., PDF and HTML) for different audiences, for example, content such as low-level details omitted for general readers or just the code for developers.

We will not go into great detail about Docbook, XML, and XSL. We will, however, illustrate the basics of each and discuss how the pieces are connected for creating dynamic documents. Docbook has extensive documentation, including two online books that cover all major aspects of its use. Knowing only about 15 Docbook elements, an author only needs to additionally know the basic structure of an XML document to be able to create a Docbook document. To write the Docbook content, one can use any text editor. We use emacs and nxml-mode. Alternatively, one can use an XML content editor such as XMLSpy. To "project" a (dynamic) Docbook document into HTML or PDF, one does not need to know any XSL but just the command to apply XSL to the document.

The author might start by creating an article that looks something like the following:

```
<article xmlns:r="http://www.r-project.org">
<title>Analyzing Traffic Flow</title>
<section><title>Introduction</title>
<para>
This article looks at the flow of cars along a
section of Highway 80 in California just outside of
Sacramento.
</para>
</section>
</article>
```

Hopefully the meaning of the XML elements such as `<section>`, `<title>`, and `<para>` (paragraph) are self-explanatory. The key thing to note is that this must be legitimate, well-formed XML. All elements are of the form <name>…</name>, that is, with opening and closing named tags and properly nested. Elements can have child elements, for example, <article> has <title> and <section>, and <section> has <title> and <para>. The resulting document is a hierarchical tree structure. Other Docbook elements used frequently include <ulink> for hyperlinks, <emphasis>, <table>, <figure>, <xref> for cross-references within and between documents, <itemizedlist> and <listitem>.

XML permits extending a vocabulary and we have added new elements to the Docbook vocabulary to introduce new concepts. These are <r:code>, <r:plot>, <r:lattice>, <r:function>, and <r:expr>, which are used to represent R commands/

code with different types of output, just as we had styles in Word. We would use these something like:

```
<para>We plot all
<r:expr>length(levels(lanes))</r:expr> levels of the
categorical variable.
<r:lattice width="5in">
   bwplot(Flow ~ hour | lane, rtraffic)
</r:lattice>
and compute a numerical summary
<r:code>
   with(rtraffic, by (Flow, lane, summary))
</r:code>
</para>
```

The author can display output from R using the r:output element, either nested within r:code elements or immediately following it.

The XML elements <r:func>, <r:pkg>, <r:class> are used to refer to R functions, packages, and classes. We can identify the package for a function or class using an XML attribute, for example, `<r:func pkg="graphics">hist</r:func>`. Function parameters and variables are identified using <r:param> and <r:var>, respectively. There are also XML elements to represent R constants such as <r:true>, <r:false>, <r:null>, and <r:na>.

Note that we have used the prefix r: for all of the R elements. This is a name space in XML to avoid conflicts with other vocabularies that we might want to mix in the same document. The name space is declared via the xmlns:r="http://www.r-project.org" content in the <article> element, with the prefix "r" being the author's choice.

The Docbook markup is relatively simple and one learns new "words" as one needs them. While XML is generally more verbose than LaTeX and other languages, there is a close correspondence between the Docbook and LaTeX vocabularies. What XML and Docbook give us over LaTeX is an array of technologies that provide much more flexibility in constructing documents and a rich set of tools for processing them in many different, programmatic manners which significantly improve the entire document production process. As we mentioned in the introduction, the extensibility by allowing us to introduce new markup and customize and extend the tools is perhaps the most important aspect for us if we are to go further in developing new paradigms for reproducible research documents.

20.3.1 Transforming the R-Docbook Document

Once the author has created an R-Docbook document, she will want to project it into a form that can contain the results of the embedded code and can be given to readers. Because this is XML, it can readily be converted into any form the author wants. She can extract all the code segments or just those in a particular section. The author can remove entire sections or discard the code, leaving only the text.

Typically, the author wants to create either an HTML or PDF version of the document. The Docbook software contains XSL libraries for transforming regular Docbook documents to either HTML or another XML format – Formatting Objects (FO) (Pawson 2002) – used for describing high-quality printed material, similar in concept to LaTeX. FO content can then be transformed directly to PDF using fop (http://www.apachage.org/fop).

There are two approaches to providing the dynamic aspect to these documents, that is, evaluate the code and render the results in the output document. We can use a two-step processor that (a) reads the documents in R and processes only the R code nodes and inserts the resulting output using Docbook markup, for example, <programlisting>, <table>, and <figure>. Alternatively, the second approach (b) uses a single, regular XSL transformation for Docbook to create the final transformation by processing both Docbook and R-specific XML nodes all at once. The second approach integrates R with an XSL engine – libxslt http://www.libxslt.org and allows us to do the processing in a single step. This is the approach we use in the XDynDocs package.

The user calls the function dynDoc() with the name of the input XML document. The second argument specifies the target format. This can be "HTML," "FO," or "latex." If "FO" is specified, this will also create the resulting PDF if the fop program is available. The user can also specify the name of the file to create, but the default is to use the input file name and change the extension to that of the target format.

The `dynDoc()` function calls the embedded XSL engine. It determines the appropriate XSL style sheet to use for the target format, but one can also explicitly specify a different XSL file to use one's own or other customizations. One can also specify XSL parameters as *name=value* pairs. These provide run-time customization of the different XSL rules and can be used, for example, to specify a different CSS to use for the output HTML document, control margins for FO and PDF, enable a table of contents, and specify the bibliography format.

The `dynDoc()` function passes control to the XSL engine. An XSL transformation is made up of a collection of templates. Each template identifies to which XML nodes it applies, and actions that process that XML node and creates new output. For example, the XSL template for a <title> node when rendering HTML would create an <h1> node and then process its child nodes, that is, the text of the title including any child nodes such as links and formatting. We can extend and override any XSL template by providing our own XSL style sheet and specifying templates that match particular XML nodes. (The Sxslt package also allows us to provide XSL templates locally within the XML document, like macro definitions for LaTeX.)

By integrating R and the XSL engine, we have the ability to call R functions from XSL templates. We can pass XML nodes from XSL template actions to R functions in order to generate content for the output document. We use this to implement the XSL templates for <r:code> and other XML elements. We pass the node to an R function that extracts the code, evaluates it, and then converts the result(s) to the target format. As with Word, there is a generic function (`convert()`) that transforms the R object to that format and one can define methods for different R types and different targets (HTML, Docbook, FO, text).

We can also implement facilities such as caching computations by providing our own XSL templates.

This XDynDocs package is available from the Omegahat Web site as an R package. It can be installed with the command:

```
install.packages ("XDynDocs", repos = "http://www.omegahat.org/R",
                  dependencies = TRUE)
```

This will take care of installing the necessary XML and Sxslt packages. While the approach may seem very complex for users of word processors, it will be quite familiar to LaTeX users. More importantly, we feel it provides a very rich and flexible framework for working with documents in very new ways.

20.4 Future Work

The work we have described provides the foundations for more ambitious work on richer structured documents. The aim is to capture more aspects of computational-based research that makes the process significantly more reproducible and informative to the researchers, collaborators, reviewers, and general audience. We have been working on ideas for representing the research process and identifying different alternative approaches to analyses within the document. We have also been developing interactive techniques for readers to be able to explore a dynamic document at different levels of resolution. We have done some work on making dynamic documents interactive by providing interactive controls that the reader can manipulate to change the computations at run-time, for example, vary tuning parameters in statistical methods or introduce alternative data sets. This allows them to do "what-if" analysis and to explore different ideas from what the author presents. We have embedded Web browsers within R and R within browsers and hope to soon make these robust so that researchers can use these to publish and disseminate in rich new ways.

References

Leisch F (2002) Sweave: dynamic generation of statistical reports using literate data analysis. In: Wolfgang Hardle BR (ed) Compstat 2002 proceedings in computational statistics. Physica Verlag, Heidelberg, pp 575–580

Max Kuhn SW (2008) odfWeave: sweave processing of open document format (odf) files

Mittelbach F, Goossens M et al (2006). The LaTeX companion, 2nd edn. Addison Wesley, Boston, MA

Pawson D (2002) XSL-FO: making XML look good in print. O'Reilly, Sebastopol, CA

Peng RD (2008) Caching and distributing statistical analyses in R. J Stat Software 26(7):1–24

Ramsey N (1994) Literate programming simplified. IEEE Software 11(5):97–105

R Development Core Team (2008) A language and environment for statistical computing, Vienna, Austria

Vugt WV (2007) Open XML: the markup explained. Retrieved 2009, from http://openxmldeveloper.org: http://openxmldeveloper.org/attachment/1970.ashx

Walsh N, Muellner L (1999) DocBook: The definitive guide. O'Reilly, Sebastopol, CA

Index