Riccardo Rizzo
Paulo J.G. Lisboa (Eds.)

LNBI 6685

Computational Intelligence Methods for Bioinformatics and Biostatistics

7th International Meeting, CIBB 2010
Palermo, Italy, September 2010
Revised Selected Papers

2010
CiBB

Springer

Lecture Notes in Bioinformatics 6685

Edited by S. Istrail, P. Pevzner, and M. Waterman

Editorial Board: A. Apostolico S. Brunak M. Gelfand

T. Lengauer S. Miyano G. Myers M.-F. Sagot D. Sankoff

R. Shamir T. Speed M. Vingron W. Wong

Subseries of Lecture Notes in Computer Science

Riccardo Rizzo   Paulo J.G. Lisboa (Eds.)

Computational Intelligence Methods for Bioinformatics and Biostatistics

7th International Meeting, CIBB 2010
Palermo, Italy, September 16-18, 2010
Revised Selected Papers

Springer

Series Editors

Sorin Istrail, Brown University, Providence, RI, USA
Pavel Pevzner, University of California, San Diego, CA, USA
Michael Waterman, University of Southern California, Los Angeles, CA, USA

Volume Editors

Riccardo Rizzo
ICAR-CNR
Consiglio Nazionale delle Ricerche
90128 Palermo, Italy
E-mail: ricrizzo@pa.icar.cnr.it

Paulo J.G. Lisboa
Liverpool John Moores University
School of Computing and Mathematical Sciences
Liverpool, L3 3AF, UK
E-mail: p.j.lisboa@ljmu.ac.uk

ISSN 0302-9743 e-ISSN 1611-3349
ISBN 978-3-642-21945-0 e-ISBN 978-3-642-21946-7
DOI 10.1007/978-3-642-21946-7
Springer Heidelberg Dordrecht London New York

Library of Congress Control Number: 2011931466

CR Subject Classification (1998): F.1, J.3, I.4, I.2, I.5, F.2

LNCS Sublibrary: SL 8 – Bioinformatics

© Springer-Verlag Berlin Heidelberg 2011
This work is subject to copyright. All rights are reserved, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, re-use of illustrations, recitation, broadcasting, reproduction on microfilms or in any other way, and storage in data banks. Duplication of this publication or parts thereof is permitted only under the provisions of the German Copyright Law of September 9, 1965, in its current version, and permission for use must always be obtained from Springer. Violations are liable to prosecution under the German Copyright Law.
The use of general descriptive names, registered names, trademarks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India

Printed on acid-free paper

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

Preface

This volume contains selected contributions delivered at the 7th International Meeting on Computational Intelligence Methods for Bioinformatics and Biostatistics (CIBB 2010) held in Palermo, Palazzo Comitini, during September 16–18, 2010.

The CIBB meeting series is organized by the Special Interest Group on Bioinformatics of the International Neural Networks Society (INNS) to provide a forum open to researchers from different disciplines to present and discuss problems concerning computational techniques in bioinformatics, system biology and medical informatics with a particular focus on neural networks, machine learning, fuzzy logic, and evolutionary computational methods. From 2004 to 2007, CIBB meetings were held with an increasing number of participants in the format of a special session of bigger conferences, namely, WIRN 2004 in Perugia, WILF 2005 in Crema, FLINS 2006 in Genoa, and WILF 2007 in Camogli. With the great success of the special session at WILF 2007 that included 26 strongly rated papers, we launched the first autonomous CIBB conference edition starting with the 2008 conference in Vietri.

CIBB 2010 attracted 24 papers submissions from all over the world. A rigorous peer-reviewed selection process was applied to ultimately select the papers included in the program of the conference. This volume collects the best contributions presented at the conference. Moreover, the volume also includes two presentations from keynote speakers and one tutorial presentation.

The success of CIBB 2010 is to be credited to the contribution of many people. First, we would like to thank the organizers of the special sessions for attracting so many good papers which extended the focus of the main topics of CIBB. Second, special thanks are due to the Program Committee members and reviewers for providing high-quality reviews. Last but not least, we would like to thank the keynote speakers Raffaele Giancarlo (University of Palermo, Italy), Paulo J.G. Lisboa (John Moores University, UK), and Gianluca Pollastri (University College of Dublin, Ireland).

April 2011

Paulo J.G. Lisboa
Riccardo Rizzo

Organization

The 7th CIBB meeting was a joint operation of the Special Interest Groups on Bioinformatics and Biopatterns of INNS and of the Task Force on Neural Networks of the IEEE CIS Technical Committee on Bioinformatics and Bioengineering with the collaboration of the Istituto di Calcolo e Reti ad Alte Prestazioni (ICAR), CNR, Palermo, Italy; Provincia Regionale of Palermo; the Dipartimento di Matematica ed Informatica, University of Palermo, Italy; and supported by the Fondazione Banco di Sicilia, Palermo, Italy.

Conference Chairs

Paulo J.G. Lisboa	John Moores University, Liverpool, UK
Riccardo Rizzo	Istituto di Calcolo e Reti ad Alte Prestazioni (ICAR), sede di Palermo, CNR, Palermo, Italy

Biostatistics Technical Chair

Elia Biganzoli	Istituto Nazionale per lo Studio e la Cura dei Tumori, Milan, Italy

Bioinformatics Technical Chair

Francesco Masulli	University of Genoa, Italy and Temple University Philadelphia, USA

CIBB Steering Committee

Pierre Baldi	University of California Irvine, USA
Alexandru Floares	Oncological Institute, Cluj-Napoca, Romania
Jon Garibaldi	University of Nottingham, UK
Francesco Masulli	University of Genoa, Italy and Temple University Philadelphia, USA
Roberto Tagliaferri	University of Salerno, Italy

Scientific Committee

Sansanee Auephanwiriyakul	Chiang Mai University, Thailand
Federico Ambrogi	University of Milan, Italy
Claudia Angelini	IAC-CNR Naples, Italy
Sanghamitra Bandyopadhyay	Indian Statistical Institute, Kolkata, India

Gilles G. Bernot	University of Nice Sophia Antipolis, France
Chengpeng Bi	Childrens Mercy Hospital, Kansas City, USA
Mario Cannataro	University Magna Graecia of Catanzaro, Italy
Xuewen Chen	University of Kansas, Lawrence, USA
Giuseppe Di Fatta	The University of Reading, UK
Enrico Formenti	University of Nice Sophia Antipolis, France
Salvatore Gaglio	University of Palermo, Italy
Christoph Friedrich	Fraunhofer Institute for Algorithms and Scientific Computing, Sankt Augustin, Germany
Antonio Giordano	University of Siena, Italy and Sbarro Institute for Cancer Research and Molecular Medicine, Center for Biotechnology, Temple University, Philadelphia, USA
Saman Halgamuge	The University of Melbourne, Australia
Pietro Lio'	University of Cambridge, UK
Michael Lones	University of York, UK
Elena Marchiori	Radboud University Nijmegen, The Netherlands
Giancarlo Mauri	University of Milano-Bicocca, Italy
Luciano Milanesi	ITB-CNR Milan, Italy
David A. Pelta	University of Granada, Spain
Leif E. Peterson	Methodist Hospital Research Institute, Houston, USA
Gianluca Pollastri	University College Dublin, Ireland
Mihail Popescu	University of Missouri - Columbia, USA
Volker Roth	University of Basel, Switzerland
Giuseppe Russo	Sbarro Institute for Cancer Research and Molecular Medicine, Center for Biotechnology, Temple University, USA
Jennifer Smith	Boise State University, USA
Federico Mattia Stefanini	University of Florence, Italy
Alfonso Urso	ICAR-CNR Palermo, Italy
Giorgio Valentini	University of Milan, Italy
Kay Wiese	Simon Fraser University, Surrey, Canada
Yanqing Zhang	Georgia State University, Atlanta, USA

Special Session Organizers

A. Floares, L. Peterson	*Intelligent Clinical Decision Support Systems (i-CDSS)*
V.P. Plagianakos, D.K. Tasoulis	*Data Clustering and Bioinformatics*

Referees

F. Ambrogi
C. Angelini
S. Auephanwiriyakul
C. Bi
M. Cannataro
E. Formenti
C.M. Friedrich
M. Lones
G. Mauri
L. Milanesi
D. Pelta
L. Peterson
G. Pollastri
M. Popescu
G. Russo
J. Smith
A.M. Urso

Local Organizing Committee

Alfonso Urso	Istituto di Calcolo e Reti ad Alte Prestazioni (ICAR), sede di Palermo, CNR, Palermo, Italy
Pietro Storniolo	Istituto di Calcolo e Reti ad Alte Prestazioni (ICAR), sede di Palermo, CNR, Palermo, Italy
Giosué Lo Bosco	Dipartimento di Matematica ed Informatica, University of Palermo, Italy

Congress Management

Fabio Ferrara	Istituto di Calcolo e Reti ad Alte Prestazioni (ICAR), sede di Palermo, CNR, Palermo, Italy
Giampiero Rizzo	Istituto di Calcolo e Reti ad Alte Prestazioni (ICAR), sede di Palermo, CNR, Palermo, Italy

Financing Institutions

Provincia Regionale di Palermo, Italy
Department of Information and Communication Technology, CNR, Italy
Istituto di Calcolo e Reti ad Alte Prestazioni (ICAR), CNR, Italy
E4 Computer Engineering, Italy
Informatica Commerciale, Palermo, Italy
MISCO, Italy

Table of Contents

Tutorial

Plenary Talks

Sequence Analysis, Promoter Analysis and Identification of Transcription Factor Binding Sites

Methods for the Unsupervised Analysis, Validation and Visualization of Structures Discovered in Bio-molecular Data – Prediction of Secondary and Tertiary Protein Structures

Gene Expression Data Analysis

Bio-medical Text Mining and Imaging – Methods for Diagnosis and Prognosis

Mathematical Modelling and Simulation of Biological Systems

Intelligent Clinical Decision Support Systems (i-CDSS)

Management and Analysis of Protein-to-Protein Interaction Data

Mario Cannataro* and Pietro Hiram Guzzi

Department of Experimental Medicine and Clinic,
University Magna Græcia of Catanzaro, Italy
{cannataro,hguzzi}@unicz.it

Abstract. This paper introduces technologies, standards and databases for generating, representing and storing Protein-to-Protein Interaction (PPI) data. Emerging algorithms for the analysis, comparison and knowledge extraction from Protein-to-Protein Interaction Networks (PINs) are also presented. Finally, the paper presents a methodology for the ontology-based analysis of Protein-to-Protein Interaction data by discussing a real case study.

Keywords: Protein-to-Protein Interaction Data, Protein-to-Protein Interaction Networks, Protein Complex Prediction, Motifs Extraction, Network Alignment, Gene Ontology.

1 Introduction

A recent trend in biology and medicine tries to elucidate the behaviour of complex systems looking at their elementary building blocks and the relationships among them.

For instance, the study of complex biological systems, such as the cells, may be faced by studying both their basic components, e.g. their proteins, and the way they interact each other, e.g. protein interactions. Interactomics is a new discipline in the omics world that studies the interactome, i.e. the set of all interactions among biological molecules taking place in an organism. In this multidisciplinary scenario wet lab experiments are used to produce data that are examined *in silico* with computational methods that try to explain the behaviour of biological systems.

Considering the analysis of interactions among proteins that take place in living organisms, we have four main tasks: (i) experimental assays to produce PPI data, (ii) representation, storage, querying and analysis of PPI data, (iii) bioinformatics methods for the analysis of PINs, (iv) mathematics models to describe PINs. Consequently the flow of data in interactomics can be schematized as depicted in Figure 1. Wet-lab technologies are used to produce data that are stored in different databases. Then data are modelled by using graphs and usually the

* Corresponding author.

R. Rizzo and P.J.G. Lisboa (Eds.): CIBB 2010, LNBI 6685, pp. 1–12, 2011.
© Springer-Verlag Berlin Heidelberg 2011

whole set of interactions of an organism is merged together in a comprehensive protein interaction network represented as a graph. Finally, computational methods mine such graphs producing biological meaningful knowledge [7].

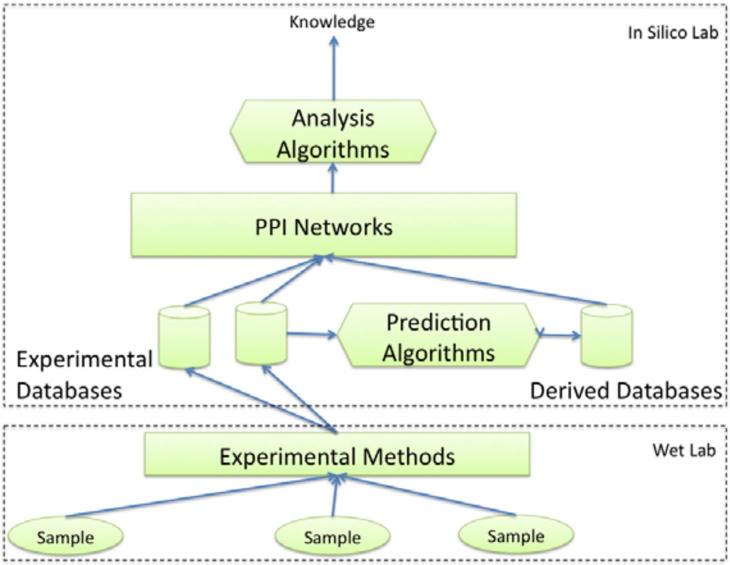


Fig. 1. Protein interactions are discovered through wet lab experiments and stored in Experimental Databases. Prediction algorithms produce derived data stored in Databases of predicted interactions. The combination of both data forms PPI networks that are further analysed by graph-based algorithms that extract knowledge from PPI data and PINs.

Following this flow of information, this paper describes the main data management and analysis approaches regarding protein-to-protein interactions using a bottom-up approach: from data generation, performed through wet lab technologies, to data representation, mainly based on XML-based standards, to data storage and querying, offered by a large set of protein-to-protein interaction databases, and finally to protein-to-protein interaction networks representation, analysis and visualization, offered by specialized algorithms and sophisticated visualization tools.

The rest of the paper is organised as follows: Section 2 summarises experimental methods for PPI data generation and standards for data representation and exchange. Section 3 presents PPI databases. Section 4 describes main algorithms for analysing and comparing PINs, while Section 5 presents tools for their visualisation. Finally, Section 6 presents an ontology-based methodology for the analysis of PPI data produced in Mass Spectrometry (MS) experiments. Section 7 concludes the paper and outlines future work.

2 Generation and Management of Protein-to-Protein Interaction Data

2.1 Experimental Methods for PPI Data Generation

The accumulation of protein interaction data for building a comprehensive network is an iterative process that requires many different experiments. Each experiment may reveal a binary or a multiple interaction, i.e. an interaction regarding three or more proteins, so a complete investigation requires the planning of a set of assays. Each determined interaction moreover has a given probability to really exist in real organism.

Considering the experimental techniques we here report two main procedures: (i) the yeast-two-hybrid (Y2H) technique, and (ii) mass-spectrometry. These experimental platforms share a general schema in which a so called *bait* protein is used as test to demonstrate its relations with one or more proteins said *preys.* Both single interactions and exhaustive screenings have been realized following this schema [18].

With respect to the quality of produced data, each assay can be evaluated on the basis of the number of false positives, i.e. the number of false interactions that are determined. Such parameter is usually referred to as *reliability* in literature. Usually reliability is calculated on the basis of complex mathematical models that takes into account many different parameters. Consequently reliability is defined as the fraction of real interactions with respect to the interactions reported in previous datasets and measurements are based mainly on the correlation of gene expression data. Three parameters are evaluated: (i) the distribution of gene expression correlation coefficients, (ii) the reliability based on gene expression correlation coefficients, and (iii) the accuracy of function predictions [10].

2.2 Standards for PPI Data

As a consequence of many experiments performed using different techniques, the amount of data and information regarding protein-protein interactions at the proteomic level is increasing in a constant way. This increase results hence in two main effects: (i) an accumulation of a large amount of data in existing databases, (ii) the introduction of new databases focusing, for instance, on a single organism, or on the integration with other sources of biological information.

As a consequence, researchers that need to retrieve data about interactions have to face not only with different data but also with different data sources and formats. The scenario is made more complicated by the absence of a common accepted systems of identifiers for interactions that are identified using the identifiers of the interactors that may be encoded using different database identifiers.

In order to standardize the representation of interactions, different standards have been proposed: the HUPO PSI-MI [20], the main emerging standard for storing and exchanging PPI data, and IMEx [27], an ongoing effort based on

HUPO PSI-MI, that aims to standardize the process of data curation and exchange between interaction databases, as happened in genomics.

The former, referred to as PSI-MI XML 1.0, has represented the first step towards the introduction of a standard for representing molecular interaction data but focused only on protein interaction data. It has been based on the use of XML as encoding language and on the use of controlled vocabularies for representing concepts. Successively, it has been extended to capture also interaction among other molecules (e.g. enzymes and nucleic acids) and recently the 2.5 version (namely PSI-MI XML2.5) has been released. Successively, a tabular version of this format, for allowing a more efficient way to exchange data has been introduced and implemented by the PSI-MI organization. This format, referred to as MITAB 2.5 [20], is a tabular format and provides a simple representation of a dataset. Both standards are used for data exchange and download but they lack in the definition of a workflow for data sharing and curation among interaction databases.

The International Molecular Exchange (IMEx) Consortium is a project that aims to develop both standards and tools to manage the process of data curation and exchange between interaction databases. It is based on the HUPO PSI-MI format for data encoding. Databases that participate in this consortium accept the deposition of interaction data from authors, helping the researcher to annotate the dataset through a set of ad hoc developed tools. Partners of IMEX produce separately their data and maintain them at first, then by using an ad hoc realized network structure, they make available all the data following the IMEx rules. Finally, the end user can retrieve such data by using a single interface available through the IMEx webserver.

3 Protein-to-Protein Interaction Databases

The accumulation of protein interaction data caused the introduction of several databases. Here we distinct on **databases of experimental determined interactions**, that include all the databases storing interactions extracted from both literature and high-throughput experiments, and databases of **predicted interactions** that store data obtained by in silico prediction. Another important class that we report is constituted by **integrated databases** or meta-databases, i.e. databases that aim to integrate data stored in other publicly available datasets. Currently, there exist many databases that differ on biological and information science criteria: the covered organism, the kind of interactions, the kind of interface, the query language, the file format and the visualization of results.

Data produced in low or high-throughput experiments are stored in **databases of experimental determined interactions** after a successive verification by a committee of database curators. Researchers can submit directly their own data to the databases, e.g. to Intact [16], or they can publish data as happens in the literature and then the database curators will extract them, e.g. MINT database [35]. Table 1 summarizes main existing databases, for a more complete description see the annual issue of the Nucleic Acid Research Journal [29].

For simpler organisms, such as yeast, worm or fly, the process of the whole coverage of the interaction network seems to be almost completed. This process caused the introduction of a huge amount of data that may be mined for many objectives. Conversely, the complexity of the interactomes of higher eukaryotes has prevented these experiments for humans. From this scenario the need for the introduction of algorithms and tools able to use the experimental data to predict protein-interactions arose. Thus, starting from existing databases of verified interactions, a number of algorithms have been developed to predict putative interactions that are accumulated into **databases of predicted interactions**. The common approach is based on the consideration that the interaction mechanisms are conserved through the evolution, i.e. if two protein A and B interact in a simple organism, then the corresponding orthologs proteins A_1, and B_1 may interact in a complex organism. Thus, starting from the interacting proteins in a simple organism, prediction are made for other organisms.

Although the existence of many databases the resulting amount of data presents three main problems [9]: the low overlap among databases, the resulting lack of completeness with respect to the real interactome and the absence of integration. Consequently, in order to perform an exhaustive data collection, (e.g. for an experiment), researchers should query manually different data sources. This problem is faced with the introduction of databases based on the integration of existing ones. Nevertheless, in the interactomics field, the integration of existing databases is not easy to solve.

The integration of data from different laboratories and sources can be done through the adoption of an accepted interaction identifiers system. It should be noted that while in other biological database systems, such as the sequence databases, there exists a common system of identifiers, and cross-reference is used to retrieve the same biological entity from different databases, PPI interactions are currently not identified by a unique identifier, but through the names of corresponding partners.

Although the existence of these problems, different approaches for data integration and the building of larger interaction maps have been proposed. The rationale of these approaches is based on a three-step process: (i) collection of data from different datasources; (ii) transformation of data into a common model; (iii) annotation and scoring of the resulting dataset.

All the existing databases go beyond the storing of the interactions, but integrate them with functional annotations, sequence information and references to corresponding genes. Finally, they generally provide some visualization that presents a subset of interactions in a comprehensive graph.

Nevertheless, the current scenario has some common problems and characteristics that are shared from almost all the databases: (i) errors in the databases, (ii) lack of naming standards, and (iii) little overlap among interactions.

Any published dataset may contain errors so any database may contain false interactions, often called false positives, i.e. proteins erroneously reported as interacting. This may be due, for instance, to technical, (i.e. false positives that are due to the detection method), and biological problems, (i.e. proteins that are reported to be interacting in vitro but they are never co-located).

In other biological database communities, such as those storing protein sequences or structures, there exist many projects providing common accepted identifiers for biological object, or at least a system for the cross-references of the same object in almost all the databases. In interactomics there is not a common identifier, and in general interaction are not identified by a single code but using the identifiers of interacting proteins.

It has been noted [9] that existing databases present little overlap with respect to the dimension of the interactomes. Despite this, the integration of databases is still an open problem due to the difficulties resulting from the absence of a naming standard.

Conversely, common aspects of existing datasets are: (i) simple web-based interface for querying, (ii) simple visualization of results in both tabular and graphical way, (iii) data are available for download in different format. It should be noted that almost all the databases offer the user the possibility of retrieving data and some annotations through a simple web-based interface. Despite this, the querying of protein networks aims to go beyond the simple retrieval of a set of interactions stored in databases. Databases can actually be queried through simple key-based searches, e.g. by inserting one or more protein identifiers.

The output of such a query is in general a list of interacting protein pairs. These pairs share a protein, the query one. Such an approach, despite the conceptual simplicity and the easy practical use, presents some limitations. Let us consider, for instance, a researcher that would compare patterns of interactions among species or a researcher that would search interactions related to a given biological compartment or a biological process. The existing query interfaces, in general, do not enables such queries.

Thus a more powerful querying system should provide a semantically more expressive language, e.g. retrieve all the interaction patterns that share the same structure. Then the query system should map the query, expressed in an high-level language (e.g., using a graph formalism), into suitable graph structures and should search for them by applying appropriate algorithms. Unfortunately this problem is not easy from a computational point of view, and it requires: (i) the modeling of the PPI network in a suitable data structure; (ii) the existence of appropriate algorithms for mapping, that is, the identification of the correspondence of nodes in a subnetwork and those stored in the database [36].

Table 1. Main Databases and Support of PSI-MI standard

Databases	
Supporting PSI-MI	Non supporting PSI-MI
DIP [30], MINT [8] MIPS [7], INTACT [2]	BIND [1] I2D [6] IntNetDB [34] STRING [33]

4 Analysis and Comparison of Protein-to-Protein Interaction Networks

PPI data can be represented by using graphs [12,14], where nodes are associated to proteins, and edges represent interactions among proteins. The most simple representation uses undirected graph, while more refined models use directed and labeled edges to integrate the information about the kind of biochemical association and its direction. Starting from a dataset of binary interactions a graph is easily built in an iterative way. As starting point, the global topology of an interaction network, i.e. the study of the clustering coefficient or of the diameter, can reveal main properties of the network. In addition to analysis of global properties, the study of recurring local topological features and the extraction of relevant modules, i.e. cliques, has found an increasing interest. For instance, it has been demonstrated that dense subgraphs as well as cliques may be associated to protein complexes, a set of mutually interacting proteins that play important biological roles [4].

For the purposes of this work, we categorize these studies in two main classes: (i) algorithms that mine a single interaction network, and (ii) algorithms for the comparison of two or more networks, also referred to as network alignment algorithms. Methods belonging to the first class try to extract motifs, i.e. recurrent regions in a graph, under the hypothesis that they could encode biological meaningful objects. In this case the structure of the motif, i.e. clique, quasi-clique, linear path, and stars determine the nature of biological knowledge. Differently, algorithms belonging to the second class compare two or more networks evidencing conserved substructures, (*local alignment algorithms*), or global similarities, (*global alignment algorithms*). Tables 2 and 3 summarize, respectively, main algorithms for network analysis and comparison.

The interest in finding motifs in networks resides on two main reason: (i) individuating small subnetworks that play important roles, and (ii) unravelling the evolutionary mechanism. The approach for studying protein networks is similar to biological sequence analysis in which the motif analysis has determined the existence of particular subsequences playing important biological roles [24]. For instance an important class of algorithms is deputed to the prediction of protein complexes, i.e. set of mutually interacting proteins. Starting from a PPI network, complexes may be identified by searching for small and highly interconnected regions, said *cliques*. Predicted complexes can be already known, i.e. their composition are known, or can denote a new protein complex. In this case, if the experiments confirm this relation, the algorithms can be used as predictors.

Finally, algorithms belonging to the second class investigate conservation and divergence of interactions between different species [5], so they usually receive in input two or more PPI networks (i.e. two or more graphs) and produce as output a set of conserved subgraphs among them. We can organize the existing algorithms on the basis of different criteria, considering for instance the number of the input networks (pairwise or multiple), the topology of the revealed structure (linear paths or dense subnetworks), the alignment strategy (local or global), or the goal (prediction of orthologs or identification of conserved

Table 2. Network Analysis Algorithms

Algorithm	*Task*	*Method*
MCODE [4]	Complex Prediction	Clustering
MCL [11]	Complex Prediction	Clustering
RNSC [21]	Complex Prediction	Clustering

Table 3. Network Alignment Algorithms

Name	Description
PathBlast[19]	Pairwise Local Alignmnent
Mawish [22]	Pairwise Local Alignmnent
Graemlin [13]	Pairwise and Multiple Local Alignmnent
ISORANK [23]	Multiple Global Alignmnent

subnetworks). The **local alignment strategy** aims to find many correspondences among small subnetworks of the input ones. Such subnetworks correspond to conserved patterns of interaction that can represent conserved functional components, e.g. complexes or pathways.

For instance, PathBlast [19] is a pairwise alignment algorithm that aims to extract conserved linear paths among two species, while MaWish [22] aims to find locally high similar subgraphs, and Graemlin [13], generalizes the previous approaches allowing the search of more general topologies with respect to linear paths and dense subnetworks in two or more organisms. ISORANK [31] is a global alignment algorithm used to find global similarities among networks that reveal functional orthologs across the input networks.

5 Visualization of Large Protein-to-Protein Interaction Networks

Although graphs are well known and studied data structures, a main complexity in the visualization of protein interaction networks is related to the high number of nodes and connections. Another issue regards the heterogeneity of nodes (proteins) and edges (interactions), since in many applications may be useful to represent different classes of proteins/interactions with different colours. Finally, annotation of proteins and interactions enriches the protein interaction networks with functional information, thus complicating their visualization.

Many software tools for the visualization of protein interaction networks have been developed. They offer basic visualisation of PINs and recently they have been enriched with new functions for PPI data management and PIN analysis. A current trend is the deployment of open, extensible visualization tools (e.g. Cytoscape), that may be incrementally enriched by the interactomics community through the development of plugins [28].

6 Ontology-Based Analysis of Protein-to-Protein Interaction Data

A typical experimental workflow of interactomics starts in the wet lab with the determination of one or more interactions among proteins. Then such interactions can be integrated with interactions already known stored in databases. This process causes the formation of protein interaction networks. Finally, protein interaction networks modelled as graphs can be mined to obtain biological relevant information. This case study describes a typical workflow of analysis of an interaction network obtained from a proteomic experiment. It starts discussing the technological platform that has produced such data. Then it explains the reconstruction of the networks through the integration of experimental data and databases. The whole process of analysis can be structured as depicted in Figure 2 (see [25,26] for more details).

The experiment starts with the data production by using, for instance, Tandem Affinity Purification coupled to Mass Spectrometry (MS-TAP) [32]. The results of a MS-TAP experiment is a set of identified proteins organized in a list, the **Dataset** hereafter. Starting from the Dataset, a network can be iteratively built by querying publicly available databases. The search can be delimited only to the interactions involving proteins within **Dataset**, or can be expanded to the interactions regarding a pair of proteins where one belongs to Dataset. After the identification of the interaction a network can be built and visualized, for example in Cytoscape. At this point, the interaction network can be analysed to extract main topological parameters. Common measurements are, but are not limited to, the following parameters: number of nodes (N) and edges (E), average clustering coefficient (cc), node-degree (k) and its distribution among nodes, average node degree avk, diameter (d) and closeness centrality of each node (ccl) [3].

Finally, the network can be mined in order to individuate the biological mechanisms and processes that are related to the Dataset. Such analysis, often referred

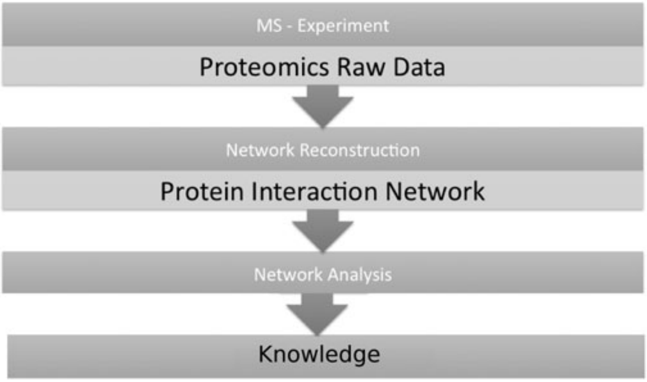


Fig. 2. Workflow of analysis of a PIN reconstructed from a proteomic experiment

to as **functional enrichment analysis**, aims to individuate a set of shared annotations among selected proteins, in order to assign a biological meaning to the selected genes/proteins. Annotations are often stored as controlled vocabularies or organized in taxonomies, e.g. Gene Ontology is a controlled-vocabulary of the molecular-biology domain to describe and organize hierarchically concepts [15]. There exists more than 60 bioinformatics tool that perform such analysis that can be cathegorized on the basis of different criteria [17]: the kind of statistical model, the annotation databases, or the kind of biological data in input.

7 Conclusion and Future Work

Analysis of protein interaction network is becoming a wide research area. This paper surveyed the whole workflow of analysis of interaction networks, from wet lab to knowledge. Paper initially introduced technologies, standards and databases for generating, representing and storing Protein-to-Protein Interaction Data. Then main algorithms of analysis are discussed, evidencing open issues and future challenges. Finally, the paper presented a methodology for the ontology-based analysis of Protein-to-Protein Interaction data.

Acknowledgments. We wish to thank Pierangelo Veltri, who shared with us many discussions and work, Concettina Guerra and Alessandro Weisz who collaborated with us on some research activities in interactomics.

References

1. Alfarano, C., Andrade, C.E., Anthony, K., Bahroos, N., Bajec, M., Bantoft, K., Bete, D., Bobechko, B., Boutilier, K., Burgess, E., Buzadzija, K., Cavero, R., D'Abreo, C., Donaldson, I., Dorairajoo, D., Dumontie, M.J., Dumontier, M.R., Earles, V., Farral, R., Feldman, H., Garderman, E., Gong, Y., Gonzaga, R., Grytsan, V., Gryz, E., Gu, V., Haldorsen, E., Halupa, A., Haw, R., Hrvojic, A., Hurrell, L., Isserlin, R., Jack, F., Juma, F., Khan, A., Kon, T., Konopinsky, S., Le, V., Lee, E., Ling, S., Magidin, M., Moniakis, J., Montojo, J., Moore, S., Muskat, B., Ng, I., Paraiso, J.P., Parker, B., Pintilie, G., Pirone, R., Salama, J.J., Sgro, S., Shan, T., Shu, Y., Siew, J., Skinner, D., Snyder, K., Stasiuk, R., Strumpf, D., Tuekam, B., Tao, S., Wang, Z., White, M., Willis, R., Wolting, C., Wong, S., Wrong, A., Xin, C., Yao, R., Yates, B., Zhang, S., Zheng, K., Pawson, T., Ouellette, B.F., Hogue, C.W.: The biomolecular interaction network database and related tools 2005 update. Nucleic Acids Res. 33(database issue), 418–424 (2005)
2. Aranda, B., Achuthan, P., Alam-Faruque, Y., Armean, I., Bridge, A., Derow, C., Feuermann, M., Ghanbarian, A.T., Kerrien, S., Khadake, J., Kerssemakers, J., Leroy, C., Menden, M., Michaut, M., Montecchi-Palazzi, L., Neuhauser, S.N., Orchard, S., Perreau, V., Roechert, B., van Eijk, K., Hermjakob, H.: The IntAct molecular interaction database in 2010. Nucleic Acids Research 38(database issue), gkp878–531 (2010)
3. Assenov, Y., Ramirez, F., Schelhorn, S.-E., Lengauer, T., Albrecht, M.: Computing topological parameters of biological networks. Bioinformatics 24(2), 282–284 (2008)

4. Bader, G., Hogue, C.: An automated method for finding molecular complexes in large protein interaction networks. BMC Bioinformatics 4(1), 2 (2003)
5. Berg, J., Lassig, M.: Local graph alignment and motif search in biological networks. Proc. Natl. Acad. Sci. 41(101), 14689–14694 (2004)
6. Brown, K., Jurisica, I.: Unequal evolutionary conservation of human protein interactions in interologous networks. Genome Biology 8(5), R95+ (2007)
7. Cannataro, M., Guzzi, P.H., Veltri, P.: Protein-to-protein interactions: Technologies, databases, and algorithms. ACM Comput. Surv. 43, 1:1–1:36 (2010)
8. Chatr-Aryamontri, A., Ceol, A., Palazzi, L.M., Nardelli, G., Schneider, M.V., Castagnoli, L., Cesareni, G.: Mint: the molecular interaction database. Nucleic Acids Res. 35, D572–D574 (2007)
9. Chaurasia, G., Iqbal, Y., Hanig, C., Herzel, H., Wanker, E.E., Futschik, M.E.: UniHI: an entry gate to the human protein interactome. Nucl. Acids Res. 35(suppl. 1), D590–D594 (2007)
10. Deng, M., Sun, F., Chen, T.: Assessment of the reliability of proteinprotein interactions and protein function prediction. In: Proc. of Pacific Symposium Biocomputing, Grand Wailea, Maui, Hawaii, pp. 140–151. World Scientific, Singapore (2003)
11. Enright, S., Van Dongen, A.J., Ouzounis, C.A.: An efficient algorithm for large-scale detection of protein families. Nucleic Acids Research 30(7), 1575–1584 (2002)
12. Fell, D.A., Wagner, A.: The small world of metabolism. Nat. Biotechnol. 18(11), 1121–1122 (2000)
13. Flannick, J., Novak, A., Srinivasan, B.S., McAdams, H.H., Batzoglou, S.: Graemlin: general and robust alignment of multiple large interaction networks. Genome Res. 16(9), 1169–1181 (2006)
14. Golumbic, M.C.: Algorithmic graph theory and perfect graphs. Academic Press, New York (1980)
15. Harris, M.A., et al.: The gene ontology (go) database and informatics resource. Nucleic Acids Res. 32(database issue), 258–261 (2004)
16. Hermjakob, H., Montecchi-Palazzi, L., Lewington, C., Mudali, S., Kerrien, S., Orchard, S., Vingron, M., Roechert, B., Roepstorff, P., Valencia, A., Margalit, H., Armstrong, J., Bairoch, A., Cesareni, G., Sherman, D., Apweiler, R.: Intact: an open source molecular interaction database. Nucleic Acids Res. 1(32)(database issue), 452–455 (2004)
17. Huang, D.W., Sherman, B.T., Lempicki, R.A.: Bioinformatics enrichment tools: paths toward the comprehensive functional analysis of large gene lists. Nucleic Acids Research 37(1), 1–13 (2009)
18. Ito, T., Chiba, T., Ozawa, R., Yoshida, M., Hattor, M., Sakaki, Y.: A comprehensive two-hybrid analysis to explore the yeast protein interactome. Proc. Natl. Acad. Sci. USA 98, 4569–4574 (2001)
19. Kelley, B.P., Yuan, B., Lewitter, F., Sharan, R., Stockwell, B.R., Ideker, T.: PathBLAST: a tool for alignment of protein interaction networks. Nucl. Acids Res. 32(suppl. 2), W83–W88 (2004)
20. Kerrien, S., Orchard, S., Montecchi-Palazzi, L., Aranda, B., Quinn, A., Vinod, N., Bader, G., Xenarios, I., Wojcik, J., Sherman, D., Tyers, M., Salama, J., Moore, S., Ceol, A., Chatraryamontri, A., Oesterheld, M., Stumpflen, V., Salwinski, L., Nerothin, J., Cerami, E., Cusick, M., Vidal, M., Gilson, M., Armstrong, J., Woollard, P., Hogue, C., Eisenberg, D., Cesareni, G., Apweiler, R., Hermjakob, H.: Broadening the horizon - level 2.5 of the hupo-psi format for molecular interactions. BMC Biology 44(1), 44 (2007)

21. King, A.D.: Graph clustering with restricted neighbourhood search, Master's thesis, University of Toronto (2004)
22. Koyutürk, M., Kim, Y., Subramaniam, S., Szpankowski, W., Grama, A.: Detecting conserved interaction patterns in biological networks. J. Comput. Biol. 13(7), 1299–1322 (2006)
23. Liao, C.-S., Lu, K., Baym, M., Singh, R., Berger, B.: IsoRankN: spectral methods for global alignment of multiple protein networks. Bioinformatics 25(12), i253–i258 (2009)
24. Milo, R., Shen-Orr, S., Itzkovitz, S., Kashtan, N., Chklovskii, D., Alon, U.: Network motifs: simple building blocks of complex networks. Science 298(5594), 824–827 (2002)
25. Nassa, G., Tarallo, R., Ambrosino, C., Bamundo, A., Ferraro, L., Paris, O., Ravo, M., Guzzi, P.H., Cannataro, M., Baumann, M., Nyman, T.A., Nola, E., Weisz, A.: A large set of estrogen receptor interacting proteins identified by tandem affinity purification in hormone-responsive human breast cancer cell nuclei. Proteomics 43, 159–165 (2011)
26. Nassa, G., Tarallo, R., Guzzi, P.H., Ferraro, L., Cirillo, F., Ravo, M., Nola, E., Baumann, M., Nyman, T.A., Cannataro, M., Ambrosino, C., Weisz, A.: Comparative analysis of nuclear estrogen receptor alpha and beta interactomes in breast cancer cells. Mol. BioSyst. 7(3), 667–676 (2011)
27. Orchard, S., Kerrien, S., Jones, P., Ceol, A., Chatr-Aryamontri, A., Salwinski, L., Nerothin, J., Hermjakob, H.: Submit your interaction data the imex way: a step by step guide to trouble-free deposition. Proteomics 7(S1), 28–34 (2007)
28. Pavlopoulos, G., Wegener, A.L., Schneider, R.: A survey of visualization tools for biological network analysis. BioData Mining 1(1), 12+ (2008)
29. Nucleic Acids Res. (Nar. database issue) (January 2007)
30. Salwinski, L., Miller, C.S., Smith, A.J., Pettit, F.K., Bowie, J.U., Eisenberg, D.: The Database of Interacting Proteins: 2004 update. Nucl. Acids Res. 32(suppl. 1), D449–D451 (2004)
31. Singh, R., Xu, J., Berger, B.: Global alignment of multiple protein interaction networks with application to functional orthology detection. Proceedings of the National Academy of Sciences 105(35), 12763–12768 (2008)
32. Siuzdak, G.: The expanding role of mass spectrometry in biotechnology. MCC Press (2006)
33. Szklarczyk, D., Franceschini, A., Kuhn, M., Simonovic, M., Roth, A., Minguez, P., Doerks, T., Stark, M., Muller, J., Bork, P., Jensen, L.J., von Mering, C.: The STRING database in 2011: functional interaction networks of proteins, globally integrated and scored. Nucleic Acids Research 39(database issue), D561–D568 (2011)
34. Xia, K., Dong, D., Han, J.-D.D.: IntNetDB v1.0: an integrated protein-protein interaction network database generated by a probabilistic model. BMC Bioinformatics 7, 508 (2006)
35. Zanzoni, A., Montecchi-Palazzi, L., Quondam, M., Ausiello, G., Helmer-Citterich, M., Cesareni, G.: Mint: a molecular interaction database. FEBS Lett. 513(1), 135–140 (2002)
36. Zhang, S., Zhang, X.-S., Chen, L.: Biomolecular network querying: a promising approach in systems biology. BMC Systems Biology 2(1), 5 (2008)

The Three Steps of Clustering in the Post-Genomic Era: A Synopsis

R. Giancarlo[1], G. Lo Bosco[1], L. Pinello[1], and F. Utro[2]

[1] Dipartimento di Matematica ed Informatica, Universitá di Palermo, Via Archirafi 34, 90123 Palermo, Italy
raffaele@math.unipa.it, {lobosco,pinello}@unipa.it

[2] Computational Genomics Group, IBM T.J. Watson Research Center, Yorktown Heights, NY, USA
futro@us.ibm.com

Abstract. Clustering is one of the most well known activities in scientific investigation and the object of research in many disciplines, ranging from Statistics to Computer Science. Following Handl et al., it can be summarized as a three step process: (a) choice of a distance function; (b) choice of a clustering algorithm; (c) choice of a validation method. Although such a purist approach to clustering is hardly seen in many areas of science, genomic data require that level of attention, if inferences made from cluster analysis have to be of some relevance to biomedical research. Unfortunately, the high dimensionality of the data and their noisy nature makes cluster analysis of genomic data particularly difficult. This paper highlights new findings that seem to address a few relevant problems in each of the three mentioned steps, both in regard to the intrinsic predictive power of methods and algorithms and their time performance. Inclusion of this latter aspect into the evaluation process is quite novel, since it is hardly considered in genomic data analysis.

1 Introduction

In recent years, the advent of high density arrays of oligonucleotides and cDNAs has had a deep impact on biological and medical research. Indeed, the new technology enables the acquisition of data that is proving to be fundamental in many areas of the biological sciences, ranging from the understanding of complex biological systems to diagnosis (e.g. [3]).

Although clustering microarray expression data is by now a fundamental aspect of microarray data analysis [10,32], the application of such a powerful and well established methodology to post-genomic data seems to be rather *ad hoc*. Motivated by such an observation, Handl et al. [20] have produced a key paper, with the intent to bring to the attention of both bioinformatics researchers and end-users some of the fundamental aspects of the methodology. In order to place this paper in the proper context, it is useful to recall from Handl et al. that clustering can be seen as a three step process: (1) choice of a distance function; (2) choice of a clustering algorithm and (3) choice of a methodology to assess the

R. Rizzo and P.J.G. Lisboa (Eds.): CIBB 2010, LNBI 6685, pp. 13–30, 2011.
© Springer-Verlag Berlin Heidelberg 2011

statistical significance of clustering solutions. Points (2) and (3) lead into two well established and rich research areas in data analysis. Unfortunately, point (1) has been hardly investigated regarding this new type of data. Indeed, there are very few results on this topic (see [10,16,30] and references therein). Although computational methods for the analysis of microarray data have witnessed an exponential growth, very little has been done in trying to assess their merits [28]. Consequently, the need for a through evaluation of the entire analysis process for microarray data is being recognized and a few benchmarking studies start to appear, i.e., [15,17]. Following that novel research trend, we consider all the three steps of clustering, devoting a separate section to each of them. In particular, this paper is organized as follows.

The experimental set-up we have used for the results reported here is presented in Section 2. Those results are simply highlighted here since space limitation do not allow to report the entire set of experiments and results, which will be presented in more extended form elsewhere.

Section 3 describes in detail a new approach to assess both the intrinsic separation ability of many standard distance functions and their use in conjunction with clustering algorithms. The main results are: (a) a qualitative visualization method to describe the interplay between distances and clustering algorithms and (b) a quantitative method to assess the performance of a clustering algorithm and the intrinsic separability of a distance function via a new external validation index.

Section 4 is devoted to clustering algorithms, in particular to Non-negative Matrix Factorization (NMF for short) [27]. Indeed, following the work by Brunet et al. [7] on molecular pattern discovery, NMF has become a reference pattern discovery method in bioinformatics [8]. Although it is well known that it is quite demanding in terms of computer time, its worthiness compared to other clustering algorithms has not been studied. It involves the assessment of the method with respect to both its ability to cluster data and the time it takes for that task, as well as comparison with other clustering algorithms. Unfortunately, our experiments show that NMF is not competitive with respect to classic clustering algorithms, both in terms of prediction power and of time performance. Therefore, the power of the method seems to be mostly for pattern discovery tasks and its use as a clustering algorithm is inappropriate.

Section 5 discusses the assessment of the statistical significance of a clustering solution. Here we concentrate on a particular aspect of this rather general question [24]: the identification of the correct number of clusters in a given dataset. We refer to this class of statistical methods as internal validation measures. Moreover, we focus on data-driven internal validation measures, particularly on those designed for and tested on microarray data. Measures in this class assume nothing about the structure of the dataset, which is inferred directly from the data. The main result in this section is a highlight of Fast Consensus (`FC` for short), a speed-up of Consensus Clustering (`Consensus` for short) [29]. This latter measure turns out to be the one of choice, among a set of quite representative measures recently benchmarked on microarray data [17]. Since it has also a paradigmatic nature for stability-based validation measures, it is a natural candidate for a speed-up. The new method FC has the same predictive power as Consensus, but it is at least one order of magnitude faster in time. Even more remarkably, it reduces to one order of magnitude the time gap between the fastest and most precise internal validation measures available in the literature.

The last section offers some conclusions and some directions of future research.

2 Experimental Set-Up

2.1 Datasets

Technically speaking, a *gold solution* for a dataset is a partition of the data in a number of classes known *a priori*. Membership of a class is established by assigning the appropriate class label to each element. In less formal terms, the partition of the dataset in classes is based on external knowledge that leaves no ambiguity on the actual number of classes and on the membership of elements to classes. Although there exist real microarray datasets for which such an *a priori* division is known, in a few previous studies of relevance here, a more relaxed criterion has been adopted to allow also datasets with high quality partitions that have been inferred by analyzing the data, i.e., by the use of internal knowledge via data analysis tools such as clustering algorithms. In strict technical terms, there is a difference between the two types of gold solutions. For their datasets, Dudoit and Fridlyand [12] elegantly make clear that difference in a related study and we closely follow their approach here.

Each dataset is a matrix, in which each row corresponds to an element to be clustered and each column to an experimental condition. The four datasets, together with the acronyms used in this paper, are reported next. For conciseness, we mention only some relevant facts about them. The interested reader can find additional information in Dudoit and Fridlyand [12] for the Lymphoma and NCI60 datasets, Monti et al. for the Normal Tissue dataset [29] and in Di Gesú et al. [11], for the remaining ones.

Lymphoma: The dataset comes from the study of Alizadeh et al. [4] on the three most common adult lymphoma tumors. It is an 80×100 matrix, where each row corresponds to a tissue sample and each column to a gene. There is an *a priori* partition into three classes and we take that as the gold solution. The dataset has been obtained from the original microarray experiments as described by Dudoit and Fridlyand [12].

NCI60: This dataset originates from a microarray study in gene expression variation among the sixty cell lines of the National Cancer Institute anti-cancer drug screen [2]. It is a 57×200 data matrix, where each row corresponds to a cell line and each column to a gene. There is an *a priori* partition of the dataset into eight classes and we take that as the gold solution. The dataset has been obtained from the original microarray experiments as described by Dudoit and Fridlyand [12].

Normal Tissue: It is a 90×1277 data matrix, where each row corresponds to a tissue sample and each column to a gene. The dataset comes from the study of Su et al. [33] on the four distinct cancer types. There is a partition into four classes and we take that as the gold solution.

PBM: The dataset contains 2329 cDNAs with a fingerprint of 139 oligos. This gives a 2329 × 139 data matrix. According to Hartuv et al. [21], the cDNAs in the dataset originated from 18 distinct genes, i.e., the *a priori* classes are known. The partition of the dataset into 18 groups was obtained by lab experiments at Novartis in Vienna. Following that study, we take those classes and the class labels assigned to the elements as the gold solution. It was used by Hartuv et al. to test their clustering algorithm.

2.2 Distances

For our experiments, among the plethora of distance functions available in the mathematical literature [9], we have used Euclidean distance, Pearson correlation and Mutual Information (MI for short), since they have been shown to be the most suitable for microarray data [16]. In what follows, we refer to distance, similarity and dissimilarity functions with the generic term distance functions.

2.3 Algorithms and Hardware

For our experiments, we have used our own C/C++ implementation of NMF, which is based on the Matlab script available at the Broad institute [1]. Indeed, it has been converted to a C/C++ version, then validated by ensuring it produces the same results as for the Matlab version in a number of simulations. In addition to NMF, we have chosen a suite of algorithms, i.e., K-means among *Partitional Methods*, Average Link among the *Hierarchical Methods*. Since they are standard and well known clustering algorithm, they are not described here. The interested reader, however, will find a detailed description of them in a classic book on the subject by Jain and Dubes [23]. It goes without saying that each of the above algorithms has already been used for data analysis of microarray data, e.g. [11,18,36]. All experiments were performed in part on several state-of-the-art PCs and in part on a 64-bit AMD Athlon 2.2 GHz bi-processor with 1 GB of main memory running Windows Server 2003. All the timing experiments reported were performed on the bi-processor, using one processor per run. The use of several machines for the experimentation was deemed necessary in order to complete the full set of experiments in a reasonable amount of time. Indeed, as detailed later, some experiments would require weeks to complete execution on Normal Tissue and PBM, the largest dataset we have used. We also point out that all the Operating Systems supervising the computations have a 32 bits precision.

3 Evaluating Distance-Clustering Performance via ROC Analysis

There are very few studies in the specialistic literature that shed light on the proper choice of a distance function for clustering of microarray data. That involves addressing the following points:

(A) Assessment of the intrinsic separation ability of a distance. That is, how well a distance discriminates independently of its use within a clustering algorithm.
(B) Assessment of the predictive clustering algorithm ability of a distance. That is, which distance function grants the best performance when used within a clustering algorithm.
(C) The interplay between (A) and (B).

Points (A) and (B) have been studied before (see [16] and references therein) with some useful insights. Unfortunately, very little is known about (C), one of the difficulties being a fair comparison between the performance of a distance function and a clustering algorithm measured in terms of classification ability. Here we address this latter problem by extending techniques in [16].

We need to introduce some terminology and recall some definitions. Given a clustering solution $C = \{c_1, \ldots, c_r\}$, it can be represented by a binary matrix J, referred to as connectivity matrix, where each entry of J is defined as follows:

$$J(i,j) = \begin{cases} 1 & \text{if } \boldsymbol{x}_i \text{ and } \boldsymbol{x}_j \text{ belongs to the same cluster,} \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

A useful tool to assesses the performance of a classifier, not necessarily binary, is the *confusion matrix*, which is a matrix where each row represents the instances in a predicted class, while each column represents the instances in an actual class. In the case of a binary classification, the 2×2 confusion matrix stores the number of elements of class 0 classified as 0, denoted $T0$, and the number of elements of class 0 classified as 1, denoted $F1$. One defines $T1$ and $F0$ analogously. In this context, the *Sensitivity* TPR and *Specificity* TNR are defined as follows:

$$TPR = \frac{T0}{T0 + F1}$$

$$TNR = \frac{T1}{T1 + F0}$$

A ROC plane [22] is a plane where $y = TPR$ and $x = FPR = 1 - TNR$, and it is useful to measure a classification in terms of TPR and FPR rates, once having established to represent with 0 the positive class. Note that, since a classifier assigns data items to classes, the TPR represents the percentage of item pairs correctly assigned to different classes, while the FPR is the percentage of item pairs incorrectly assigned to different classes. In the ROC plane, it is possible to define a particular curve, which is referred to as ROC curve, that allows to assess the performance of a classifier. Indeed, the area under this curve (AUC for short) is defined in the range $[0, 1]$, where a value of 0.5 corresponds to the performance of a classifier with a random assignment rule, while the closer is AUC to one, the better is the performance of the classifier.

We address point (C) by:

(C.1) showing how to map a clustering solution into the ROC plane (see Section 3.2);

(C.2) introducing a distance between a clustering solution and the gold solution (see Section 3.2);

(C.3) showing how (C.1) and (C.2) can be used to fairly compare the intrinsic ability of distance function and a clustering algorithms (see Section 3.3).

3.1 Clustering Solutions and ROC Plane

Given a gold solution GS, it is possible to map a clustering solution s into the ROC plane as follows:

1. Compute the connectivity matrix J_s for s.
2. Starting from J_s, compute the confusion matrix with respect to GS using the definition of confusion matrix stated at the beginning of this section.
3. Use this confusion matrix to compute TPR and FPR for s. Those two variables naturally identify a point into the ROC plane, associated to s.

A few remarks are in order. The above approach naturally leads to measure a clustering solution in terms of TPR and FPR rates. The point into the ROC plane associated with GS is $P_{GS} = (0, 1)$ (see the square marker in Figures 1-3). Finally, each point in the ROC plane can be obtained via a clustering solution.

3.2 The Distance between Clustering Solutions in ROC Plane

We now provide a method to assess the relative merits of different clustering algorithms using a given distance function. Given a clustering solution s, let $P_s = (x, y)$ be the point in the ROC plane corresponding to it.

The performance of s is proportional to the proximity of P_s to P_{GS}, as we now explain. Let E_m be the *Misclassification error rate* defined as the sum between FPR (x) and False negative rate ($FNR = 1 - y$). That is, E_m is the L1 metric (d_1) computed between P_s and P_{GS}, i.e., $d_1(P_{GS}, P_s) = |x + 1 - y|$. Then, the closer P_s and P_{GS} with respect to d_1 are, the better the clustering solution with respect to E_m.

It is worth pointing out that P_s gives a measure on the agglomerative and divisive tendency of a generic clustering algorithm. Indeed, the greater the x value, the more divisive the clustering algorithm is. Analogously, the smaller the y value, the more agglomerative the clustering algorithm is. Indeed, we can actually devise an index that measures such a tendency.

Let E_b be the *Balancing error rate* defined as the measures of how much FPR and FNR are unbalanced. The *Balanced Misclassification Index* (BMI for short) for a generic clustering solution is:

$$BMI = \alpha \times (E_m)^2 + \beta \times (E_b)^2 \tag{2}$$

where the weights α and β allow to tune the importance between balance and misclassification.

It is of interest and relevance here to notice that by setting $\alpha = \beta = 0.5$, BMI corresponds to $d_2(P_{GS}, P_s)$. That is, the L_2 (Euclidean) metric between the points P_{GS} and P_s. This means that the closer P_s and P_{GS} are with respect to d_2, the better the clustering solution, in equal measure ($\alpha = \beta = 0.5$) between misclassification error rate E_m and balancing error rate E_b.

Operationally, once fixed $\alpha = \beta = 0.5$, if we want to compute the BMI of a clustering algorithm producing a clustering solution with $x = FPR$ and $y = TPR$, respectively, one needs only to compute the Euclidean distance between the points P_s and P_{GS} in the ROC plane. It is obvious that such a technique can also be used to compare the performances of several clustering solutions by considering the Euclidean distances between the associated points into the ROC plane and P_{GS}.

3.3 A Procedure to Compare Distance Functions and Clustering Algorithms Via ROC Analysis

We recall from [16] that starting from a distance matrix D and a gold solution GS, it is possible to derive a ROC curve into the ROC plane, as we now briefly outline. Given a threshold value $\phi \in [0, 1]$ and the distance matrix D, let J_ϕ be a matrix in which each entry is defined as follows:

$$J_\phi(i,j) = \begin{cases} 1 & \text{if } D(i,j) \leq \phi, \\ 0 & \text{otherwise.} \end{cases} \tag{3}$$

As for the connectivity matrix defined in (1) it is possible also for J_ϕ to compute a confusion matrix. Therefore, by considering all the points corresponding to different threshold values, we obtain the ROC curve for the distance function d. If each point in the ROC curve so obtained corresponded to a proper partition of the items, i.e., a clustering solution, we could use it to address (C). Unfortunately, that is not the case, as we now argue. In fact, J_ϕ does not correspond to a partition of the dataset since there could be three items i, j, k such that (i, j) and (j, k) belong to different clusters, that is two clusters could have a non-empty intersection. Therefore, in order to properly compare a distance function with a clustering algorithm, via ROC analysis, we need to "convert" the matrix J_ϕ into a matrix J'_ϕ representing a clustering solution. This can be done in several ways: here we have adopted an approach based on the connected components induced by the matrix J_ϕ. Intuitively, the process is the following: if J_ϕ does not correspond to a partition, i.e., at least two sets a and b have non-empty intersection, then they are merged into a new set $c = a \cup b$. This allows to transform the ROC curve associated to a distance function into a new curve in which each point corresponds to a proper clustering solution. We refer to this curve as the *corrected ROC curve* (CROC for short) of a distance, useful to measure the *intrinsic separation ability of a distance function with respect to clustering*.

Using the CROC curve, one can find the best clustering solution associated to a distance function with respect to BMI, as the closest point to P_{GS} into the

CROC curve (see the $*$ marker in Figures 1-3 as an example). One can now fairly compare a distance function and a clustering solution produced by an algorithm, in terms of their classification ability:

1. Compute the ROC curve for a distance function d.
2. Calculate the CROC curve starting from the ROC curve computed in the previous point.
3. Find the best point into the CROC curve, i.e., the point with the lowest value of BMI, and mark it.
4. Map one ore more clustering solutions in the ROC plane (as described in Section 3.1) and mark the corresponding points.
5. Rank the performance of each marked points in the ROC plane, as described in Section 3.2.

3.4 Results

Figures 1-3 show the results of the experiments. In particular, each figure reports the performance of the clustering algorithms with the use of the same distance function for each dataset. Such analysis shows that the K-means and

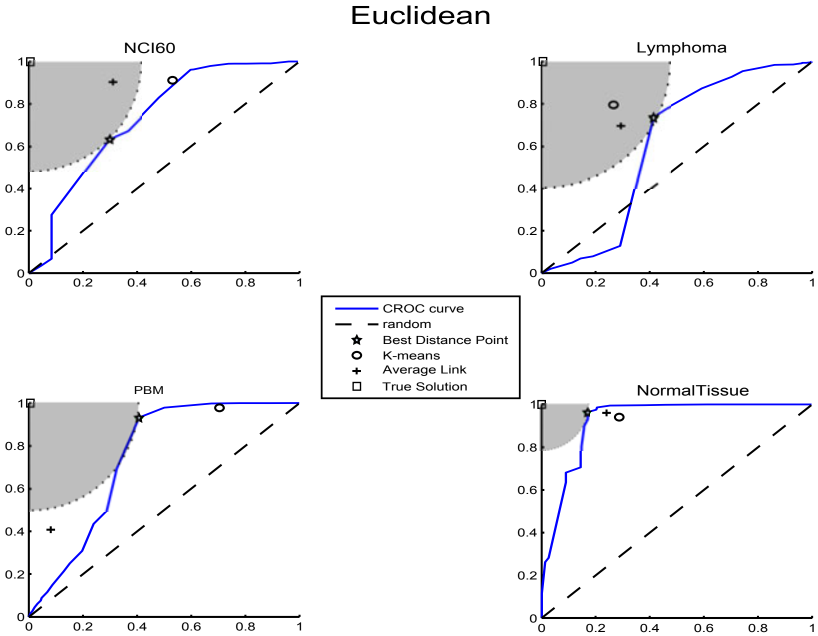


Fig. 1. The CROC curve and plot of the clustering solutions for each dataset in the case of the Euclidean distance. The markers show TPR versus FPR of each clustering solution. The area in gray represents the set of points that has a better performance with respect to the best distance point for BMI, while the dotted line represents set of points with the same performance.

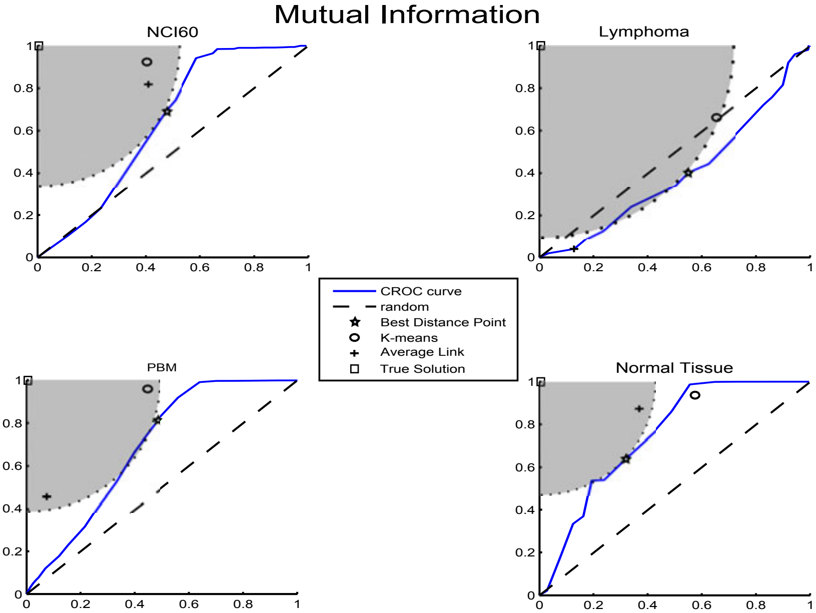


Fig. 2. The CROC curve and plot of the clustering solutions for each dataset in the case of Mutual Information. The markers show TPR versus FPR of each clustering solution.

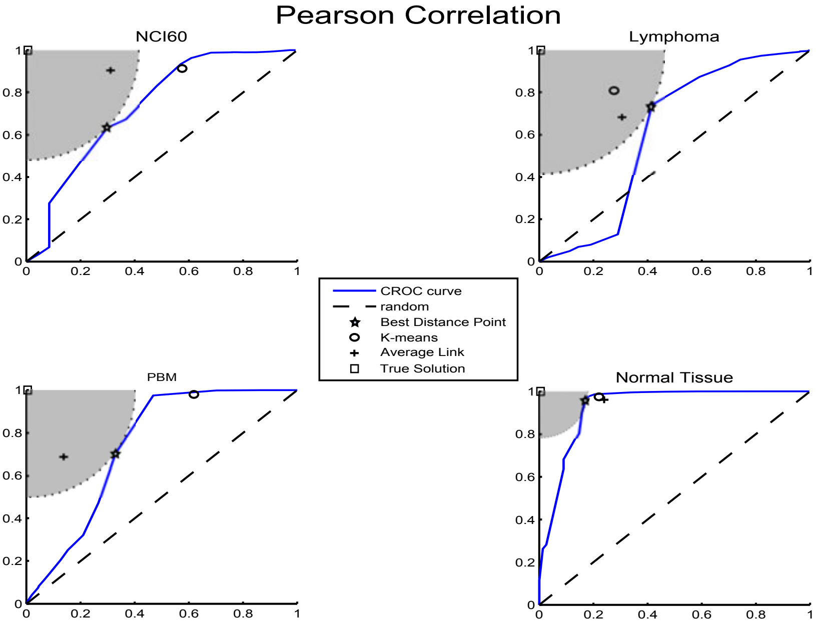


Fig. 3. The CROC curve and plot of the clustering solutions for each dataset in the case of Pearson Correlation. The markers show TPR versus FPR of each clustering solution.

Table 1. The *BMI* value for the best point distance and each clustering algorithm for all datasets for the Euclidean distance. The numbers in bold show *BMI* values lower than the best distance point into the CROC curve, i.e., when the performance of the algorithm is better then of the performance of the distance alone.

	Euclidean			
	NCI60	Lymphoma	PBM	Normal Tissue
Average Link	**0.321903637**	**0.419453466**	0.597614387	0.242813873
K-means	0.538007588	**0.334160756**	0.704107566	0.2937045
Best distance Point	0.469881192	0.488040102	0.411447355	0.170227836

Table 2. The *BMI* value for the best point distance and each clustering algorithm for all datasets for the Mutual Information. The legend is as in Table 1.

	Mutual Information			
	NCI60	Lymphoma	PBM	Normal Tissue
Average Link	**0.446332725**	0.968440209	0.54845816	**0.385645494**
K-means	**0.409391975**	**0.73217916**	**0.44437129**	0.578532332
Best distance Point	0.564302857	0.809560795	0.515610657	0.478981757

Table 3. The *BMI* value for the best point distance and each clustering algorithm for all datasets for the Pearson correlation. The legend is as in Table 1.

	Pearson Correlation			
	NCI60	Lymphoma	PBM	Normal Tissue
Average Link	**0.321903637**	**0.438963921**	**0.3415215**	**0.242813873**
K-means	0.581929376	**0.336156536**	0.619121891	**0.221943924**
Best distance Point	0.469881192	0.488040102	0.441733534	0.515610657

Average Link clustering algorithms are both able to improve the intrinsic separation ability of a distance function with respect to clustering. In particular, the *BMI* values of Average Link is 8/12 greater then the *BMI* of the corresponding best point for the distance in the CROC curve (see also Tables 1-3). It is worth pointing out that the intrinsic separation ability of the *MI* and Pearson correlation are improved by K-means and Average Link in most of the cases.

4 Clustering Algorithms

The second step of the clustering process is to compute a partition of X via a clustering algorithm. Recall that a choice of a distance function may be required. In the general data mining literature, there is a great proliferation of research on clustering algorithms, in particular for gene expression data [10]. Some of those studies concentrate both on the ability of an algorithm to obtain a high quality partition of the data and on its performance in terms of computational resources, mainly CPU time. For instance, hierarchical clustering and K-means algorithms [24] have been the object of several speed-ups (see [6,14,31] and references therein). Moreover, the need for computational performance in the area of clustering for microarray data is so acute that implementations of well known algorithms, such as K-means, specific for multi-core architectures, are being proposed [26]. It is worth recalling that, putting together the study by D'haeseleer [10] and a more recent one by Freyhult et al. [15], hierarchical algorithms (with the exception of Single Link) and K-means seem to be the most appropriate for microarray clustering.

Here we consider NMF as a clustering algorithm. One of the main differences between NMF and the already mentioned algorithms is that it produces a partition of X without the use of a distance function. Another difference is its generality as a pattern discovery tool in bioinformatics.

In what follows, we limit ourselves giving a brief description of NMF. The interested reader can find relevant references for a more in-depth description in [7,8,27]. Assume that one is given m n-dimensional non-negative data vectors, organized in an $n \times m$ matrix V. Assume also that one is given an integer r, usually chosen such that $(n+m)r < nm$. NMF computes two matrices W and H such that $V \simeq WH$, where the first is of dimension $n \times r$ and the second of dimension $r \times m$. Notice that each data vector can be written as $v \simeq Wh$, where v and h are homologous columns in V and H. That is, each data vector can be seen as a linear combination of the columns of W weighted by the components of h. Following such an observation, W is considered the basis of a new space of size r, describing the data, and H contains the coefficients. Following the same notation as in Brunet et al. [7] and Devarajan [8], assume that V represents the outcome of a microarray experiment, where we have m samples and each of them is composed of measurements of n genes (e.g. V would be the transpose of the datasets we use here-see Section 2.1). In this case, W and H assume two very intuitive roles. W is a matrix whose columns are "metagenes" and H is a matrix whose rows are "meta expression patterns". If one is interested in clustering the samples in r groups, as we do here, then one can place sample i in cluster j if the expression level of sample i is the maximum in metagene j. That is, $h_{i,j}$ is maximum in the i-th column of H.

As for methods implementing NMF, the most popular follow at least one of the following principles and techniques: alternating direction iterations, projected Newton, reduced quadratic approximation, and descent search. The interested reader is referred to [35] for a compendium on the state of the art, including available software.

Recall from Section 2.1 that all of our datasets have a gold solution and from [18] that an external index measures how well a clustering solution computed by an algorithm agrees with the gold solution for a given dataset. In particular, we use the Adjusted Rand index (R_A for short) for our experiments. It is worth recalling that it has a maximum value of one, indicating a perfect agreement between the two partitions, while it has an expected value of zero indicating a level of agreement due to chance. Note that R_A may be negative [13,37]. So, for

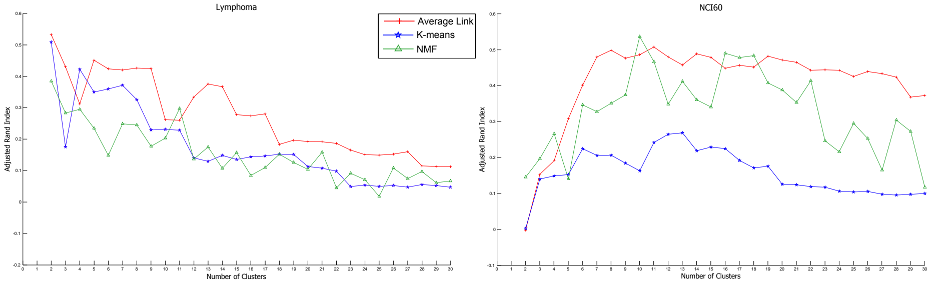


Fig. 4. The R_A index curves, for each dataset. In each figure, the plot of each algorithm is drawn according to the legend. For Normal Tissue and PBM, the experiments with NMF were terminated due to their high computational demand and the corresponding partial plots are not reported here. It is worth recalling that Lymphoma and NCI60 have 3 and 9 classes in the gold solution.

two partitions to be in significant agreement, R_A must assume a non-negative value substantially away from zero.

In order to evaluate the precision of a clustering algorithm, i.e., its ability to identify a good partition of the data, we use the following methodology for our experiments. With the use of a given clustering algorithm, a partition of X into i clusters is generated, for $i \in [2, 30]$, and its agreement with the gold solution is measured via R_A. Then, all those values are plotted to obtain a curve. For a good algorithm, one expects that curve to have a maximum in the proximity of the number of clusters in the gold solution and to decrease sharply after that point. We also record the time, in milliseconds, that the algorithm takes in order to generate all of those solutions.

The results corresponding to our experiments are reported in Fig. 4 for the precision part (plots of R_A) and in Table 4 for the timing results. For precision, all algorithms perform very well on the NCI60 dataset, while their performance is somewhat mixed on the Lymphoma dataset. Those results highlight that the predictive power of NMF does not seem to be superior to that of the other two classic algorithms. Moreover, it is from two to four orders of magnitude slower in time. A full set of experiments on nine benchmark microarray datasets, including the four reported here, are presented in [35] and they provide strong evidence

Table 4. Timing results in milliseconds for all the algorithms on all datasets. For NMF on PBM and Normal Tisue, the experiments were terminated due to their high computational demand (weeks to complete).

	NCI60	Lymphoma	PBM	Normal Tissue
Average Link	500	921	4.4×10^5	2.0×10^3
K-means	3.2×10^3	7.2×10^3	1.1×10^6	7.5×10^4
NMF	3.9×10^5	5.2×10^5	-	-

that NMF does not seem to be competitive with respect to the other two classic algorithms.

Therefore, the use of NMF as a clustering algorithm is not suggested, in particular for large datasets. Indeed, given the steep computational price one has to afford, its use does not seem to be justified since Average Link is at least four orders of magnitude faster and with a better precision. In fact, the main power of NMF rests on its pattern discovery ability, and its use as a clustering algorithm seems to be very limiting of the technique.

5 Internal Validation Measures

The last step of the clustering process is to assess the statistical significance of a clustering solution via a validation measure. As stated in the Introduction, we concentrate on data-driven internal measures that predict the correct number of clusters in a given dataset. Giancarlo et al. [17] have recently proposed an extensive comparative analysis of measures taken from the most relevant paradigms in the area: (a) hypothesis testing in statistics, e.g., [34]; (b) stability-based techniques, e.g., [5,12,29] and (c) jackknife techniques, e.g., [38]. These benchmarks consider both the ability of a measure to predict the correct number of clusters in a dataset and, departing from the current state of the art in that area, the computer time it takes for a measure to complete its task. Since the findings of that study are essential to place this research in a proper context, we highlight them next:

(1) There is a very natural hierarchy of internal validation measures, with the fastest and less precise at the top. In terms of time, there is a gap of at least two orders of magnitude between the fastest, Within Cluster Sum of Squares (`WCSS` for short) [34], and the slowest ones.
(2) All measures considered in that study have severe limitations on large datasets with a large number of clusters, either in their ability to predict the correct number of clusters or to finish their execution in a reasonable amount of time, e.g, a few days.
(3) Although among the slowest, `Consensus` displays some quite remarkable properties that, accounting for (1) and (2), make it the measure of choice for small and medium sized datasets. Indeed, it is very reliable in terms of its ability to predict the correct number of clusters in a dataset, in particular when used in conjunction with hierarchical clustering algorithms. Moreover, such a performance is stable across the choice of basic clustering algorithms, i.e., various versions of hierarchical clustering and K-means, used to produce clustering solutions.

5.1 FC and Consensus Validation Measures

Since validation procedures are the computational bottleneck in many data analysis processes involving microarrays, more efficient ones would be a substantial contribution to this area [25]. One way to tackle such an admittedly difficult question is to design "approximation" algorithms, i.e., algorithms that provide essentially the same precision of a measure, while being substantially faster in time. Giancarlo et al. [17] have proved the validity of such an approach and they have also started a systematic investigation of its potentiality. Here we contribute FC, an "approximation" of Consensus, to that line of research.

In what follows, we limit ourselves to an intuitive description of both Consensus and FC, referring the reader to [29] and [19,35] for a description of each, respectively. Indeed, a large number of clustering solutions, each obtained via a sample of the original dataset, seem to be required in order to identify the correct number of clusters. However, there is no theoretic reason indicating that those clustering solutions must each be generated from a *different* sample of the input dataset, as Consensus does. Such an independent set of "sampling and clustering steps" generates duplications in the computation and therefore loss of efficiency. Based on this observation, FC performs, first, a sampling step to generate a data matrix, which is then used to generate all the required clustering solutions. This allows to obtain a speed-up since costly computational duplications are avoided when the clustering algorithm is hierarchical. Indeed, it becomes possible to interleave the computation of the measure with the level bottom-up construction of the hierarchical tree underlying the clustering algorithms. Specifically, only one dendogram construction is required rather than the repeated and partial construction of dendograms as in the Consensus procedure. Therefore, we use, in full, the main characteristic of agglomerative algorithms.

Tables 5 and 6 report the precision and timing results regarding FC and Consensus on all datasets and clustering algorithms, respectively. Note that, in terms of precision, FC and Consensus provide nearly identical predictions on all the datasets. Moreover, in terms of time, note that FC is faster then Consensus by at least one order of magnitude on hierarchical algorithm. In particular, FC is able to complete execution on the PBM dataset, as opposed to Consensus, with all algorithms.

Moreover, in order to compare FC with other internal validation measures proposed in the literature, we take into account the benchmarking proposed by Giancarlo et al. [17]. From that study we extract and report, in Tables 7 and 8, the fastest and best performing measures, with the addition of FC. The interested reader can find in [17] a more in-depth description of the measures used in that study and reported here. As is self-evident from that latter table, FC with Average Link is within a one order of magnitude difference in speed with respect to the fastest measures, i.e., WCSS and G-Gap [17] (an approximation of the Gap Statistics). Quite remarkably, it grants a better precision in terms of its ability to identify the underlying structure in each of the benchmark datasets. It is also of relevance to point out that FC with Average Link has a time performance comparable to that of FOM [38], but again it has a better precision performance. Notice that, none of the three just-mentioned measures depends on any parameter setting, implying that no speed-up will result from a tuning of the algorithms.

Table 5. A summary of the precision results of Consensus and FC on all datasets and clustering algorithms. A number in a circle with a black background indicates a prediction in agreement with the number of classes in the dataset; while a number in a circle with a white background indicates a prediction that differs, in absolute value, by 1 from the number of classes in the dataset; a number not in a circle/square indicates the remaining predictions; a dash indicates that the experiments were terminated due to their high computational demand.

	Precision			
	Lymphoma	NCI60	Normal Tissue	PBM
FC-Average Link		❽	10	2
FC-K-means	④		10	16
Consensus-Average Link		❽	10	-
Consensus-K-means	④		10	-
Gold solution	**3**	**8**	**13**	**18**

Table 6. A summary of the timing results of Consensus and FC on all datasets and clustering algorithms. Each cell reports timing in milliseconds, while a dash indicates that the experiments were terminated due to their high computational demand.

	Timing			
	Lymphoma	NCI60	Normal Tissue	PBM
FC-Average Link	6.8×10^4	7.0×10^4	3.4×10^5	4.2×10^7
FC-K-means	1.1×10^6	1.2×10^6	2.0×10^6	1.6×10^8
Consensus-Average Link	1.3×10^6	1.4×10^6	9.5×10^6	-
Consensus-K-means	1.1×10^6	1.2×10^6	6.3×10^6	-

Table 7. A summary of the fastest performing measures taken into account by Giancarlo et al., with the addition FC. The table legend is as in Table 5.

	Precision				
	CNS Rat	Leukemia	NCI60	Lymphoma	Yeast
WCSS-R-R0					
G-Gap-K-means			4		
G-Gap-R-R5			2		
FOM-R-R5				5	
FOM-Average Link				6	
FC-Average Link					
FC-K-means					
Gold solution	**6**	**3**	**8**	**3**	**5**

The results outlined above are particularly significant since (i) FOM is one of the most established and highly-referenced measures specifically designed for microarray data; (ii) in purely algorithmic terms, WCSS and G-Gap, are so simple as to represent a "lower bound" in terms of the time performance that is achievable by any data-driven internal validation measure. In conclusion, our experiments show that FC is quite close in time performance to three of the fastest data-driven validation measures available in the Literature, while also granting better

Table 8. A summary of the timing results for the fastest performing measures taken into account by Giancarlo et al., with the addition FC. The table legend is as in Table 6.

	Timing			
	CNS Rat	Leukemia	NCI60	Lymphoma
WCSS-R-R0	1.2×10^3	8.0×10^2	4.1×10^3	3.0×10^3
G-Gap-K-means	2.4×10^3	2.0×10^3	8.3×10^4	8.4×10^3
G-Gap-R-R5	1.2×10^3	8.0×10^2	4.5×10^4	3.2×10^3
FOM-R-R5	3.9×10^3	3.7×10^4	2.1×10^5	7.6×10^4
FOM-Average Link	1.6×10^3	7.5×10^3	5.1×10^4	1.8×10^4
FC-Average Link	4.7×10^4	3.5×10^4	5.2×10^4	1.3×10^5
FC-K-means	7.2×10^5	7.7×10^5	2.5×10^6	2.3×10^6

precision results. In view of the fact that the former measures are considered reference points in this area, the speed-up of Consensus proposed here seems to be a non-trivial step forward in the area of data-driven internal validation measures.

6 Conclusions

The results in this paper extend the ones reported in the literature in several ways. Namely:

- A new approach is proposed for the assessment of the relationship between the classification ability of a distance function and of a clustering algorithm. That is achieved via the introduction of BMI, a new validation index. Moreover, the comparative methodology associated to BMI is able to establish that K-means and Average Link clustering are able to improve upon the intrinsic separation ability of a distance with respect to clustering. That is, they amplify the discriminative ability of a distance function.
- NMF is a computationally expensive procedure, even on datasets of moderate size and quite manageable by other algorithms. Its use as a clustering algorithm is discouraged.
- FC is perceived as a non-trivial step forward in the identification of a validation measure for microarray data analysis that is both fast in execution time and accurate in its prediction of the number of clusters in a dataset.

References

1. Broad institute,
http://www.broadinstitute.org/cgi-bin/cancer/publications/pub_paper.cgi?mode=view&paper_id=89
2. NCI 60 Cancer Microarray Project, http://genome-www.stanford.edu/NCI60
3. Stanford microarray database, http://genome-www5.stanford.edu/

4. Alizadeh, A.A., Eisen, M.B., Davis, R.E., Ma, C., Lossos, I.S., Rosenwald, A., Boldrick, J.C., Sabet, H., Tran, T., Yu, X., Powell, J.I., Yang, L., Marti, G.E., Moore, T., Hudson, J.J., Lu, L., Lewis, D.B., Tibshirani, R., Sherlock, G., Chan, W.C., Greiner, T.C., Weisenburger, D.D., Armitage, J.O., Warnke, R., Levy, R., Wilson, W., Grever, M.R., Byrd, J.C., Botstein, D., Brown, P.O., Staudt, L.M.: Distinct types of diffuse large b-cell lymphoma identified by gene expression profiling. Nature 403, 503–511 (2000)
5. Ben-Hur, A., Elisseeff, A., Guyon, I.: A stability based method for discovering structure in clustering data. In: Seventh Pacific Symposium on Biocomputing, pp. 6–17. ISCB (2002)
6. Borodin, A., Ostrovsky, R., Rabani, Y.: Subquadratic approximation algorithms for clustering problems in high dimensional space. Machine Learning 56, 153–167 (2004)
7. Brunet, J.-P., Tamayo, P., Golub, T.R., Mesirov, J.P.: Metagenes and molecular pattern discovery using matrix factorization. Proc. of the National Academy of Sciences of the United States of America 101, 4164–4169 (2004)
8. Devarajan, K.: Nonnegative Matrix Factorization: An Analytical and Interpretive Tool in Computational Biology. PLoS Comput. Biol. 4, e1000029 (2008)
9. Deza, E., Deza, M.: Dictionary of distances. Elsevier, Amsterdam (2006)
10. D'haeseleer, P.: How does gene expression cluster work? Nature Biotechnology 23, 1499–1501 (2006)
11. Di Gesú, V., Giancarlo, R., Lo Bosco, G., Raimondi, A., Scaturro, D.: Genclust: A genetic algorithm for clustering gene expression data. BMC Bioinformatics 6, 289 (2005)
12. Dudoit, S., Fridlyand, J.: A prediction-based resampling method for estimating the number of clusters in a dataset. Genome Biology 3 (2002)
13. Fisher, D., Hoffman, P.: The Adjusted Rand Statistic: A SAS macro. Psychometrika 53, 417–423 (1988)
14. Frahling, G., Sohler, C.: A fast K-means implementation using coresets. In: Proceedings of the Twenty-Second Annual Symposium on Computational Geometry, pp. 135–143. ACM, New York (2006)
15. Freyhult, E., Landfors, M., Önskog, J., Hvidsten, T.R., Rydén, P.: Challenges in microarray class discovery: a comprehensive examination of normalization, gene selection and clustering. BMC Bioinformatics 11, 503 (2010)
16. Giancarlo, R., Lo Bosco, G., Pinello, L.: Distance Functions, Clustering Algorithms and Microarray Data Analysis. In: Blum, C., Battiti, R. (eds.) LION 4. LNCS, vol. 6073, pp. 125–138. Springer, Heidelberg (2010)
17. Giancarlo, R., Scaturro, D., Utro, F.: Computational cluster validation for microarray data analysis: experimental assessment of Clest, Consensus Clustering, Figure of Merit, Gap Statistics and Model Explorer. BMC Bioinformatics 9, 462 (2008)
18. Giancarlo, R., Scaturro, D., Utro, F.: Statistical Indices for Computational and Data Driven Class Discovery in Microarray Data. In: Biological Data Mining, pp. 295–335. CRC Press, Boca Raton (2009)
19. Giancarlo, R., Utro, F.: Speeding up the Consensus Clustering methodology for microarray data analysis. Algorithms for Molecular Biology 6, 1 (2011)
20. Handl, J., Knowles, J., Kell, D.B.: Computational cluster validation in post-genomic data analysis. Bioinformatics 21, 3201–3212 (2005)
21. Hartuv, E., Schmitt, A., Lange, J., Meier-Ewert, S., Lehrach, H., Shamir, R.: An algorithm for clustering of cDNAs for gene expression analysis using short oligonucleotide fingerprints. Genomics 66, 249–256 (2000)

22. Hanley, J.A., McNeil, B.J.: The meaning and use of the area under a receiver operating characteristic (ROC) curve. Radiology 143, 29–36 (1982)
23. Jain, A.K., Murty, M.N., Flynn, P.J.: Data Clustering: a Review. ACM Computing Surveys 31, 264–323 (1999)
24. Jain, A., Dubes, R.: Algorithms for Clustering Data. Prentice-Hall, Englewood Cliffs (1988)
25. Klie, S., Nikoloski, Z., Selbig, J.: Biological cluster evaluation for gene function prediction. Journal of Computational Biology 17, 1–18 (2010)
26. Kraus, J., Kestler, H.: A highly efficient multi-core algorithm for clustering extremely large datasets. BMC Bioinformatics 11, 169 (2010)
27. Lee, D.D., Seung, H.S.: Learning the parts of objects by Non-negative Matrix Factorization. Nature 401, 788–791 (1999)
28. Mehta, T., Tanik, M., Allison, D.B.: Towards sound epistemological foundations of statistical methods for high-dimensional biology. Nature Genetics 36, 943–947 (2004)
29. Monti, S., Tamayo, P., Mesirov, J., Golub, T.: Consensus clustering: A resampling-based method for class discovery and visualization of gene expression microarray data. Machine Learning 52, 91–118 (2003)
30. Priness, I., Maimon, O., Ben-Gal, I.: Evaluation of gene-expression clustering via mutual information distance measure. BMC Bioinformatics 8, 1–12 (2007)
31. Seal, S., Comarina, S., Aluru, S.: An optimal hierarchical clustering algorithm for gene expression data. Information Processing Letters 93, 143–147 (2004)
32. Speed, T.P.: Statistical analysis of gene expression microarray data. Chapman & Hall/CRC (2003)
33. Su, A.I., Cooke, M.P., Ching, K.A., Hakak, Y., Walker, J.R., Wiltshire, T., Orth, A.P., Vega, R.G., Sapinoso, L.M., Moqrich, A., Patapoutian, A., Hampton, G.M., Schultz, P.G., Hogenesch, J.B.: Large-scale analysis of the human and mouse transcriptomes. Proceedings of the National Academy of Sciences of the United States of America 99, 4465–4470 (2002)
34. Tibshirani, R., Walther, G., Hastie, T.: Estimating the number of clusters in a dataset via the gap statistics. Journal Royal Statistical Society B 2, 411–423 (2001)
35. Utro, F.: Algorithms for internal validation clustering measures in the Post Genomic Era, Doctoral Dissertation, University of Palermo (2011), `http://arxiv.org/abs/1102.2915v1`
36. Xu, Y., Olman, V., Xu, D.: Clustering gene expression data using a graph-theoretic approach: An application of minimum spanning tree. Bioinformatics 18, 526–535 (2002)
37. Yeung, K.Y.: Cluster Analysis of Gene Expression Data. Ph.D. thesis, University of Washington (2001)
38. Yeung, K.Y., Haynor, D.R., Ruzzo, W.L.: Validating clustering for gene expression data. Bioinformatics 17, 309–318 (2001)

De Novo Protein Subcellular Localization Prediction by N-to-1 Neural Networks

Catherine Mooney[1], Yong-Hong Wang[2], and Gianluca Pollastri[1,⋆]

[1] Complex and Adaptive Systems Laboratory and School of Computer Science and Informatics, University College Dublin, Belfield, Dublin 4

[2] Biophysics Institute, Hebei University of Technology, Tianjin, China

Abstract. Knowledge of the subcellular location of a protein provides valuable information about its function and possible interaction with other proteins. In the post-genomic era, fast and accurate predictors of subcellular location are required if this abundance of sequence data is to be fully exploited. We have developed a subcellular localization predictor (SCL_pred) which predicts the location of a protein into four classes for animals and fungi and five classes for plants (secretory pathway, cytoplasm, nucleus, mitochondrion and chloroplast) using high throughput machine learning techniques trained on large non-redundant sets of protein sequences. The algorithm powering SCL_pred is a novel Neural Network (N-to-1 Neural Network, or N1-NN) which is capable of mapping whole sequences into single properties (a functional class, in this work) without resorting to predefined transformations, but rather by adaptively compressing the sequence into a hidden feature vector. We benchmark SCL_pred against other publicly available predictors using two benchmarks including a new subset of Swiss-Prot release 57. We show that SCL_pred compares favourably to the other state-of-the-art predictors. Moreover, the N1-NN algorithm is fully general and may be applied to a host of problems of similar shape, that is, in which a whole sequence needs to be mapped into a fixed-size array of properties, and the adaptive compression it operates may even shed light on the space of protein sequences. The predictive systems described in this paper are publicly available at http://distill.ucd.ie/distill/.

1 Introduction

With the recent advances in high throughput sequencing technology there has been a rapid increase in the availability of sequence information. To fully exploit this information sequences need to be annotated quickly and accurately, which has led to the development of automated annotation systems. A major step towards determining the function of a protein is determining its Subcellular Localization (SCL). Knowledge of the location of the protein sheds light not only on where it might function but also what other proteins it might interact with, as, in order to interact, proteins must inhabit the same location or physically

[⋆] Corresponding author.

R. Rizzo and P.J.G. Lisboa (Eds.): CIBB 2010, LNBI 6685, pp. 31–43, 2011.
© Springer-Verlag Berlin Heidelberg 2011

adjacent compartments, at least temporarily. At present there is a growing gap between the number of proteins that have reliable SCL annotations and the number of known protein sequences. Experimental approaches to SCL prediction are time-consuming and expensive, whereas computational methods can provide fast and increasingly more accurate localization predictions.

There are various different mechanisms by which a protein is directed to a particular location in the cell and there are many possible compartments in which eukaryotics protein may be located. Here we consider four for animals and fungi and five for plants: nucleus, cytoplasm, mitochondria, chloroplast and the secretory pathway. Some nuclear proteins have a nuclear localisation signal (NLS) which may occur anywhere in the sequence. Most secretory pathway, mitochondrial and chloroplastic proteins have N-terminal peptides (SP, mTP and cTP respectively) but many proteins have no known motif [8,16]. However, it would appear that for most proteins the sequence of the protein alone has sufficient information to predict the protein's location in the cell.

There are many methods for the prediction of SCL which can be roughly divided into two groups: homology-based, that rely on similarity to another sequence of known location; and *de novo* or *ab initio*, sequence-based methods, which may use evolutionary information in the form of multiple sequence alignments (MSA), but do not depend on sequences of known location. The method we describe in this article falls into this latter category.

We predict SCL for eukaryotes only, which we divide into animals, plants and fungi. In a first series of tests we adopt essentially the same experimental setting and 4/5 location classes as in [6,18], to which we compare our predictor. We then take a further step by developing new, redundancy reduced training and testing sets starting from Swiss Prot 57 [5], and benchmark SCL_pred on these sets against five state-of-the-art, publicly available predictors of SCL: BaCelLo, LOCtree, Protein Prowler, TARGETp and WoLF PSORT.

BaCelLo. BaCelLo [18] uses a hierarchy of binary SVMs to predict SCL for three eukaryotic kingdoms into four classes for animals and fungi and five classes for plants: secreted, cytoplasm, nucleus, mitochondrion and chloroplast. The predictor is trained on a non-redundant set of experimentally annotated sequences from release 48 of Swiss-Prot. Predictions are made from the full protein sequence, from the N- and C-terminal regions and evolutionary information in the form of a MSA. In [6] the performance of BaCelLo is benchmarked against LOCtree, Protein Prowler, TARGETp and WoLF PSORT with a test set of protein sequences derived from a subset of Swiss-Prot 54. BaCelLo is available at `http://gpcr.biocomp.unibo.it/bacello/`

LOCtree. Similarly to BaCelLo, LOCtree [16] uses binary SVMs to predict SCL. Three versions of the predictor are available, trained specifically for plants, non-plants and prokaryotes. For prokaryotes predictions are into three classes: secreted, periplasm and cytoplasm. In the case of eukaryotes predictions are into six classes: extracellular space, nucleus, cytoplasm, chloroplast, mitochondrion and other organelles. LOCtree is trained on a redundancy reduced subset of release 40 of Swiss-Prot. Predictions are made from the full sequence of the protein, a 50-residue N-terminal region, predicted secondary structure and the output of SIGNALp (for eukaryotes). LOCtree is available at `http://cubic.bioc.columbia.edu/services/loctree/`

Protein Prowler. Protein Prowler [4,10] is based on the ideas behind TargetP and trained on the same datasets, a redundancy reduced subset of Swiss-Prot releases 37 and 38. The predictor uses a series of neural networks and SVMs specialised for the prediction of plants or non-plants and predicts into the following classes: secretory pathway (presence of a signal peptide), mitochondrion (presence of a mitochondrial targeting peptide), chloroplast (presence of a chloroplast transit peptide) and other. Protein Prowler is available at `http://pprowler.itee.uq.edu.au/`

TargetP. TargetP [7] uses a feed-forward neural network specialised for the prediction of plant and non-plant SCL into three and four classes respectively based on the N-terminal amino acid sequence. The prediction is based on the presence of a chloroplast transit peptide (cTP), a mitochondrial targeting peptide (mTP) or a secretory pathway signal peptide (SP). TargetP is available at `http://www.cbs.dtu.dk/services/TargetP/`

WoLF PSORT. WoLF PSORT [11] is a version of the PSORT family of SCL predictors for the prediction of eukaryotic proteins based on their amino acid sequence. Based on a number of features (amino acid composition, the presence of known sorting signal and target peptides etc, with different features for animals, fungi and plants) WoLF PSORT uses a k-nearest neighbour classifier, comparing these features to other Swiss-Prot annotated proteins, resulting in a ranked list of up to 12 possible locations: chloroplast, cytosol, cytoskeleton, endoplasmic reticulum, extracellular, Golgi apparatus, lysosome, mitochondria, nuclear, peroxisome, plasma membrane, vacuolar membrane. WoLF PSORT is available at `http://wolfpsort.org/`

2 Materials and Methods

2.1 Datasets

The first dataset which we use to train and test SCL_pred is the dataset used by [18] to train BaCelLo in ten-fold cross validation, for a direct comparison with this predictor. We call this set the BaCelLo set. We also test this version of SCL_pred on the test datasets used in [6] (BaCelLo_2008 set), which is based on Swiss-Prot 54. Next we combine the BaCelLo and BaCelLo_2008 sets and redundancy reduce the union by an all-against-all PSI-BLAST [1] search (with $e = 10^{-3}$) removing any sequence with a hit with more than 30% sequence identity to any other sequence in the set. Table 1 shows the number of sequences per class for each of the three kingdoms in this new set (BaCelLo_union set).

Using the BaCelLo_union set we re-train SCL_pred in ten-fold cross validation and test it on a independent set extracted from Swiss-Prot 57 (SP_57 set). To create this we first remove from Swiss-Prot 57 any protein present in Swiss-Prot 54 (from which BaCelLo_union is obtained), which leaves 203,860 sequences out of the original 462,764 entries. We then search for metazoa, fungi and viridiplante with an appropriate SCL, that is entries with the keywords "nucleus", "cytoplasm","mitochondrion", "Plastid, chloroplast" or "secreted" in the SUBCELLULAR LOCATION subfield. We exclude membrane proteins, entries with multiple keywords and non-experimental qualifiers (Potential, Probable, By similarity) and sequences with fewer than 30 residues. Then we PSI-BLAST the remaining sequences against Swiss-Prot 54 with $e = 10^{-3}$ and remove any sequences with a hit with more than 30% sequence identity to any sequence in Swiss-Prot 54. Finally we run an internal redundancy reduction on the remaining sequences, removing any entry with more than 90% sequence identity to another sequence in the set. Table 2 shows the number of sequences per class for each of the three kingdoms.

All the BaCelLo datasets are publicly available on the BaCelLo website: `http://gpcr.biocomp.unibo.it/bacello/dataset.htm`

Table 1. Number of sequences per class for each of the three kingdoms in the BaCelLo_union set. See text for details.

	Animals	Fungi	Plants
Cytoplasm	689	466	89
Mitochondrion	263	271	72
Nucleus	1488	884	155
Secreted	881	881	48
Chloroplast			277
Total	3321	1717	641

Table 2. Number of sequences per class for each of the three kingdoms in the SP_57 set. See text for details.

	Animals	Fungi	Plants
Cytoplasm	29	82	1
Mitochondrion	6	55	9
Nucleus	78	84	65
Secreted	107	2	3
Chloroplast			18
Total	220	223	96

MSA. Multiple sequence alignments are extracted from a redundancy reduced, 2004 version of the NR dataset containing over 1 million sequences. The alignments are generated by three runs of PSI-BLAST with parameters $b = 3000$

(maximum number of hits), $e = 10^{-3}$ (expectation of a random hit) and $h = 10^{-10}$ (expectation of a random hit for sequences used to generate the PSSM).

Input coding. Similarly to in [20] the input at each residue is coded as a letter out of an alphabet of 25. Beside the 20 standard amino acids, B (aspartic acid or asparagine), U (selenocysteine), X (unknown), Z (glutamic acid or glutamine) and . (gap) are considered. The input presented to the networks is the frequency of each of the 24 non-gap symbols, plus the total frequency of gaps in each column of the alignment.

2.2 Predictive Architecture: N1-NN

We will operationally call the model we describe in this work N-to-1 Neural Network, or N1-NN. The model is loosely based on previous models we have developed (e.g. [21]) and on our framework to design Neural Networks for structured data [2,23]. In this case, instead of compressing all the information of a sequence into a handful of predefined features (e.g. k-mer frequencies, sequence length, etc.), we decide beforehand only *how many* features we want to compress a sequence into. If these features are stored in a vector $f = (f_1, \ldots, f_h)$, and if we represent the i-th residue in the sequence as r_i, then f is obtained as:

$$f = k \sum_{i=1}^{N} \mathcal{N}^{(h)}(r_{i-c}, \ldots, r_{i+c}) \tag{1}$$

where $\mathcal{N}^{(h)}$ is a non-linear function, which we implement by a two-layered feed-forward Neural Network with h non-linear output units. $\mathcal{N}^{(h)}$ is replicated N times (N being the sequence length), and k is a normalisation constant. Notice that the feature vector f is obtained by combining information coming directly from all windows of $2c + 1$ residues in the protein, and is based on motifs that may be fairly long (e.g. if $c = 10$, as in all the tests in this article, the motifs have a length of 21 residues). The feature vector f thus obtained, is mapped into the property of interest o (for instance, cellular component class), as follows:

$$o = \mathcal{N}^{(o)}(f) \tag{2}$$

where $\mathcal{N}^{(o)}$ is a non linear function which we implement by another 2-layered feed-forward neural network. The whole, compound neural network (the cascade of N sequence to feature vector networks and one feature vector to output network) is itself a feed-forward neural network, thus can trained by gradient descent via the back-propagation algorithm. As there are N copies of $\mathcal{N}^{(h)}$ for a sequence of length N, there will be N contributions to the gradient for this network, which are simply added together. Notice that the feature vector f is not a predefined transformation/compression of the sequence, but instead is automatically learned in order to minimise the output error, hence to be most informative to predict the property of interest. Although there is a daunting number of possible motifs of length $2c + 1$, the model has only a relatively small number of free parameters to represent them, hence does not suffer from overparametrisation problems that arise when one counts frequencies of k-mers as soon as $k > 2-3$. If training is successful, only (soft) motifs relevant to the task at hand will be represented in f. Thus f is effectively a compressed version of the sequence into a fixed-size array, and the compression is property-driven.

The number of free parameters in the overall N1-NN can be controlled by: the number of units in the hidden layer of the sequence-to-feature network $\mathcal{N}^{(h)}()$, N_f^H; the number of hidden units in the feature-to-output network $\mathcal{N}^{(o)}()$, N_o^H; the number of hidden states in the feature vector f, which is also the number of output units in the sequence-to-feature network, N_f. Given that only one instance of the sequence-to-feature network (i.e. only one set of free parameters) is replicated for all positions in the sequence, and there is only one feature-to-output network, the overall number of free parameters N_p of the N1-NN is:

$$N_p = (N_i + 1)N_f^H + (N_f^H + 1)N_f + (N_f + 1)N_f^H + (N_f^H + 1)N_o \qquad (3)$$

where N_i is the size of the input vector representing one residue and N_o is the number of output classes.

Training, Ensembling. For each training experiment (i.e. training on the BaCelLo set and training on the BaCelLo_union set) we implement three predictors, one for each of the three kingdoms of animals, fungi and plants. Each training is conducted by 10 fold-cross validation, i.e. 10 different sets of training runs are performed in which a different tenth of the overall set is reserved for testing. The 10 tenths are roughly equally sized, disjoint, and their union covers the whole set. For each training the 9/10 of the set that are not reserved for testing are further split into a validation set (1/10 of the overall set) and a proper training set. The training set is used to learn the free parameters of the network by gradient descent, while the validation set is used to choose model and hyperparameters (network size and architecture, i.e. N_f^H, N_f and N_o^H). For each different architecture we run three trainings, which differ only in the training vs. validation split. We choose the architecture which performs best on validation. For each fold the three networks for the best architecture are ensemble averaged and evaluated on the corresponding test set. The final results for each 10-cross validation (different kingdoms, BaCelLo and BaCelLo_union sets) are the average of the results on each test set. When testing on an entirely different set from the one used during training (BaCelLo for training and BaCelLo_2008 for testing, BaCelLo_union for training and SP_57 for testing) we ensemble-combine *all* the models from all cross-validation folds of the best architecture. Table 3 shows details of network size and training times for each of the three predictors.

Training is performed by gradient descent on the error, which we model as the relative entropy between the target class and the output of the network. The overall output of the network (output layer of $\mathcal{N}^{(o)}()$) is implemented as a softmax function, while all internal squashing functions are implemented as hyperbolic tangents. Training terminates when either the walltime on the server is reached (6 days for fungi and plants, 10 days for animals) or the epoch limit is reached (40k for fungi and plants and 20k for animals). The gradient is updated 360 times for each epoch (or once every 2-6 examples, depending on the set), and the examples are shuffled between epochs. The learning rate is halved every time a reduction of the error is not observed for more than 50 epochs. Models are saved at regular intervals (every 100 epochs) during training. When training is complete the model with the best performance on the validation set is chosen to be part of the final ensemble for each predictor.

Table 3. Network size and training times for the three network architectures. N_f: size of the feature vector; N_o^H: number of hidden units in the feature-to-output network; N_f^H: number of hidden units in the sequence-to-feature network.

	Animals	Fungi	Plants
N_f	12	10	8
N_o^H	11	7	4
N_f^H	13	11	9
Epochs limit	20k	40k	40k
Walltime	10 days	6 days	6 days

2.3 Evaluating Performance

We measure accuracy/specificity (Cov) and coverage/sensitivity (Acc) per class as in [16,4,7,18] and the geometric average (GAv) as in [16,18]:

$$\begin{aligned} Cov &= 100\frac{TP}{TP+FP} \\ Acc &= 100\frac{TP}{TP+FN} \\ GAv &= \sqrt{Acc.Cov} \end{aligned} \tag{4}$$

where:

- True positives (TP): the number of sequences predicted in a class that are observed in that class.
- False positives (FP): the number of sequences predicted in a class that are not observed in that class.
- False negatives (FN): the number of sequences predicted not to be in a class that are observed in that class.

The overall accuracy of the predictors is measured by Q%:

$$Q\% = 100\frac{\text{number of proteins correctly predicted}}{\text{number of proteins in data set}} \tag{5}$$

3 Results and Discussion

In previous tests BaCelLo [18] was shown to outperform the following publicly available methods for the prediction of the subcellular localization: Loctree [16], Psort II [17], SubLoc [12], ESLpred [3], LOCSVMpsi [24], SLP-local [13], Protein Prowler [4], TARGETp [7], PredoTar [22] and pTARGET [9]. In Table 4 we show the performance of SCL_pred compared to BaCelLo on the BaCelLo sets [18]. Both predictors are assessed by ten-fold cross-validation. Overall SCL_pred is more accurate for animals (77.7% versus 73.8%) and fungi (76% versus 70.1%) while the accuracy for plants is similar (68% versus 68.2%). The accuracy per class differs somewhat, with BaCelLo being more accurate for mitochondrial proteins, especially in the case of plants where the SCL_pred prediction is poor (16.4% versus 54%). However SCL_pred is more accurate for secreted proteins in fungi and plants by 9.5% and 20.6% respectively. Prediction accuracy is similar for proteins in the cytoplasm and nucleus and SCL_pred is again more accurate for chloroplastic proteins by 7%.

Table 5 shows the accuracy of the same version of SCL_pred tested on the two test datasets from [6] compared with the other five SCL predictors tested on the same dataset (results from [6]). SCL_pred performs well for animals and plants (better than all the other servers in 5 out of 8 cases), however it performs less

Table 4. Q% for BaCelLo and SCL_pred, trained and tested in ten-fold cross validation on the BaCelLo set [18], extracted from Swiss-Prot 48

	Animals		Fungi		Plants	
	BaCelLo	SCL_pred	BaCelLo	SCL_pred	BaCelLo	SCL_pred
Cytoplasm	41.4	44.4	39.4	36.0	46.9	46.6
Mitochondrion	66.2	58.5	69.5	67.6	54.0	16.4
Nucleus	84.9	84.8	87.0	88.8	75.7	75.2
Secreted	90.7	90.1	76.9	86.4	64.8	85.4
Chloroplast					76.4	83.4
Q	73.8	**77.7**	70.1	**76.0**	**68.2**	68.0

Table 5. Q% for SCL_pred compared to BaCelLo [18], LOCtree [16], WoLF PSORT [11], Protein Prowler [10] and TARGETp [7]. Tested on the full and reduced (in brackets) BaCelLo_2008 dataset (from Swiss-Prot 54) (see [6] for details). Results for the predictors other than SCL_pred taken from [6].

	Animals		Fungi		Plants	
Predictor	3 Class	4 Class	3 Class	4 Class	4 Class	5 Class
SCL_pred	**92** (89)	**82** (**74**)	78 (79)	55 (52)	**85** (69)	**84** (67)
BaCelLo	89 (**91**)	75 (64)	82 (84)	**59** (**57**)	77 (**76**)	76 (69)
LOCtree	90 (81)	78 (62)	81 (75)	57 (47)	53 (**76**)	52 (**70**)
WoLF PSORT	91 (88)	81 (71)	86 (82)	58 (51)	25 (69)	24 (57)
PProwler	89 (**91**)		**89** (**86**)		19 (63)	
TARGETp	86 (88)		84 (82)		24 (67)	

well on fungi. We interpret these mixed results as a consequence of the small size of the sets: given we take into account the whole, unprocessed sequence, rather than a handful of features extracted from it, the networks we use have at least a few thousand adjustable parameters and SCL_pred is more prone to overfitting the training set than the other systems.

To check whether larger datasets may alleviate the problem, we repeat the experiments on the BaCelLo_union set, which is approximately 30% larger than the BaCelLo set (3321 proteins for Animals, 1717 for fungi, 641 for plants). The accuracy of this new version of SCL_pred is shown in Table 6. We then retest this version of SCL_pred on the SP_57 (a subset of Swiss-Prot 57, described in the dataset section) and again compare its accuracy with BaCelLo, LOCtree, WoLF PSORT, Protein Prowler and TARGETp (Table 7). We obtained results for WoLF PSORT and Protein Prowler through their respective web servers, and results for TARGETp were obtained by downloading the stand alone version of TARGETp available from the TARGETp website, which we then ran locally, hence we have no control on the sequence identity cutoffs between the training sets of these predictors and SP_57. BaCelLo results were kindly provided by Dr Pierleoni. We could not obtain results for LOCtree in this case. In five out of the six cases SCL_pred is the most accurate predictor overall (Table 7).

In Table 8 we show a more detailed analysis of these results for SCL_pred, BaCelLo and WoLF PSORT. It is important to note that due to efforts to reduce

Table 6. Coverage (Cov), accuracy (Acc) and geometric average (GAv) per class for SCL_pred, trained and tested in ten-fold cross validation on the combined and redundancy reduced datasets from [18] and [6] (Swiss-Prot 48 and 54)

	Animals			Fungi			Plants		
	Cov	Acc	GAv	Cov	Acc	GAv	Cov	Acc	GAv
Cytoplasm	53.1	50.5	51.8	60.5	47.0	53.3	46.9	42.7	44.8
Mitochondrion	65.6	64.6	65.1	69.2	66.4	67.8	46.7	19.4	30.1
Nucleus	79.2	81.5	80.3	72.8	82.4	77.4	77.6	78.1	77.8
Secreted	88.8	88.1	88.4	88.4	87.5	88.0	72.7	66.7	69.6
Chloroplast							66.7	79.4	72.8
Q	75.5			70.5			66.3		

Table 7. Q(%) for SCL_pred compared to BaCelLo [18], WoLF PSORT [11], Protein Prowler [10] and TARGETp [7] tested on the SP_57 set

	Animals		Fungi		Plants	
Predictor	3 Class	4 Class	3 Class	4 Class	4 Class	5 Class
SCL_pred	84.5	**68.6**	**89.2**	**68.6**	**82.3**	**82.3**
BaCelLo	**90.0**	66.8	87.9	57.4	76.0	76.0
WoLF PSORT	83.6	68.2	75.8	52.9	71.9	67.7
PProwler	70.5		82.5		77.1	
TARGETp	65.0		80.7		71.9	

Table 8. Coverage (Cov), accuracy (Acc) and geometric average (GAv) per class for the three four-class animal and fungus predictors and the five-class plant predictors tested on the new subset of Swiss-Prot 57

Animals													
	Cytoplasm			Mitochondrion			Nucleus			Secreted			Q
Predictor	Cov	Acc	GAv	Cov	Acc	GAv	Cov	Acc	GAv	Cov	Acc	GAv	
SCL_pred	**25.0**	27.6	**26.3**	25.0	16.7	20.4	65.3	62.8	64.1	85.3	86.9	86.1	68.6
BaCelLo	17.7	**31.0**	23.4	30.0	**50.0**	38.7	67.3	47.4	56.5	94.2	**91.6**	**92.9**	66.8
WoLF PSORT	18.9	24.1	21.4	16.7	33.3	23.6	**72.2**	**66.7**	**69.4**	**95.7**	83.2	89.2	68.2
Consensus	21.2	24.1	22.6	**50.0**	**50.0**	**50.0**	68.5	64.1	66.3	89.8	90.7	90.2	**71.4**

Fungi													
	Cytoplasm			Mitochondrion			Nucleus			Secreted			Q
Predictor	Cov	Acc	GAv	Cov	Acc	GAv	Cov	Acc	GAv	Cov	Acc	GAv	
SCL_pred	**75.6**	**41.5**	**56.0**	**84.0**	76.4	80.1	**61.1**	**91.7**	**74.8**	0	0	0	**68.6**
BaCelLo	48.2	32.9	39.8	78.9	81.8	80.4	52.9	64.3	58.3	**25.0**	**100**	**50.0**	57.4
WoLF PSORT	46.8	26.8	35.4	77.3	61.8	69.1	57.5	72.6	64.6	12.5	50.0	25.0	52.9
Consensus	73.0	32.9	49.0	81.4	**87.3**	**84.3**	60.7	88.1	73.1	20.0	50.0	31.6	67.3

Plants															
	Cytoplasm			Mitochondrion			Nucleus			Secreted			Chloroplast			Q
Predictor	Cov	Acc	GAv	Cov	Acc	GAv	Cov	Acc	GAv	Cov	Acc	GAv	Cov	Acc	GAv	
SCL_pred	0.0	0.0	0.0	0.0	0.0	0.0	**96.9**	**95.4**	**96.1**	**100**	**66.7**	**81.6**	**53.6**	**83.3**	**66.8**	**82.3**
BaCelLo	0.0	0.0	0.0	0.0	0.0	0.0	88.1	90.8	89.4	50.0	**66.7**	57.7	50.0	66.7	57.7	76.0
WoLF PSORT	0.0	0.0	0.0	**50.0**	**22.2**	**33.3**	93.3	86.1	89.7	0.0	0.0	0.0	33.3	38.9	36.0	67.7
Consensus	0.0	0.0	0.0	0.0	0.0	0.0	96.8	92.3	94.5	100.0	**66.7**	**81.6**	50.0	**83.3**	64.5	80.2

redundancy between this new subset of Swiss-Prot 57 and Swiss-Prot 54 (used in training) the number of samples per class is very small in some instances (only one sequence for plant cytoplasm, two for secreted proteins in fungi and three in plants and only six and nine animal and plant mitochondrial proteins respectively).

The most accurately predicted classes for each predictor are the classes with the greatest number of examples: nucleus and secreted in animals; cytoplasm, mitochondrion and nucleus in fungi; and nucleus and chloroplast in plants. Overall SCL_pred continues to perform well, comfortably outperforming BaCelLo and WoLF PSORT in the most densely populated classes for plants and fungi (nucleus and chloroplast, and nucleus and cytoplasm respectively) and also performing well for mitochondrial proteins in fungi. Performance of the animal predictor is more mixed, with none of the three predictors performing well in the less densely populated classes of cytoplasm and mitochondrion. In the other two classes of nuclear and secreted proteins the performance of the three predictors for coverage, accuracy and geometric average is the same for SCL_pred and BaCelLo (75%) when averaged across these three measures for the two classes and 79% for WoLF PSORT. The overall Q performance is slightly better for SCL_pred (68.6%) than for WoLF PSORT (68.2%) and BaCelLo (66.8%). Given the small size (96-223 proteins) of these sets, and their unbalanced nature, further testing on larger, more balanced sets would be desirable when such sets become available.

We also test the accuracy of a consensus prediction between SCL_pred, BaCelLo and WoLF PSORT. The combination of several prediction methods has been used successfully in many cases, for instance for structure predictions at CASP [15]. Here we take a majority vote between the three predictors, and where there is a tie (i.e. each of the three predictors predicts a different class) we trust SCL_pred. The consensus predictor is more accurate for the animal predictor but SCL_pred is more accurate than the consensus for fungi and plants. We do consider that this is an area worth further investigation and a SCL meta-server may be of use to the community of biologists.

4 Conclusion and Future Work

As the amount of sequence information churned out by experimental methods keeps expanding at an ever-increasing pace, it is crucial to develop and make available fast and accurate computational methods to make sense of it. SCL prediction is a step towards bridging the gap between a protein sequence and the protein's function and can provide information about potential protein-protein interactions and insight into possible drug targets and disease processes. As more SCL predictors become available predictions may be combined through the development of meta servers or consensus prediction methods similar to those developed for protein structure prediction and which have been shown to be successful at CASP. As different SCL predictors are specialised for prediction into different classes and number of classes, and as some predictors are more accurate than others at prediction into any one class, this information can be exploited to lead to more accurate overall predictions, especially if the predictors are diverse in their behaviour.

In this article we have developed a new method for SCL prediction (SCL_pred) based on a novel Neural Network architecture (N1-NN). The architecture can map a sequence of any length into a set of individual properties for the whole sequence. We have developed three kingdom specific predictors for animals, fungi and plants and predict into four classes for animals and fungi (nucleus, cytoplasm, mitochondria and the secretory pathway) and an additional fifth class for plants (chloroplast). We have trained SCL_pred in ten-fold cross-validation on two large non-redundant subsets of annotated proteins from Swiss-Prot releases 48 and 54 and benchmarked them against five other state-of-the-art SCL prediction servers on independent sets. SCL_pred performs favourably on these benchmarks and we expect that its prediction accuracy will continue to improve with frequent re-trainings to take advantage of larger, more diverse, datasets of annotated proteins as they become available, and as our understanding of the underlying biological mechanisms improves. We expect larger datasets to be especially beneficial to our models, as these incorporate information from the whole sequence and normally have a higher number of free parameters than the alternatives.

Although here we have only used as input to the network information about the primary sequence and multiple sequence alignments, other residue-level information may be input to the model, such as predicted secondary structure, solvent accessibility, location of predicted binding sites, etc. Incorporating diverse information into the input to SCL_pred is one of our future directions of investigation, as it is the inclusion of putative homology to "templates", or proteins of known localisation/structure (e.g. by techniques similar to those we have developed in [19,14]). A further direction of research is studying the space of *f* vectors (i.e. compressed, property-driven representations of whole proteins as fixed-size arrays) induced by different output targets (functional classes, protein folds/families), to determine whether they are satisfactory representations towards protein comparison, and whether they yield insights into the structure of the protein space.

SCL_pred is available as part of our webservers for protein sequence annotation. Our server is designed to allow fast and reliable annotation of protein sequences on a genomic-scale: up to 32,768 residues can be handled in a single submission. The servers are freely available for academic users at `http://distill.ucd.ie/distill/`. Linux binaries and the benchmarking sets are freely available for academic users upon request.

Funding

This work is supported by Science Foundation Ireland grants 10/RFP/GEN2749 and 05/RFP/CMS0029 and grant RP/2005/219 from the Health Research Board of Ireland.

Acknowledgements

We thank authors of BaCelLo for making their datasets publicly available and in particular we thank Dr Andrea Pierleoni for providing the BaCelLo predictions.

References

1. Altschul, S., Madden, T., Schäffer, A., Zhang, J., Zhang, Z., Miller, W., Lipman, D.: Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. Nucleic Acids Res. 25(17), 3389–3402 (1997)
2. Baldi, P., Pollastri, G.: The principled design of large-scale recursive neural network architectures – DAG-RNNs and the protein structure prediction problem. Journal of Machine Learning Research 4, 575–602 (2003)
3. Bhasin, M., Raghava, G.P.: ESLpred: SVM-based method for subcellular localization of eukaryotic proteins using dipeptide composition and PSI-BLAST. Nucleic Acids Res. 32, W414–W419 (2004)
4. Bóden, M., Hawkins, J.: Prediction of subcellular localization using sequence-biased recurrent networks. Bioinformatics 21(10), 2279–2286 (2005)
5. Boeckmann, B., Bairoch, A., Apweiler, R., Blatter, M., Estreicher, A., Gasteiger, E., Martin, M., Michoud, K., O'Donovan, C., Phan, I., Pilbout, S., Schneider, M.: The Swiss-Prot protein knowledgebase and its supplement TrEMBL in 2003. Nucleic Acids Res. 31, 365–370 (2003)
6. Casadio, R., Martelli, P.L., Pierleoni, A.: The prediction of protein subcellular localization from sequence: a shortcut to functional genome annotation. Brief Funct. Genomic Proteomic 7(1), 63–73 (2008)

7. Emanuelsson, O., Nielsen, H., Brunak, S., von Heijne, G.: Predicting subcellular localization of proteins based on their N-terminal amino acid sequence. J. Mol. Biol. 300, 1005–1016 (2000)
8. Emanuelsson, O.: Predicting protein subcellular localisation from amino acid sequence information. Brief Bioinform. 3(4), 361–376 (2002)
9. Guda, C., Subramaniam, S.: pTARGET: a new method for predicting protein subcellular localization in eukaryotes. Bioinformatics 21, 3963–3969 (2005)
10. Hawkins, J., Bóden, M.: Detecting and sorting targeting peptides with recurrent networks and support vector machines. Journal of Bioinformatics and Computational Biology 4(1), 1–18 (2006)
11. Horton, P., Park, K.J., Obayashi, T., Fujita, N., Harada, H., Adams-Collier, C.J., Naka, K.: WoLF PSORT:protein localization predictor. Nucleic Acids Res. 35, W5857 (2007)
12. Hua, S., Sun, Z.: Support vector machine approach for protein subcellular localization prediction. Bioinformatics 17, 721–728 (2001)
13. Matsuda, S., Vert, J.P., Saigo, H., Ueda, N., Toh, H., Akutsu, T.: A novel representation of protein sequences for prediction of subcellular location using support vector machines. Protein Sci. 14, 2804–2813 (2005)
14. Mooney, C., Pollastri: Beyond the twilight zone: Automated prediction of structural properties of proteins by recursive neural networks and remote homology information. Proteins 77(1), 181–190 (2009)
15. Moult, J.: A decade of CASP: progress, bottlenecks and prognosis in protein structure prediction. Curr. Opin. Struct. Biol. 15(3), 285–289 (2005)
16. Nair, R., Rost, B.: Mimicking cellular sorting improves prediction of subcellular localization. J. Mol. Biol. 348, 85–100 (2005)
17. Nakai, K., Horton, P.: PSORT: a program for detecting the sorting signals of proteins and predicting their subcellular localization. Trends Biochem. Sci. 24, 34–35 (1999)
18. Pierleoni, A., Martelli, P.L., Fariselli, P., Casadio, R.: BaCelLo: a balanced subcellular localization predictor. Bioinformatics 422(14), 408–416 (2006)
19. Pollastri, G., Martin, A.J.M., Mooney, C., Vullo, A.: Accurate prediction of protein secondary structure and solvent accessibility by consensus combiners of sequence and structure information. BMC Bioinformatics 8(201), 12 (2007)
20. Pollastri, G., McLysaght, A.: Porter: a new, accurate server for protein secondary structure prediction. Bioinformatics 21(8), 1719–1720 (2005)
21. Pollastri, G., Vullo, A., Frasconi, P., Baldi, P.: Modular DAG-RNN architectures for assembling coarse protein structures. Journal of Computational Biology 13(3), 631–650 (2006)
22. Small, I., Peeters, N., Legeai, F., Lurin, C.: Predotar: A tool for rapidly screening proteomes for N-terminal targeting sequences. Proteomics 6, 1581–1590 (2004)
23. Walsh, I., Vullo, A., Pollastri, G.: Recursive neural networks for undirected graphs for learning molecular endpoints. In: Kadirkamanathan, V., Sanguinetti, G., Girolami, M., Niranjan, M., Noirel, J. (eds.) PRIB 2009. LNCS, vol. 5780, pp. 391–403. Springer, Heidelberg (2009)
24. Xie, D., Li, A., Wang, M., Fan, Z., Feng, H.: LOCSVMPSI: a web server for subcellular localization of eukaryotic proteins using SVM and profile of PSI-BLAST. Nucleic Acids Res. 33, W105–W110 (2005)

Osmoprotectants in the Sugarcane (*Saccharum* spp.) Transcriptome Revealed by in Silico Evaluation

Petra Barros dos Santos, Nina da Mota Soares-Cavalcanti,
Gabriela S. Vieira-de-Melo, and Ana Maria Benko-Iseppon

Universidade Federal de Pernambuco, Center of Biological Sciences,
Department of Genetics, Laboratory of Plant Genetics and Biotechnology,
R. Prof. Moraes Rêgo 1235, CEP 50670-420, Recife, PE, Brazil
ana.benko.iseppon@pq.cnpq

Abstract. Environmental stresses such as drought and salinity limit crop productivity in worldwide level. These stresses often lead to the accumulation of osmoprotectants in most organisms, including plants. In the present work, a search of known osmoprotectants (*P5CS*, *P5CR*, *INPS1*, *BADH*, *CMO*, *TPS*, *TPP*, *OASTL* and *SAT*) was carried out in the sugarcane transcriptome (237,954 expressed sequence tags) using *in silico* procedures. Alignments revealed that sugarcane presents a high number of osmoprotectant candidate genes, with 56 clusters found. *In silico* expression revealed higher expression in stressed callus tissues and those infected by *Herbaspirilum rubrisubalbicans* (HR), confirming the multi-function character of the osmoprotectants. As expected, the phylogenetic analysis revealed distinct groups among angiosperms, algae, animals, fungi and bacteria, in almost all dendrograms, with high degree of sequence conservation among angiosperms. As observed in comparative analysis between the ORFs of sugarcane and other organisms, the genic structure of these plants was relatively conserved suggesting that the accumulation of compatible solutes is an ancient metabolic adaptation.

Keywords: data mining, salt stress, drought stress, crop evolution, differential expression.

1 Introduction

Among several abiotic stresses, drought and salinity are the most important factors influencing the growth, survival, yield and natural distribution of plants worldwide [1]. Plants have developed mechanisms like stress avoidance to cope with low water content and the synthesis of compatible osmolytes, being one of the main mechanisms that organisms, including plants, have to prevent the harmful effects caused by abiotic stresses [2]. Osmoprotectants are also termed compatible solutes because they are accumulated by the plants without disturbance or interference in the cellular metabolism, besides their protective properties. Proline (Pro), glycinebetaine (GB), myo-inositol, trehalose and cysteine

R. Rizzo and P.J.G. Lisboa (Eds.): CIBB 2010, LNBI 6685, pp. 44–58, 2011.
© Springer-Verlag Berlin Heidelberg 2011

(Cys) are important osmoprotectants that in a general way stabilize proteins and membranes against the denaturing effects of high salt concentrations and other harmful solutes, facilitating the retention of water in the cytoplasm, raising cellular osmotic pressure, and allowing the protection of membranes, protein complexes and cellular constituents [3].

A key to progress towards breeding better crops under abiotic stress has been to understand the changes in cellular, biochemical and molecular machinery of the plants. It is believed that osmoregulation would be the best strategy for abiotic stress tolerance, especially if osmoregulatory genes could be triggered in response to drought, salinity and high temperature [4]. To date, various approaches in genetic engineering have allowed the introduction of new pathways for the biosynthesis of various compatible solutes in plants [3].

The present work performed a data-mining based identification of osmoprotectant genes in the sugarcane database (SUCEST project) [5], using well know rice sequences as templates, and comparing the obtained clusters with sequences from public databases and literature data. Furthermore, the expression profile of the different osmoprotectant categories was evaluated, aiming to identify their role under different conditions in this crop. An *in silico* routine for this purpose is also presented, being applicable for the search of osmoprotectants in monocots in general. As little information is available about these molecules from sugarcane, that is cultived mainly in tropical and subtropical areas and considered one of the world's most important crop plants, it is believed that this research may contribute for breeding purposes in pulse crops like sugarcane.

2 Material and Methods

Protein seed sequences of P5CS, P5CR, TPP, TPS, BADH, CMO and INPS1 from rice were obtained at TIGR (The Institute for Genomic Research) database [6]. Each sequence was compared against the SUCEST database using tBLASTn tool and sequences with e-value equal to e^{-05} or less were annotated. After that a reverse alignment with BLASTx at NCBI was made to confirm de identity of the sequences. Sugarcane clusters were translated using the ORF Finder tool at NCBI and screened for conserved domains using RPS-BLAST CD-search tool [7]. Multiple alignments with proteins from sugarcane and other organisms that presented complete domains were generated at the CLUSTALx program [8]. To avoid influence of sequence sizes in the alignments, the non aligned 5' and 3' extremities were excluded, as well as autapomorphic, non informative internal sequence regions.

A phylogenetic maximum parsimony analysis using bootstrap function (2,000 replications) was performed, generating a consensus tree with a cut-off of 50 using the program MEGA Version 4 for Windows [9]. For each cladogram the most basal organisms have been manually settled as outgroup, in all cases non angiosperms.

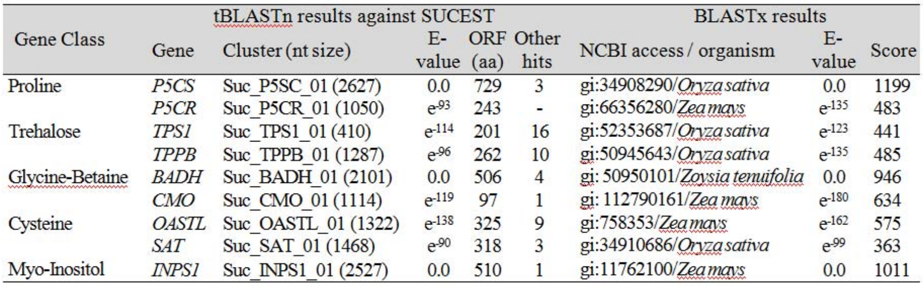
The prevalence of sugarcane clusters were verified by direct counting of the reads that composed each cluster, followed by data normalization (considering the total number of reads sequenced in each library) and calculation of the relative frequency (reads per library). A hierarchical clustering approach was applied using normalized data, from all identified osmoprotectant transcripts, allowing the generation of a graphic representation with aid of the CLUSTER program [10]. The resulting dendrograms including both axes (using the weighted pair group for each gene class and library) were generated using the TreeView program for Windows [11].

3 Results

3.1 Sugarcane Orthologs

Results for all gene categories are summarized in Table 1. The analysis of *P5CS* and *P5CR* osmoprotectant candidates revealed four and one clusters with high sequence conservation, respectively. The clusters for P5CS had high degree of similarity with e-values ranging from 0.0 to e^{-49}, while the single P5CR candidate identified presented an e-value of e^{-93} (Tab. 1). Considering the searched CDs, the best hit from P5CS presented both complete CDs (AA_kinase and ProA) while the other candidates presented partial CDs. Regarding the P5CR candidate the ProC domain was complete.

Table 1. Clusters of sugarcane (best hits) in the SUCEST database as compared with osmoprotectants known from rice, including sugarcane candidates as well as data from best matches from other species. Abbreviations: nt, nucleotide; aa, amino-acid; Suc_, sugarcane candidates.



Gene Class	Gene	tBLASTn results against SUCEST Cluster (nt size)	E-value	ORF (aa)	Other hits	BLASTx results NCBI access / organism	E-value	Score
Proline	*P5CS*	Suc_P5SC_01 (2627)	0.0	729	3	gi:34908290/*Oryza sativa*	0.0	1199
	P5CR	Suc_P5CR_01 (1050)	e^{-93}	243	-	gi:66356280/*Zea mays*	e^{-135}	483
Trehalose	*TPS1*	Suc_TPS1_01 (410)	e^{-114}	201	16	gi:52353687/*Oryza sativa*	e^{-123}	441
	TPPB	Suc_TPPB_01 (1287)	e^{-96}	262	10	gi:50945643/*Oryza sativa*	e^{-135}	485
Glycine-Betaine	*BADH*	Suc_BADH_01 (2101)	0.0	506	4	gi: 50950101/*Zoysia tenuifolia*	0.0	946
	CMO	Suc_CMO_01 (1114)	e^{-119}	97	1	gi: 112790161/*Zea mays*	e^{-180}	634
Cysteine	*OASTL*	Suc_OASTL_01 (1322)	e^{-138}	325	9	gi:758353/*Zea mays*	e^{-162}	575
	SAT	Suc_SAT_01 (1468)	e^{-90}	318	3	gi:34910686/*Oryza sativa*	e^{-99}	363
Myo-Inositol	*INPS1*	Suc_INPS1_01 (2527)	0.0	510	1	gi:11762100/*Zea mays*	0.0	1011

After tBLASTn with the BADH seed sequence the results revealed 24 sequences (e-values from 0.0 to e^{-11}), however after reverse alignment the results revealed five *BADH* candidates, that presented similarity to Poaceae organisms (Tab. 1). With respect to the *CMO* candidates (Tab. 1), the results indicated two candidates, with the best hit presenting high similarity to maize (e^{-119}). Considering the searched CDs, the *BADH* candidates presented complete CDs while the CMO clusters presented only a partial HcaE domain.

The search for orthologous to *TPS* and *TPP* genes revealed the presence of several clusters with high similarity (Tab. 1); 17 and 11 candidates (respectively) were obtained with e-values ranging from e^{-114} to e^{-20}. After reverse alignment most clusters showed similarity to the respective protein from *O. sativa*. Regarding the CD analysis only one TPS sequence had both domains, although the Glico_transferase was incomplete, while all other five sequences had the complete HAD CD. The search for TPP revealed three candidates with the complete HAD-like domain.

The search for *OASTL* in sugarcane database revealed 10 candidates, while the search for *SAT* showed four (Tab. 1). The tBLASTn results for OASTL revealed in most cases great similarity with the used seed sequence (*OASTL*: 12003.m06618, e-values from e^{-138} to e e^{-06}). After BLASTx analysis the identity of all obtained sequences from OASTL and SAT clusters was confirmed, associating them with proteins mainly from *O. sativa*. Considering the CDs in *OASTL* candidates, four clusters beard the complete PALP domain, while in two sequences no CD was found. Regarding *SAT*, two clusters had both complete procured domains (Satase_N and LbetaH).

After tBLASTn in SUCEST two *INPS1* candidates were identified with the best hit presenting an e-value of 0.0. The reverse alignments revealed that these two sequences were similar to myo-inositol 1-phosphate synthase protein from *Z. mays*. After RPS-BLAST the procured domain Inos_1-P_synth was complete in the best hit, being absent at the other sequence.

3.2 Expression Pattern Analysis of Sugarcane Orthologous

After evaluation of the 483 reads identified, it was possible to observe that all SUCEST libraries presented at least one osmoprotectant representative. Considering their distribution, higher prevalence was observed in flower tissues (FL=20%), roots and stem-root transition (RZ and RT=12%, each) (Fig. 1A). Regarding correlation of the distribution among reads and osmoprotectant categories, it was clear that reads involved in the trehalose pathway (TPS+TPP) were most abundant in the SUCEST libraries, followed by OASTL+SAT, and by BADH+CMO (Fig. 1B). After data normalization, a higher expression in calli tissues (CL3 and CL4 libraries) submitted to light/dark and temperature (4^{o}C and 37^{o}C) stresses was observed. Regarding the spatial co-expression among libraries (gray upper dendrogram) it was possible to detect a stronger relation among RZ1/LV1 and CL4/AD1 with LB2 (Fig. 1C). Concerning the co-expression of genes the analysis revealed two groups [INPS1/P5CS] and [SAT/OASTL + BADH + TPS] (Fig. 1C).

3.3 Phylogenetic Analysis

The multiple alignments revealed a considerable degree of conservation among bacteria, fungi, protozoan, algae, plants and animals, considering each osmoprotectant gene category, positioning these groups in distinct branches in most

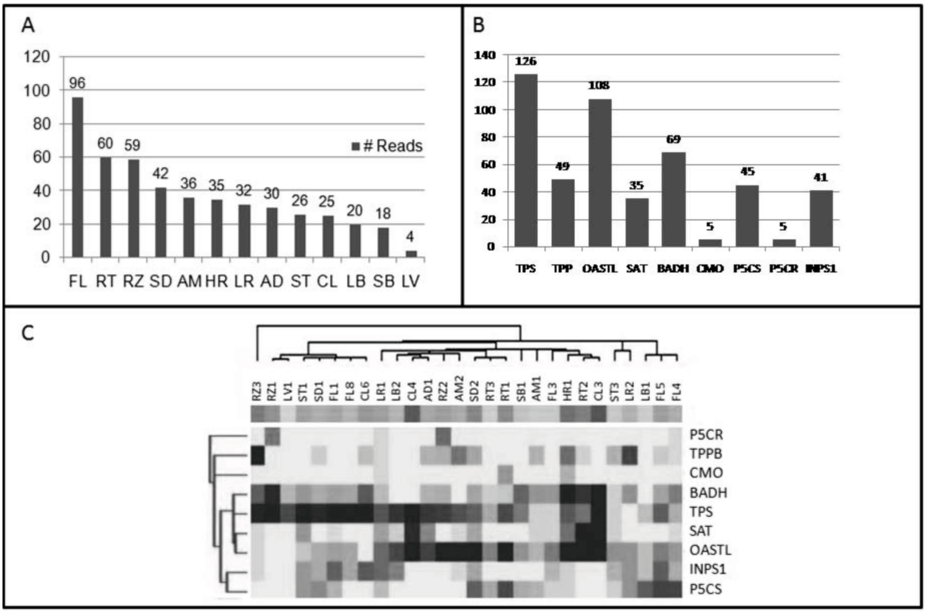


Fig. 1. (A) General distribution of all osmoprotectant transcripts (number of reads) in the SUCEST libraries. (B) Prevalence of reads per osmoprotectant category. Numbers outside the columns refer to the absolute number of reads found in each gene. (C) Differential display of standard sugarcane transcripts representing selected osmoprotectant genes. Graphic represents the expression in P5CS, P5CR, BADH, CMO, TPS, TPP, OASTL, SAT and INPS1 clusters. White means no expression and black to gray mean all levels of expression. Library codes: AD/AD1: tissues infected by *Gluconacetobacter diazotroficans*, AM: Apical meristem from mature plants (AM1) and immature plants (AM2); CL: Calli tissues treated for 12h at 4 to 37^{o}C in the dark or light (CL3, CL4 and CL6); FL: Flowers harvested at different developmental stages (FL1, FL3, FL4, FL5 and FL8); HR: tissues infected with *Herbaspirillum rubrisubalbicans*; and LB: Lateral buds from mature plants (LB1+2); LR: Leaf Roll from immature plants, large insert (LR1) and small insert (LR2); LV/LV1: Etiolated leaves from plantlets grown *in vitro*; RT: Roots from 0.3 cm length (RT1) to mature plants (RT2) and root apex (RT3); RZ Root to shoot transition of young plants zone 1, 2 and 3 (RZ1, RZ2 and RZ3); SB/SB1: Stalk bark from mature plants; SD: Seeds in different stages (SD1 and 2); ST: Stem first internodes (ST1) and fourth internodes (ST3).

generated dendrograms, as expected. For P5CR dendrogram (Fig. 2A) the angiosperms remained as an isolated group (clade V), where the Magnoliopsida (VA) and Liliopsida (VB) formed subgroups. Furthermore, *Chlamydomonas reinhardtii* (Clorophyta) was separated in clade IV, with animals in clade III, while fungi and bacteria remained in the clades II and I, respectively.

Regarding the INPS1 dendrogram (Fig. 2B) the bacteria members were determined as an outgroup (branch I), with remaining organisms positioned in four

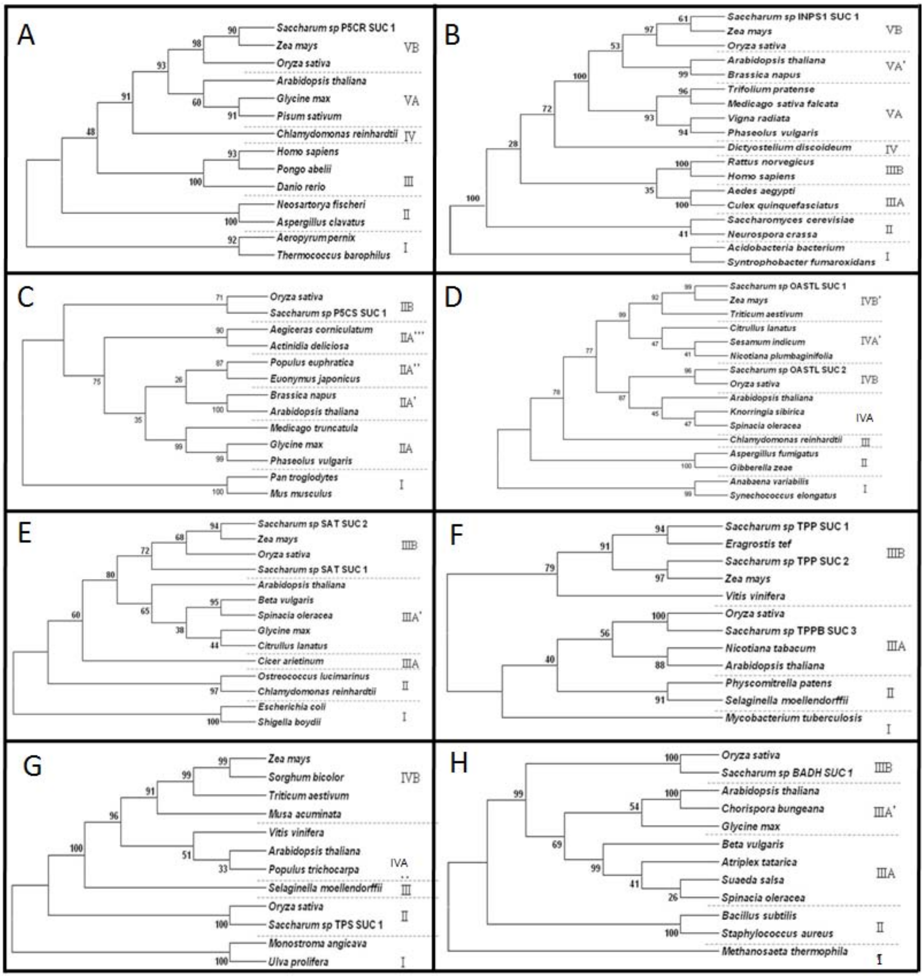


Fig. 2. Dendrograms generated after maximum parsimony analysis illustrating relationships revealed by conserved domains similarity in (A) P5CR-like, (B) INPS1-like, (D) OASTL-like, (E) SAT-like, (F) TPP-like, (G) TPS-like and (H) BADH-like sequences and putative sugarcane orthologous. The numbers in the base of clades regard bootstrap values (2,000 replications).

clades, with two of these subdivided with grouping and subgrouping following a clear segregation according to their classical taxonomic grouping. The subclade VB grouped the monocots, VA and VA' the dicots, IV the amoeba *Dictyostelium discoideum* and III the animals, being this last one subdivided in two subclades: IIIA with the class Insecta and IIIB with Vertevrata class, while the group II comprised only Fungi representative. The other dendrograms for P5SC, OASTL, SAT, TPP, TPS and BADH are available in the figures 3C, 3D, 3E, 3F, 3G and 3H, respectively.

4 Discussion

4.1 Sugarcane Osmoprotectants

Amino acids and their derivatives are the dominant compatible solutes in phylogenetically distant taxa [12]. Among these, almost all plants accumulate high concentrations of proline in response to the imposition of a wide range of biotic and abiotic stresses [13] this is also the case of sugarcane where 56 genes of this category could be uncovered in the present approach. The great similarity of *P5CS* and *P5CR* sugarcane candidates was in accordance to the classic taxonomic relationships, since all significant alignments occurred within the Poaceae family. The presence of proline pathway can be supported by the fact that its accumulation was previously reported for rice [14], maize [15] and sugarcane cultivars [16] subjected to abiotic stress without exogenous proline supply to the medium.

Despite of the fact that no *CMO* candidates had the complete CD, the high similarity of the identified sequences to the same genes of maize supports the existence of this gene in sugarcane. The low number of *CMO* candidates can be explained since (I) the rigidity in choline utilization is not uncommon in plant metabolism and (II) the main factor that limits the accumulation of GB is the availability of choline to allow the oxidation reaction [17]. Many studies have proved that the number of BADH transcripts is higher than CMO in almost all organisms [18], a fact also confirmed by our results.

Orthologues of TPS and TPP, both involved in trehalose synthesis, were found in sugarcane transcriptome. In fact, the trehalose pathway may be activated in many organisms after some type of stress or serving as an accumulation of storage carbohydrates [19]. In angiosperms the trehalose accumulation may occur in plants with diseases or colonized by microorganisms.

According to Hesse et al. (2004) [20] cysteine is required for the production of diverse key metabolites in different pathways: protein synthesis, glutathione, phytochelatins, etc. SAT and OASTL are directly involved in the cysteine biosynthesis, that in plants takes place in cytosol, plastids and mitochondria. A great amount of OASTL transcripts was found in sugarcane, more than SAT candidates, a result supported by evidences from Ruffet et al. (1994) [21], which concluded that in plants endogenous levels of OASTL are far in excess as compared to SAT products, an occurrence explained by the fact that SAT is generally found in association with OASTL, forming the Cysteine Synthase Complex (CSC), while OASTL may also be found in its free form in the cell [22].

Compounds containing inositol are abundant in plant cells [23]. Among these, the myo-inositol has a crucial role in plant physiology and development being involved in signal transduction, phytic acid biosynthesis, auxin storage and transport, cell wall biosynthesis and osmoprotection under salt-water stress [24]. Despite of the absence of induced conditions, 41 candidates could be identified, revealing the presence and diversity of this gene category in sugarcane.

Together these results point to the presence of all important gene categories for the synthesis of compatible osmolytes in sugarcane transcriptome, considering the available knowledge from rice model plant.

4.2 Expression Pattern

The results from SUCEST database revealed the existence of a higher osmoprotectant expression in two callus libraries (CL3 and CL4 = calli submitted to light/dark and temperature stress) indicating that these genes are recruited under additional adverse conditions such as absence of light and temperature changes. Moreover, it is important to highlight that the *in vitro* environment is largely known as an abiotic stress condition [25]. Errabi et al. (2006) [26] have reported higher amounts of proline in sugarcane calli exposed to different osmotic stress intensities. In callus of wheat, increased amounts of glycine betaine were reported in tolerant as well as in sensitive cultivars in response to water stress [27]. The second sugarcane library that had high number of transcripts was HR1, a library constructed with plantlets inoculated with *H. rubrisubalbicans* [28]. In the SUCEST project the infection with this bacteria occurred during *in vitro* cultivation, so the environmental stress cannot be discarded as an inductive factor of compatible osmolytes expression.

Some studies using the hierarquical clustering for expression pattern analysis suggested that genes active in the same pathways are expect to have similar patterns of gene expression and could also be physically clustered in the chromosomes [29] [30]. In *A. thaliana* genome, for example, genes that participate of the same pathway (as *TPS-TPP* and *SAT-OASTL*) are localized in the same chromosome [31]. In contrast, at the rice genome the *CMO* gene is localized on chromosome 6, and two copies of a gene for *BADH* on chromosomes 4 and 8 [32]. It is interesting to note that our results demonstrated co-expression of some libraries and closer co-expression of *SAT/OASTL* sequences, while *CMO* and *BADH* stayed at different branches, in accordance to their distribution in model plants.

4.3 Dendrograms/Phylogenetic Analysis

According to Benko-Iseppon et al. (2005) [33] dendrograms made with genes related to abiotic stress often show groups of sequences in functional arrangements, with common signatures probably reflecting adaptation or selection driven by environmental pressure. The fact that phylogenetically diverse organisms, as bacteria, unicellular algae, fungi, vascular plants, invertebrates and vertebrates utilize the same families of organic osmolytes suggests that these genes are ancient and suffered strong selective pressures associated with convergent evolution [12] [34]. The here generated dendrograms revealed high degree of conservation among sequences of sugarcane and other organisms. As expected, almost all dendrograms reflected the evolutionary history of the analyzed taxa. For example, the P5CR dendrogram (see Fig. 2A) also revealed high conservation among protein sequences of different eukaryotic groups divided in distinct monophyletic branches. Regarding angiosperms it was possible to note that monocots and dicots formed a monophyletic group, subdivided into two groups with the legumes together in the VA subbranch.

The dendrogram based on the INPS1 multiple alignment showed a clear segregation of these proteins into five different monophyletic groups (Fig. 2B) following their phylogenetic position (bacteria, fungi, vertebrates, invertebrates and plants). The subclade VA included legume species, possibly reflecting ancestral characters shared by Fabaceae. Besides, medicago (a salt sensitive plant [35]) grouped with *Trifolium pratense*, a plant affected by moderate salt levels [36], while legumes grouped also together, as expected, since they are more tolerant to water deficit [37].

Concerning the P5CS dendrogram (Fig. 2C) the presence of a symplesiomorphic character among the monophyletic clades of plants and animals was detected. In fact, Fujita et al. (1998) [38] revealed that P5CS proteins from higher eukaryotes are clearly monophyletic. The observed topology for the plants clade clearly showed the existence of two groups, separating the P5CS proteins from monocots and dicots. In bacteria two genes are responsible for proline biosynthesis (*proB* and *proA*), with the two enzymatic domains of P5CS corresponding to the ProB and ProA proteins of *Escherichia coli* [39]. It has been proposed that the corresponding plant genes may have fused and originated the bifunctional enzyme present nowadays in plant genomes [40]. In animal systems, a similar event of domain fusion must have occurred since P5CS activity has been detected in mammalian cells and a single-gene encodes both functional enzymatic activities [41].

Regarding OASTL dendrogram (Fig. 2D) the sequences from bacteria were placed as an outgroup (I), sharing synarcheomorphic characters with the remaining organisms. The Chlorophyta and Embryophyta presented a common archeomorphyc character, as expected, since they belong to the Archaeplastida, the major line of eukaryotes. Concerning the angiosperms, it is clear that different OASTL isoforms were represented in two distinct groups (IVA+IVB and IVA'+IVB'), both formed by Liliopsida and Magnoliopsida species. Considering the monocots subgroup IVB', the OASTL SUC 1 sequence grouped with the cytosolic OASTL isoform from *Z. mays* (gi:758353), as well as the IVA' subgroup that included also sequences from the cytosolic isoform. The second group of monocots (IVB) included the OASTL SUC 2 sequence with the OASTL plastid isoform from *O. sativa* (gi:57899533), which was also the best match in the reverse alignment. Finally, the IVA subgroup included sequences from *S. oleracea* (gi:303902) as well as from *Knorringia sibirica* (gi:186688080) that are described as a plastid isoform of OASTL. Thus, it is interesting to note that the organisms from the Angiospermae division were grouped according to their expression sites, showing the divergent evolution of such isoforms.

The multiple alignments with SAT proteins showed great degree of conservation among different organisms, revealing its ancestral character. The allosteric feedback inhibition of SATase activity by L-Cys is a important regulatory mechanism for OAS levels. Other mechanism is the modulation os SATase activity through reversible formation of protein complex with Cys synthase [42] [43]. The presence of feedback regulation in SATase isoforms differs within plant species and subcellular compartments and has been reported in watermelon (*Citrullus vulgaris* [44]; arabidopsis [45]; *S. oleracea* [46]; and *Glycine max* [47]). It is noteworthy that all these organisms grouped together in the dendrogram, clade IIIA' (Fig. 2E). In contrast, Droux (2003) [48] showed that the cytosolic and mitochondrial isoforms from pea presented insensitivity to Cys inhibition, an interesting feature considering that *Cicer arietinum* (chickpea) stayed in a basal position (IIIA) possibly reflecting these features.

In the TPP dendrogram (Fig. 2F) two distinct groups were formed regarding the flowering plants (both including dicot and monocot representatives), probably due to the different TPP isoforms, in accordance with the data of Lunn (2007) [49] that described up to 10 types of TPP isoforms in plants as arabidopsis, rice and poplar. Moreover in phylogenetic analyses using plant TPP protein sequences, the same authors found that the angiosperm sequences were consistently divided into two major groups, providing robust support for a fundamental dichotomy within the angiosperm TPP family. In the branch IIIB, separated groupings could be observed in the monocot subbranch: the *Saccharum* sp. 1 sequence presented higher similarity with *Eragrostis tef* sequence, while *Saccharum* sp. 2 shared a larger number of characters with a sequence from maize; it is important to stick out that both *E. tef* and maize are recognized by their active TPP enzyme (designated as Ramosa3) that is involved in the control of inflorescence branching by modification of a sugar signal that moves into axillary meristems [50]. Comparing the sequences within branch IIIA it was possible to identify similarities among sugarcane and rice (gi: 45544517) sequences justifying their position in the same branch. According to Pramanik and Imai (2005) [51], this rice TPP isoform is highly induced by chilling stress in shoot and root tissues of seedlings and also by salt and drought stress. Looking for the reads that composed the cluster where *Saccharum* sp. 3 aligned, it was evident that two of the four reads came from root to shoot zone transition of young plants (RZ3 library) a tissue very sensitive to abiotic factors, in consistence with the information given by the authors. Thus, it is possible to infer that these two branches represented two distinct isoforms, the first (IIIB) with proteins involved with inflorescence architecture and the second (IIIA) involved in stress responses.

Regarding the TPP from the *Mycobacterium tuberculosis*, it was observed that it stayed as an out-group, presenting less common traits with the remaining taxa (Fig. 2F). Lunn (2007) [49] suggested that the plant TPP sequences were most closely related to those from bacteria, in consonance with Brown et al. (2001) [52] that proposed that plant TPP genes may have originated from the endosymbiotic ancestor of mitochondria, which is thought to be similar to the actual gene present in some bacteria. With respect to the branch II, it is interesting to note that the group included *Physcomitrella patens* and *Selaginella moellendorffii*, two species strongly resistant to adverse environmental conditions. According Frank et al. (2005) [53], studies with *P. patens* (Bryophyta) showed that this species presents a high salt and drought tolerance, while the *Selaginella* species (Lycopodiophyta) is known as resurrection fern that can recover from almost complete desiccation.

The complete sequencing of the arabidopsis genome revealed 11 genes (AtTPS111) encoding TPS enzymes [31], which are formed by the glucosyltransferase-like and HAD-like domains. In contrast with the other analyzed osmoprotectant genes the generated TPS dendrogram (Fig. 2G) using sequences bearing these two domains did not follow the current phylogenetic systematic. Avonce et al (2006) [54] explained that these inconsistencies are evident only for genes displaying multiple paralogs copies in a single organism, probably due to either lateral gene transfer or differential loss of paralogs. They also showed that all plant TPS genes are under selection pressure suggesting that each of them have a particular function, which could probably be related to other processes, not necessarily related to osmoprotection.

In the BADH dendrogram (Fig. 2H) monocots and dicots were positioned in distinct branches with a high bootstrap value (99). Regarding the dicots, the subclades IIIA and IIIA' comprised members of the Rosidae subclass (Brassicaceae and Fabaceae families) and only members of the Caryophyllidae subclass (respectively), which is known to accumulate far more GB than do other plants in response to osmotic stresses [55]. With respect to the monocots, previous studies suggested that BADH is localized in peroxisomes [56] while, differently, in the dicots the BADH is localized in the chloroplasts [57]. Another difference between monocots and dicots regards the RNA processing pattern. The sequencing data from cDNA clones from diverse monocots demonstrated that all tested sequences had an unusual posttranscriptional processing resulting in deletion(s) of the 5' exonic sequences [58].

5 Concluding Remarks

Together these results point to the presence of all important gene categories for the synthesis of compatible osmolytes in sugarcane transcriptome, considering available knowledge from rice model plant. Many crops lack the ability to efficiently synthesize some types of osmoprotectants that are naturally accumulated by stress-tolerant plants. The production of transgenic plants that can accumulate osmoprotectants will allow the transference of this defense mechanism to important crop plants, such as sugarcane. Therefore, additional studies involving transgenic plants tolerant to stress conditions will help to verify their utility potential in crop-breeding programs. In conclusion, this survey of osmoprotectant-related genes in the sugarcane transcriptome can provide new insights into the study of the genetic and also synthetic pathways of compatible osmolytes, biosynthesis regulation of these compounds and phylogenetic relationship among their natural producers, continuing to expand our knowledge on the evolution and adaptations of flowering-plants to abiotic stresses and providing a framework on which future studies into the function(s) of the osmoprotectants in the plants metabolism may be based.

Acknowledgments. The authors thank CNPq (Conselho Nacional de Desenvolvimento Cientifico e Tecnologico) and FACEPE (Fundacão de Amparo à Pesquisa do Estado de Pernambuco) for the concession of fellowships. To FAPESP (Fundacão de Amparo à Pesquisa do Estado de São Paulo) and SUCEST coordination we thank for the accession to the Sugarcane EST data bank.

References

1. Verslues, P.E., Agarwal, M., Katiyar-Agarwal, S., Zhu, J., Zhu, J.-K.: Techniques for Molecular Analysis. Methods and Concepts in Quantifying Resistance to Drought, Salt and Freezing, Abiotic Stresses that Affect Plant Water Status. Plant J. 45, 523–539 (2006)
2. Yancey, P.H.: Water stress, Osmolytes and Proteins. Amer. Zool. 41, 699-70 (2001)
3. Cherian, S., Reddy, M.P., Ferreira, R.B.: Transgenic Plants with Improved Dehydration Stress Tolerance: Progress and Future Prospects. Biol. Plant. 50(4), 481–495 (2006)
4. Bhatnagar-Mathur, P., Vadez, V., Sharma, K.K.: Transgenic Approaches for Abiotic Stress Tolerance in Plants: Retrospect and Prospects. Plant Cell Rep. 27, 411–424 (2008)
5. Sugar Cane EST Genome Project, `http://www.sucest.lad.dcc.unicamp.br/en/`
6. The Institute for Genomic Research, `http://www.tigr.org`
7. Altschul, S.F., Gish, W., Miller, W., Myers, E., Lipman, D.J.: Basic Local Alignment Search Tool. J. Mol. Biol. 215, 403–410 (1990)
8. Thompson, J.D., Gibson, T.J., Plewniak, F., Jeanmougin, J., Higgins, D.G.: The ClustalX Windows Interface: Flexible Strategies for Multiple Sequence Alignment Aided by Quality Analysis Tools. Nucleic Acids Res. 25, 4876–4882 (1997)
9. Kumar, S., Tamura, K., Nei, M.: MEGA4: Molecular Evolutionary Genetics Analysis (MEGA) Software Version 4.0. Mol. Biol. Evol. 24(8), 1596–1599 (2007)
10. Eisen, M.B., Spellman, P.T., Brown, P.O., Botstein, D.: Cluster Analysis and Display of Genomic-Wide Expression Pattern. Proc. Natl. Acad. Sci. USA. 95, 14863–14868 (1998)
11. Page, R.D.M.: TreeView: An Application to Display Phylogenetic Trees on Personal Computers. Comp. Appl. Biosci. 12, 357–358 (1996)
12. Yancey, P.H., Clark, M.E., Hand, S.C., Bowlus, R.D., Somero, G.N.: Living with water stress: evolution of osmolyte systems. Science 217(24), 1214–1222 (1982)
13. Hare, P.D., Cress, W.A.: Metabolic implications of stress-induced proline accumulation in plants. Plant Growth Regul. 21, 79–102 (1997)
14. Kavi-Kishor, P.B.: Effect of Salt Stress on Callus Cultures of Oryza sativa L. J. Exp. Bot. 39, 235–240 (1988)
15. Ober, E., Sharp, R.: Proline Accumulation in Maize (Zea mays L.) Primary Root at Low Water Potentials. Plant Physiol. 105(3), 981–987 (1994)
16. Kumari, A., Patade, V.Y., Suprasanna, P.: In Silico Analysis of *P5CS* Gene Evolution in Plants. Online J. Bioinf. 9(1), 1–11 (2008)
17. Rontein, D., Basset, G., Hanson, A.D.: Metabolic Engineering of Osmoprotectant Accumulation in Plants. Metab. Eng. 4(1), 49–56 (2002)
18. Jagendorf, A.T., Takabe, T.: Inducers of Glycine-betaine Synthesis in Barley. Plant Physiol. 127, 1827–1835 (2001)

19. Elbein, A.D.: The Metabolism of Alpha-Trehalose. Adv. Carb. Chem. Biochem. 30, 227–256 (1974)
20. Hesse, H., Nikiforova, V., Gakière, B., Hoefgen, R.: Molecular Analysis and Control of Cysteine Biosynthesis: Integration of Nitrogen and Sulphur Metabolism. J. Exp. Bot. 55(401), 1283–1292 (2004)
21. Ruffet, M.L., Droux, M., Douce, R.: Purification and Kinetic Properties of Serine Acetyltransferase Free of O-Acetylserine (thiol) Lyase from Spinach Chloroplasts. Plant Physiol. 104, 597–604 (1994)
22. Hell, R., Wirtz, M., Berkowitz, O., Droux, M.: The Cysteine Synthase Complex from Plants. Mitochondrial Serine Acetyltransferase from Arabidopsis thaliana Carries a Bifunctional Domain for Catalysis and protein-protein Interaction. Eur. J. Biochem. 268, 683–686 (2001)
23. Loewus, F., Loewus, M.W.: Myo-inositol: Its Biosynthesis and Metabolism. Annu. Rev. Plant Physiol. 34, 137–161 (1984)
24. Loewus, F., Murthy, P.: Myo-inositol Metabolism in Plants. Plant Sci. 150, 1–19 (2000)
25. Soares-Cavalcanti, N.M.: Estudo in Silico de Genes que Codificam Fatores de Transcricão Responsivos à Seca, Salinidade e Congelamento nos Genomas do Eucalipto, Cana e Arroz". Master's thesis. Universidade Federal de Pernambuco, Recife-PE, Brazil (2007)
26. Errabii, T., Gandonou, C.B., Essalmani, H., Abrini, J., Idaomar, M., Skali-Senhaji, N.: Growth, Proline and Ion Accumulation in Sugarcane Callus Cultures Under Drought-Induced Osmotic Stress and its Subsequent Relief. Afric. J. Biotech. 5, 1488–1493 (2006)
27. Nayyar, H., Walia, D.P.: Genotypic Variation in Wheat in Response to Water Stress and Abscisic Acid Induced Accumulation of Osmolytes in Developing Grains. J. Agron. Crop. Sci. 190, 39–45 (2004)
28. Lee, S., Reth, A., Meletzus, D., Sevilla, M., Kennedy, C.: Characterization of a Major Cluster of nif, fix, and Associated Genes in a Sugarcane Endophyte, Acetobacter diazotrophicus. J. Bacteriol. 182, 7088–7091 (2000)
29. Lambais, M.R.: In Silico Differential Display of Defense-Related ESTs from Sugarcane Tissues Infected with Diazothrophic endophytes. Genet. Mol. Biol. 24, 103–111 (2001)
30. Wanderley-Nogueira, A.C., Soares-Cavalcanti, N.M., Lima-Morais, D., Silva, L.C.B., Barbosa-Silva, A., Benko-Iseppon, A.M.: Abundance and Diversity of Resistance (R) Genes in the Sugarcane Transcriptome. Genet. Mol. Res. 4, 866–889 (2007)
31. The Arabidopsis Genome Initiative.: Analysis of the Genome Sequence of the Flowering Plant Arabidopsis thaliana. Nature 408(6814), 796–815 (2000)
32. International Rice Genome Sequencing Project. The Map-Based Sequence of the Rice Genome. Nature 436, 793–800 (2005)
33. Benko-Iseppon, A.M., Soares-Cavalcanti, N.M., Wanderley-Nogueira, A.C., Berlarmino, L.C.S., Silva, R., Almeida, P., Brunelli, K., Kido, L., Kido, E.: Genes Associated to Biotic and Abiotic Stresses in Cowpea [Vigna unguiculata (l.) Walp.] and other angiosperms. In: Nogueira, R.J.M.C., Araujo, E.L., Willadino, L.C., Cavalcante, U.M.T. (eds.) Environmental Stresses: Damages and Benefits to Plants, pp. 350–359. UFRPE University Press, Recife-PE (2005)
34. Rhodes, D., Hanson, A.D.: Quaternary ammonium and tertiary sulfonium compounds in higher plants. Annu. Rev. Plant Physiol. 44, 357–384 (1993)

35. Veatch, M.E., Smith, S.E., Vandemark, G.: Shoot biomass production among accessions of Medicago truncatula exposed to NaCl. Crop Sci. Soc. Amer. 44, 1008–1013 (2004)
36. Winter, E., Lauchli, A.: Salt Tolerance of Trifolium alexandrinum L. I. Comparison of the Salt Response of T. alexandrinum and T. pretense. Aust. J. Plant Physiol. 9(2), 221–226 (1982)
37. Omae, H., Kumar, A., Egawa, Y., Kashiwaba, K., Mariko, S.: Adaptation to heat and drought stress in snap bean (Phaseolus vulgaris) during the reproductive stage of development. Jpn. Agric. Res. Q. 40(3), 213–216 (2006)
38. Fujita, T., Maggio, A., Garcia-Rios, M., Bressan, R.A., Csonka, L.N.: Comparative analysis of the regulation of expression and structures of two evolutionarily divergent genes for delta-1-pyrroline-5-carboxylate synthetase from tomato. Plant Physiol. 118, 661–674 (1998)
39. Kishor, P.B.K., Sangam, S., Amrutha, R.N., Laxmi, P.S., Naidu, K.R., Rao, K.R.S.S., Rao, S., Reddy, K.J., Theriappan, P., Sreenivasulu, N.: Regulation of Proline Biosynthesis, Degradation, Uptake and Transport in Higher Plants: its Implications in Plant Growth and Abiotic Stress Tolerance. Curr. Sci. 88(3), 424–438 (2005)
40. Hu, C.A., Delauney, A.J., Verma, D.P.: A bifunctional Enzyme (Delta-1-Pyrroline-5-Carboxylate Synthetase) Catalyzes the First two Steps in Proline Biosynthesis in Plants. Proc. Nat. Acad. Sci. USA 89, 9354–9358 (1992)
41. Turchetto-Zolet, A.C., Margis-Pinheiro, M., Margis, R.: The Evolution of Pyrroline-5-Carboxylate Synthase in Plants: A Key Enzyme in Proline Synthesis. Mol. Gen. Genom. 281, 87–97 (2009)
42. Saito, K., Kurosawa, M., Tatsuguchi, K., Takagi, Y., Murakoshi, I.: Modulation of Cysteine Biosynthesis in Chloroplasts of Transgenic Tobacco Overexpressing Cysteine Synthase [o-acetylserine(thiol)-lyase]. Plant Physiol. 106, 887–895 (1995)
43. Kawashima, C.G., Berkowitz, O., Hell, R., Noji, M., Saito, K.: Characterization and Expression Analysis of a Serine Acetyltransferase Gene Family Involved in a Key Step of the Sulfur Assimilation Pathway in Arabidopsis. Plant Physiol. 137(1), 220–230 (2005)
44. Saito, K., Kurosawa, M., Tatsuguchi, K., Takagi, Y., Murakoshi, I.: Modulation of Cysteine Biosynthesis in Chloroplasts of Transgenic Tobacco Overexpressing Cysteine Synthase [O-Acetylserine(thiol)-Lyase]. Plant Physiol. 106, 887–895 (1995)
45. Noji, M., Inoue, K., Kimura, N., Gouda, A., Saito, K.: Isoform-Dependent Differences in Feedback Regulation and Subcellular Localization of Serine-Acetyltransferase Involved in Cysteine Biosynthesis from Arabidopsis thaliana. J. Biol. Chem. 273, 32739–32745 (1998)
46. Noji, M., Takagi, Y., Kimura, N., Inoue, K., Saito, M., Horikoshi, M., Saito, F., Takahashi, H., Sai, K.: Serine Acetyltransferase Involved in Cysteine Biosynthesis from Spinach: Molecular Cloning, Characterization and Expression Analysis of cDNA Encoding a Plastidic Isoform. Plant Cell Physiol. 42, 627–634 (2001)
47. Chronis, D., Krishnan, H.B.: Sulfur Assimilation in Soybean (Glycine max [L.] Merr.): Molecular Cloning and Characterization of a Cytosolic Isoform of Serine Acetyltransferase. Planta. 218, 417–426 (2004)
48. Droux, M.: Plant Serine Acetyltransferase: New Insights for Regulation of Sulphur Metabolism in Plant Cells. Plant Physiol. Biochem. 41, 619–627 (2003)
49. Lunn, J.E.: Gene Families and Evolution of Trehalose Metabolism in Plants. Funct. Plant Biol. 34, 550–563 (2007)

50. Satoh-Nagasawa, N., Nagasawa, N., Malcomber, S., Sakai, H., Jackson, D.: A Trehalose Metabolic Enzyme Controls Inflorescence Architecture in Maize. Nature 441, 227–230 (2006)
51. Pramanik, M.H., Imai, R.: Functional Identification of a Trehalose 6-Phosphate Phosphatase Gene that is Involved in Transient Induction of Trehalose Biosynthesis During Chilling Stress in Rice. Plant Mol. Biol. 58(6), 751–762 (2005)
52. Brown, J.R., Douady, C.J., Italia, M.J., Marshall, W.E., Stanhope, M.J.: Universal Trees Based on Large Combined Protein Sequence Data Sets. Nature Genet. 28, 281–285 (2001)
53. Frank, W., Ratnadewi, D., Reski, R.: Physcomitrella patens is highly tolerant against drought, salt and osmotic stress. Planta. 220(3), 384–394 (2005)
54. Avonce, N., Mendoza-Vargas, A., Morett, E., Iturriaga, G.: Insights on the Evolution of Trehalose Biosynthesis. BMC Evol. Biol. 6, 109 (2006)
55. Weretilnyk, E.A., Bednarek, S., McCue, K.F., Rhodes, D., Hanson, A.D.: Comparative Biochemical and Immunological Studies of the Glycine Betaine Synthesis Pathway in Diverse Families of Dicotyledons. Planta. 178, 342–352 (1989)
56. Nakamura, T., Yokota, S., Muramoto, Y., Tsutsui, K., Oguri, Y., Fukui, K., Takabe, T.: Expression of a Betaine Aldehyde Dehydrogenase Gene in Rice, a Glycinebetaine Non Accumulator, and Possible Localization of its Protein in Peroxissomes. Plant J. 11(5), 1115–1120 (1997)
57. Rontein, D., Basset, G., Hanson, A.D.: Metabolic Engineering of Osmoprotectant Accumulation in Plants. Metab. Engin. 4(1), 49–56 (2002)
58. Niu, X., Zheng, W., Lu, B., Ren, G., Huang, W., Wang, S., Liu, J., Tang, Z., Luo, D., Wang, Y., Liu, Y.: An Unusual Posttranscriptional Processing in Two betaine aldehyde dehydrogenase Loci of Cereal Crops Directed by Short, Direct Repeats in Response to Stress Conditions. Plant Physiol. 143, 1929–1942 (2007)

IP6K Gene Discovery in Plant mtDNA

Fabio Fassetti[1], Ofelia Leone[1], Luigi Palopoli[1],
Simona E. Rombo[1,2], and Adolfo Saiardi[3]

[1] DEIS, Università della Calabria, Via Pietro Bucci 41C Rende (CS) Italy
{f.fassetti,oleone,palopoli,simona.rombo}@deis.unical.it
[2] ICAR-CNR, Via Pietro Bucci 41C Rende (CS) Italy
[3] LMCB, MRC Cell Biology Unit & Department of Developmental Biology, UCL
dmcbado@ucl.ac.uk

Abstract. IP6 Kinases (IP6Ks) are important mammalian enzymes involved in inositol phosphates metabolism. Although IP6Ks have not yet been identified in plant chromosomes, there are many clues suggesting that the corresponding gene might be found in plant mtDNA, encrypted and hidden by virtue of editing and/or trans-splicing processes. In this paper, we propose an approach to search for the gene *IP6K* and applied it on mitocondrial DNA (mtDNA) of plants. To search for the gene IP6K, we applied a technique based on motif discovery by considering the nucleotide sequence corresponding to a specific tag of the IP6K family. Such a tag has been found in all *IP6K* genes identified up to now, as well as in all genes belonging to the Inositol Polyphosphates Kinases (IPK) superfamily. IPK tag sequence corresponds to the catalytic site of the enzyme and it can be considered as an identifier of IPK genes.

The analysis we conducted provided the relevant negative answer that *IP6K* does not actually occur in vegetable mtDNA.

Finally, we also validated our approach by searching for the known *Ipk1* gene in *Arabidopsis thaliana* genome.

1 Introduction

In the last few years several genomes of different organisms have been completely sequenced. Despite the availability of many sequence data, much of their informational content still remains undiscovered. Many genes are not yet been identified, and the gene number contained in sequenced genomes is still unclear. Computational methods for gene discovering are based on "ab-initio" gene finding (detecting genes by looking for distinct patterns that define where a gene begins and ends), or on comparative gene finding (looking for genes by comparing segments of sequence with those of known genes and proteins). Even if several genes have been discovered by computational genome analysis, many challenges still remain open. Indeed, although such computational methods are very helpful in finding canonic genes, there are situations in which they fail in discovering genes encrypted in the genome due to several complications that may possibly arise. For instance, it is known that from the same gene several proteins can be generated, that two genes can partially overlap, and so on. Furthermore, there are some

R. Rizzo and P.J.G. Lisboa (Eds.): CIBB 2010, LNBI 6685, pp. 59–71, 2011.
© Springer-Verlag Berlin Heidelberg 2011

mechanisms at RNA level, like RNA editing and trans-splicing, that increase the complexity of a gene thus expanding the diversity of proteins of the organism.

Very intriguing is the case of genes whose existence is supposed on the basis of biological considerations, but they are not yet been discovered. One of these genes is IP6 Kinase (*IP6K*) in plants, the gene that encodes the enzyme converting inositol hexakisphosphate (IP_6 or phitic acid) in diphosphoinositol pentakisphosphate (IP_7 or PP-IP_5). Although *IP6K* has not yet been identified in plant chromosomes, there are many clues suggesting its presence in plant cells. Inositol polyphosphates are an important class of regulatory molecules involved in a variety of intracellular signaling pathways, and IP6 Kinases are the mammalian enzymes responsible of their synthesis.

IP6K is the most abundant inositol polyphosphate in eukaryotic cells. It is the precursor of a class of more anionic inositol polyphosphate, the inositol pyrophosphates, in which the fully phosphorylated IP_6 ring is further phosphorylated to create high-energy pyrophosphate group. The best characterized inositol pyrophosphates are the diphosphoinositol pentakisphosphate (IP_7 or PP-IP_5) and the bis-diphosphoinositol tetrakisphosphate (IP_8 or $[PP]_2$-IP_4), with one and two pyrophosphate group, respectively [3].

Inositol pyrophosphates are important cellular messengers that control a wide range of cellular function, including endocytosis [31], apoptosis [22], telomere length [30], and have been argued to be able to drive a new kind of protein post-trasductional modification (protein pyro-phosphorylation) [5]. Since their discovery in the early 1990s, inositol pyrophosphates have been found in *all* analyzed eukaryotic cells, from yeast to mammalian neuron, along with the widespread conservation of the enzymes responsible for their synthesis. The mammalian enzymes responsible for IP_7 synthesis are called IP6 Kinases (IP6Ks); they are able to convert IP_6 plus ATP to IP_7 [28]. It is now known that IP6Ks belong to a superfamily of Inositol Polyphosphates Kinase ($PFAM$ accession number $PF03770$), that evolved from a common ancestor, comprising IP6Ks, Inositol Polyphosphate Multikinase (IPMK) and IP_3-3Ks that specifically convert I(1,4,5)P_3 to I(1,3,4,5)P_4. Interestingly, the presence of pyrophosphate IP_7 has been demonstrated also in vegetable organisms, both in monocotyledonous and in dycotiledonous plants [12,7]. Furthermore, the conversion of IP_6 to IP_7 has been detected in *Arabidopsis* cells and leaf tissue in the presence of ATP, demonstrating IP6-kinase activity in plant extracts[1]. These findings, together with the observed high conservation through the evolution of IP6K, strongly suggest the presence of this enzyme in vegetable cells.

Therefore, IP6K enzyme was searched in plant genomes by homology based methods, but all studies have failed to reveal its presence. Only two IPMK proteins (called AtIPK2_a and AtIPK2_b in *Arabidopsis thaliana*) have been identified so far [34,40]. These two enzymes contribute to inositol 1,3,4,5,6-pentakisphosphate (IP_5) production in Arabidopsis, but do not show any inositol pyrophosphate enzymatic activity [34,40].

[1] Adolfo Saiardi and Cristina Azavedo unpublished manuscript.

However, there are many clues connecting *IP6K* to cell mitochondria. It was shown that human *IP6K2* moves from nuclei to mitochondria and provides physiologic regulation of apoptotic process by generating IP_7 [23]. Furthermore, yeasts deficient in KCS1 (yeast IP6-Kinase), $kcsl\Delta mutants$, do not survive if they are grown in conditions in which survival is dependent from mitochondrial function, thus demonstrating the importance of *IP6K*[2]. Summarizing, to date *IP6K* has not been identified in plant chromosomes, but there are many clues suggesting its presence in vegetable cells. Some further observations could suggest that the corresponding gene might be found in plant mtDNA, probably encrypted and hidden by virtue of editing and/or trans-splicing processes.

It is known that most of mtDNA information concerns genic products acting inside the mitochondrion itself. Plant mitochondrial genomes have several peculiar characteristics with respect to the mammalian ones such as the larger size (from 200Kb to 2400Kb), the presence of introns and genetic material of chloroplast or nuclear origin [25].

Also, mitochondrial genome is characterized by occurrence of phenomena (like RNA editing) enlarging protein variability [36].

On the basis of the above considerations, we decided to search IP6K in mtDNA of plants. Because of the considerable sequence heterogeneity among the several IPKs known, common homology search programs are not useful to this aim. Thus, we decided to use a new approach, looking not for the gene sequence as whole, but for a specific tag sequence, characterizing *IPK* gene family.

In order to search for the IPK family tag in mtDNA sequences, we exploited L-SME [11], a software for motif extraction which allows for the motif structure to be only partially specified by the user. As the main result of our analysis, we can conclude that IP6K does not actually occur in vegetable mtDNA. The negative answer that IP6K gene does not actually occur in vegetable mtDNA is an important result that restricts the search of the gene to plant nuclear DNA. Moreover, to show the goodness of our method, we tested it searching a known gene, already identified by biological techniques. Such a gene is the one coding for inositol 1,3,4,5,6-pentakisphosphate2-kinase (InsP5 2-kinase or Ipk1), the enzyme responsible for the production of inositol hexakisphosphate (IP_6). Ipk1s are unique among inositol phosphate kinases in that they phosphorylate the axial 2-position of the inositide ring, whereas other enzymes act on equatorial position of the ring [14].

The family of enzymes responsible for the synthesis of IP_6 from IP_5 are known as Ipk1. The first *Ipk1* gene was identified in yeast [15] and in other different fungal species. Although functionally conserved, *IPK* genes present very low sequence homology in different organisms, with less than 24% identity in pairwise combinations across the fungal proteins [9]. The sequence identity is limited to a few small regions with high homology. This lack of significant homology initially disallowed the discovery of non-fungal *Ipk1*. After characterization of human *Ipk1* [16], the gene was cloned in *Arabidopsis thaliana* using molecular strategy based on the presence of specific tags in the protein [35]. As a consequence, in this

[2] Adolfo Saiardi unpublished manuscript.

paper, we searched *Ipk1* in *Arabidopsis thaliana* genome with L-SME software, exploiting the presence of specific tag in *Ipk1* gene family. This approach leads us to easily find the gene, allowing us to validate the method, that appears general and very useful when homology search strategies cannot be used.

The paper is organized as follows. In Section 2 we describe our approach and the tuning settings of DLME that we exploited for our purposes. In Section 3 we illustrate the main results of our analysis, while in Section 4 we discuss their biological relevance. Finally, in Section 5, we draw our conclusion.

2 Methods

Common software for sequence search is based on sequence homology, but it is not very useful when the expected homology between the gene searched for and the known sequences is low. Furthermore, this software cannot detect possible changes in nucleotide sequences due to RNA editing mechanisms. The intuition behind our work is that all *IPK* genes are characterized by the presence of specific tags, short sequences of few amino acids corresponding to enzyme functional regions. The most important one is the `P-XXX-D-X-K-X-G` domain, corresponding to the catalytic site of the enzyme. For the identification of *IP6K* gene in plant mtDNA, we focused on the nucleotide sequence corresponding to this specific IPK tag. In particular, we analyzed all the published mtDNA sequences (available at `http://www.ncbi.nlm.nih.gov/sites/ entrez`) and performed motif extraction from them, since a tag can be viewed as a subsequence whose structure is not completely specified a priori, that is, a special kind of *motif*. Among the different algorithms and tools available for motif discovery (e.g., see [6,39,21,10,2,27]), we chose L-SME [11] since it is able to handle different complex kinds of pattern variabilities, as will be better recalled in the following.

For each identified tag, we extracted a sequence of about 1200 nucleotides surrounding the consensus sequence and examined it as a candidate *IP6K* gene. Nucleotidic sequences were translated into amino acid sequences by using the *Transeq* [26] software. Then, in order to detect possible homologies, we performed sequence alignments using *ClustalW* [17] and BLAST [1]. Finally, using the TBLASTX and TBALSTN algorithms, we screened expressed sequence tag (EST) databases for proteins containing the mitochondrial sequences identified by our tag search.

In the following, we briefly describe the L-SME systems, the methodology to perform the tag search and the settings we exploited for our goals.

2.1 L-SME

L-SME [11] is a system designed to mine general kinds of motifs where several "exceptions" may be tolerated; that is, it is able to handle different complex kinds of pattern variabilities. In particular, L-SME is able to search for patterns composed of any number of short subsequences (boxes, in the following), where both the region lengths and the distances between regions can be specified by the user as an interval ranging from a minimum to a maximum value. Moreover, mismatches are taken into, as well as “skips” (deletions) and box “swaps” (box invertions), that possibly affect box occurrences. Furthermore, in L-SME, it is possible to specify boxes where some symbols are “anchored” to get a fixed value. Despite the complexity of the addressed pattern variabilities, the system is able to exhibit very good performances.

The flexibility of the method in specifying variable distances between two boxes, easily allows possible introns to be taken into account.

In order to limit the great variability introduced by considering introns, we adopted an incremental approach consisting in iteratively increasing the number of introns. In particular, we did not take care of any intron at the beginning and, then, we considered the presence of n introns by incrementing the distance between n pair of boxes of the typically maximum length of an intron (e.g., 100 bases for *Arabidopsis thaliana*).

The adopted method is illustrated in Figure 1.

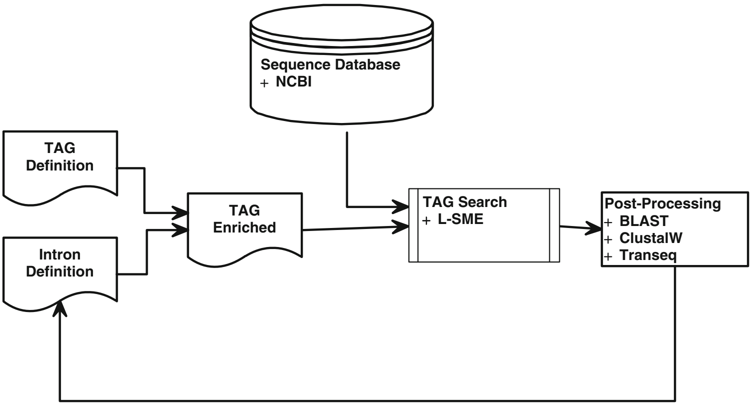


Fig. 1. Summary of the Method

2.2 L-SME Settings

For the purposes of this research, we looked for the pattern:

[CC{T,C,A,G}]---------[GA{T,C}]---[AA{A,G}]---[GG{T,C,A,G}],

where the square brackets delimite the boxes and the hyphens denote the distances between boxes. The configuration parameters of L-SME are reported in Figure 2.

Distance:	Hamming
Number of skips:	0
Number of swaps:	0
Number of boxes:	4

First box length:	3	*Second box length*:	3
Distance from second box:	9	*Distance from third box*:	3
First box anchors:	CCT CCC CCA CCG	*Second box anchors*:	GAT GAC
Third box length:	3	*Fourth box length*:	3
Distance from fourth box:	3	*Fourth box anchors*:	GGT GGC GGA GGG
Third box anchors:	AAA AAG		

Fig. 2. L-SME parameters configuration

As for the validation carried on the *Ipk1* gene, we looked for the pattern composed by the two regions EIKPK and R-XX-MHQ-X-LK characterizing the searched gene. Such regions typically occur at a distance ranging from 9 to 19 amino acids. Then, the corresponding pattern in the genome sequence is:

```
GA{A,C}AT{T,C,A}AA{A,G}CC{T,C,A,G}AA{A,G}--...--
{AGA,AGG,CGT,CGC,CGA,CGG}------{ATG}CA{T,C}CA{A,G}---
{TTA,TTG,CTT,CTC,CTA,CTG}AA{A,G}
```

where the number of boxes is 11 and the length of each box is 3. The distance between the fifth and the sixth box, corresponding to the distance between the region EIKPK and the region R-XX-MHQ-X-LK, is set to the interval [27 – 57]. As for the distances between the other boxes, they are set according to the above described method.

3 Results

The full mitochondrial genome sequence is known for 39 different vegetable organisms, belonging to various Phyla, even very distant from an evolutionary point of view. The specific IP6Ks tag (P-XXX-D-X-K-X-G) search was performed over all sequenced mitochondrial genomes available to date and both DNA strands were analyzed. Twentythree genomes out of 46 gave at least one positive match. Interestingly, we noted that some tag sequences (all 9 amino acids) were identical among different organisms. For each identified tag we extracted a sequence of about 1200 nucleotides surrounding it. To find out possible relevant homologies, we performed alignments among the sequences found in different vegetable organisms. All the sequences sharing the same tag showed high homology in the region surrounding the consensus sequence, while alignment with *IP6K* known genes (yeast *KCS1* or human *IP6K1*) showed only a weak similarity. Furthermore, in order to confirm the identity of our putative hit, we looked for other IP6Ks conserved domains in the identified putative amino acid sequences. These domains comprise the ATP binding site, first characterized in IP3-3K [8], the C-terminal domain (last 19 amino acids), important for the catalytic activity [37], and the "SSLL" domain, required for enzymatic activity of IP6K [29]. These preliminary analysis led us to focus on the sequence `PVGTDRKGG`, that was found in mtDNA of *Tripsacum dactyloides*, *Sorghum bicolour*, and three different species of *Zea genus* (*Zea mays*, *Zea perennis* and *Zea luxurians*). Alignment between the 410 amino acids around the `PVGTDRKGG` sequence of *Tripsacum dactyloides* and the human *IP6K* gene showed an interesting correspondence of the consensus region (see Figure 3). Furthermore, in the sequence around the tag, there are many identitical and conserved or semiconserved amino acids. With regard to other conserved domains, we did not find a good correspondence between Tripsacum sequence and known *IP6K* genes, but these regions are not as conserved as the `P-XXX-D-X-K-X-G` tag.

In order to confirm the goodness of our method, we applied it on nuclear DNA of Arabidopsis thaliana, searching *Ipk1* gene. We looked for the `EIKPK` motif (box A) and the `R-XX-MHQ-X-LK` motif, both present in all *Ipk1* genes discovered up to now; they are separated each other from a variable number of amino acids (19 amino acids in human and rat *Ipk1*, 9 in yeast *Ipk1*). We found the conserved sequence on chromosome 5 of *Arabidopsis thaliana* genome. In particular we had only one positive match when we considered the possibility of occurrence of an intron between the third and fourth amino acid of EIKPK motif. The intron length resulted to be 82 nucleotides and distance between two `EIKPK` and `R-XX-MHQ-X-LK` motifs 63bp. We extracted a sequence of about 2000 nucleotides around the tags. BLAST allignements showed that the sequence was *Arabidopsis thaliana Ipk1* gene.

To verify if the *Tripsacum dactyloides* sequence is an actively transcribed gene, we analyzed the Expressed Sequence Tags (ESTs) databases. These databases include short fragment of DNA derived from a longer cDNA sequence and representing part of the expressed genome. In order to confirm the expression of the considered mtDNA sequence, we screened EST databases using the region surrounding the `PVGTDRKGG` tag of *Tripsacum dactyloides.* This search failed to find any EST matching indicating that our putative hit is unlikely to be transcribed in mtRNA. Finally, we used a region of 50 amino acids of *Tripsacum dactyloides* mtDNA surrounding the consensus sequence to perform a multiple alignment with corresponding regions of inositol phosphate kinase (IPMK, IP6K, IP3-3K) from different organism using *ClustalW2.* As shown in Figure 4, the found sequence resulted to be an outsider, thus that it does not belong to any subgroup of kinase composing the *IPK* gene family.

SeqA	Name	Len(aa)	SeqB	Name	Len(aa)	Score
1	Tripsacum	410	2	Human	410	6

```
CLUSTAL 2.0.12 multiple sequence alignment

Tripsacum -PTEILSEY-K--KAISLWWYTSRQFWNFQFQSEHIDPSMDLYV-PLQSCSSFLATSSIF 55
Human     MVVQNSADAGDMRAGVQLEPFLHQVGGHMSVMKYDEHTVCKPLVSREQRFYESLPQAMKR 60
            .:  ::  .   .:.*  :  :   ::.. . . ..  .  *   *   . *. :

Tripsacum FLGTC--TRNSYVRDSSSENLPVFHSHMR--QESLWATGRHEVIHHVQT-TFRSLGTVYK 110
Human     FTPQYKGTVTVHLWKDSTGHLSLVANPVKESQEPFKVSTESAAVAIWQT-LQQTTGSNGS 119
          *      * . :: ..*: :*.:. . ::  **.: .: .  .:   **   :: *:  .

Tripsacum S-NSHKWNEKWVHVNDRDLANNNVPSPYGVRDPKAIVE-TVSYSY-LT-AAPLLQGWG-S 165
Human     DCTLAQWPHAQLARSPKESPAKALLRSEPHLNTPAFSLVEDTNGNQVE--RKSFNPWGLQ 177
          . .  :* .  :  . :: . : :  .    :. *:     : .  :      :: ** .

Tripsacum ASEPYVGRVSASYSSGRRPGKLRRD--GETQLLPVGTDRKGG------GDKLVKKQAYCP 217
Human     CHQAHLTRLCSEYPENKRHRFLLLENVVSQYTHPCVLDLKMGTRQHGDDASEEKKARHMR 237
          . :.:: *:.:.*...:*   *  :   .    *   * * *      . .  **  :

Tripsacum TPKKQTKKTYAL-QQSAHPLLVASRFHPSP--FRDRRLI-YVQ-SSSDQSARTPDRLCPP 272
Human     KCAQSTSACLGVRICGMQVYQTDKKYFLCKDKYYGRKLSVEGFRQALYQFLHNGSHLRRE 297
          .  :.*.   .:   . :   . .::. .   : .*:*      .:  *  :. .:*

Tripsacum ILSRTKWNGSLILVTLCPDPSPHVRFYPPATRPTQH-GRPPPHSMLTRAGARFLGSPFPP 331
Human     LLEPILHQLRALLSVIRSQSS--YRFYSSSLLVIYD-GQEPPE--------RAPGSPHPH 346
          :*.    :   :* .: .:.*   ***..:     . *: **.        *  ***.*

Tripsacum RS-RPGWPACGSGNSPVPW-KKGWLDAGSTPRGAVRT-MISSRPLFAYR-GCLTPLRQLA 387
Human     EAPQAAHGSSPGGLTKVDIRMIDFAHTTYKGYWNEHTTYDGPDPGYIFG---LENLIRIL 403
          .: :..  :. .* : *     .: .:  .     :*   .. * : :    *  * ::

Tripsacum LPALSCL                                                      394
Human     QDIQEGE                                                      410
              .
```

Fig. 3. Alignment between the 410 amino acids around the PVGTDRKGG sequence of *Tripsacum dactyloides* and the human *IP6K* gene (Clustal W)

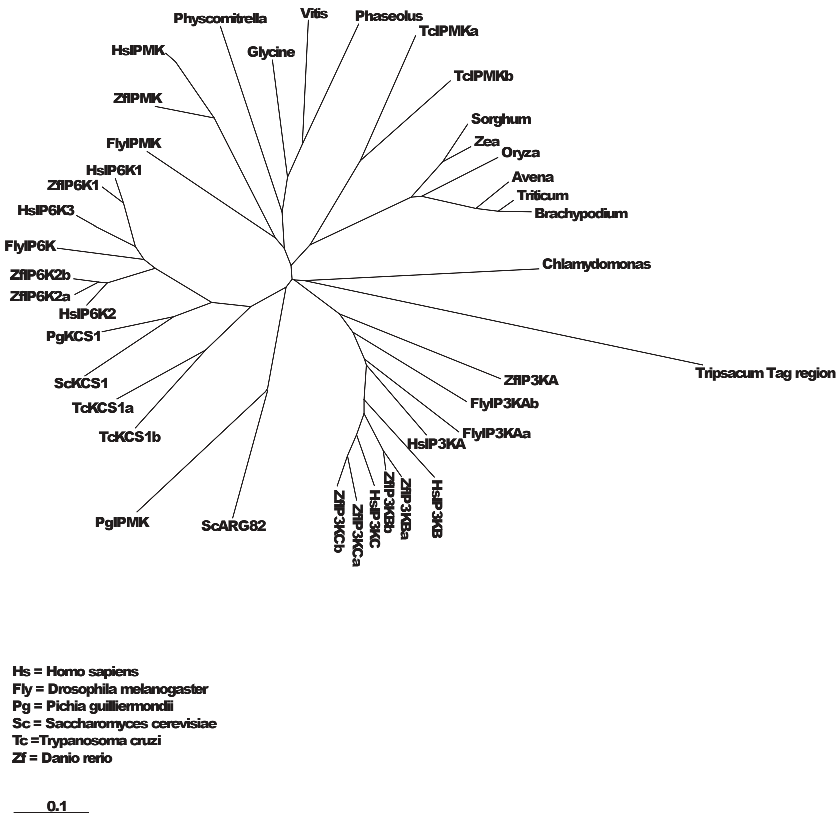


Fig. 4. Multiple alignment of a 50 aa region of *Tripsacum dactyloides* mtDNA surrounding the tag with corresponding regions of inositol phosphate kinase (IPMK, IP6K, IP3-3K) from different organism (ClustalW2)

4 Discussion

IP6K is a gene found in many different organisms but it has not yet been identified in plant genomes. The enzyme catalyzes the conversion of IP_6 to IP_7 using ATP as phosphate donor. It belongs to an inositol polyphosphate kinase superfamily, the IPKs (Pfam *PF*03770), that evolved from a common ancestor. It is thought that a primordial IPMK may have been the evolutionary precursor of the IP3-3Ks and the IP6Ks, all of which contain the `P-XXX-D-X-K-X-G` domain [32]. Thus this domain represents a unique consensus sequence for the IPK family, with four key amino acids very conserved among different inositol phosphate kinases, despite their considerable sequence heterogeneity. This domain modulates the catalytic site for phosphate transfer from ATP to the inositol ring [4]. The inositol pyrophosphate IP_7 is present in all eukaryotic cells analyzed thus far, from amoeba to human; it is not surprising that the enzyme responsible for its synthesis is highly conserved through evolution. Indeed, after the first IP6K purification from rat brain [38], the enzyme was cloned in other mammalians, and its high evolutionary conservation was regularly observed, thus facilitating the identification and cloning of IP6K enzymes from distant organisms, including yeast and the amoeba *Dictyostelium* [20]. It is notable that *Dictyiostelium* diverted from the evolutionary mainstream after the diversion of yeast but before the splitting between animals and plants [19]. Furthermore, the only *IPK* gene present in the ancient eukaryote diplomate *Giardia lamblia* has been demonstrated to be *IP6K* [18]. Thus, on the basis of evolutionary considerations, *IP6K* is expected to be found also in vegetable organisms.

Moreover, pyrophosphate IP_7 is present in vegetable organisms, and IP6-kinase activity has been demonstrated in plants. However, bioinformatics analysis failed to identify any IP6 kinase in the complete *Arabidopsis thaliana* nuclear genome. We hypothesized that *IP6K* gene actually occurs nested in vegetable mtDNA, where phenomena enlarging protein variability occur more frequently. Tags identification in mtDNA could indicate the presence of *IP6K* gene, even if not in a canonic form. Indeed there are some mechanisms that can alter the linearity of genetic information transport from DNA to protein. Most of them occur at RNA level and the most known is RNA editing. Trans-splicing is a further process generating genetic variability, in which two RNA molecules, produced by different DNA regions (even very distant from each other), are joined in a single RNA molecule able to produce a protein. Indeed, trans-splicing mechanisms could compact a gene consisting of more segments dislocated in different mtDNA regions, and editing phenomena could account for the failure of homology searches. Indeed mechanisms altering the linearity of genetic information transport, like RNA editing and trans-splicing, can generate RNA molecules much different from DNA producing them, so that DNA sequence can be not immediately referable to *IP6K* gene in its transcript. The occurrence of these phenomena could account for the failure of homology searches. Thus, the search of a gene starting from its characterizing consensus sequence represents a promising approach to find an encrypted gene.

To confirm the goodness of our method we searched *Ipk1* gene in *Arabidopsis thaliana* genome. We found both the `EIKPK` and the `R-XX-MHQ-X-LK` region, separated from 21 amino acids on *Arabidpsis thaliana* chromosome 5. Surprisingly we found an intron of 82 nucleotides between the third and the fourth amino acid of the region `EIKPK`. The presence of introns inside a tag is very unlikely because tags are very short sequence, usually less then ten amino acids. Furthermore, the most reliable theory about the function of introns is that they separate gene segments that code functional domains of proteins [33]. This organization of discontinuous genes provides insights on the evolution of complex eukaryotic genes. According to the model of Gilbert [13], exons represent protein domains that were "marshalled" together during evolution [24]. Consequently, introns can exist between tags, but the presence of a intron inside one single tag is highly improbable. Thus, even if the `EIKPK` region is very conserved and it has been proposed to be functionally important, the presence of an intron inside it could indicate that the domain is actually restricted to the first three amino acids.

5 Conclusion

There are many suggestions, both theoretical and experimental, indicating the presence of IP6K gene in plant cells. Furthermore several clues connect *IP6K* to cell mitochondria, like the experimental observation that IP6K2 moves from nuclei to mitochondria and the demonstration that *IP6K* is essential for these organelles. Thus, we decided to perform the *IP6K* gene search on mitochondrial DNA of plants, where we argue it could have been contained in an encrypted form.

We searched for a specific IP6K tag within all available plant mtDNA sequences using L-SME, a very flexible system for motif discovery, allowing us to deal with genetic code degeneration and possible occurrences of editing events. The search we performed pointed out the presence of several tags in mtDNA of examined plants, but an accurate analysis of sequences surrounding the consensus motifs led us to conclude that our hits do not belong to the *IPK* gene family. The `P-XXX-D-X-K-X-G` consensus sequence is a characterizing motif of IP kinases, and it was found in all members of the family. Thus, the presence of the tag is essential to assign a gene to IPK family, but a sequence containing the tag is not necessarily an *IPK* gene. Our search discovered several tags in mtDNA of plants, but any sequence containing the tag was not ascribable to *IP6K* gene and therefore we are led to conclude that *IP6K* gene is not present in plant mtDNA. We finally validated the effectiveness of our approach by searching for the known *Ipk1* gene in *Arabidopsis thaliana* genome.

As the future work, we plan to extend the search of the IP6K tag on the nuclear genome of plants.

References

1. Altschul, S.F., Madden, T.L., Schaffer, A.A., Zhang, J., Zhang, Z., Miller, W., Lipman, D.J.: Gapped blast and psi-blast: a new generation of protein database search programs. NAR 25(17), 3389–3402 (1997)
2. Apostolico, A., Gong, F.-C., Lonardi, S.: Verbumculus and the discovery of unusual words. Journal of Computer Science and Technology 19(1), 22–41 (2004)
3. Bennett, M., Onnebo, S.M., Azevedo, C., Saiardi, A.: Inositol pyrophosphates: metabolism and signaling. Cell Mol. Life Sci. 63, 552–564 (2006)
4. Bertsch, U., Deschermeier, C., Fanick, W., Girkontaite, I., Hillemeier, K., Johnen, H., Weglhner, W., Emmrich, F., Mayr, G.W.: The second messenger binding site of inositol 1,4,5-trisphosphate 3-kinase is centered in the catalytic domain and related to the inositol trisphosphate receptor site. J. Biol. Chem. 275, 1557–1564 (2000)
5. Bhandari, R., Saiardi, A., Ahmadibeni, Y., Snowman, A.M., Resnick, A.C., Kristiansen, T.Z., Molina, H., Pandey, A., Werner, J.K., Juluri, K.R., Xu, Y., Prestwich, G.D., Parang, K., Snyder, S.H.: Protein pyrophosphorylation by inositol pyrophosphates is a posttranslational event. Proc. Natl. Acad. Sci. U S A 104(39), 15305–15310 (2007)
6. Brazma, A., Jonassen, I., Eidhammer, I., Gilbert, D.: Approaches to the automatic discovery of patterns in biosequences. Journal of Computational Biology 5(2), 277–304 (1998)

7. Brearley, C.A., Hanke, D.E.: Inositol phosphates in barley (hordeum vul. l.) aleurone tissue are stereochemically similar to the products of breakdown of insp6 in vitro by wheat-bran phytase. Bioch. J. 318(1), 279–286 (1996)
8. Communi, D., Takazawa, K., Erneux, C.: Lys-197 and asp-414 are critical residues for binding of atp/mg2+ by rat brain inositol 1,4,5-trisphosphate 3-kinase. Biochem J. 291, 811–816 (1993)
9. Ives, E.B., Nichols, J., Wente, S.R., York, J.D.: Biochemical and functional characterization of inositol 1,3,4,5,6-pentakiphosphate 2-kinases. The Journal of Biological Chemistry 275, 36575–36583 (2000)
10. Eskin, E., Pevzner, P.A.: Finding composite regulatory patterns in DNA sequences. Bioinformatics 18, S354–S363 (2002)
11. Fassetti, F., Greco, G., Terracina, G.: Mining loosely structured motifs from biological data. IEEE Trans. Knowl. Data Eng. 20(11), 1472–1489 (2008)
12. Flores, S., Smart, C.C.: Abscisic acid-induced changes in inositol metabolism in spirodela polyrrhiza. Planta. 211, 823–832 (2000)
13. Gilbert, P.: Why genes in pieces? Nature 271(5645), 501 (1978)
14. Gonzales, B., Banos-Sanz, J.I., Villate, M., Brearley, C.A., Sanz-Aparicio, J.: Inositol 1,3,4,5,6-pentakisphosphate 2-kinase is a distant ipk member with a singular inosite binding site for axial 2-oh recognition. Proc. Natl. Acad. Sci. U S A 107(21), 9608–9613 (2010)
15. York, J.D., Odom, A.R., Murphy, R., Ives, E.B., Wente, S.R.: A phospholipase c-dependent inositol polyphosphate kinase pathway required for efficient messanger rna export. Science 285, 96–100 (1999)
16. Verbsky, J.W., Wilson, M.P., Kisseleva, M.V., Majerus, P.W., Wenter, S.R.: The synthesis of inositol hexakiphosphate. characterization of human inositol 1,3,4,5,6-pentakiphosphate 2-kinase. The Journal of Biological Chemistry 277, 31857–31862 (2002)
17. Larkin, M.A., Blackshields, G., Brown, N.P.: ClustalW and ClustalX version 2. Bioinf. 23(21), 2947–2948 (2007)
18. Letcher, A.J., Schell, M.J., Irvine, R.F.: Do mammals make all their own inositol hexakisphosphate? Biochem. J. 416(2), 263–270 (2008)
19. Loomis, W.F., Smith, D.W.: Consensus phylogeny of Dictyostelium. Experientia. 51(12), 1110–1115 (1995)
20. Luo, H.R., Huang, Y.E., Chen, J.C., Saiardi, A., Iijima, M., Ye, K., Huang, Y., Nagata, E., Devreotes, P., Snyder, S.H.: Inositol pyrophosphates mediate chemotaxis in Dictyostelium via pleckstrin homology domain-PtdIns(3,4,5)P3 interactions. Cell 114(5), 559–572 (2003)
21. Marsan, L., Sagot, M.-F.: Algorithms for extracting structured motifs using a suffix tree with application to promoter and regulatory site consensus identification. J. of Comput. Biol. 7, 345–360 (2000)
22. Morrison, B.H., Bauer, J.A., Hu, J., Grane, R.W., Ozdemir, A.M., Chawla-Sarkar, M., Gong, B., Almasan, A., Kalvakolanu, D.V., Lindner, D.J.: Inositol hexakisphosphate kinase 2 sensitizes ovarian carcinoma cells to multiple cancer therapeutics. Oncogene 21(12), 1882–1889 (2002)
23. Nagata, E., Luo, H.R., Saiardi, A., Bae, B., Suzuki, N., Snyder, S.H.: Inositol hexakisphosphate kinase-2, a physiologic mediator of cell death. J. Biol. Chem. 280(2), 1634–1640 (2005)
24. OMalley, B.W., Stein, J.P., Means, A.R.: The evolution of a complex eukaryotic gene. Metabolism 31(7), 646–653 (1982)

25. Palmer, J.D., Adams, K.L., Cho, Y., Parkinson, C.L., Qiu, Y.L., Song, K.: Dynamic evolution of plant mitochondrial genomes: mobile genes and introns and highly variable mutation rates. Proc. Natl. Acad. Sci. U S A 97(13), 6960–6966 (2000)
26. Rice, P., Longden, I., Bleasby, A.: EMBOSS: the European Molecular Biology Open Software Suite. Trends in Genetics 16(6), 276–277 (2000)
27. Rombo, S.E., Palopoli, L.: Pattern discovery in biosequences: From simple to complex patterns. In: Masseglia, F., Poncelet, P., Teisseire, M. (eds.) Data Mining Patterns: New Methods and Applications. IGI Global (2007)
28. Saiardi, A., Erdjument-Bromage, H., Snowman, A.M., Tempst, P., Snyder, S.H.: Synthesis of diphosphoinositol pentakisphosphate by a newly identified family of higher inositol polyphosphate kinases. Curr. Biol. 9(22), 1323–1326 (1999)
29. Saiardi, A., Nagata, E., Luo, H.R., Sawa, A., Luo, X., Snowman, A.M., Snyder, S.H.: Mammalian inositol polyphosphate multikinase synthesizes inositol 1,4,5-trisphosphate and an inositol pyrophosphate. Proc. Natl. Acad. Sci. U S A 98(5), 2306–2311 (2001)
30. Saiardi, A., Resnick, A.C., Snowman, A.M.: Inositol pyrophosphates regulate cell death and telomere length through phosphoinositide 3-kinase-related protein kinases. Proc. Natl. Acad. Sci. U S A 102, 1911–1914 (2005)
31. Saiardi, A., Sciambi, C., McCaffery, J.M.: Inositol pyrophosphates regulate endocytic trafficking. Proc. Natl. Acad. Sci. U S A 99, 14206–14211 (2002)
32. Shears, S.B.: How versatile are inositol phosphate kinases? Biochem. J. 377, 265–280 (2004)
33. Stein, J.P., Catterall, J.F., Kisto, P., Means, A.R., O'Malley, B.W.: Ovomucoid intervening sequences specify functional domains and generate protein polymorphism. Cell 21, 681–687 (1980)
34. Stevenson-Paulik, J., Odom, A., York, J.: Molecular and biochemical characterization of two plant inositol polyposphate 6-/3-5- kinases. J. Biol. Chem. 277, 42711–42718 (2002)
35. Sweetman, D., Johnson, S., Caddick, S.E., Hanke, D.E., Brearley, C.A.: Characteryzation of an arabidopsis inositol 1,3,4,5,6-pentakisphosphate 2-kinase (atipk1). Biochem. J. 394, 95–103 (2006)
36. Takenaka, M., Verbitskiya, D., van der Merwea, J.A., Zehrmanna, A., Brennickea, A.: The process of rna editing in plant mitochondria. Mitochondrion 8, 35–46 (2008)
37. Togashi, S., Takazawa, K., Endo, T., Erneux, C., Onaya, T.: Structural identification of the myo-inositol 1,4,5-trisphosphate-binding domain in rat brain inositol 1,4,5-trisphosphate 3-kinase. Biochem. J. 326, 221–225 (1997)
38. Voglmaier, S.M., Bembenek, M.E., Kaplin, A.I., Dorman, G., Olszewski, J.D., Prestwich, G.D., Snyder, S.H.: Purified inositol hexakisphosphate kinase is an atp synthase: diphosphoinositol pentakisphosphate as a high-energy phosphate donor. Proc. Natl. Acad. Sci. U S A 15, 4305–4310 (1996)
39. Wang, J., Shapiro, B., Shasha, D.: Pattern DiscoVery in Biomolecular Data: Tools, Techniques and Applications. Oxford University Press, NY (1999)
40. Xia, H.J., Brearley, C., Elge, S., Kaplan, B., Fromm, H., Mueller-Roeber, B.: Arabidopsis inositol polyphosphate 6-/3-kinase is a nuclear protein that complements a yeast mutant lacking a functional argr-mcm1 transcription complex. Plant Cell 15, 449–463 (2003)

Identification and Expression of Early Nodulin in Sugarcane Transcriptome Revealed by in Silico Analysis

Gabriela Souto Vieira-de-Mello, Petra Barros dos Santos,
Nina da Mota Soares-Cavalcanti, and Ana Maria Benko-Iseppon

Universidade Federal de Pernambuco, Center of Biological Sciences,
Department of Genetics, Laboratory of Plant Genetics and Biotechnology,
R. Prof. Moraes Rêgo 1235, CEP 50670-420, Recife, PE, Brazil
ana.benko.iseppon@pq.cnpq

Abstract. Nodulin genes have been defined as plant genes that are induced during nodule formation in legumes. Many studies, however, revealed a number of nodulins in non-legumes, including monocot plants, suggesting that these genes play additional roles besides nodulation. The presence and expression profile of the early nodulin genes (*Annexin*, *DMI3*, *NIN*, *NORK*, *CCS52A*, and *ENOD8*) was evaluated in the sugarcane transcriptome (237,954 ESTs) using in silico procedures. 129 sugarcane clusters were identified (out of 1,476 transcripts) and their expression profile was evaluated. Higher expression was observed in libraries of flowers, roots and normalized mix of tissues, confirming their multi-function character besides the plant-bacteria endophytic interaction in sugarcane. The multiple alignments revealed high homology among sugarcane sequences and respective proteins from other plants, mainly monocots, revealing a relatively conserved genetic structure among species, probably regarding ancient genetic processes.

Keywords: early nodulins, nitrogen fixation, expression pattern.

1 Introduction

Sugarcane is one of the most important sources of sugar and alcohol in the world and is cultivated in tropical and subtropical areas in more than 80 countries around the globe. Several sugarcane varieties have the ability to grow with low nitrogen fertilizer inputs, being selected for high yields with low inputs of inorganic nitrogen fertilizer [1]. This important crop establishes association with endophytic diazotrophic bacteria, including *Gluconacetobacter diazotrophicus*, *Herbaspirillum seropedicae* and *H. rubrisubalbicans*, showing unique features when compared with other nitrogen-fixing associations. Bacteria colonizes the intercellular spaces and vascular tissues of most organs of the infected symbiont promoting plant growth, without causing visible disease symptoms [1] [2]. It is still unclear which mechanisms are involved in the establishment of this particular type of interaction and what kind of molecules mediate signaling between plant and bacteria [1].

R. Rizzo and P.J.G. Lisboa (Eds.): CIBB 2010, LNBI 6685, pp. 72–85, 2011.
© Springer-Verlag Berlin Heidelberg 2011

Nodulins have been defined as plant genes that are exclusively induced during nodule formation in legume plants. Many studies, however, revealed a number of nodulin-related sequences in non-legumes, as *ENOD40* [3]. Moreover some non leguminous plants, including rice, have the ability to perceive lipochitooligosaccharide nodulation signal molecules (nod factors) produced by the rhizobia, suggesting that nodulation related processes are present in non legumes [4]. Recent evaluations indicated that the molecular communication between sugarcane and the microbes might involve lipopolysaccharides present in the outer membrane of these gram-negative bacteria [2]. In addition, some sugarcane genes involved in plant-bacteria signalization during the association and nitrogen metabolism are probably activated by the endophytic bacteria in the early steps of plant colonization, allowing the plant to assimilate the nitrogen fixed by the bacteria [5]. These genes also seem to act as nodule activators, once they present homology with some legume nodulins [1].

No previous evaluation of early nodulin genes was carried out in the sugarcane transcriptome, despite of the potential application of these genes in increasing the efficiency of the association between this plant and diazotrophic bacteria. In this context the present work aimed to perform an in silico identification and characterization of early nodulins in the sugarcane transcriptome, using known legume nodulin sequences as templates, evaluating also the expression profiles of nodulin-related sequences in this organism.

2 Material and Methods

For the annotation routine (Fig. 1) known full length cDNA sequences of early nodulin genes (*Annexin*, *DMI3*, *NIN*, *NORK*, *CCS52A* and *ENOD8*) from legumes (Tab. 1) were used as seed sequences against the SUCEST database with aid of a local tBLASTn search tool. Only sequences with e-value up to e^{-10} or less were used for a homology screening in Genbank using the BLASTx tool [6]. The cluster frame of the tBLASTn alignment was used to predict the Open Reading Frames (ORFs) for each selected cluster. Sugarcane clusters were translated using the ORF-finder tool at NCBI and screened for conserved motifs with aid of the RPS-BLAST CD-search tool.

The prevalence of sugarcane early nodulins was based on the number of reads that composed each cluster, followed by data normalization and calculation of the relative frequency (reads per library). A hierarchical clustering approach was applied using normalized data, from all identified nodulins transcripts, allowing the generation of a graphic representation with aid of the CLUSTER program [7]. The resulting dendrograms including both axes (using the weighted pair group for each gene class and library) were generated using the TreeView program for Windows [8]. Multiple alignments (CLUSTALx program) were generated using sugarcane complete sequences together with sequences from other organisms, searched NCBI. The phylogenetic analysis was performed using the MEGA program (Version 4 for Windows [9]) using maximum parsimony method, with bootstrap of 2,000 replications and pairwise deletion for the treatment of

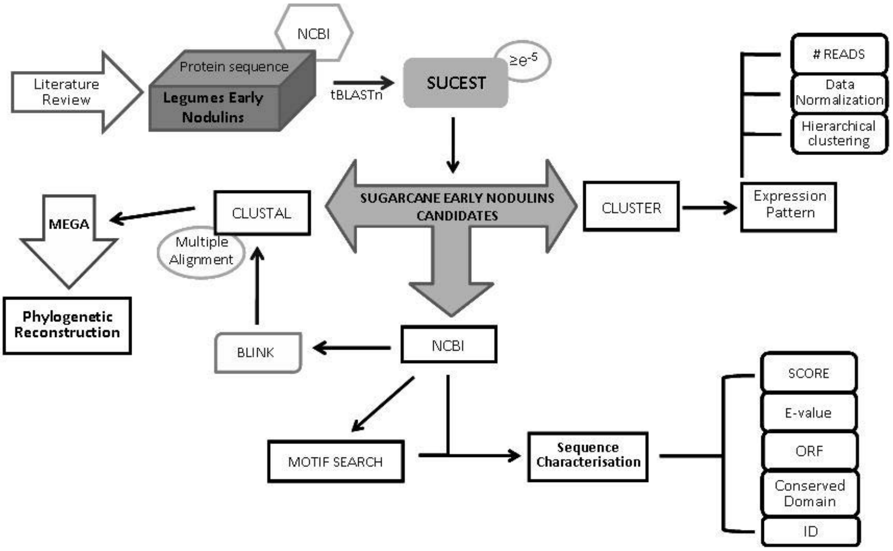


Fig. 1. Schematic representation of the routine application. Annotation was accomplished using seed sequences of *NIN*, *ENOD8*, *NORK*, *CCS52A*, *ENOD40*, *DMI3* and *Annexin* from legumes, followed by identification of sugarcane candidates, confirmation of identity, sequence characterization, differential expression profiling and phylogenetic analyses.

GAPs during the alignments, generating a consensus tree with a cut-off of 50 (50% more parsimonious trees).

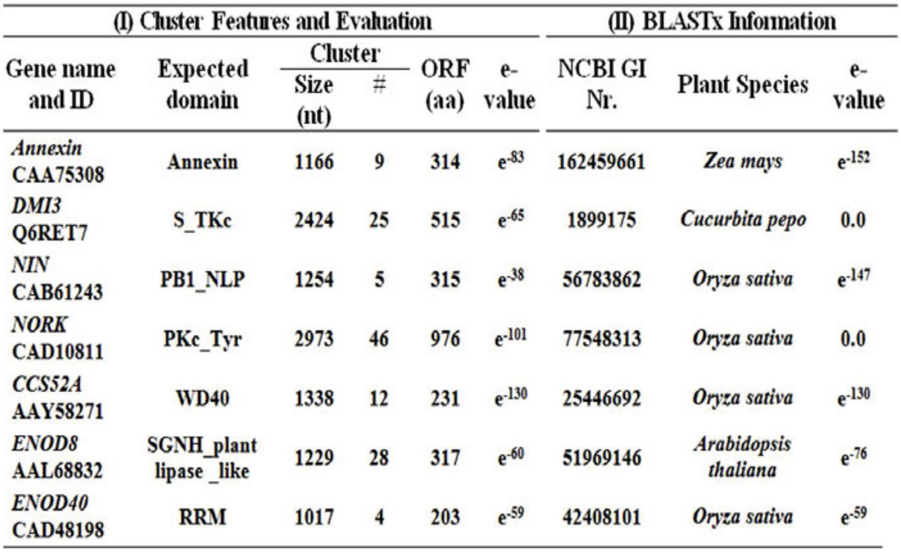
3 Results

3.1 Sugarcane Orthologs

129 candidate clusters (from 1,476 reads) could be identified in the SUCEST database, with e-values ranging from 0.0 to e^{-10} (Tab. 1). **Annexin**: a high degree of similarity was found (e-value up to 4e^{-83}). All nine clusters presented best matches with their respective proteins after BLASTx (seven with monocots - *Z. mays* and *O. sativa* - and two with dicots - *A. thaliana* and *Cicer arietinum*). The searched annexin domain was found in two clusters, being structurally conserved. **DMI3**: 25 sugarcane clusters were identified (e-values from 8e^{-65} to e^{-10}) of which seven presented the complete S_TKc domain. After reverse alignments 19 sugarcane sequences exhibited best similarity with monocots, including *Z. mays* (five) and *O. sativa* (14 alignments), and six were similar to dicots (Cucurbitaceae, Rosaceae, Fabaceae and Brassicaceae). **CCS52A**: 12 clusters were identified (2e^{-130} to 3e^{-11}). After reverse alignment 83.3% presented similarity with *O. sativa* while 16.7% were similar to *Lotus japonicus* and *A. thaliana*. The WD40 conserved domain was found in three *Saccharum officinarum* orthologs.

NIN: search revealed five clusters with high degree of similarity; two with the searched RWP-RK domain complete, one incomplete and two with no domain. After BLASTx the clusters showed similarity with *O. sativa* and *L. japonicus* (e-values from 7e^{-147} to 8e^{-23}). **NORK**: 46 representatives were found. After BLASTx 93.5% showed similarity with monocots (mainly *O. sativa*) while 6.5% were similar to *A. thaliana*. Regarding the integrity of the PKc_Tyr conserved domains, in 28 and 18 clusters they were complete and incomplete, respectively. **ENOD8**: the 28 sequences obtained using BLASTn showed high degree of conservation with *O. sativa* proteins after BLASTx, with more than 75% of the selected clusters similar to this monocot. The procured SGNH_plant_lipase_like domain was detected in 12 clusters. **ENOD40**: four clusters were selected in the SUCEST database, three showing high similarity with a respective protein from *O. sativa*. The procured RRM domain was found complete and incomplete in three and one clusters, respectively.

Table 1. Main sugarcane clusters similar to nodulins genes. tBLASTn results and sequence evaluation of sugarcane nodulins genes including the best match of each gene: (I) Features and evaluation results with sugarcane cluster size in nucleotides (nt), ORF (Open Reading Frame) size in amino-acids (aa), e-value; numbers (#) of matched clusters. (II) Data about BLASTx best alignment: NCBI GI number and plant species.



(I) Cluster Features and Evaluation						(II) BLASTx Information		
Gene name and ID	Expected domain	Cluster		ORF (aa)	e-value	NCBI GI Nr.	Plant Species	e-value
		Size (nt)	#					
Annexin CAA75308	Annexin	1166	9	314	e^{-83}	162459661	*Zea mays*	e^{-152}
DMI3 Q6RET7	S_TKc	2424	25	515	e^{-65}	1899175	*Cucurbita pepo*	0.0
NIN CAB61243	PB1_NLP	1254	5	315	e^{-38}	56783862	*Oryza sativa*	e^{-147}
NORK CAD10811	PKc_Tyr	2973	46	976	e^{-101}	77548313	*Oryza sativa*	0.0
CCS52A AAY58271	WD40	1338	12	231	e^{-130}	25446692	*Oryza sativa*	e^{-130}
ENOD8 AAL68832	SGNH_plant lipase _like	1229	28	317	e^{-60}	51969146	*Arabidopsis thaliana*	e^{-76}
ENOD40 CAD48198	RRM	1017	4	203	e^{-59}	42408101	*Oryza sativa*	e^{-59}

3.2 Expression Pattern

Regarding the prevalence of early nodulins (Fig. 2A), it was clear that NORK reads were most abundant, with 612 reads (representing 42% of nodulins transcripts), followed by DMI3 with 320 reads (22% of the total number). The lowest

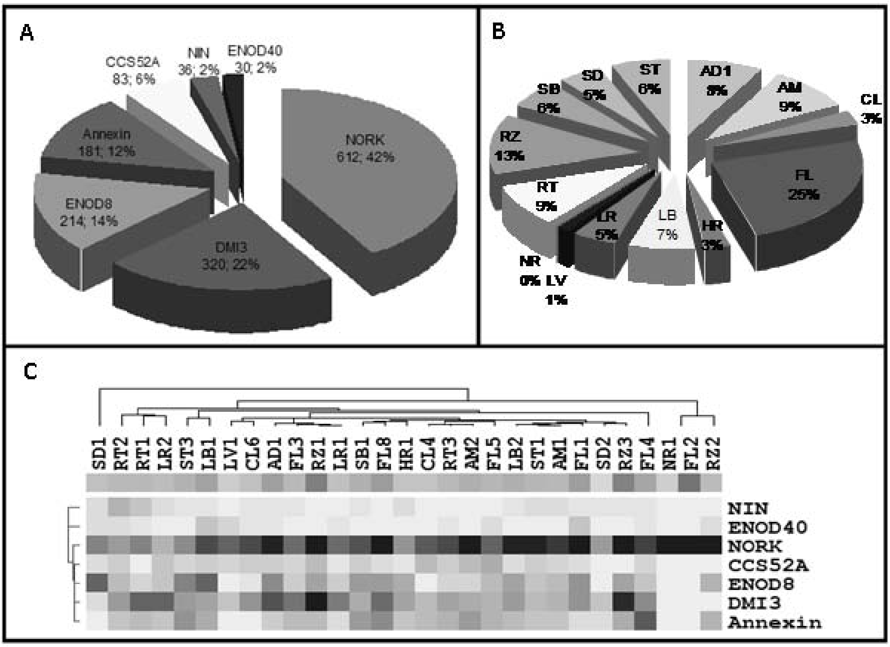


Fig. 2. (A) Prevalence of early nodulin transcripts in the SUCEST database. Numbers above refer to the absolute number of reads found and the numbers below refer to the percentage of reads that compose each nodulin class. (B) Occurrence of sugarcane early nodulins in the SUCEST libraries. Numbers refer to the percentage of reads in each library. (C) Heat map representing expression pattern of sugarcane early nodulin genes. White means no expression and black means all levels of expression. Library codes: AD/AD1: tissues infected by *Gluconacetobacter diazotroficans*, AM: Apical meristems (AM1/AM2); CL: Calli treated for 12 h at 4 to 37^{o}C in the dark or light (CL3/CL4/CL6); FL: Flowers at different developmental stages (FL1/FL3/FL4/FL5/FL8); HR: tissues infected with *Herbaspirillum rubrisubalbicans*; LB: Lateral buds from mature plants (LB12); LR: Leaf Roll from immature plants (LR1/, LR2); LV/LV1: Etiolated leaves from plantlets grown in vitro; NR: All normalized tissues; RT: Roots (RT1/,RT2) and root apex (RT3); RZ Root to shoot transition (RZ1/RZ2/RZ3); SB/SB1: Stalk bark; SD: Seeds (SD1/SD2); ST: Stem (ST1/ST3).

number was observed for NIN and ENOD40, representing 2% each. Considering the distribution of the 1,476 nodulin transcripts in the 14 analyzed libraries, in general a higher prevalence could be observed in flower (FL= 25%) and stem-root transition (RZ=13%; Fig. 2B) tissues.

It is interesting that all 29 analyzed libraries from SUCEST database comprised at least one read. The heat map (Fig. 2C) revealed a higher expression in flower (FL2) and in Stem-Root transition (RZ1/RZ3).

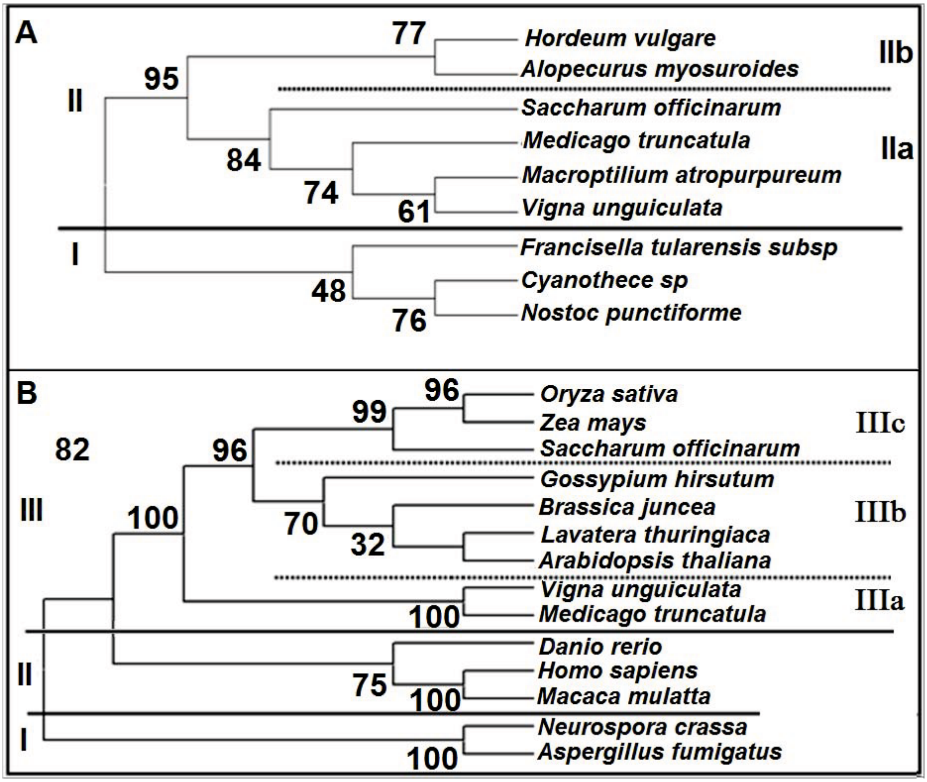


Fig. 3. Dendrograms generated after maximum parsimony analysis illustrating relationships revealed by conserved domains similarity among (A) ENOD8 and (B) Annexin sequences and putative sugarcane orthologs. The numbers in the base of clades regard to bootstrap values (2,000 replications).

3.3 Dendrograms

The multiple alignments generated using Annexin and ENOD8 sequences showed a high degree of conservation among the nodulin orthologs from diverse organisms. In the resulted dendrograms (Fig. 3A-B) it was possible to observe the placement of different organisms (fungi, protists, plants and animals) in separated clades according their kingdom classification. For ENOD8 they were distributed into two groups; the first one (outgroup) comprising the protozoans (I) and the other including plants species (II). The group II showed two subclades, grouping dicots and sugarcane (sister group IIa, bootstrap 84) and monocots (sister group IIb) in different branches. In the annexin dendrogram (Fig. 3B) fungi figured as an outgroup (branch I), while animals and plants were positioned in two clades (II and III, respectively) according to their higher taxonomic classification. In group III the Fabaceae family (IIIa) was segregated from the remaining dicots (IIIb), which were placed together with the monocots (IIIc).

4 Discussion

4.1 Sugarcane Orthologs

Annexins comprise a multigene and multifunctional family of amphipathic proteins presenting a broad taxonomic distribution covering prokaryotes, fungi, protists, plants and higher vertebrates. Regarding Magnoliophyta this proteins are conserved in both dicots and monocots [10]. Concerning their functions, legume annexins are upregulated by Nod factors and play a role in nodulation signaling [11]. Besides the role in the symbioses, annexins from non-legumes are associated with different cellular processes. For example in maize, annexins are considered to be multifunctional proteins capable of peroxidase activity, elevation of cytosolic calcium and direct formation of a passive Ca^{2+}- and K^{+}-permeable conductance [12]. As expected, the sugarcane Annexin orthologs showed higher sequence conservation with Poaceae organisms following the taxonomic proximity, as expected. In addition two sequences presented the conserved domain complete, indicating the existence and conservation of Annexin genes in sugarcane.

Another early nodulin, DMI3, is a plant-specific protein that belongs to the CCaMK group of serine-threonine protein kinases present from the moss *Physcomitrella patens* to higher plants, including dicots and monocots [13]. Many plant-specific *DMI3* orthologs were found in sugarcane transcriptome bearing high similarity with known genes. Most sugarcane CCaMKs presented high similarity with rice sequences. Regarding this resemblance, it is suggested that legume *DMI3* also beard high similarity to rice and lily. Little is known about the biological role of CCaMKs in plants, and it is suggested that a CCaMK is required by mycorrhized plants to interpret a complex calcium signature elicited in response to fungus signals [14]. This could be also the case of sugarcane that besides the interaction with endophytic bacteria is able to establish mycorrhizal associations [5].

Regarding CCS52A, our findings confirm that this gene is present in sugarcane, since we found clusters with complete domains and best hits with high degree of similarity with rice, a organism with well characterized CCS52A [15]. These findings can be supported by Foucher and Kondorosi [16] that proposed that CCS52A is an ubiquitous regulator of cell cycle transition to differentiation in non legume plants cells. Moreover, orthologs of this protein have been also found in various other plant species like medicago, arabidopsis, tomato, wheat and rice, indicating a strong conservation of the CCS52A proteins in the plant kingdom [17].

With respect to the *NIN* family, the results indicated the presence of at least five isoforms with high similarity with rice NIN-like proteins (NLPs), what can reinforce the findings of Riechman *et al.* [18], which theorized that there are no close relatives to the legume NIN proteins in rice or arabidopsis. Instead, these non-legumes presented NLPs regarding the closest relatives of legume NINs. In addition, the NLPs are multidomain proteins with a high degree of conservation; the phylogenetic tree inferred from the NLP alignment suggested that there are at least three variants of this gene in the common ancestor of mono- and eudicots [19].

Sugarcane's most abundant nodulin regarded the NORK gene class, with 46 clusters. The extracellular domain of NORK protein presented three LRR (Leucine Rich Repeat) domains, which are required for perception of a liposaccharide nodulation signal in legumes. Proteins that possess similarity to the unique NORK extracellular domain are found in monocots and dicots, suggesting that this region may have a biological role that is not limited to nodulation [20]. The RLKs (Receptor Like Kinases) comprise the largest gene family of receptors in plants, with more than 600 homologs in arabidopsis and 1,100 in rice [21]. In both organisms these RLKs might have roles in plant development and in signal transduction during interactions with endophytic organisms and pathogens [22]. In addition Vinagre *et al.* [23] identified in sugarcane a LRR-RLK whose expression is regulated in response to interactions with beneficial bacteria. Together, these facts confirm our findings in sugarcane transcriptome and explain the high number of clusters found.

ENOD8 is a member of the GDSL family of lipolytic enzymes present in plant and bacteria that have the putative active serine site, which is not perfectly conserved in all members of the GDSL gene family [24]. In plants, GDSL lipase candidates of species like arabidopsis, *Rauvolfia serpentina*, *Medicago sativa*, *Hevea brasiliensis* and *Alopecurus myosuroides* have been isolated, cloned and characterized, revealing that they are conserved among these species [25]. Studying infected libraries of SUCEST, Nogueira *et al.* [1] found that sugarcane ENOD8 is similar to myrosinase-associated protein (MyAP) related with the plant defense responses. In our differential expression results, sequences of ENOD8 protein were also found in non-infected tissues, suggesting that this protein plays a role in other functions besides the interaction with endophytic organisms; however, in monocots these functions remain unknown. In addition, the similarity with rice and arabidopsis sequences found in our alignments can be explained by the fact that few sequences from other non-legumes are available in NCBI database.

The occurrence of ENOD40 sequences in monocots and different clades within the core eudicots is indicative that ENOD40 is an ancient gene that has been maintained in these plants after divergent evolution [26]. This gene was also functionally characterized in *Z. mays* [27]; in addition, previous studies have identified isoforms in the sugarcane genome, using southern analysis [4], a fact confirmed by our findings. Additionally, the low number of clusters found can be explained by evidences presented by Compaan *et al.* [27] that suggested that this gene category is expressed in low levels in most non-legume plants. In legumes the ENOD40 is a critical gene responsible for cortical cell divisions leading to the initiation of nodule development in rhizobial association [28]; playing a role in the interaction with arbuscular mycorrhiza in the fungal growth in the root cortex, increasing the frequency of arbuscule formation [29]. ENOD40 genes present regions that are highly conserved among distantly related plant species [27]. In accordance to this fact ENOD40 from *O. sativa* encodes peptides that are homologous to proteins encoded by the corresponding genes in legumes, even thought their expression is not associated with symbiotic interactions [4].

4.2 Expression Pattern

In legumes, transcripts of the ENOD-like genes were identified in roots, stems and flowers, suggesting that these genes might have roles in the development of different organs involved principally in the regulation of plant development, morphogenesis, secondary metabolites synthesis and defense responses [30]. In addition, functional evaluations showed that many of these genes are in fact expressed in nonsymbiotic tissues and/or during nonsymbiotic conditions also presenting a number of homologs in non-legume plants, as arabidopsis and rice that are unable to form nodules [31].

Thus, it is hypothesized that nodulin genes have arisen as a result of the recruitment of pre-existing non-symbiotic genes which might have roles in other physiological processes, common to all plants, like controlling growth and development [32]. In fact, the presence of nodulin transcripts in non-infected tissues in the SUCEST libraries confirms this hypothesis. In sugarcane several genes possibly involved in nitrogen metabolism and plant-bacteria signaling during endophytic diazotrophic associations seem to act as nodule-enhanced genes [1]. Regarding the expression pattern of early nodulins, the observed majority of nodulin transcripts was found in flower libraries, an expected result, since this is the largest SUCEST library, as compared with other tissues. Regarding NORK results, a significant expression could be detected in most SUCEST libraries; what is in agreement with the fact that genes encoding RLKs isoforms, besides their roles in organism interactions, are very closely related to plant developmental processes, being present in tissues under growth and differentiation, like seeds, plantlets in different stages of development, in flowers, leaves and root-to-shoot transition regions, confirming the crucial importance of these proteins for plants [22].

Our results have shown that the sugarcane ENOD40 gene presented a similar expression pattern as previously found in rice, where the expression in the developing vascular bundles of the stem prevailed [33]. In legumes the expression of ENOD40 is induced within hours of *Rhizobium* inoculation and it appears to be critical for proper nodule development; however, transcripts are also localized in the stem, lateral roots and other tissues in these plants [28].

The occurrence of annexin transcripts in almost all SUCEST libraries occurred in accordance to Proust *et al.* [34] that using northern-blotting analysis revealed that annexins from plants have a fairly widespread expression. Concerning monocot annexins, Smallwood *et al.* [35] showed that the transcripts were found in root tissues, stem and young expanding leaves of barley, while Carroll *et al.* (1998) [38] reported that the maize annexin was expressed in root cap cells and differentiating vascular tissues in roots [36], both similar to the annexin expression in sugarcane found in our analysis.

Besides the nodulins described above, many early nodulins presented an expression related to organ differentiation in monocots, like DMI3, ENOD8, and CCS52A [13] [16] [37]. Based on the distribution and prevalence of these early nodulins in sugarcane transcriptome, we suggest that these genes also play a role in organ development, at least, in this monocot. In a general view, the fact that different nodulins are expressed in most SUCEST libraries support the assumption that these genes are expressed not only in plant-microbe interactions, revealing their importance for all angiosperms.

4.3 Dendrograms

The multiple alignments with sugarcane orthologs and nodulin from other species showed a relative degree of conservation among sequences as expected, confirming that a significant proportion of nodule-specific functions are performed by recruiting preexisting genes common to non-legume plants. Additionally it is now known that many of the nodulation genes have been acquired following duplication of those with related functions [38].

In ENOD8 dendrogram (Fig. 1A) a clear segregation of the bacteria (clade I) and plant (clade II) in monophyletic groups was evident. Regarding the plant kingdom, the separation of the dicots (IIa) and monocots (IIb), except for the sugarcane, was expected, since numerous ENOD8 sequences were found in plants that are not able to fix nitrogen, being all probably induced by exogenous signals or regulated in a tissue specific manner. Regarding the group IIa, exclusive synapomorphies characterized the sugarcane and the legumes (IIb) clade, as expected, since both organisms establish symbiotic relationships with microorganisms. In addition, the legumes were grouped together, as expected, since these proteins are strongly involved in nitrogen fixation processes, being associated with the symbiosome membrane in root nodules [39].

The generated annexin dendrogram reflected the evolutionary history of the plants. Both legumes sequences (*V. unguiculata* and *M. truncatula*) stayed as separated subclades within the dicots clade, confirming the hypothesis of Doyle and Luckow [40] that nodulation specific genes arose within the legume family already among the earliest lineages. In a general view, members of the annexin family are composed by a variable N-terminal region and a highly conserved C-terminal core [41]. However, plant annexins share common biological activities and functions with their animal counterparts, such as the ability to stimulate Ca^{2+}-dependent exocytosis [36]. The here obtained annexin dendrogram (Fig. 1B) is in accordance to this divergent evolution, showing animal annexins (branch II) as a monophyletic group and also as a sister-group of plants, probably sharing synarcheomorphic characters.

According to Moss [42] plant annexins make up a monophyletic cluster whose members generally lack amino-terminal domains and functional calcium-binding sites in their second and third repeats. As seen in the present results, the non-legume families (Brassicaceae and Malvaceae) formed a paraphyletic merophyletic group. In addition, the presence of specific features in annexins from monocots and dicots could be seen in the aligments, which resulted in the separation of these classes in smaller clades. In reports regarding non legume plants annexins have been associated with different cellular processes. Annexins purified from plant species such as maize, cotton and celery presented different characteristics [43]; for example, a cotton annexin was associated with the modulation of callose synthase activity located in plasma membrane [44], while maize annexins are capable of peroxidase activity, elevation of cytosolic calcium and direct formation of a passive Ca^{2+}- and K^{+}-permeable conductance [12]. These functional differences could justify the diverging positions among legume and non legume annexins, also here observed.

5 Concluding Remarks

With aid of bioinformatic tools it was possible to identify all seven early nodulin gene categories out of 195 sugarcane contigs and 1,746 transcripts, allowing also inferences regarding their expression pattern. All nodulin candidates bearing the respective conserved domains could be identified in sugarcane, most of them putatively involved in tissue development and growth, besides plant-host interactions. Considering the low amount of previously described nodulins in monocots, the identified sequences represent valuable resources for structural and functional evaluations including expression assays and may lead to significant benefits for sugarcane production.

Acknowledgments. We thank CNPq (Conselho Nacional de Desenvolvimento Cientifico e Tecnologico, Brazil), CAPES (Coordenacão de Aperfeicoamento de Pessoal de Nivel Superior, Brazil) and FACEPE (Fundacão de Amparo à Pesquisa do Estado de Pernambuco, Brazil) for the concession of fellowships.

References

1. Nogueira, E.M., Vinagre, F., Masuda, H.P., Vargas, C., Padua, V.L.M., Silva, F.R., Santos, R.V., Baldani, J.I., Ferreira, P.C.G., Hemerly, A.S.: Expression of Sugarcane Genes Induced by Inoculation With Gluconacetobacter diazotrophicus and Herbaspirillum rubrisubalbicans. Gen. and Mol. Biol. 24, 199–206 (2001)
2. Serrato, R.V., Sassaki, G.L., Cruz, L.M., Carlson, R.W., Muszynski, A., Monteiro, R.A., Pedrosa, F.O., Souza, E.M., Iacomini, M.: Chemical Composition of Lipopolysaccharides Isolated From Various Endophytic Nitrogen-fixing Bacteria of the Genus Herbaspirillum. Can. J. Microbiol. 56, 342–347 (2010)
3. Kouchi, H., Takane, K., So, R.B., Ladha, J.K., Reddy, P.M.: Rice ENOD40: Isolation and Expression Analysis in Rice and Transgenic Soybean Root Nodules. Plant J. 18, 121–129 (1999)
4. Reddy, P.M., Aggarwal, R.K., Ramos, M.C., Ladha, J.K., Brar, D.S., Kouchi, K.: Widespread Occurrence of the Homologs of the Early Nodulin (ENOD) Genes in Oryza Species, Related Grasses. Bioch. and Bioph. Res. Com. 258, 148–154 (1999)
5. Vargas, C., Padua, V.L.M., Nogueira, E.M., Vinagre, F., Masuda, H.P., Da Silva, L.: Signaling Pathways Mediating the Association Between Sugarcane and Endophytic Diazotrophic Bacteria: A Genomic Approach. Symb. 35, 159–180 (2003)

6. Altschul, S.F., Madden, T.L., Schaffer, A.A., Zhang, J., Zhang, Z., Miller, M., Lipman, D.J.: Gapped BLAST and PSI-BLAST: A New Generation of Protein Database Search Programs. Nucl. Acid Res. 25, 3389–3402 (1997)
7. Eisen, M.B., Spellman, P.T., Brown, P.O., Botstein, D.: Cluster Analysis and Display of Genomic-wide Expression Pattern. PNAS 95, 14863–14868 (1998)
8. Page, R.D.: Treeview Program Version 161. Com. App. Biosc. 12, 357–358 (1996)
9. Tamura, K., Dudley, J., Nei, M., Kumar, S.: Mega4: Molecular Evolutionary Genetics Analysis (MEGA) Software Version 4.0. Mol. Biol. and Evol. 24, 1596–1599 (2004)
10. Morgan, R.O., Martin-Almedina, S., Iglesias, J.M., Gonzalez-Florez, M.I., Fernandez, M.P.: Evolutionary Perspective on Annexin Calcium-binding Domains. Bioch. Bioph. Acta 1742, 133–140 (2004)
11. Niebel, F.C., Lescure, N., Cullimore, J.V., Gamas, P.: The Medicago truncatula MtAnn1 Gene Encoding an Annexin is Induced by Nod Factors and During the Symbiotic Interaction With Rhizobium meliloti. Mol. Plant-Microbe Interac. 1, 504–513 (1998)
12. Laohavisit, A., Mortimer, J.C., Demidchik, V., Coxon, K.M., Stancombe, A.M., Macpherson, N., Brownlee, C., Hofmann, A., Webb, A.A.R., Miedema, H., Battey, N.H., Davies, J.M.: Zea mays Annexins Modulate Cytosolic Free Ca^{2+} and Generate a Ca^{2+}-Permeable Conductance. The Plant Cell 21, 479–493 (2009)
13. Messinese, E., Mun, J.-H., Yeun, L.H., Jayaraman, D., Rouge, P., Barre, A., Lougnon, G., Schornack, S., Bono, J.-J., Cook, D.R., Ane, J.-M.: A Novel Nuclear Protein Interacts With the Symbiotic DMI3 Calcium- and Calmodulin-dependent Protein Kinase of Medicago truncatula. Mol. Plant-Microbe Interac. 20, 912–921 (2007)
14. Yang, T., Poovaiah, B.W.: Calcium/calmodulin-mediated Signal Network in Plants. Trends in Plant Sci. 8, 505–512 (2003)
15. Cebolla, A., Vinardell, J.M., Kiss, E., Olah, B., Roudier, F., Kondorosi, A., Kondorosi, E.: The Mitotic Inhibitor ccs52 is Required for Endoreduplication and Ploidy-dependent Cell Enlargement in Plants. Europ. Mol. Biol. J. 18, 4476–4484 (1999)
16. Foucher, F., Kondorosi, A.: Cell Cycle Regulation in Course of Nodule Organogenesis in Medicago. Plant Mol. Biol. 43, 773–786 (2000)
17. Gonzalez-Sama, A., de la Peña, T.C., Kevei, Z., Mergaert, P., Lucas, M.M., Felipe, M.R., Kondorosi, E., Pueyo, J.J.: Nuclear DNA Endoreduplication and Expression of the Mitotic Inhibitor CCS52 Associated to Determinate and Lupinoid Nodule Organogenesis. Mol. Plant-Microbe Interac. 19, 173–180 (2006)
18. Riechman, J.L., Heard, J., Martin, G., Reuber, L., Keddie, C.Z.J., Pineda, L.A.O., Ratcliffe, O.J., Samaha, R.R., Creelman, R., Pilgrim, M., Broun, P., Zhang, J.Z., Ghandehari, D., Sherman, B.K., Yu, G.L.: Arabidopsis Transcription Factors: Genome-wide Comparative Analysis Among Eukaryotes. Sci. 290, 2105–2110 (2000)
19. Schauser, L., Wieloch, W., Stougaard, J.: Evolution of NIN-Like Proteins in Arabidopsis, Rice, and Lotus japonicas. J. Mol. Evol. 60, 229–237 (2005)
20. Endre, G., Kereszt, A., Kevei, Z., Mihacea, S., Kalo, P., Kiss, G.B.: A receptor Kinase Gene Regulating Symbiotic Nodule Development. Nature 417, 962–966 (2002)
21. Shiu, S.H., Karlowski, W.M., Pan, R., Tzeng, Y.H., Mayer, K.F.X., Li, W.-H.: Comparative Analysis of the Receptor-like Kinase Family in Arabidopsis and Rice. The Plant Cell 16, 1220–1234 (2004)

22. Morillo, S.A., Tax, F.E.: Functional Analysis of Receptor-like Kinases in Monocots and Dicots. Cur. Op. in Plant Biol. 9, 460–469 (2006)
23. Vinagre, F., Vargas, C., Schwarcz, K., Cavalcante, J., Nogueira, E.M., Baldani, J.I., Ferreira, P.C.G., Hemerly, A.S.: SHR5: A Novel Plant Receptor Kinase Involved in Plant-N2-fixing Endophytic Bacteria Association. J. Exp. Bot. 57, 559–569 (2006)
24. Gyorgyey, J., Vaubert, D., Jimenez-Zurdo, J.I., Charon, C., Troussard, T., Kondorosi, A., Kondorosi, E.: Analysis of Medicago truncatula Nodule Expressed Tags. Mol. Plant-Microbe Interac. 13, 62–71 (2000)
25. Ruppert, M., Woll, J., Giritch, A., Genady, E., Ma, X., Stockigt, J.: Functional Expression of an Ajmaline Pathway-specific Esterase From Rauvolfia in a Novel Plant-virus Expression System. Planta 222, 888–898 (2005)
26. Ruttink, T.: ENOD40 Affects Phytohormone Cross-talk. PhD Thesis, Wageningen University (2003) ISBN:9058089797
27. Compaan, B., Ruttinka, T., Albrecht, C., Meeley, R.: Identification and Characterization of a Zea mays Line Carrying a Transposon-tagged ENOD40. Bioch. Bioph. Acta 1629, 84–91 (2003)
28. Charon, C., Sousa, C., Crespi, M., Kondorosi, A.: Alteration of enod40 Expression Modifies Medicago truncatula Root Nodule Development Induced by Sinorhizobium meliloti. The Plant Cell 11, 1953–1966 (1999)
29. Sinvany, G., Kapulnik, Y., Wininger, S., Badani, H., Jurkevitch, E.: The Early Nodulin ENOD40 is Induced by, And Also Promotes Arbuscular Mycorrhizal Root Colonization. Physiol. Mol. Plant Pathol. 60, 103–109 (2002)
30. Ling, H., Zhao, J., Zuo, K., Qiu, C., Yao, H., Qin, J., Sun, X., Tang, K.: Isolation and Expression Analysis of a GDSL-like lipase Gene From Brassica napus L. J. Bioch. Mol. Biol. 39, 297–303 (2006)
31. Miyao, A., Iwasaki, Y., Kitano, H., Itoh, J., Maekawa, M., Murata, K., Yatou, O., Nagato, Y., Hirochika, H.: A Large-scale Collection of Phenotypic Data Describing an Insertional Mutant Population to Facilitate Functional Analysis of Rice Genes. Plant Mol. Biol. 63, 625–635 (2007)
32. Andersson, C., Ostergaard Jensen, E., Llewellyn, D., Dennis, E., Peacock, W.J.: A New Hemoglobin Gene From Soybean: a Role for Hemoglobin in all Plants. PNAS 93, 5682–5687 (1996)
33. Kawahara, H., Chonan, N.: Studies on Morphogenesis in Rice Plants. Histological Observation on the Maturing Process of Vascular Bundles in Culm. Japan J. Crop Sci. 37, 399–410 (1968)
34. Proust, J., Houlne, G., Schantz, M.-L., Schantz, R.: Characterization and Gene Expression of an Annexin During Fruit Development in Capsicum annum. FEBS Letters 383, 208–212 (1996)
35. Smallwood, M.F., Gurr, S.J., McPherson, M.J., Roberts, K., Bowles, D.J.: The Pattern of Plant Annexin Gene Expression. Bioch. 281, 501–505 (1992)
36. Carroll, A.D., Moyen, C., Van Kesteren, P., Tooke, F., Battey, N.H., Brownlee, C.: Ca$^{(}$2+), Annexins, and GTP Modulate Exocytosis From Maize Root Cap Protoplasts. The Plant Cell 10, 1267–1276 (1998)
37. Peng, T., Dickstein, R.: Regulation of Plant Nodule-specific Genes Expressed in Alfalfa Nodules Arrested at an Early Stage of Development. Plant Sci. 101, 65–73 (1994)
38. Heckmann, A.B., Lombardo, F., Miwa, H., Perry, J.A., Bunnewell, S., Parniske, M., Wang, T.L., Downie, A.: Lotus japonicas Nodulation Requires two GRAS Domain Regulators, One of Which is Functionally Conserved in a Non-legume. Plant Physiol. 142, 1739–1750 (2006)

39. Catalano, C.M., Lane, W.S., Sherrier, D.J.: Biochemical Characterization of Symbiosome Membrane Proteins From Medicago truncatula Root Nodules. Electroph. 25, 519–531 (2004)
40. Doyle, J.J., Luckow, M.A.: The Rest of the Iceberg. Legume Diversity in a Phylogenetic Context. Plant Physiol. 131, 900–910 (2003)
41. Morgan, S.O., Fernandez, M.P.: Distinct Annexin Subfamilies in Plants and Protists Diverged Prior to Animal Annexins and From a Common Ancestor. J. Mol. Evol. 44, 178–188 (1997)
42. Moss, S.E.: Annexins. Trends in Cell Biol. 7, 87–89 (1997)
43. Clark, G.B., Roux, S.J.: Annexins of Plant Cells. Plant Physiol. 109, 1133–1139 (1995)
44. Andrawis, A., Solomon, M., Delmer, D.P.: Cotton Fiber Annexins: A Potential Role in the Regulation of Callose Synthase. The Plant J.: for Cell and Mol. Biol. 18, 763–772 (1993)

An Interactive Method of Anatomical Segmentation and Gene Expression Estimation for an Experimental Mouse Brain Slice

Anton Osokin[1], Dmitry Vetrov[1,2], Alexey Lebedev[1], Vladimir Galatenko[1], Dmitry Kropotov[2], and Konstantin Anokhin[3]

[1] Moscow State University, Russia, 119991, Moscow, Vorobyevy gory, 1
anton.osokin@gmail.com, VetrovD@yandex.ru
[2] Dorodnicyn Computing Centre of Russian Academy of Sciences,
Russia, 119333, Moscow, Vavilova str., 40
dmitry.kropotov@gmail.com
[3] P.K. Anokhin Institute of Normal Physiology,
Russia, 103008, Moscow, Bolshaya Nikitskaya, 6, bld. 4
k_anokhin@yahoo.com

Abstract. We consider the problem of statistical analysis of gene expression in a mouse brain during cognitive processes. In particular we focus on the problems of anatomical segmentation of a histological brain slice and estimation of slice's gene expression level. The first problem is solved by interactive registration of an experimental brain slice into 3D brain model constructed using Allen Brain Atlas. The second problem is solved by special image filtering and further smart resolution reduction. We also describe the procedure of non-linear correction of atlas slices which improves the quality of the 3D-model significantly.

Keywords: mouse brain studies, gene expression detection, image segmentation, morphing, Allen Brain Atlas.

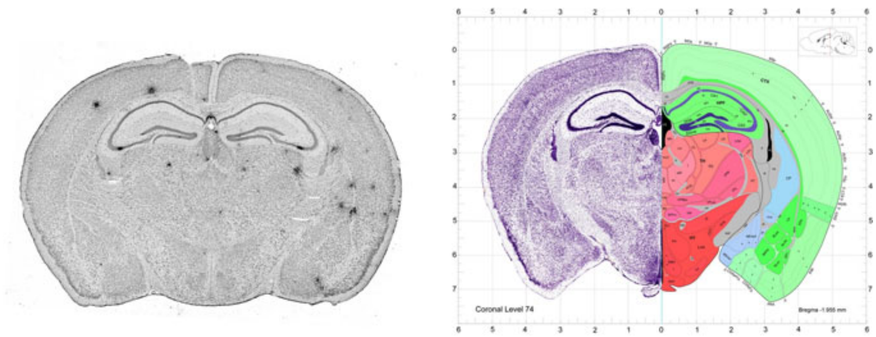
1 Introduction

The analysis of gene expression in a brain is extremely important for cognitive research. Many cognitive functions (e.g. memory consolidation) are regulated by specific genes whose expression starts during some intellectual activity, e.g. training. On the other hand it is known that changes in activity in specific anatomical brain zones reflect the cognitive processes. The combination of anatomical brain map with gene expression patterns and its further statistical processing would allow researchers to discover new genes that are responsible for cognitive processes and new anatomical structures where the functional activity takes place.

Up to the current moment gene expression in animal brains is measured using the following technique. A brain is extracted, frozen and then cut into slices. Each slice is double-stained by Nissl method to highlight histology (see Fig. 1a) and by

R. Rizzo and P.J.G. Lisboa (Eds.): CIBB 2010, LNBI 6685, pp. 86–97, 2011.

© Springer-Verlag Berlin Heidelberg 2011



(a) (b)

Fig. 1. An example of an experimental slice in Nissl stain (a) and an example of brain slice from Allen Brain Atlas with both histological and anatomical views (b)

ISH[1] method which reveals the neurons with expression of corresponding genes (see Fig. 2)[2]. The main problem is to determine brain structures where active genes are located. This problem is difficult even for human experts especially when the slices are obtained using a non-standard section-plane. However there are several atlases for various animals which contain both histological images and the corresponding images where all brain structures are marked by experts [1]. Here we use Allen Mouse Brain Atlas (ABA) [7], which contains a set of 2D coronal mouse brain slices (see Fig. 1b).

In the paper we address the problem of semi-automatic segmentation of a brain slice (given by Nissl image, ISH image, or both) into anatomical structures. For this reason we propose the following procedure. First a 3D-model of a mouse brain is constructed using 2D ABA slices. Hence this model contains both histological and anatomical views. Then for the experimental slice we obtain the most similar slice from the constructed 3D model (hereinafter we denote it as a virtual slice) manually by means of special software BrainTravel. At last we transfer the anatomic segmentation of the virtual slice to the experimental slice using a non-linear transformation. For the 3D model we update our previous result [5] by adding a non-linear correction of atlas slices (section 2.1). This correction improves the quality of the 3D model significantly. The virtual slice search process is described in section 2.2 while anatomical segmentation of an experimental slice is given in section 2.3.

In the paper we also consider the problem of numerical estimation of expression level for ISH slices. This is done by a series of filters and detailed in section 3. Finally the paper finishes with a discussion of further work in the area of statistical analysis of gene expression in the brain during cognitive activity.

[1] In Situ Hybridization.

[2] In practice it is usually difficult to make a double-stain of one slice and hence the neighboring slices are stained by different methods.

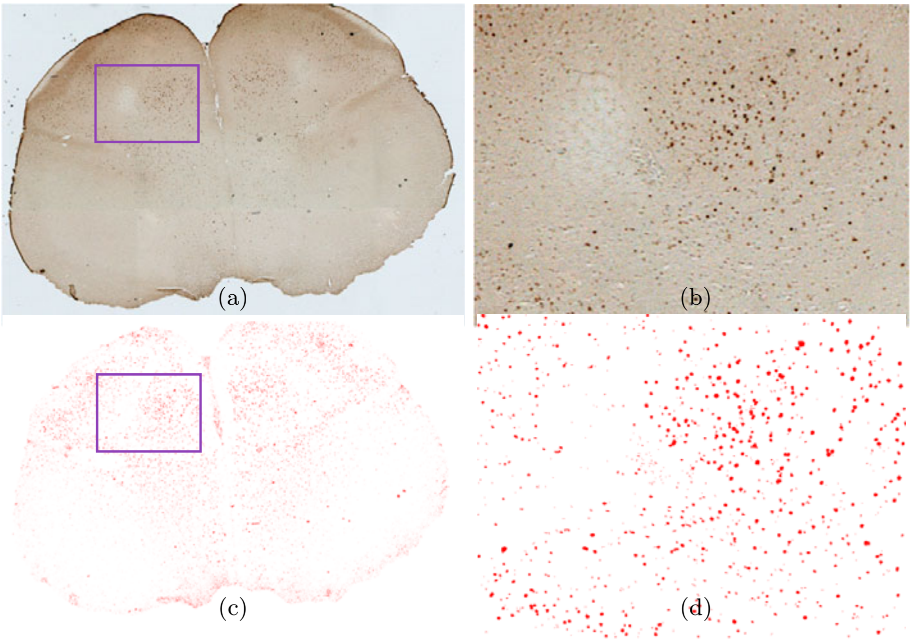


Fig. 2. An example of an experimental slice with gene expression in ISH stain. (a) — initial experimental slice; (b) — enlarged part of slice (a); (c) — processed slice (a) using the method from section 3; (d) — enlarged part of image (c).

2 Anatomical Segmentation

The procedure of anatomical segmentation of an experimental brain slice consists of several steps. First we register a 2D experimental slice into the 3D-model of mouse brain with information about anatomical structures. During the registration we obtain a virtual slice from the 3D-model that is similar to the experimental one. The virtual slice can be seen in both histological and anatomical modes. Afterwords we find a non-linear deformation of the virtual slice by putting pairs of corresponding keypoints on the histological virtual slice and the experimental slice. Finally we apply the non-linear deformation to the anatomical virtual slice.

2.1 Non-linear Correction of Atlas Slices

As a 3D-model of a mouse brain we use the one from [5]. It was constructed from the set of images from Allen Brain Atlas (ABA). Those images contain information about histology (Nissl stain) and anatomy and are available on-line.[3] At first the slices were aligned linearly, afterwords pairwise deformations were found between all the neighboring slices using cubic B-splines. Finally these deformations

[3] http://mouse.brain-map.org/atlas/ARA/Coronal/browser.html

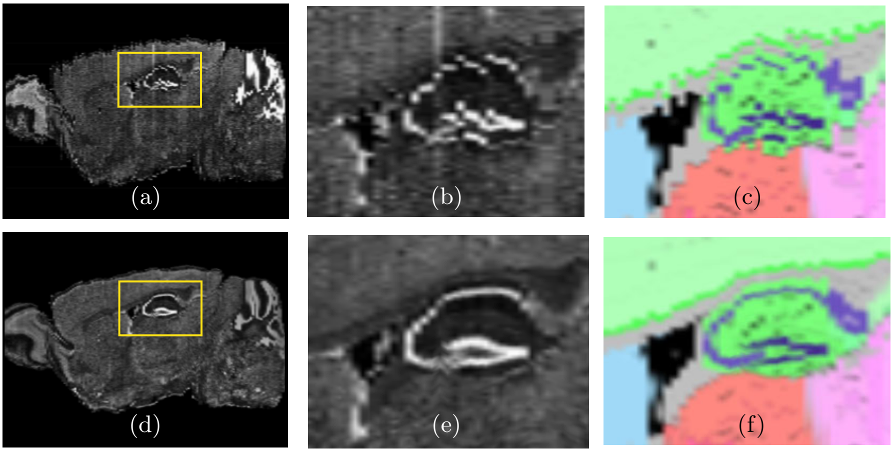


Fig. 3. Comparison of different 3D brain models, constructed from 2D ABA slices. (a)-(c) – standard 3D model (simple concatenation of all slices with morphing); (d)-(f) – 3D model, constructed by method from [5] plus the non-linear correction of atlas slices.

were used to fill in the gaps between the atlas slices using morphing transform. The details of the algorithm are given in [5]. Here we have modified the algorithm by adding one more step – a non-linear correction of atlas slices. This step is necessary because all the slices from the atlas are independently distorted due to the peculiarities of technological process of their cutting and staining. The application of morphing to uncorrected atlas slices leads to non-smooth histological and anatomical structures which are clearly seen in sagittal projection (see fig. 3a-c). To overcome this problem we used the following approach.

Consider the k^{th} uncorrected slice of the atlas $f_k : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$. Our goal is to obtain a corrected slice F_k that is harmonized with its neighbors. First we compute non-linear deformations $g_{k,i}$ between it and its d previous and d following slices so that

$$f_k \circ g_{k,i} \approx f_i, \quad \forall i = k-d, \dots, k+d.$$

The distance between neighboring slices in ABA is 100 μm, and so the slices lying at the distance of several hundred microns may differ significantly and direct deformation between them may appear inadequate. In the paper we follow [9] and establish the recurrent scheme

$$g_{k,i} = \begin{cases} g_{k,i+1} \circ g_{i+1,i}, & k-d \le i < k; \\ g_{k,k}, & i = k; \\ g_{k,i-1} \circ g_{i-1,i}, & k < i \le k+d. \end{cases} \tag{1}$$

The non-linear correction of atlas slices is performed by application of weighted deformation to each slice

$$F_k = f_k \circ \sum_{i=k-d}^{k+d} \gamma_i g_{k,i},$$

where $\gamma_i = \frac{C_{2d}^{i-k+d}}{2^{2d}}$. Such non-linear correction of atlas slices before morphing transform yields a much better spatial interpolation (see fig. 3,d-f).

2.2 Registration of an Experimental Slice into the 3D Model of Mouse Brain

Suppose we have only ISH experimental slice. Evidently it is impossible to find the correspondent virtual slice in the 3D model automatically, since Nissl and ISH stains differ significantly. Note that Nissl stained slices obtained by different research groups could also differ significantly (e.g. in our case Nissl experimental slice differs from Nissl ABA slice). Therefore we developed a special software BrainTravel which allows to navigate through the 3D model and find the correspondent virtual slice manually. We did not expect any difficulties with manual registration of experimental slices using BrainTravel. It usually took us about 2-3 minutes per slice. Note that after registration is finished the virtual and experimental slices are not identical due to the variability of individual brains. Direct transfer of anatomical segmentation to the experimental slice is hence unsuitable (see. fig. 4).

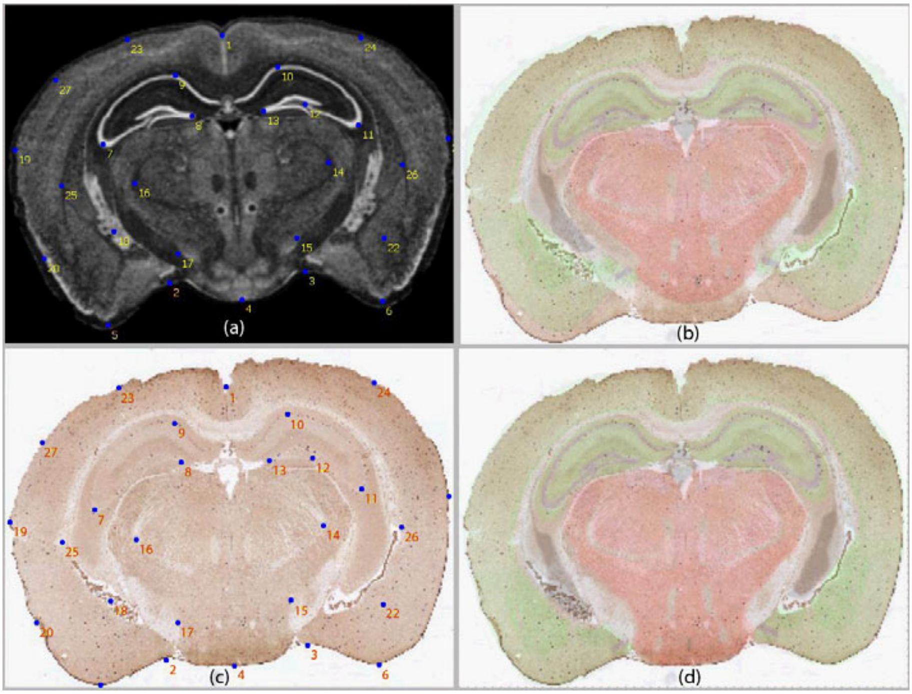


Fig. 4. Example of anatomical segmentation of an ISH experimental slice. (c) – an ISH experimental slice, (a) – the corresponding to (c) virtual slice, (b) – a result of segmentation by direct transfer of anatomical map from the virtual slice, (d) – a result of segmentation with non-linear transform of the virtual slice using keypoints.

2.3 Deformation of Virtual Slice with the Aid of Keypoints

In order to obtain a valid anatomical segmentation of the experimental slice we have to deform the virtual slice which was found during the registration process. The difference in stains makes it impossible to apply any method which minimizes the pixel-wise difference between the two images. A possible way out could be the use of information-based methods [4] which are capable to match images of different modalities. In the paper we focused on an alternative approach which seems to be more reliable and is based on the keypoints location. The user assigns keypoints to the corresponding positions in both images. Let $\left\{(x_k^1, y_k^1)\right\}_{k=1}^K$ and $\left\{(x_k^2, y_k^2)\right\}_{k=1}^K$ be the sets of keypoints on the experimental and virtual slices respectively (see fig. 4a,c). We are looking for the deformation $g : \mathbb{R}^2 \to \mathbb{R}^2$ of the virtual slice such that it minimizes the following criterion

$$E(g) = \sum_{k=1}^{K} d\left((x_k^1, y_k^1), g(x_k^2, y_k^2)\right) \to \min_{g(.,.)\in\mathcal{G}},$$

where $d((x_1, y_1), (x_2, y_2))$ stands for Euclidean distance between the two points. As before we use the family of cubic B-spline[4] deformations

$$\mathcal{G} = \left\{ g(x, y) = \sum_{k=1}^{K} \sum_{m=1}^{M} w_{km}\beta_3(x/h - k)\beta_3(y/h - m) \right\}.$$

After the deformation is found it is applied to the anatomical segmentation of the virtual slice and the deformed segmentation is treated as an anatomical segmentation of experimental slice. An example of such segmentation is shown in figure 4.

3 Expression Detection and Image Resolution Reduction

The detection of expressing cells on ISH slices and the reduction of expression map resolution consists of the following steps:

- image preprocessing;
- extraction of the high-resolution expression map:
  - evaluating the measures of expression presence or absence;
  - combining of expression presence or absence measures into one integral characteristic to create a high resolution expression map;
- reducing expression map resolution.

There is also an additional (optional) step that includes automated quality control of the resulting map and the evaluation of integral characteristics of the resulting expression map.

[4] Cubic B-spline is a function

$$\beta_3(x) = \begin{cases} 2/3 - (1 - |x|/2)x^2, & 0 < |x| \leq 1, \\ (2 - |x|)^3/6, & 1 < |x| < 2, \\ 0, & |x| \geq 2. \end{cases}$$

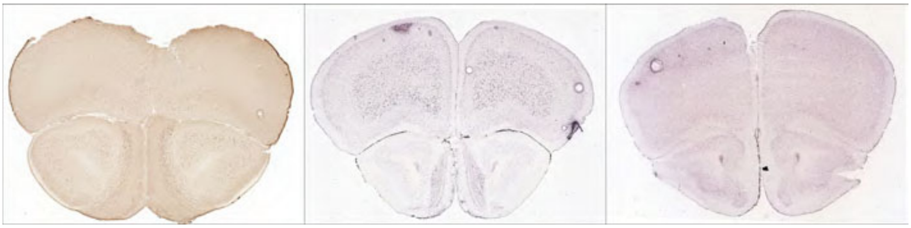


Fig. 5. Examples of slice images. Left: an image from experiments carried out in Anokhin Institute of Normal Physiology (The Russian Academy of Medical Sciences); middle and right: images from Allen Mouse Brain Atlas [6,7].

3.1 Image Preprocessing

Image preprocessing is an optional step that is performed in a fully automated or semi-automated mode. Formally, this step can be omitted, but in this case manual tuning of parameters may be required in order to achieve good quality of expression map extraction.

The main goal of image preprocessing is color and intensity normalization. Due to technical reasons (different illuminance conditions, non-identical coloring agents, etc.) slice images may differ (see Fig. 5), and hence different thresholds and other parameters are required in order to find expression map of good quality for different slice images. Individual estimation of parameters can be replaced by preprocessing that unifies global characteristics of an image.

The preprocessing is based on the analysis of color intensity histograms. At first we construct the grey-scale intensity histogram, separate out the background and recolor it into the fixed color (e.g. white) and in fact exclude it from the further analysis. Then, we perform the intensity/color normalization using affine transformation applied to the intensities of red, green and blue color components (r, g, b). The transformation A_{opt} that minimizes the difference between smoothed color histograms with the predefined "standard" ones is used.

The deviation of the transformation A_{opt} from the identity transformation (in other words, the deviation of transformation matrix from the identity matrix and shift vector from zero one) can be treated as an indirect image quality control characteristic that shows the difference between "standard" experiment conditions (such as illuminance, color-agent properties, etc.) and the conditions of the studied experiment. This deviation can be measured with the discrete analogue of the following formula:

$$\int_0^1 \int_0^1 \int_0^1 \|(r, g, b) - A_{opt}^{-1}(r, g, b)\|_w \; p(r, g, b) \, dr \, dg \, db,$$

where $p(r, g, b)$ is the probability density of observed color (r, g, b) in "standard" slice images and $\| \cdot \|_w$ is a weighted l_1-norm with weights $w = (w_r, w_g, w_b)$ (all color intensities are assumed to be in $[0; 1]$).

3.2 Extraction of the High-Resolution Expression Map

The high-resolution expression map is an image of the same size as the initial image whose pixels show the probabilistic measure of the fact that the corresponding pixel in the initial image is located in the expressing nucleus. Instead of using discrete categories (expression presence / expression absence, or high / moderate / low / no expression), we use continuous (probabilistic) expression measure. In particular, it allows to deal with intermediate categories in a more correct way and hence to achieve better quality after resolution reduction.

To obtain the final expression measure we combine several rough measures. Two types of rough measures are used: “pointwise measures” and “neighborhood measures”.

Pointwise measures evaluate the level of expression presence or absence based on the color intensities of the studied pixel only. In order to reduce computational expenses (such reduction is important due to very high image resolution) we use the point-wise measure: $\sigma(c \cdot r) = \sigma(w_r r + w_g g + w_b b)$, where $c = (r, g, b)$ stands for the color of the pyxel, $w = (w_r, w_g, w_b)$ are some coefficients, and σ is a sigmoid-type function. A sigmoid-type function is a non-decreasing continuous function which equals zero if the argument is less than a threshold and equals one if the argument is greater than another threshold. In the simplest case it is a piecewise-linear (first degree spline) function:

$$\sigma(x) = \begin{cases} 0, & x \le T_{min}, \\ \frac{x - T_{min}}{T_{max} - T_{min}}, & x \in (T_{min}, T_{max}), \\ 1, & x \ge T_{max}. \end{cases}$$

In this paper we use more smooth sigmoid-type functions that are second and third degree splines.

The usage of several pointwise measures is explained by the fact that the projections of regions in the three-dimensional color cube corresponding to color intensities typical for the expression presence and absence on a one-dimensional subspace are partially mixed. Projecting the data to several one-dimensional subspaces helps to reduce the number of cases when expression presence or absence is unclear. In [3], e.g., two projections (or, equivalently, two pointwise measures) are used: the first one allows to determine the pixels where the expression presence is clear, and the second one — the pixels where the expression absence is clear.

Neighborhood measures are also based on the one-dimensional (grey-scale) color $(w_r r + w_g g + w_b b)$ instead of the three-dimensional color intensities (r, g, b). Neighborhood measures are evaluated based on the neighborhood of the analyzed point: the sigmoid-type functions are applied either to the difference between the color of the analyzed pixel and the mean color of the neighborhood, or to the mean color of the neighborhood (it allows to detect and shadow certain types of the artifacts).

After individual measures are evaluated, they are combined into the resulting measure in the following way:

$$P = \exp(\alpha_1 \ln P_1 + \alpha_2 \ln P_2 + \ \ldots \ + \alpha_N \ln P_N) = P_1^{\alpha_1} \cdot P_2^{\alpha_2} \cdot \ \ldots \ \cdot P_N^{\alpha_N},$$

where N is the number of individual measures, P_1, P_2, … , P_N are the values of individual measures, and α_1, α_2, … , α_N are the weights ($\alpha_j \geq 0$). By default, all weights α_j are equal to $1/N$, and the resulting measure is the geometrical mean. Such averaging is more efficient in comparison to the (weighted) arithmetical mean due to the probabilistic nature of measures. If required, in the semi-automated mode user can adjust the weights of individual measures. For example, user may turn off the measure that mainly shadows the artifacts in case of high-quality images with no artifacts, and, on the contrary, increase the weight of this measure in case of poor-quality images. Note that adjustment of weights can be replaced by changing sigmoid-type functions used for the evaluation of individual measures as a positive power of a sigmoid-type function is also a sigmoid-type function.

3.3 Reduction of the Expression Map Resolution

Shortly speaking, the reduction of expression map resolution is performed by the convolution with an appropriate kernel and rescaling by sigmoid-type function. The initial (high-resolution) expression map is divided into rectangles corresponding to the pixels of low-resolution expression field, and for each rectangle $\varDelta_{j,k}$ the measure of expression is calculated as an integral of the product of the high-resolution expression field $\mathrm{Expr}_{\mathrm{highres}}(x, y)$ and the kernel function:

$$\mathrm{Expr}_{\mathrm{lowres}}(j,k) = \sigma\left(\int\int_{(x,y)\,\in\,\mathrm{supp}\,K_{j,k}} \mathrm{Expr}_{\mathrm{highres}}(x,y)\,K_{j,k}(x,y)\,dx\,dy\right),$$

where σ is a sigmoid-type function, $\mathrm{supp}\,K_{j,k}$ is the support of the kernel $K_{j,k}$ or, in other words, the neighborhood of the rectangle $\varDelta_{j,k}$. We used kernel $K_{j,k}$ given by formula $K_{j,k}(x, y) = C(1 - \sigma(\mathrm{dist}\{\varDelta_{j,k}; (x, y\}))$. This measure equals a constant C inside the rectangle $\varDelta_{j,k}$, equals zero outside the neighborhood of this rectangle, and continuously goes to zero in the middle zone.

Note that in the low-resolution expression field the value in each pixel is not the probabilistic measure of expression presence or absence, but a characteristic of total expression in the corresponding region. For a certain kernel and sigmoid-type function this characteristic is a total area of expressing nuclei in the region. Under certain assumptions on the cells comprising the region this characteristic can be easily converted to the approximate number of expressing nuclei of this region.

The results of the reduction of the expression map resolution are illustrated in Fig. 6.

3.4 Postprocessing Using Anatomical Segmentation

In case if anatomical segmentation is available, this segmentation can be used for postprocessing that includes the evaluation of integral characteristics, automated quality control and the correction of expression fields.

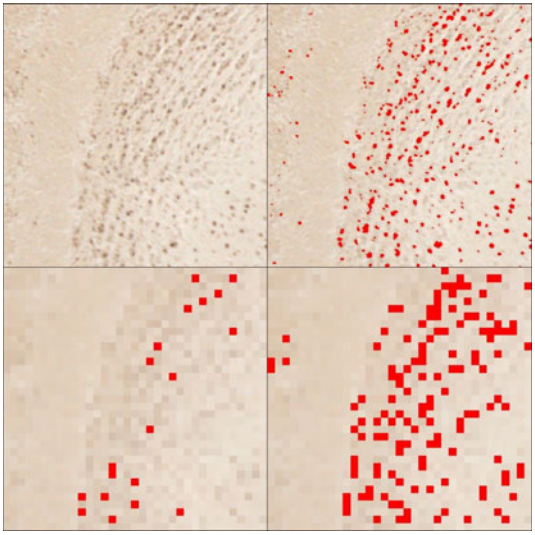


Fig. 6. Examples of expression map resolution reduction. Upper–left: a fragment of a high-resolution ISH slice image; upper–right: the result of expression map extraction; lower–left: the results of expression field extraction from the low-resolution slice image; lower–right: low-resolution expression field.

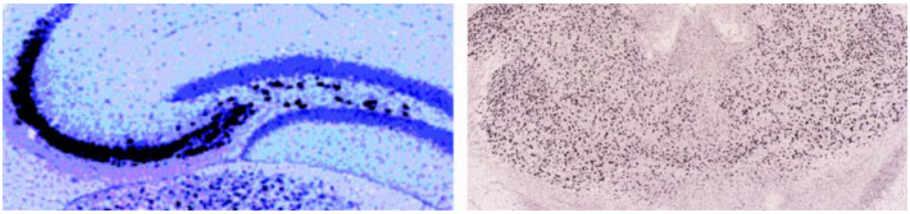


Fig. 7. Example of different properties of expression fields for different structures. Left: ISH image of hippocampal region (source: [8]) — individual expressing nuclei are not separated; right: thalamus region (source: Allen Mouse Brain Atlas [6,7]) — expressing nuclei are separated.

The main integral characteristics that can be evaluated at this step are the total expression characteristics of regions corresponding to individual anatomical structures and substructures. If images of similar slices are available for several representatives of different groups (e.g., “experiment” and “control” groups), these values can be used for rough statistical analysis that determines anatomical structures in which the activity (expression) differs significantly for different groups.

The automated quality control is based on the analysis of connectivity components of high-resolution expression map. For certain anatomical structures these components satisfy certain limitations on size and shape, but these limitations are different for different structures. For example, in many structures expressing nuclei are comparatively small and always separated by non-expressing (non-nucleic) zones, but this property is violated for several structures including hippocampus and olfactory bulb [8] (see Fig. 7). If the anatomical segmentation is available, the compliance of the properties of high-resolution expression map connectivity components with theoretical limitations that hold for the corresponding anatomical structure can be tested. The quality control module reports cases of violations of the limitations and optionally shadows the corresponding connectivity component in the expression map because generally violations are explained by incorrect expression extraction caused by image artifacts.

4 Discussion

We have presented algorithms for anatomical segmentation of Nissl/ISH experimental brain slices and for estimation of gene expression level on them. The solution of the first problem involved the exploitation of the 3D-model of mouse brain which we had built earlier [5] based on Allen mouse brain atlas. The BrainTravel software which we developed allows us to register an experimental slice within the model by finding the most similar virtual slice, and to deform the anatomical structures on the virtual slice from the model so that it matches the experimental slice. Thus we obtain a valid anatomical segmentation of the experimental slice. Note that despite the alternative methods based on the use of Markov Random Fields [2] this method is not limited by the number of anatomical structures it may discover. The estimation of expression level is performed in a separate procedure and the results can be loaded into the BrainTravel. Finally we can unite the results of segmentation with the expression level (see fig. 4d). This union opens vast perspectives for further cognitive research since it becomes possible to include the knowledge about anatomical structures in the statistical analysis of gene expression. Potentially this should lead to more elaborate bioexperimental design and the discovery of new genes and anatomical zones which are activated during cognitive activity of an animal.

Acknowledgments. The work was supported by the Russian Foundation for Basic Research (projects Nos. 10-01-90015, 10-01-90419) and Federal Target Program ”Scientific and scientific-pedagogical personnel of innovative Russia in 2009-2013” (contracts Nos. P1265, P1295).

References

1. Bolyne, J., Lee, E.F., Toga, A.W.: Digital Atlases as a Framework for Data Sharing. Frontiers in Neuroscience 2, 100–106 (2008)
2. Bae, M., Pan, R., Wu, T., Badea, A.: Automated Segmentation of Mouse Brain Images Using Extended MRF. NeuroImage 46(3), 717–725 (2009)
3. Carson, J., Ju, T., Thaller, C., Warren, J., Bello, M., Kakadiaris, I.A., Chiu, W., Eichelle, G.: Automated characterization of gene expression patterns with an atlas of the mouse brain. In: Proc. 26th Annual Int. Conf. of the IEEE Engineering in Medicine and Biology Society, vol. 2, pp. 2917–2920 (2004)

4. Hermosillo, G., Chefdhotel, C., Faugeras, O.: Variational methods for multimodal image matching. Int. J. Comput. Vis. 50, 329–343 (2002)
5. Osokin, A., Vetrov, D., Kropotov, D.: 3-D Mouse Brain Model Reconstruction from a Sequence of 2-D Slices in Application to Allen Brain Atlas. In: Masulli, F., Peterson, L.E., Tagliaferri, R. (eds.) CIBB 2009. LNCS, vol. 6160, pp. 291–303. Springer, Heidelberg (2010)
6. Lein, E.S., et al.: Genome-wide atlas of gene expression in the adult mouse brain. Nature 445, 168–176 (2007)
7. Allen Mouse Brain Atlas [Internet]. Seattle (WA): Allen Institute for Brain Science (2009), `http://mouse.brain-map.org`
8. Ng, L., Pathak, S.D., Hawrykycz, M., et al.: Neuroinformatics for Genome–Wide 3D Gene Expression Mapping in the Mouse Brain. IEEE/ACM Transactions on Computational Biology and Bioinformatics 4(3), 382–393 (2007)
9. Ju, T., Warren, J., Carson, J., et al.: 3D volume reconstruction of a mouse brain from histological sections using warp filtering. Journal of Neuroscience Methods 156(1-2), 84–100 (2006)

Prediction of the Bonding State of Cysteine Residues in Proteins with Machine-Learning Methods

Castrense Savojardo[1,2], Piero Fariselli[1,2], Pier Luigi Martelli[1],
Priyank Shukla[1,2], and Rita Casadio[1]

[1] Biocomputing Group University of Bologna
via Irnerio 42, 40126 Bologna, Italy
[2] Department of Computer Science, University of Bologna
Via Mura Anteo Zamboni 7, 40127 Bologna, Italy
{savojard,piero,gigi,priyank,casadio}@biocomp.unibo.it
http://www.biocomp.unibo.it

Abstract. In this paper we evaluate the performance of machine learning methods in the task of predicting the bonding state of cysteines starting from protein sequences. This task is the first step for the identification of disulfide bonds in proteins. We score the performance of three different approaches: 1) Hidden Support Vector Machines (HSVMs) which integrate the SVM predictions with a Hidden Markov Model; 2) SVM-HMMs which discriminatively train models that are isomorphic to a kth-order hidden Markov model; 3) Grammatical-Restrained Hidden Conditional Random Fields (GRHCRFs) that we recently introduced. We evaluate two different encoding schemes based on sequence profile and position specific scoring matrix (PSSM) as computed with the PSI-BLAST program and we show that when the evolutionary information is encoded with PSSM all the methods perform better than with sequence profile. Among the different methods it appears that GRHCRFs perform slightly better than the others achieving a per protein accuracy of 87% with a Matthews correlation coefficient (C) of 0.73. Finally, we investigate the difference between disulfide bonding state predictions in Eukaryotes and Prokaryotes. Our analysis shows that the per-protein accuracy in Prokaryotic proteins is higher than that in Eukaryotes (0.88 vs 0.83). However, given the paucity of bonded cysteines in Prokaryotes as compared to Eukaryotes the Matthews correlation coefficient is drastically reduced (0.48 vs 0.80).

Keywords: Machine Learning, Conditional Random Fields, Disulfide Prediction, Disulfide Bonding State, Protein Structure Prediction, Protein Folding.

1 Introduction

Disulfide bonds may link the thiol groups of cysteine residues in membrane and globular proteins [8]. Their formation is reversible and can be modulated by the

R. Rizzo and P.J.G. Lisboa (Eds.): CIBB 2010, LNBI 6685, pp. 98–111, 2011.
© Springer-Verlag Berlin Heidelberg 2011

redox ambient potential and mediated by specific proteins [8]. The bonding state of cysteines plays a relevant role in stabilizing the tertiary folds of proteins, in defining protein functions and in triggering functionally relevant conformational changes [8,16].

In recent years, the prediction of the bonding state of cysteine residues has emerged as an increasingly important task due to its relevance in constraining the structure and function of proteins. By this, cysteines likely to make disulfide bonds in the folded protein structure starting from its sequence can be highlighted. In recent years a variety of computational approaches has been proposed to address this problem. The first advancement was the introduction of evolutionary information to train a neural-network based predictor [10]. Fiser and Simon [13] observing that the great majority of the proteins tend to have cysteines all-bonded or all-free, introduced an "all-or-none" rule. This however hampers the applicability of the method to proteins containing cysteines of both types. Methods based on local residue and global protein descriptors were also developed and indicate that protein composition is a relevant piece of information [18,21]. Based on the observation that the number of bonded cysteines is even, Martelli et al. [17] first introduced a Hidden Neural Network approach that takes advantage of both local and global characteristics of the protein. Other machine learning methods were also described [7,3,5,23].

A comparison among different machine learning approaches is difficult since authors reported different scoring indices on different datasets and also they often do not specify if the training data set comprises or not trivial cases (proteins with only one cysteine in the sequence that obviously cannot contain disulfide bridges). This can bias the scoring values. Here we compare three different state-of-the-art machine-learning methods on a newly generated non redundant dataset specifically built for the purpose of predicting the cysteine bonding state.

2 Materials and Methods

2.1 Dataset Description

Our dataset is derived from September 2009 PDB release. We selected all proteins whose structure resolution was higher than 2.5 Angstrom with at least two cysteines. We ended up with a dataset of 3940 sequences whose details are reported in Table 1. Since our selected proteins may contain some local sequence similarity we further clustered the remaining protein chains using the transitive closure algorithm by defining a graph whose nodes represent the proteins. An edge connects two nodes if and only if sequence identity between the corresponding protein sequences is $> 25\%$ (according to an all-against-all protein BLAST search). The transitive closure algorithm defines clusters as the connected components of the graph. Thus, clusters can contain proteins whose pairwise sequence identity is very low. With these clusters we created 20 different balanced sets to perform 20-fold cross validation tests.

Table 1. Dataset description

Type	Number
Protein sequences	3940
Total cysteine residues	18918
Bonded cysteine residues	4110
Free cysteine residues	14808

2.2 Machine-Learning Based Predictors

The prediction of the disulfide bonding state of cysteines can be formulated as a sequence labeling task. Protein cysteine residues were extracted and only the list of ordered cysteines (with their neighboring residues) was considered. A machine learning technique for sequence labeling was applied in order to label each cysteine as bonded or free.

In this paper, we considered three different approaches that for convenience are named as:

- Hidden Support Vector Machines (HSVMs)
- Hidden Markov Support Vector Machines (SVM-HMMs)
- Grammatical-Restrained Hidden Conditional Random Fields (GRHCRFs)

HSVMs. HSVMs are a natural extension of the previously described Hidden Neural Networks [17] that were based on the combination of two popular machine learning algorithms, Neural Networks and Hidden Markov Models. In this hybrid system, a standard feed-forward NN is firstly trained in order to estimate the probability of each cysteine of being in the bonding or free state. The output of NN is then used as state emission probability of a HMM. The probabilistic state model assures that only meaningful predictions (an even number of bonding cysteines) are provided by means of the Viterbi decoding. Here NNs are replaced by SVMs. In order to obtain probabilistic output from SVMs, we adopted the LIBSVM package [6]. We report results obtained with a RBF kernel, the best performing kernel function among those tested.

SVM-HMMs. Hidden Markov Support Vector Machines [2] have been developed within the emerging framework of large margin methods for structured output learning [22]. To implement these models we use the package SVM-HMM [15] and we refer to them with this name. SVM-HMMs combine the maximum margin principle typical of SVMs with a HMM as kernel and by this is possible to run efficient dynamic programming algorithms and probabilistic dependency between adjacent labels. In addition, SVM-HMMs follow the discriminative learning approach and they discriminatively train models that are isomorphic to a kth-order hidden Markov model using the Structural Support Vector Machine (SVM) formulation. Although in principle SVM-HMM can learn a non-ambiguous grammar, the current implementation sometimes fails and the final predictions are not always biologically meaningful.

Given an observation sequence $\mathbf{x} = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T)$, SVM-HMM predicts a label sequence $\mathbf{y} = (y_1, y_2, \ldots, y_T)$ by means of the following linear discriminant function:

$$\mathbf{y}^* = \arg\max_{\mathbf{y}} \textstyle\sum_{i=1}^{T} \sum_{j=1}^{k} \langle \mathbf{x}_i, \mathbf{w}^e_{y_{i-j},\ldots,y_i} \rangle + \sum_{i=1}^{T-1} w^t_{y_{i-j},\ldots,y_i} \tag{1}$$

where k is the order of the HMM. The model learns one emission weight vector $\mathbf{w}^e_{y_{i-j},\ldots,y_i}$ for each kth-order label sequence $y_{i-j}, \ldots, y_i$ and one transition weight $w^t_{y_{i-j},\ldots,y_i}$ for each sequence of adjacent labels.

In the training phase, given a set of training examples $\mathcal{D} = \{(\mathbf{x}^n, \mathbf{y}^n) \in \mathcal{X} \times \mathcal{Y} \mid n = 1, \ldots, m\}$, SVM-HMM solves the following quadratic optimization problem:

$$\begin{aligned}
&\min_{\mathbf{w},\xi} \tfrac{1}{2} \|\mathbf{w}\|^2 + \tfrac{C}{n} \textstyle\sum_{n=1}^{m} \xi_n \\
&s.t. \textstyle\sum_{i=1}^{T} \sum_{j=1}^{k} \langle \mathbf{x}^n_i, \mathbf{w}^e_{y^n_{i-j},\ldots,y^n_i} \rangle + \sum_{i=1}^{T-1} w^t_{y^n_{i-j},\ldots,y^n_i} \geq \\
&\textstyle\sum_{i=1}^{T} \sum_{j=1}^{k} \langle \mathbf{x}^n_i, \mathbf{w}^e_{y'_{i-j},\ldots,y'_i} \rangle + \sum_{i=1}^{T-1} w^t_{y'_{i-j},\ldots,y'_i} + \mathit{\Delta}(\mathbf{y}^n, \mathbf{y}') - \xi_n \\
&\forall n, \forall \mathbf{y}' \neq \mathbf{y}^n, \xi_n \geq 0
\end{aligned} \tag{2}$$

where C is a parameter that governs the trade-off between training error and margin size, ξ_n is a slack variable and $\mathit{\Delta}(\mathbf{y}^n, \mathbf{y}')$ is a loss function that computes the per label loss for each individual label sequence $\mathbf{y}'$ (Hamming loss).

Since the number of constraints in the above optimization problem exponentially increases with the leangth of the sequence (the length of $\mathbf{y}$), the cutting-plane algorithm is used to solve it up to a precision of ϵ in polynomial time [22].

GRHCRFs. We also benchmark our recently developed Grammatical-Restrained Hidden Conditional Random Field (GRHCRFs) [12]. GRHCRFs can incorporate regular grammar production rules in order to consider, during both training and prediction phases, only those labeling that are in agreement with the user defined grammar. In addition, the discriminative nature of GRHCRFs offers several advantages over generative approaches such as Hidden Markov Models (HMMs), including the relaxation of the strong independence assumptions [12].

Like HMMs, GRHCRFs can be represented through Finite State Machines (FSMs). The structure of FSMs is determined by the specific grammar used for the problem at hand. In a typical sequence labeling task, given an observation sequence $\mathbf{x}$, we want to obtain the most probable label sequence $\mathbf{y}$. In order to better generalize, GRHCRFs (as well as HMMs) define a one-to-many mapping between labels and FSM states and, at the same time, restrict the accepted predictions to only those corresponding to an allowed path.

Let $\mathbf{x}$ be the random variable over observation sequences to be labelled, $\mathbf{y}$ the random variable over label sequences and $\mathbf{s}$ the random variable over state sequences. A function $\mathit{\Lambda}(s) = y$ is defined to map each state s to a given label y. In a first order model, explicit feature functions $f(s_{j-1}, s_j, \mathbf{x})$ unrolled over each sequence position j are considered. The probability of a label sequence $\mathbf{y}$, given an observation sequence $\mathbf{x}$, is obtained as:

$$p(\mathbf{y}|\mathbf{x}) = \frac{Z(\mathbf{y}, \mathbf{x})}{Z(\mathbf{x})} \tag{3}$$

where

$$Z(\mathbf{y},\mathbf{x}) = \sum_{\mathbf{s}} \prod_j \exp(\sum_k \lambda_k f_k(s_{j-1}, s_j, \mathbf{x}))) \cdot \mathit{\Gamma}(s_{j-1}, s_j) \cdot \mathit{\Omega}(s_j, y_j) \tag{4}$$

$$Z(\mathbf{x}) = \sum_{\mathbf{y}} \sum_{\mathbf{s}} \prod_j \exp(\sum_k \lambda_k f_k(s_{j-1}, s_j, \mathbf{x}))) \cdot \mathit{\Gamma}(s_{j-1}, s_j) \cdot \mathit{\Omega}(s_j, y_j) \tag{5}$$

are normalization factors (partition functions) that can be computed using a standard Forward-Backward procedure. The constraints

$$\mathit{\Gamma}(s, s') \begin{cases} 1 \text{ if } (s, s') \text{ is a valid transition} \\ 0 \qquad\qquad \text{otherwise} \end{cases} \tag{6}$$

$$\mathit{\Omega}(s, y) \begin{cases} 1 \text{ if } \mathit{\Delta}(s) = y \\ 0 \;\; \text{otherwise} \end{cases} \tag{7}$$

have been introduced in order to consider only valid state paths in the FSM.

Given training data $\mathcal{D} = \{(\mathbf{x}^n, \mathbf{y}^n) \in \mathcal{X} \times \mathcal{Y} \mid n = 1, \ldots, m\}$, parameters $\mathit{\Theta} = \{\lambda_k\}$ associated to each feature f_k, are learned via maximization of the conditional log-likelihood:

$$\begin{aligned} \mathcal{L}(\mathcal{D}; \mathit{\Theta}) &= \log \prod_{n=1}^{m} p(\mathbf{y}^n | \mathbf{x}^n; \mathit{\Theta}) \\ &= \sum_{n=1}^{m} \log(Z(\mathbf{y}^n, \mathbf{x}^n)) - \log(Z(\mathbf{x}^n)) \quad . \end{aligned} \tag{8}$$

The maximization is carried out using the Limited memory Broyden-Fletcher-Goldfarb-Shanno (L-BFGS) quasi-Newton optimization algorithm [4]. For further details about GRHCRFs we refer to [12].

3 Input Description

To assign the bonding state of cysteines we encode each cysteine with a "local vector" representing the sequence neighbors. The neighborhood is based on a residue-window w ($w = 2n + 1$ residues) centered into the cysteine to be predicted. Then, the local encoding of each cysteine consists of a vector of dimension $20 \cdot w$, where the number 20 elements is the number of residue types. We used two different kinds of local encoding, both based on evolutionary information as computed by PSI-BLAST [1]: the sequence profile (frequency of the residues in the alignment positions) and the position substitution score matrix (PSSM) as internally computed by PSI-BLAST with the BLOSUM62 matrix. Furthermore, we tested several window dimensions to determine how the local encoding affects the method accuracies.

In order to obtain the profiles and the corresponding PSSMs, for each protein sequence in our dataset we run PSI-BLAST locally against the Uniref90 dataset with eight iterations (-j 8) and with an expectation equal to 0.001 (-e 0.001).

3.1 Scoring Indices

To evaluate the accuracy we define the classical label-based indices, such as:

$$Q2 = \frac{p}{N} \tag{9}$$

where p and N are the total number of correct predictions and total number of examples, respectively. The Matthews correlation coefficient (C) for a given class s is defined as:

$$C(s) = \frac{[p(s)n(s) - u(s)o(s)]}{[(p(s) + u(s))(p(s) + o(s))(n(s) + u(s))(n(s) + o(s))]^{1/2}} \quad . \tag{10}$$

$p(s)$ and $n(s)$ are respectively the true positive and true negative predictions for class s, while $o(s)$ and $u(s)$ are the numbers of false positives and false negatives with respect to that class. The Precision (coverage, Pr) for each class s is defined as

$$Pr(s) = \frac{p(s)}{[p(s) + u(s)]} \quad . \tag{11}$$

The Recall (accuracy, Re) is the probability of correct predictions and it is defined as follows:

$$Re(s) = \frac{p(s)}{[p(s) + o(s)]} \quad . \tag{12}$$

The F1-Score is defined as the harmonic mean between Precision and Recall:

$$F1(s) = \frac{2 \cdot Pr(s) \cdot Rc(s)}{Pr(s) + Re(s)} \quad . \tag{13}$$

Finally, the Q_p is defined as the number of correctly predicted proteins N_{cp} divided by the total number of proteins N_p:

$$Q_p = \frac{N_{cp}}{N_p} \quad . \tag{14}$$

4 Results and Discussion

4.1 Evaluation of the Different Methods

We evaluate the three different methods as a function of the window amplitude centered on the cysteine residue and adopting different models. While SVM-HMM and GRHCRF are single methods, HSVM consists of a cascade of two algorithms and we trained and tested it in two steps. In the first step we train and test a standard Support Vector Machine to assign bonding state probabilities to each cysteine of our dataset. We tested several standard kernel functions and different parameters using a 20-fold cross-validation split. It turned out that the best performing SVM is based on the RBF kernel, so that only the results obtained with this kernel are shown below. In the second step for each learning set we trained a HMM model where the emission probabilities are defined by the SVM predictions (fore more details see [17]). HSVM is then evaluated as a single method. Two different models were tested: a 4-state model (Figure 1) and a 7-state model (Figure 2). The basic idea behind the two models is to generate predictions that are consistent with the notion that only an even number of bonding cysteines can be assigned, since each disulfide bond requires two of them. The 4-state model (Figure 1) is the simplest automaton that fulfills the disulfide bond constraints. However, it was previously noted that the vast majority of the protein sequences tend to have cysteines all-bonded or all-free and very few protein sequences have a mixture of both states [13]. For this reason, in order to capture the different priors (and maybe different local propensities) we also test a 7-state model (Figure 2) for which the three different protein classes (all-bonded , all-free and mixed) are made explicit by three different possible paths through the automaton (Figure 2).

Figure 3 shows that the per protein accuracy (Qp) of the HSVM models increases as the input window increases and achieves a maximum value when the window length comprises 25 residues. It is worth noticing that the PSSM encoding (black curves) performs better than the sequence profile encoding (gray curves), indicating that BLOSUM62 weighted with position specific residue frequency is better for the task at hands. Finally from Figure 3 it appears that the 4-state HMM performs slightly better than the 7-state one.

SVM-HMM is a single method that optimizes the labeling assignment with a maximal margin approach [22]. The implementation allows the usage of hidden Markov models of different order from 0 to 3. We evaluate the accuracy for Markov models of order two and three (see Figure 4). We tested two different

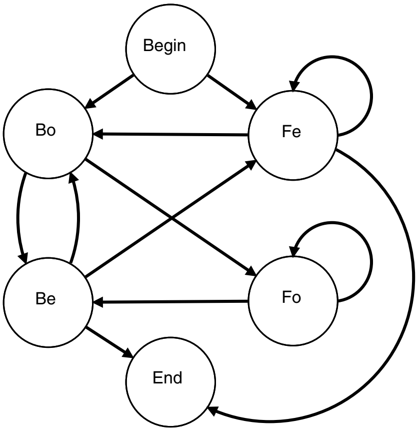


Fig. 1. Four state automaton describing the grammar of the cysteine residue in protein sequences (Begin and End states are silent and are not counted). The states *Bo* and *Be* define bonding labels while the states *Fo* and *Fe* indicate free cysteine labels.

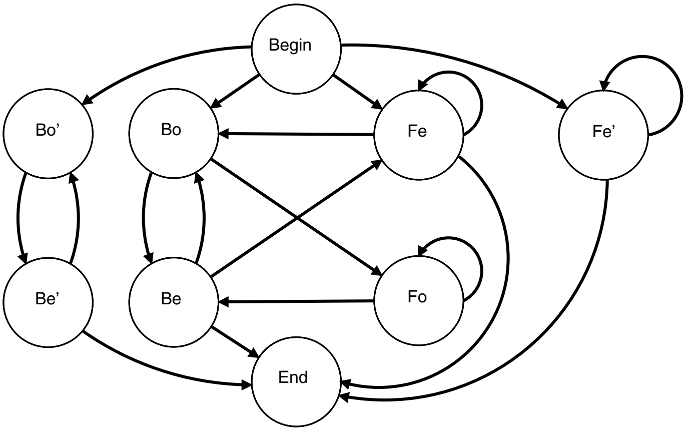


Fig. 2. Seven state automaton describing the grammar of the cysteine residue in protein sequences. The paths generated with the leftmost part of the automaton (Bo',Be') describe the proteins that contain only bonded cysteines. The paths generated with the rightmost states of the automaton (Fe') represent the proteins that contain only free cysteines. The paths generated with the central part of the automaton (Bo,$B2$,Fo,Fe) defines the proteins that contain both types of cysteines.

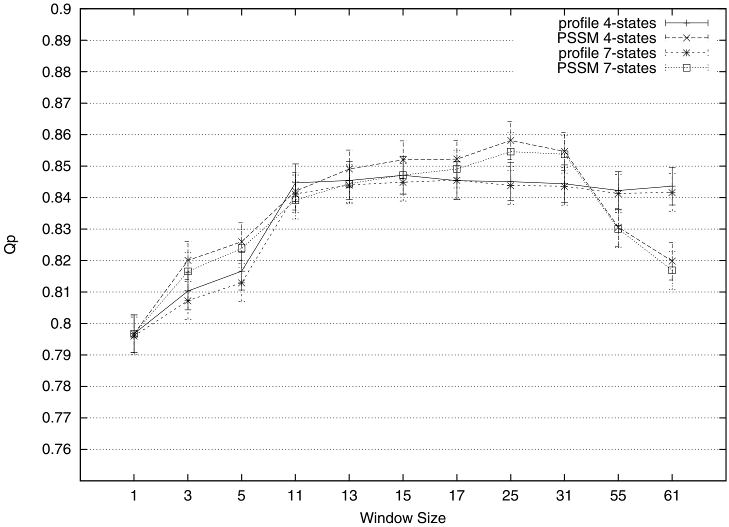


Fig. 3. HSVM cross-validation performance as a function of the cysteine local environment (window size). Two different input encoding (PSSM-based and profile-based) and two different automata (with 4-states and 7 states) are tested. Error bars are computed using the binomial standard deviation.

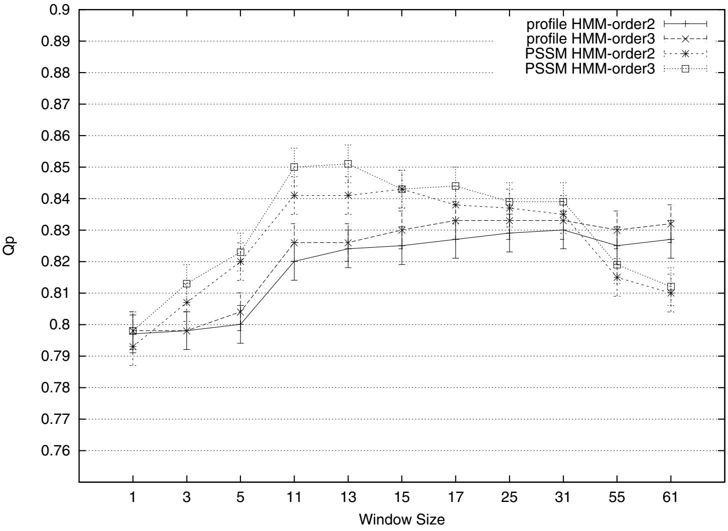


Fig. 4. SVM-HMM cross-validation performance as a function of the cysteine local environment (window size). Two different input encodings (PSSM-based and profile-based) and two different Markov chain orders (2 and 3 step back) are tested. Error bars are computed as in Figure 3.

labeling schemes for the SVM-HMM. The first one uses only two labels, Free and Bonded. However, since this labeling produced poor results we adopted the automaton of Figure 1 to assign the labels, namely: *Fo*-*Fe* (as free) and *Bo*-*Be* (as bonded). Furthermore to add information about the stating and ending probabilities we added two "dummy" cysteines at the beginnings and at the ends. In this case the performance is higher as indicated by the results shown in Figure 4 as a function of the dimension of the input window. It is worth noticing that even in this case some SVM-HMM predictions do not fulfill the parity constraints. In Figure 4 it appears that for the predictive performance the input encoding is more relevant than the Markov model order and that the PSSM encodes more information than the sequence profile also for SVM-HMM.

Figure 5 shows the per protein accuracy (Qp) of the GRHCRFs as a function of the input window size. Also in this case the PSSM-based input performs better than the profile-based one. Differently from HSVMs, the performance of the 7-state model is comparable with that of the 4-state model or even slightly better, achieving the maximal per protein accuracy with a window of 17 residues (Figure 5).

In Table 2, for each method we show detailed accuracy indices obtained with the best performing window (Figures 3, 4, 5). For sake of comparison we also report the results achieved with the best SVM to highlight that the introduction of an automaton significantly increases the performance (compare SVM with the other methods, especially HSVM). Furthermore, it appears that the GRHCRF model tends to give more balanced predictions as indicated by the C and F1

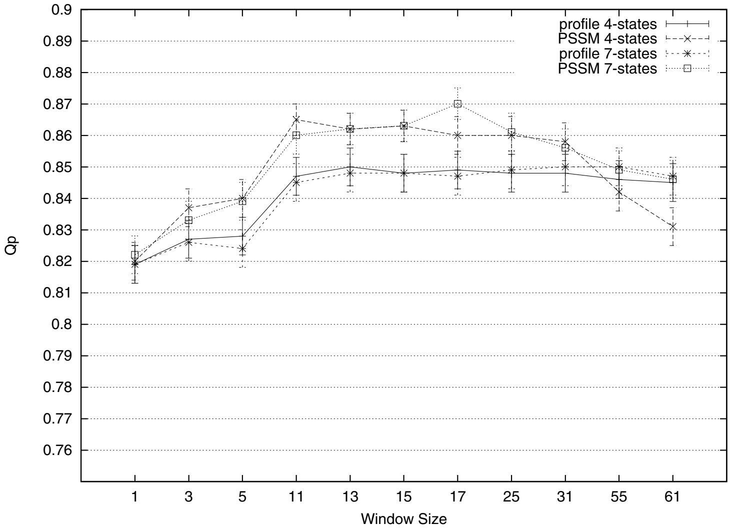


Fig. 5. GRHCRF cross-validation performance as a function of the cysteine local environment (window size). Two different input encodings (PSSM-based and profile-based) and two different automata (with 4-states and 7-states) are tested. Error bars are computed as in Figure 3.

Table 2. Cross-validation performance of the different best models

Method	Pr(b)	Re(b)	F1(b)	Pr(f)	Re(f)	F1(f)	Q2†	CC†	Qp†
SVM*	0.78	0.56	0.65	0.89	0.96	0.92	0.87	0.59	0.75
HSVM	0.92	0.56	0.69	0.89	0.98	0.94	0.89	0.66	0.86
SVM-HMM	0.86	0.64	0.73	0.91	0.97	0.94	0.90	0.69	0.85
GRHCRF	0.88	0.69	0.77	0.92	0.97	0.95	0.91	0.73	0.87

* SVM is presented here to show the improvement when it is coupled with a HMM (HSVM). † Mann-Whitney test is performed on these global indexes using the best and the second best methods to evaluate if the differences are statistically significant. The 20-fold cross-validation values are used to evaluate the Mann-Whitney scores and P-values are 0.004, 0.002 and 0.115 for Q2, CC and Qp, respectively.

indices. This is very important since the two classes are unbalanced (roughly 30% bonded and 70% free cysteines, Table 1) and all methods tend to overpredict the more abundant Free class.

4.2 Organism-Based Prediction: Eukaryotes vs. Prokaryotes

Disulfide bond stability depends on the redox properties of the environment in which the protein is located. For this reason, the proteins that contain stable disulfide bonds are usually secreted and are rarely found in the cytoplasmic

Table 3. The distribution of free and bonded cysteines in Eukariotic and Prokaryotic proteins

number of bonds	Eukaryotes		Prokaryotes	
	Bonded	Free	Bonded	Free
0	0	7116	0	6925
1	288	304	474	233
2	400	81	184	22
3	510	55	114	5
4	328	15	88	4
≥5	1668	48	56	0
All	3194	7619	916	7189

Table 4. Cross-validation performance of GRHCRF trained on chains from Eukaryotes and Prokaryotes

Set	Input	Pr(b)	Re(b)	F1(b)	Pr(f)	Re(f)	F1(f)	Q2	CC	Qp
Eukaryotes	PSSM	0.91	0.83	0.87	0.93	0.96	0.94	0.91	0.80	0.83
Eukaryotes	Profile	0.88	0.77	0.82	0.91	0.96	0.93	0.90	0.76	0.81
Prokaryotes	PSSM	0.64	0.43	0.51	0.93	0.97	0.95	0.90	0.48	0.88
Prokaryotes	Profile	0.73	0.19	0.30	0.91	0.99	0.95	0.90	0.34	0.86

For the indices see the subsection "Scoring indices". b=bonded cysteines, f=free cysteines.

compartments [9,14]. There is also a difference between Prokaryotes and Eukaryotes organisms. If proteins are from Prokaryotes the formation of disulfide bonds commonly take place in extracytoplasmatic compartments [14,19], while in Eukaryotic cells the bond formation occurs in the lumen of the endoplasmic reticulum [20]. For these reasons, it is interesting to evaluate if there are differences between cysteine-containing proteins sorted out with respect to the distinction in Eukaryotic and Prokaryotic chains. This can affect the prediction capability of our method. Starting from our dataset, in Table 3 we report the distribution of the proteins sorted out by Eukaryote/Prokaryote organisms. From the data it is evident that the distribution of disulfide bonded cysteines is very different in the two types of subsets. In Prokaryotes proteins that contain cysteine residues not only tend to have far less disulfide bonds than Eukaryotic proteins, but also the number of distinct disulfide patterns is far lower as indicated by the fact that proteins with more than 4 disulfide bonds are presently absent (Table 3). With this picture, it is clear that the prediction of disulfide bonds may be different in the two subgroups. In Table 4 we report the results obtained with the Grammatical Restrained Hidden Conditional Random Field models as a function of the cell organism type. The results shown in Table 4 indicate that the disproportion between free and bonded cysteine in

Prokaryotes generates unbalanced predictions and that this leads to a dominance of free state predictions. However, this effect leads to an increase of Q_p, the overall per-protein accuracy (Table 4). The Qp is the most relevant measure for the evaluation of the prediction capability of the methods with respect to the entire protein sequence [11]. As noted before, the adoption of a PSSM-based input increases the method performance independently of the organism type.

5 Conclusions

In this paper we evaluate the performance of different machine-learning methods on the task of predicting the disulfide bonding state of cysteines using different input encodings. We show that when evolutionary information is encoded with PSSM all the methods perform better than with sequence profile (Figures 3-5). All the machine learning models are well performing. In particular, the newly developed GRHCRF can correctly predict 87% of the proteins of our (non trivial) dataset with C equal to 0.73. The C values of GRHCRF is significantly different from that obtained by the second best SVM-HMM method (0.69), since accordingly to the Mann-Whitney test the equality hypothesis scores with a P-value of 0.002. Furthermore, we investigated the differences between disulfide bonding state prediction in classifying protein chain in relation to their Prokaryotic or Eukaryotic origin. Our analysis shows that the per-protein accuracy in Prokaryotic proteins is higher than those in Eukaryotes. This corresponds however to a large decrease of the Matthews Correlation Coefficient due to an unbalanced prediction of free state cysteines with respect to the bonded state ones. This effect is mirroring the paucity of bonding state examples present in the Prokaryotic proteins.

Summing up, our results indicate that when the different approaches are tested on a non redundant, abundant, updated and non trivial data set of proteins, the performance of GRHCRF is slightly better than that of the state of the art predictors. GRHCRF is therefore proposed as a good candidate method for filtering genomes and predicting disulfide proteomes in the different platforms for large-scale automated annotation processes.

Acknowledgments

We thank MIUR for the following grants: PNR 2001-2003 (FIRB art.8) and PNR 2003 projects (FIRB art.8) on Bioinformatics for Genomics and Proteomics, both delivered to RC. All the authors thank the LIBI-Laboratorio Internazionale di BioInformatica. PS is a recipient of 'Brain-in' PhD scholarship from Institute of Advanced Studies, University of Bologna, and '100-Young Indian Researcher' scholarship from Ministry of Instruction University and Research (MIUR), Government of Italy.

References

1. Altschul, S.F., Gish, W., Miller, W., Myers, E.W., Lipman, D.J.: Basic local alignment search tool. J. of Mol. Biol. 213(3), 403–410 (1990)
2. Altun, Y., Tsochantaridis, I., Hofmann, T.: Hidden Markov Support Vector Machines. In: Twentieth International Conference on Machine Learning (ICML 2003), Washington DC (2003)
3. Baldi, P., Cheng, J., Vullo, A.: Large-scale prediction of disulphide bond connectivity. In: Saul, L.K., Weiss, Y., Bottou, L. (eds.) Advances in Neural Information Processing Systems 17, pp. 97–104. MIT Press, Cambridge (2005)
4. Byrd, R.H., Lu, P., Nocedal, J.: A Limited Memory Algorithm for Bound Constrained Optimization. SIAM Journal on Scientific and Statistical Computing 16(5), 1190–1208 (1995)
5. Ceroni, A., Passerini, A., Vullo, A., Frasconi, P.: DISULFIND: a Disulfide Bonding State and Cysteine Connectivity Prediction Server. Nucleic Acids Research 34 (Web Server), W177–W181 (2006)
6. Chang, C.-C., Lin, C.-J.: LIBSVM : a library for support vector machines (2001), Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`
7. Chen, Y.C., Lin, Y.S., Lin, C.J., Hwang, J.K.: Prediction of the bonding states of cysteines using the support vector machines based on multiple feature vectors and cysteine state sequences. Proteins: Structure, Function, and Bioinformatics 55(4), 1036–1042 (2004)
8. Creighton, T.E.: Proteins: Structures and Molecular Properties. W.H. Freeman, New York (1992)
9. Derman, A.I., Beckwith, J.: Escherichia coli alkaline phosphatase fails to acquire disulfide bonds when retained in the cytoplasm. Journal of Bacteriology 173(23), 7719–7722 (1991)
10. Fariselli, P., Riccobelli, P., Casadio, R.: Role of evolutionary information in predicting the disulfide-bonding state of cysteine in proteins. Protein: Structure, Function, and Bioinformatics 36(3), 340–346 (1999)
11. Fariselli, P., Martelli, P.L., Casadio, R.: A neural network based method for predicting the disulfide connectivity in proteins. In: Damiani, E., et al. (eds.) Knowledge Based Intelligent Information Engineering Systems and Allied Technologies (KES 2002), vol. 1, pp. 464–468. IOS Press, Amsterdam (2002)
12. Fariselli, P., Savojardo, C., Martelli, P.L., Casadio, R.: Grammatical-Restrained Hidden Conditional Random Fields for Bioinformatics applications. Algorithms for Molecular Biology 4(13) (2009)
13. Fiser, A., Simon, I.: Predicting the oxidation state of cysteines by multiple sequence alignment. Bioinformatics 16(3), 251–256 (2000)
14. Kadokura, H., Katzen, F., Beckwith, J.: Protein disulfide bond formation in prokaryotes. Annual Review of Biochemistry 72, 111–135 (2003)
15. Joachims, T.: SVM-HMM (2010), `http://www.cs.cornell.edu/People/tj/svm_light/svm_hmm.html`
16. Liu, H.-L.: Recent Advances in Disulfide Connectivity Predictions. Current Bioinformatics 2(1), 31–47 (2007)
17. Martelli, P.L., Fariselli, P., Malaguti, L., Casadio, R.: Prediction of the disulfide bonding state of cysteines in proteins with hidden neural networks. Protein Engineering Design and Selection 15(12), 951–953 (2002)
18. Mucchielli-Giorgi, M.H., Hazout, S., Tuffery, P.: Predicting the disulfide bonding state of cysteines using protein descriptors. Proteins: Structure, Function, and Bioinformatics 46(3), 243–249 (2002)

19. Nakamoto, H., Bardwell, J.C.A.: Catalysis of disulfide bond formation and isomerization in the bacterial periplasm. Biochimica et Biophysica Acta 1694(1-3), 111–119 (2004)
20. Sevier, C.S., Qu, H., Heldman, N., Gross, E., Fass, D., Kaiser, C.A.: Modulation of cellular disulfide-bond formation and the ER redox environment by feedback regulation of Ero1. Cell 129(2), 333–344 (2007)
21. Song, J.N., Wang, M.L., Li, W.J., Xu, W.B.: Prediction of the disulfide-bonding state of cysteines in proteins based on dipeptide composition. Biochemical and Biophysical Research Communications 318(1), 142–147 (2004)
22. Tsochataridis, I., Joachims, T., Hofmann, T., Altun, Y.: Large Margin Methods for Structured and Interdependent Output Variables. Journal of Machine Learning Research 6, 1453–1484 (2005)
23. Vincent, M., Passerini, A., Labb, M., Frasconi, P.: A simplified approach to disulfide connectivity prediction from protein sequences. BMC Bioinformatics 9(20) (2008)

Supervised Classification Methods for Mining Cell Differences as Depicted by Raman Spectroscopy

Petros Xanthopoulos[1], Roberta De Asmundis[2], Mario Rosario Guarracino[2],
Georgios Pyrgiotakis[3], and Panos M. Pardalos[1,4]

[1] Center for Applied Optimization, Department of Industrial and Systems Engineering, University of Florida, Gainesville, FL, USA

[2] High Performance Computing and Networking Institute, National Research Council of Italy, Naples, IT

[3] Particle Engineering Research Center, University of Florida, Gainesville, FL, USA

[4] McKnight Brain Institute, University of Florida, Gainesville, FL, USA

Abstract. Discrimination of different cell types is very important in many medical and biological applications. Existing methodologies are based on cost inefficient technologies or tedious one-by-one empirical examination of the cells. Recently, Raman spectroscopy, a inexpensive and efficient method, has been employed for cell discrimination. Nevertheless, the traditional protocols for analyzing Raman spectra require preprocessing and peak fitting analysis which does not allow simultaneous examination of many spectra. In this paper we examine the applicability of supervised learning algorithms in the cell differentiation problem. Five different methods are presented and tested on two different datasets. Computational results show that machine learning algorithms can be employed in order to automate cell discrimination tasks.*abstract*

Keywords: Raman spectroscopy, Cell discrimination, Supervised classification.

1 Introduction

The discrimination of cells has widespread use in biomedical and biological applications. Cells can undergo different death types (e.g. apoptotic, necrotic), due to the action of a toxic substance or shock. In the case of cancer cell death, the quantity of cells subject to necrotic death, compared with those going through apoptotic death, is an indicator of the treatment effect. Another application of cell discrimination is cell line characterization, that is to confirm the identity and the purity of a group of cells that will be used for an experiment. The standard solution is either to use microarray technologies, or to relay on the knowledge of an expert. In the first case, analysis takes a long time, is subject to errors, and requires specialized equipments [1]. On the other hand, when the analysis is based only on observations, results can be highly subjective and difficult to reproduce.

R. Rizzo and P.J.G. Lisboa (Eds.): CIBB 2010, LNBI 6685, pp. 112–122, 2011.
© Springer-Verlag Berlin Heidelberg 2011

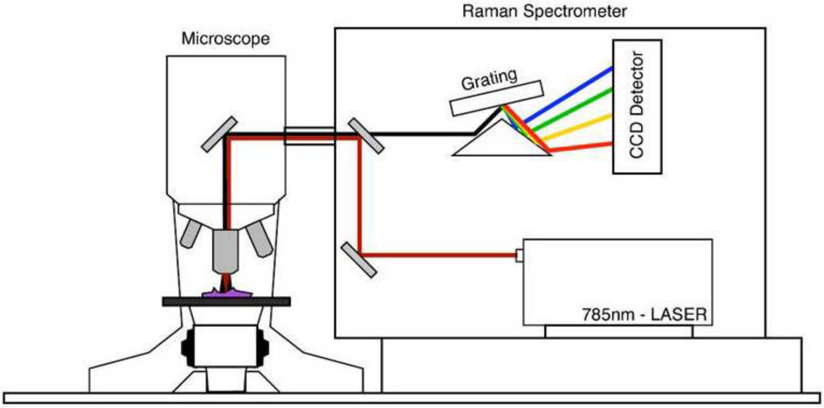


Fig. 1. Pictorial view or Raman spectrometer

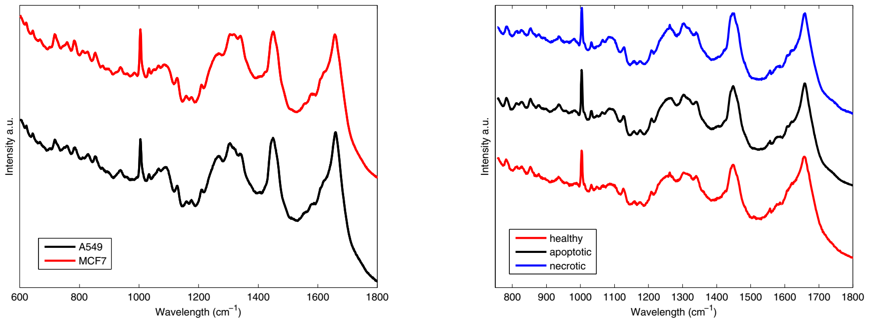
Recently, Raman spectroscopy has been applied to the analysis of cells. This method is based on a diffraction principle, called the Raman shift, that permits to estimate the quantity and quality of enzymes, proteins and DNA present in a single cell. A microscope focuses the laser through the objective lens on the sample and the scattered photons are collected by the same objective lens and travel to the Raman spectrometer, where they are analyzed by a grating and a CCD detector, as depicted in Figure 1.

Since low energy lasers do not deteriorate or kill cells, it is used in vitro and it can be used in vivo. Furthermore, Raman spectra are not affected by changes in water, which makes the results robust with respect to the natural changes in size and shape of cells. Finally, the spectrometer scan accomplish the experiment in less than a minute, and can even be brought outside a biological laboratory, which can make it potentially useful in many other applications, as in the case of biological threat detection in airports or battlefields.

Raman spectrum is usually analyzed to find peaks at specific wavelengths, which reveals the presence and abundance of a specific cell component. This in turn can be used as a biomarker for cell discrimination or cell death classification [2]. This strategy is highly dependent on the experiment and spectroscope tuning, thus giving rise to questions regarding normalization of spectra and peak detection.

In this work, we explore and compare alternative data mining algorithms that can analyze the whole spectrum. These techniques have been successfully applied to other biological and biomedical problems, and and are de facto standard methods for supervised data classification [3,4]. Methods are tested through numerical experiments on real data and their efficiency with respect to their overall classification accuracy is reported.

In this article we use the following notation: all vectors will be column vectors and for a vector $\mathbf{x}$ in the m-dimensional input space $\mathbb{R}^m$, its components will be



(a) , (b)

Fig. 2. The mean Raman spectra for each class: a) cells from A459 and MCF7 cell line and b) A549 cells treated with Etoposide (apoptotic), Triton-X (necrotic) and control cells. All spectra have been normalized so that they have zero mean and unitary standard deviation and then they were shifted for clarity.

denoted as $\mathbf{x}_i$, for $i = 1, \ldots, m$. A column vector of 1s of arbitrary dimension will be denoted by $\mathbf{e}$. Matrices are indicated with capital letters.

The rest of the paper is organized as follows. In Section 2 we discuss about the data sets used to test the different algorithms, in Section 3 we present the computational results and in Section 4 we discuss some further extensions and challenges.

2 Materials

2.1 Dataset

For evaluating the data mining algorithms, we used two different data sets. The first contains cells from two different cell lines: 30 cells from the A549 cell line and 60 from MCF7 cell line. The first are breast cancer cells, whereas the later are cancer epithelia cells. All 90 cells of this class were not treated with any substance. The aim of this experiment is to evaluate the ability of various data mining techniques in discriminating between different cell lines.

The second dataset consists uniquely of A549 cancer epithelial cells. The first 28 cells are untreated cancer cells (control), the next 27 cells were treated with Etoposide and the last 28 cells were treated with Triton-X, so that they undergo apoptotic and necrotic death correspondingly. The detailed protocols followed for the biological experiments were standard and can be found at [5]. The mean spectrum of each class for the two datasets are shown in Fig1 (a & b).

2.2 Raman Spectroscope

The Raman microscope is an InVia system by Renishaw. It consists of a Leica microscope connected to a Renishaw 2000 spectrometer. The high power diode laser (250 mW) produces laser light of 785 nm. Both data sets were acquired by Particle engineering Research Center (P.E.R.C.) at the University of Florida.

2.3 Data Preprocessing

For peak analysis Raman spectra can be preprocessed in many ways. Once they have been acquired by the instrument, the first step consists in subtracting the background noise. This is usually done subtracting to each spectrum the value of a spectrum obtained without the biological sample. Then spectra are normalized subtracting a mean spectrum obtained with a polynomial approximation of fixed order. Other techniques are used to detect peaks and to delete spikes.

In the present work, we only normalized the data along the features of the training set, to obtain features with zero mean and unit variance. Those values of mean and variance are then used to normalize the test spectra.

2.4 Methods

2.5 Support Vector Machines

Support Vector Machines (SVM) [6] are state of the art supervised classification methods, widely used in many application areas. Let us consider a dataset composed of n pairs $(\mathbf{x}_i, y_i)$ where $\mathbf{x}_i \in \mathbb{R}^m$ is the feature vector, and $y_i \in \{-1, 1\}$ is the class label. SVM find a hyperplane $\mathbf{w}^T\mathbf{x} + b = 0$ with the objective to separate the elements belonging to the two different classes. To this extend they determine two parallel hyperplanes $\mathbf{w}^T\mathbf{x} + b = \pm 1$, of maximum distance, leaving all points of the two classes on different sides. Elements with the minimum distance from both classes are called *support vectors* and are the only elements needed to train the classifier. This is equivalent to the solution of the following mathematical program:

$$\begin{array}{ll} \min_{\mathbf{w}\neq 0} & \frac{1}{2}\mathbf{w}^T\mathbf{w} \\ s.t. & \\ & y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1. \end{array} \tag{1}$$

The optimal hyperplane is the solution to the above quadratic linearly constrained problem. The advantage of this method is that a very small number of support vectors are sufficient to define the optimal separating hyperplane. The problem has a solution if exist a line which separates all points of the two classes in different half spaces. When this is not the case, then we need to allow for some errors and let some of the points to be between the two hyperplanes $\mathbf{w}^T\mathbf{x} + b = \pm 1$. For this purpose, we introduce a slack variable ξ_i for each point x_i, and we search for the hyperplane minimizing the following problem:

$$\begin{array}{ll} \min_{\mathbf{w}\neq 0} & \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{i=1}^{N}\xi_i \\ s.t. & \\ & y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 - \xi_i \\ & \xi_i \geq 0 \quad i = 1, \ldots, N \end{array} \tag{2}$$

where C is the capacity constant, $\mathbf{w}$ is again the coefficient vector of the separating hyperplane, and b the intercept.

One generalization for the multiclass case is obtained with a one against all strategy, in which a binary classifier is built for each class against all other points. For each new test point, the class label is assigned voting on the labels assigned by the single classifiers.

2.6 Regularized Generalized Eigenvalue Classifier

Mangasarian et al. [7] proposed to generalize the previous technique and to classify these two classes of points using two non parallel hyperplanes, each the closest to one set of points, and the furthest from the other. We indicate with $A \in \mathbb{R}^{n \times m}$ and $B \in \mathbb{R}^{k \times m}$ the matrices containing the points of the data set, one point of each class on each row. Let $\mathbf{x}^T\mathbf{w} - \gamma = 0$ be a hyperplane in $\mathbb{R}^m$. In order to satisfy the previous condition for all points in A, the hyperplane can be obtained by solving the following optimization problem:

$$\min_{\mathbf{w},\gamma \neq 0} \frac{||A\mathbf{w} - \mathbf{e}\gamma||^2}{||B\mathbf{w} - \mathbf{e}\gamma||^2}. \tag{3}$$

The hyperplane for cases in B can be obtained by minimizing the inverse of the objective function in (3). Now, let

$$\begin{aligned} G &= [A \quad -\mathbf{e}]^T[A \quad -\mathbf{e}], \\ H &= [B \quad -\mathbf{e}]^T[B \quad -\mathbf{e}], \\ \mathbf{z} &= [\mathbf{w}^T \quad \gamma]^T, \end{aligned} \tag{4}$$

where $[A \quad -\mathbf{e}]$ is the matrix obtained from A adding the column vector $-\mathbf{e}$. Using (4), equation (3) becomes:

$$\min_{\mathbf{z} \in \mathbb{R}^m} \frac{\mathbf{z}^T G \mathbf{z}}{\mathbf{z}^T H \mathbf{z}}. \tag{5}$$

The expression in (5) is the Raleigh quotient of the generalized eigenvalue problem $G\mathbf{x} = \lambda H\mathbf{x}$. The stationary points occur at, and only at, the eigenvectors of (5), and the value of the objective function (3) are given by the respective eigenvalues. When H is positive definite, the Raleigh quotient is bounded and it ranges over the interval determined by minimum and maximum eigenvalues [8]. Matrix H is positive definite under the assumption that the columns of $[B \quad -\mathbf{e}]$ are linearly independent. This is actually the case, since the number of features (wavelengths) is much higher than the number of spectra. The inverse of the objective function in (5) has the same eigenvectors and reciprocal eigenvalues. Let $\mathbf{z}_{min} = [\mathbf{w}_1^T \quad \gamma_1]^T$ and $\mathbf{z}_{max} = [w_2^T \quad \gamma_2]^T$ be the eigenvectors related to the eigenvalues of smallest and largest modulo, respectively. Then, $\mathbf{x}^T\mathbf{w}_1 - \gamma_1 = 0$ is the closest hyperplane to the set of points in A and the furthest from those in B, and $\mathbf{x}^T\mathbf{w}_2 - \gamma_2 = 0$ is the closest hyperplane to the set of points in B and the furthest from those in A.

For the multiclass problem, a strategy similar to the one used for SVM is applied. Suppose the problem is to build a classification model for a linearly separable data set, described by two features and divided in three classes Class i, $i = 1, 2, 3$. Following the ReGEC idea, to separate the Class 1 from the other two classes, it is possible to build two hyperplanes $\mathbf{w}_l^T \mathbf{x} - \gamma_l = 0$, $l = 2, 3$. The average of these hyperplanes is then evaluated as the average $\tilde{w}$ of the normal vectors of coefficients $\mathbf{w}_i$. The average hyperplane is obtained computing the principal components of the two normal vectors, and using the solution as the normal vector of the resulting hyperplane.

2.7 k-Nearest Neighboor

The key idea of the algorithm is to assign a new point to the class that belongs to the majority of the k closest neighbors in the training set. This is a majority voting on the class labels for every test point. When $k = 1$, the point is simply assigned to the same class of its closest neighbor. To measure the distance between two points, different distance functions can be used. In our experiments, we decided to use an Euclidean distance, in accordance with the other classification methods, and a fixed value of $k = 3$.

2.8 Linear Discriminant Analysis

Linear Discriminant analysis (LDA) provides an elegant way for classification using discriminant features [9]. We first discuss about the two-class LDA.
LDA's idea is to transform multivariate observations $\mathbf{x}$ to univariate observations $\mathbf{y}$ such that new observations derived from the two classes are separated as much as possible. Let $\mathbf{x}_1, \ldots, \mathbf{x}_p \in \mathbb{R}^m$ be a set of p samples belonging to two different classes A and B. We define the scatter matrices, with respect to A and B, as

$$\begin{array}{l} S_A = \sum_{\mathbf{x} \in A} (\mathbf{x} - \bar{\mathbf{x}}_A)(\mathbf{x} - \bar{\mathbf{x}}_A)^T, \\ S_B = \sum_{\mathbf{x} \in B} (\mathbf{x} - \bar{\mathbf{x}}_B)(\mathbf{x} - \bar{\mathbf{x}}_B)^T \end{array} \tag{6}$$

where $\bar{\mathbf{x}}_A = \frac{1}{p_A} \sum_{\mathbf{x} \in A} \mathbf{x}$ and $\bar{\mathbf{x}}_B = \frac{1}{p_B} \sum_{\mathbf{x} \in B} \mathbf{x}$, and p_A, p_B are the number of samples in A and B respectively. The total intra-class scatter matrix is given by the sum of S_A and S_B:

$$S = S_A + S_B. \tag{7}$$

Beside this, the inter-class scatter matrix is given by

$$S_{AB} = (\bar{\mathbf{x}}_A - \bar{\mathbf{x}}_B)(\bar{\mathbf{x}}_A - \bar{\mathbf{x}}_B)^T. \tag{8}$$

We can find the linear transformation ϕ which minimizes the following ratio, using the Fisher's criterion

$$\Im(\phi) = \frac{|\phi^T S_{AB} \phi|}{|\phi^T S \phi|}. \tag{9}$$

If the matrix S is non singular then Eq. (9) can be solved as a simple eigenvalue problem and ϕ is given by the eigenvectors of matrix $S^{-1}S_{AB}$.

Multi-class LDA is a natural extension of the previous case. Given n classes, we need to redefine the scatter matrices: the intra-class matrix becomes

$$S = S_1 + S_2 + \cdots + S_n \tag{10}$$

while the inter-class scatter matrix is given by

$$S_{1,\ldots,n} = \sum_{i=1}^{n} p_i(\bar{\mathbf{x}}_i - \bar{\mathbf{x}})(\bar{\mathbf{x}}_i - \bar{\mathbf{x}})^T \tag{11}$$

where p_i is the number of samples in the i-th class, $\bar{\mathbf{x}}_i$ is the mean for each class, and $\bar{\mathbf{x}}$ is the total mean vector calculated with

$$\bar{\mathbf{x}} = \frac{1}{p}\sum_{i=1}^{n} p_i\bar{\mathbf{x}}_i.$$

The linear transformation ϕ we wish to find can be obtained by solving the following generalized eigenvalue problem:

$$S_{1,\ldots,n}\phi = \lambda S\phi.$$

Once the transformation ϕ is given, the classification can be performed in the transformed space based on some distance measures d. The class of a new point $\mathbf{z}$ is determined by

$$class(\mathbf{z}) = \arg\min_{n}\{d(\mathbf{z}\phi, \bar{\mathbf{x}}_n\phi)\} \tag{12}$$

where $\bar{\mathbf{x}}_n$ is the centroid of n-th class.

2.9 Software

For the computational experiments Matlab arsenal toolbox was used for LDA, IIS, k-NN [10], whereas for SVM, libsvm was employed [11]. For ReGEC classification, the author's implementation was used [12,13].

2.10 Improved Iterative Scaling

Given a random process which produces, at each time step, some output value y which is a member of the set of possible outputs, IIS [14] computes the probability of the event y influenced by a conditioning information x. In this way we can consider, for example, in a text sequence, the probability $p(y|x)$ of the event that given a word x, the next word will be y. This leads to the following exponential model:

$$p_\Lambda(y|x) = \frac{1}{Z_\Lambda(x)}\exp(\sum_{i=1}^{m}\lambda_i f_i(x,y)), \tag{13}$$

where $f_i(x,y)$ is a binary valued function called *feature function*, $\lambda_i \in \mathbf{R}$ is the Lagrange multiplier corresponding to f_i and $|\lambda_i|$ is a measure of the importance of the feature f_i, $Z_{\Lambda(x)}$ is a normalizing factor and finally we put $\Lambda = \{\lambda_1, \dots, \lambda_m\}$.

Given a joint empirical distribution $\bar{p}(x,y)$, the log-likelihood of $\bar{p}$ according to a conditional model $p_\Lambda(y|x)$, is defined as

$$L_{\overline{(p)}}(\Lambda) = \sum_{x,y} \bar{p}(x,y) \log p_\Lambda(y|x). \tag{14}$$

This can be regarded as a measure of the quality of the model p_Λ. Clearly we have that $L_{\overline{(p)}}(\Lambda) \leq 0$ and $L_{\overline{(p)}}(\Lambda) = 0$ if and only if p_Λ is *perfect* with respect to $\bar{p}$, i.e. $p_\Lambda(y|x) = 1 \Leftrightarrow \bar{p}(x,y) > 0$.

Given the set $\{f_1, \dots, f_m\}$, the exponential form 13 and the distribution $\bar{p}$, IIS solves the maximum likelihood problem computing

$$\Lambda^* = \arg\max_\Lambda L_{\bar{p}}(\Lambda) \in \mathbf{R}^m.$$

3 Results and Discussion

3.1 Classification and Model Selection

We applied the following supervised learning algorithms on both datasets: a) soft margin SVM b) Regularized generalized eigenvalue classification and c) k nearest neighbor classification (k-NN with $k = 3$), d) Linear Discriminant Analysis and e) Improved Iterative Scaling (IIS) classification. No kernel was applied in the classifiers. In particular, for soft margin SVM classifier the parameter C was chosen to be 10 for the first dataset and 100 for the second. For ReGEC the regularization parameter δ was chosen 0.01. The tuning was done through a grid search on the on the parameter space. At every repetition 90% of the samples were used for training and 10% for testing. The average cross validation accuracies are reported on Table 1.

We can see that for both datasets only C-SVM and ReGEC achieve classification higher that 95%. Nearest neighbor classification although it performs

Table 1. Average classification accuracy for hold out cross validation (100 repetitions). With bold is the highest accuracy achieved for each dataset.

	Classification accuracy (%)	
	Cell line discrimination (two class)	Cell death discrimination (three class)
C-SVM	95.33	97.33
ReGEC	96.66	**98.44**
3-NNR	79.22	95.44
IIS	95.67	87.44
LDA	**100**	91.00

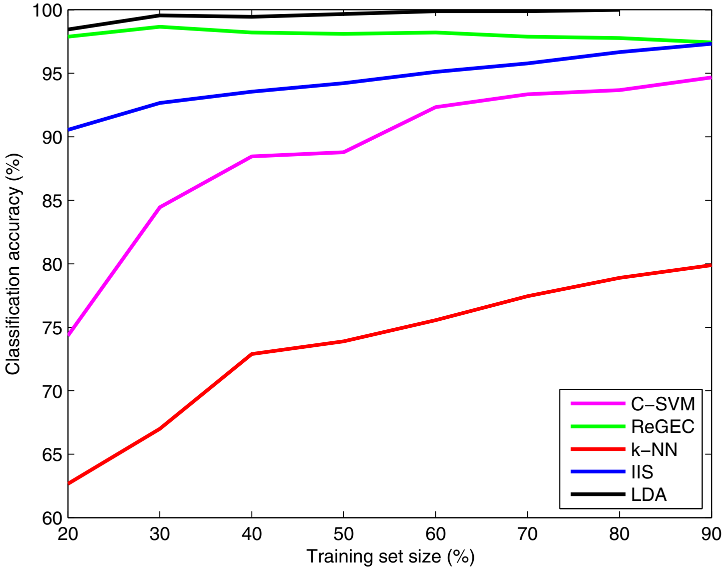


Fig. 3. Classification accuracy versus size of the training set for the binary classification task. Accuracy is evaluated for 100 cross validation runs.

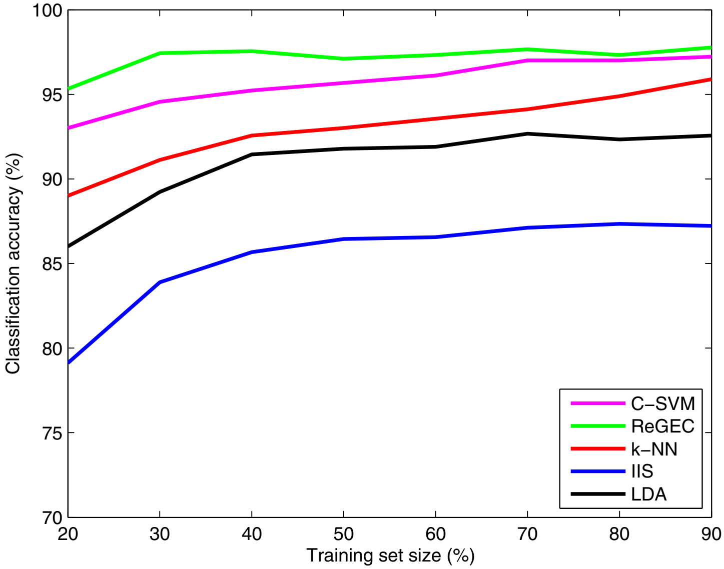


Fig. 4. Classification accuracy versus size of the training set for the multiclass classification task. Accuracy is evaluated for 100 cross validation runs.

very well for the three class problem it has poor results in the two class. This is related to the generic drawback of this method which makes it very sensitive to outliers. Linear Discriminant analysis also achieves high classification results ($> 90\%$ in both cases) justifying its use in the literature [15,16].

3.2 Impact of Training Set Dimension

Next we examined the robustness of each classifier with respect to the size of the training dataset. For this we fixed the training size dataset and we repeated the cross validation process for different sizes of the training dataset. The results were evaluated through hold out cross validation (100 repetitions). Results are shown in Fig 3 & 4. We notice that ReGEC is considerably robust to the size of the training dataset maintaining classification accuracy higher that 95% for all the cases. Overall algorithms demonstrated a smooth performance meaning that the change os the classification accuracy was proportional to the change of the classification dataset size.

4 Concluding Remarks

In this paper we compared the performance of five supervised classification algorithms in two different Raman spectra cell discrimination problems. Another very important aspect of Raman spectra analysis is to determine which area of the spectrum is responsible for the different cell discrimination. Such information will provide more insight in the biological part of cell discrimination since individual groups of features correspond to different cell compounds (DNA, RNA, lipids and proteins). This can be achieved by applying classification strategies in combination with feature selection techniques in order to determine the features that maximize the classification accuracy. Such techniques will boost the analysis and interpretation of Raman spectroscopy and will serve as a research assisting tool for biologist and clinical scientists.

Acknowledgments. Roberta De Asmundis contributed to the present work during a research training period she spent at ICAR-CNR. This work has been partially funded by PRIN 20078MHYS4. In addition, this work was also supported by Airforce grants.

References

1. Powers, K., Brown, S., Krishna, V., Wasdo, S., Moudgil, B., Roberts, S.: Research strategies for safety evaluation of nanomaterials. Part VI. Characterization of nanoscale particles for toxicological evaluation. Toxicological Sciences 90, 296–303 (2006)
2. Bhowmick, T., Pyrgiotakis, G., Finton, K., Suresh, A., Kane, S., Moudgil, B., Bellare, J.: A study of the effect of JB particles on Saccharomyces cerevisiae (yeast) cells by Raman spectroscopy. Journal of Raman Spectroscopy 39, 1859–1868 (2008)

3. Pardalos, P., Boginski, V., Vazacopoulos, A.: Data Mining in Biomedicine. Springer, Heidelberg (2007)
4. Seref, O., Kundakcioglu, O., Pardalos, P.: Data Mining, Systems Analysis, and Optimization in Biomedicine. In: AIP Conference Proceedings (2007)
5. Pyrgiotakis, G., Kundakcioglu, O.E., Finton, K., Pardalos, P.M., Powers, K., Moudgil, B.M.: Cell death discrimination with Raman spectroscopy and Support Vector Machines. Annals of Biomedical Engineering 37, 1464–1473 (2009)
6. Vapnik, V.: The Nature of Statistical Learning Theory. Springer, New York (1995)
7. Mangasarian, O.L., Wild, E.W.: Multisurface proximal Support Vector Machine classification via generalized eigenvalues. IEEE Trans. Pattern Anal. Mach. Intell. 28, 69–74 (2006)
8. Parlett, B.N.: The Symmetric Eigenvalue Problem (Classics in Applied Mathematics). SIAM, Philadelphia (1987)
9. Fisher, R.: The use of multiple measurements in taxonomic problems. Annals of Eugenics 7, 179–188 (1936)
10. Yan, R.: MATLAB Arsenal-A Matlab Package for Classification Algorithms. Informedia, School of Computer Science, Carnegie Mellon University (2006)
11. Chang, C.C., Lin, C.J.: LIBSVM: a library for support vector machines (2001), Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`
12. Guarracino, M.R., Cifarelli, C., Seref, O., Pardalos, P.M.: A classification method based on generalized eigenvalue problems. Optimization Methods and Software 22, 73–81 (2007)
13. Guarracino, M., Cifarelli, C., Seref, O., Pardalos, P.: A parallel classification method for genomic and proteomic problems. In: 20th International Conference on Advanced Information Networking and Applications, AINA 2006, vol. 2, pp. 588–592 (2006)
14. Della Pietra, S., Della Pietra, V., Lafferty, J., Technol, R., Brook, S.: Inducing features of random fields. IEEE Transactions on Pattern Analysis and Machine Intelligence 19, 380–393 (1997)
15. Notingher, I., Green, C., Dyer, C., Perkins, E., Hopkins, N., Lindsay, C., Hench, L.L.: Discrimination between ricin and sulphur mustard toxicity in vitro using Raman spectroscopy. Journal of The Royal Society Interface 1, 79–90 (2004)
16. Owen, C.A., Selvakumaran, J., Notingher, I., Jell, G., Hench, L.L., Stevens, M.M.: In vitro toxicology evaluation of pharmaceuticals using Raman micro-spectroscopy. J. Cell. Biochem. 99, 178–186 (2006)

Use of Biplots and Partial Least Squares Regression in Microarray Data Analysis for Assessing Association between Genes Involved in Different Biological Pathways

Niccoló Bassani[1], Federico Ambrogi[1], Danila Coradini[1], and Elia Biganzoli[1,2]

[1] Section of Medical Statistics and Biometry "'G.A. Maccacaro"' - University of Milan Campus "'Cascina Rosa"', via Vanzetti 5 - 20133 Milano (MI)
niccolo.bassani@unimi.it

[2] Unit of Medical Statistics, Biometry and Bioinformatics - National Cancer Institute, Milan Campus "'Cascina Rosa"', via Vanzetti 5 - 20133 Milano (MI)

Abstract. Microarrays are widely used to study expression profiles for thousand of transcripts simultaneously and to explore inter-relationships between sets of genes.

Visualization techniques and Partial Least Squares (PLS) regression have thus gained relevance in genomic. Biplots provide an aid to understand relationships between genes and samples and among genes, whereas passive projections of variables are helpful for understanding conditional relationships between sets of genes to be quantitatively evaluated via PLS regression.

62 genes involved in loss of cell polarity and 8 involved in Epithelial-Mesenchymal Transition (EMT), were selected from a study on 49 mesothelioma samples, and analysis considered EMT genes as conditioning and polarity genes as conditioned variables. PLS regression results are consistent with the PCA-based biplot of EMT genes and with passive projections of polarity genes.

Future work will address sparsity in PCA and PLS regression. PLS path modeling will be considered after specification of a detailed dependency network.

1 Introduction

Microarrays are a technology developed in the early 90's, that has now become a standard tool for analyzing expression and patterns of co-regulation of thousand of genes simultaneously [1]. Since their first appearance there has been a considerable effort to develop statistical up-to-date methods for analyzing data from genomic experiments, and several well-known techniques such as cluster analysis and Principal Components Analysis (PCA) have been widely applied for evaluating associations between genes and samples and among genes themselves [1,2,3,4]. Graphical instruments such as the PCA-based biplots proposed by Gabriel [5] are now used often in genomic research for exploring structures of association in the data.

R. Rizzo and P.J.G. Lisboa (Eds.): CIBB 2010, LNBI 6685, pp. 123–134, 2011.
© Springer-Verlag Berlin Heidelberg 2011

Another multivariate technique which is now used in several fields of application is Partial Least Squares (PLS). PLS was first introduced by Herman and Svante Wold in chemometrics [6,7], and are now becoming a widespread technique for dealing with genomic data for purposes of regression, classification and survival analysis [8]. In a nutshell, PLS is a dimension reduction approach coupled with a multivariate regression model. Both these features are extremely relevant in the context of microarrays since (1) it is common to have a data matrix containing much more genes than samples, and (2) researchers are often interested in understanding how expression of some genes influences patterns of regulation in other genes.

In recent studies the focus shifted from gene profiling to the investigation of how genes interact with each other. To address this specific issue, we compared PCA-related visualization methods and PLS regression, applying them to a subset of genes from a previously published microarray experiment. We considered only genes involved in two specific pathways, Epithelial-to-Mesenchymal transition (EMT) and loss of cell polarity, evaluating whether expression of EMT-related genes could influence expression profile for polarity genes. Specifically, associations among genes involved in different pathways were visualized by passively projecting the set of *dependent genes* (polarity pathway) on the PCA-based biplot of the set of *independent genes* (EMT pathway). Moreover, the associations were quantitatively evaluated by means of coefficients from a PLS regression.

This is the outline of the paper: in Section 2 we briefly describe mathematical details of both passive projections and PLS framework, in Section 3 application of these methods on a real subset of a microarray experiment is presented. Discussion of results and comparison between methods is discussed in Section 4 and conclusions and future perspectives are the main point of Section 5.

2 Methods

2.1 PCA-Based Biplots and Passive Projection

Principal Components Analysis is an exploratory multivariate technique which allows to reduce high-dimensional data to a lower dimensional space accounting for most of the variability of the original data [9]. This technique offers a great advantage, allowing to visualize in a bi-dimensional space higher-dimensional datasets through a PCA-based biplot.

In particular, given an $n \times p$ matrix $\mathbf{X}$, the goal of PCA is to find $m < p$ uncorrelated linear combinations of the variables which "explain" most of the variation in $\mathbf{X}$. These linear combinations will have the form

$$\alpha_k^{'} x = \alpha_{k1} x_1 + \alpha_{k2} x_2 + \cdots + \alpha_{kp} x_p = \sum_{j=1}^{p} \alpha_{kj} x_j \ . \tag{1}$$

where k indicates the general principal components and j the general variable. If columns of $\mathbf{X}$ have been centered to have zero mean, it can be shown that the α_k vectors of parameters, which we will refer to as loadings, correspond to the eigenvectors of Σ, the sample covariance matrix of $\mathbf{X}$, defined as $\frac{\mathbf{X}^{'}\mathbf{X}}{n}$. The number of components that can be estimated is equal to the minimun between n and p, but in practice only those explaining the most variance will be considered: this translates in relevant reduction of the space of the variables.

Gabriel suggested to use the biplot, a technique which allows to show variables and samples simultaneously on the same plot by means of a suitable rescaling, for visualizing results from such an analysis [5]. Specifically, using the Singular Value Decomposition (SVD) it is possible to write the $\mathbf{X}$ matrix as

$$X = USV^{'} \; . \tag{2}$$

where, $\mathbf{U}$ is a $n \times r$ matrix, $\mathbf{V}$ is a $p \times r$ matrix, both with orthonormal columns, $\mathbf{S}$ is an $r \times r$ diagonal matrix with elements $s_1^{1/2} \geq s_2^{1/2} \geq \cdots \geq s_r^{1/2}$ and r is the rank of matrix $\mathbf{X}$. If we define $\mathbf{S}^{\alpha}$ for $0 \leq \alpha \leq 1$ as the diagonal matrix whose elements are $s_1^{\alpha/2}, s_2^{\alpha/2}, \ldots, s_r^{\alpha/2}$ and similarly for matrix $\mathbf{S}^{1-\alpha}$, and let $\mathbf{G} = \mathbf{U}\mathbf{S}^{\alpha}$, $\mathbf{H}^{'} = \mathbf{S}^{1-\alpha}\mathbf{V}^{'}$, then

$$GH^{'} = US^{\alpha}\, S^{1-\alpha}V^{'} = USV^{'} = X \; . \tag{3}$$

So, the $(i,j) - th$ element of $\mathbf{X}$ can be written as

$$x_{ij} = g_i^{'}h_j = \sum_{k=1}^{r} u_{ik}S_k^{1/2}v_{jk} \; . \tag{4}$$

where $g_i^{'}, i = 1, 2, \ldots, n$ and $h_j^{'}, j = 1, 2, \ldots, p$ are the rows of $\mathbf{G}$ and $\mathbf{H}$, respectively. Equation 4 can be approximated by

$$_m\tilde{x}_{ij} = \sum_{k=1}^{m} u_{ik}s_k^{1/2}v_{jk} = \sum_{k=1}^{m} g_{ik}h_{jk} = g_i^{*}h_j^{*} \; . \tag{5}$$

where g_i^*, h_j^* contain first m elements of g_i and h_j respectively. This means that by plotting g_i^* and h_j^* on the same graphic one can deduce several informations about relationships between variables and subjects and among variables themselves [9]. Since our interest mainly lied in interpreting the biplot here described, no rotation of principal components was performed.

Sometimes one can possibly be interested also in evaluating associations between different sets of genes that are supposed (or known) to be related to one another, that is evaluating how does the expression profile for a defined set of genes influence patterns of expression for a different set of genes measured on the same samples. This task can be addressed graphically by passively projecting the new set of conditioned genes on the PCA-based biplot of the conditioning ones, by exploiting properties of the Singular Value Decomposition described in equation 2. Specifically, since matrix $\mathbf{U}$ from the SVD has orthonormal columns, equation 2 can be written as

$$U^{'}X = U^{'}USV^{'} \Longrightarrow U^{'}X = SV^{'} \; . \tag{6}$$

This means that we can project a new set of variables measured on the same subjects onto the space of the first two principal components of the **X** matrix by computing $\mathbf{U}^{'}\mathbf{Y}$, where **Y** is an $n \times q$ matrix of new q variables. The resulting matrix is an $r \times q$ which contains passive projections of **Y** variables on the space of **X**'s PCs; for matters of comparison, this matrix will have to be properly scaled by $\sqrt{n}$ before plotting, as suggested by Venables and Ripley [10].

2.2 Partial Least Squares Regression

Partial Least Squares regression is a technique which can be very useful to evaluate inter-relations between two distinct large sets of variables, **X** and **Y**. Unlike Principal Component Regression [11], which uses PCs of **X** to predict **Y** thus implicitly assuming that what is relevant to **X** (the principal components) are relevant also to **Y**, PLS regression searches for some latent components that simultaneously decompose **X** and **Y**, with the constraint that these components maximize covariance between **X** and **Y** themselves.

Specifically, **X** and **Y** are decomposed as

$$X = TP^T \quad \text{and} \quad Y = TQ^T \; . \tag{7}$$

where **T** represents the common *score* matrix whose columns represent the latent vectors and **P** and **Q**, by analogy with PCA, are the $p \times r$ and $q \times r$ loadings matrices, with p and q being respectively the number of predictors and response variables, and r the number of components estimated. These loadings represent the association between each variable and each latent component, and are extremely useful for understanding which variables contribute to each of the latent components estimated. To obtain columns of **T** one has to compute a pair of vectors t and u, such that

$$t = Xw \qquad \text{and} \qquad u = Yc \; . \tag{8}$$

where w and c are two sets of weights that create a linear combination of the columns of **X** and **Y** such that their covariance is maximized, that is $t^T u = b$ is maximal. Matrix $\mathbf{Q}^T$ can be factorized as $\mathbf{Q}^T = \mathbf{B}\mathbf{C}^T$, where **B** is a diagonal matrix containing the b values previously described and columns of **C** are the c weights introduced in equation 8. First formula in equation 7 can thus be rewritten as

$$Y = TBC^T = X\left(P^{T+}\right)BC^T = XB_{PLS} \; . \tag{9}$$

where $\mathbf{P}^{T+}$ is the Moore-Penrose pseudo-inverse of $\mathbf{P}^T$. $\mathbf{B}_{PLS}$ are the coefficients of the model, and can then be used to evaluate strength of associations between variables [12]. Moreover, it is possible to graphically evaluate association between dependent and independent variables using graphical displays similar to those used in PCA.

3 Results

3.1 Description of Dataset

The dataset used for analysis is publicly available at the ArrayExpress website, and is composed by 40 human tissues of malignant pleural mesothelioma characterized by different histotype (23 epithelial, 16 mixed and 1 not characterized)and by 9 normal tissues (4 lungs and 5 pleuras) [13]. Of about 22000 genes measured in this study only 70 were taken into account, 62 involved in the loss of cell polarity pathway, and 8 associated with epithelial-to-mesenchymal transition (EMT). Since EMT is supposed to play a role in determining different polarity profiles, our goal is to investigate levels of expression of polarity genes conditionally to the expression of EMT genes.

3.2 Biplots and Passive Projections

Summary of PCA for genes involved in EMT pathway is reported in table 1. It is possible to note that the first two components explain almost half of the total variability. Passively projecting polarity genes, according to equation (6) gives biplot shown in figure 1.

Table 1. Explained variability of Principal Components of EMT

	PC1	PC2	PC3	PC4	PC5	PC6	PC7	PC8
Standard deviation	1.596	1.191	0.994	0.937	0.846	0.769	0.683	0.630
Proportion of Variance	0.319	0.177	0.123	0.110	0.089	0.074	0.058	0.050
Cumulative Proportion	0.319	0.496	0.619	0.729	0.818	0.892	0.950	1.000

From this figure one can possibly note that EMT genes, represented by solid arrows, are negatively associated with most of epithelial mesotheliomas, and that no clear association is seen with mixed mesotheliomas. Moreover, healthy tissues seem to be weakly positively associated with SNAI1, FOXC1 and FOXC2.

As far as gene associations are considered, EMT-related genes SNAI1 and FOXC1 show positive association with polarity genes CDC42, CLDN9, JAM2, CLDN5, CLDN17 and CLDN14, whereas the same EMT genes are negatively associated with another set of polarity genes, reported in table 2.

3.3 PLS Regression

As described before in Section 2, PLS regression aims at finding a set of components decomposing simultaneously both matrix $\mathbf{X}$ of predictors variables (genes) and matrix $\mathbf{Y}$ of responses, so that latent components extracted maximize both the predictors and responses explained variance. However, whereas cumulative explained variance of these components sum to 1 for the predictors, this generally is not true for the dependent variables, and it is not verified in this case neither, as one can see from table 3.

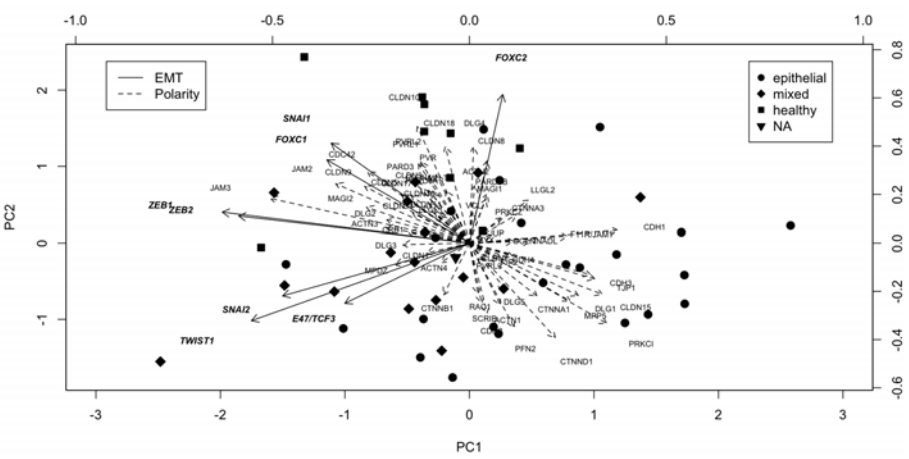


Fig. 1. PCA-based biplot of EMT genes (solid arrows and bold italic font) and passive projections of polarity genes (dashed arrows and plain font). Samples labels have been characterized by different type of points according to the histoype reported in the original dataset. For the line MPM2 no classification was available, so it was labelled as missing (NA).

Table 2. Polarity genes negatively associated with EMT-related genes SNAI1 and FOXC1, conditioning to the association structure depicted from Principal Components Analysis on EMT genes

ACTN1	CTNNA1	PRKCI
CDH2	CTNND1	PVRL3
CDH3	DLG1	RAC1
CDH4	DLG5	SCRIB
CLDN15	MPP5	TJP1
CLDN7	PFN2	TJP2

Using all possible components we are able to explain less than a half of the total response variability, so using all the components extracted can be a good choice. Moreover, the 8^{th} component seems to be the one explaining more variance among all, so it is advisable to retain it in following analysis.

To understand what's the meaning of the latent components estimated, it is useful to look at the matrices of loadings for both the **X** and the **Y**, so to evaluate which associations between **X** or **Y** and the latent components are likely to be the most relevant, and thus which model coefficients should be given more attention. Loadings above a pre-defined threshold of ±0.5 are reported in tables 4 and 5 for both **X** and **Y**. Such a threshold was chosen to focus attention only on strong associations, retaining the most relevant information.

When considering the first latent component, it is possible to see from table 4 that the only predictor gene relevantly correlated with it is ZEB2, which is

Table 3. Explained variability of components extracted with PLS regression for the **Y** matrix of responses

	PC1	PC2	PC3	PC4	PC5	PC6	PC7	PC8
SS Loadings	3.714	1.787	4.335	2.154	2.118	2.605	3.364	4.896
Proportion of Variance	0.060	0.029	0.070	0.035	0.0.034	0.042	0.054	0.079
Cumulative Proportion	0.060	0.089	0.159	0.193	0.228	0270	0.324	0.403

Table 4. PLS regression loadings for matrix X; only loadings larger than ±0.5 are reported

Gene	Comp 1	Comp 2	Comp 3	Comp 4	Comp 5	Comp 6	Comp 7	Comp 8
SNAI1					0.6621			
SNAI2			0.6496					
TWIST1		-0.7048						
ZEB1								0.7951
ZEB2	-0.5518					0.7562		
FOXC1				-0.7373				
FOXC2		0.5758		0.5354	-0.6571			

negatively correlated. From table 5 we can note that CLDN15, PRKCI and CDH3 are all positively associated with the first component. Thus, it is possible to say that ZEB1 is negatively associated with these 3 polarity genes, and this result is consistent with that depicted in figure 1, where it can be seen that ZEB1 lies on opposite side of the plot with respect to CLDN15, CDH3 and PRKCI. Similarly, also SNAI2 (positively associated to the third component) and CLDN15,CDH3 and CDH2 (all negatively related to the third component) are negatively associated in the biplot of figure 1.

Since no polarity gene was relevantly associated with the second component (i.e. no loading was larger than the specified threshold of ±0.5) the column referring to the second component in table 5 is empty, and no consideration on TWIST1 and FOXC2 (both relevantly associated to this component) can be made.

With regard to the fourth component, FOXC1-CDH2 and FOXC2-CLDN7 are actually negatively associated (see figure 1), whereas positive associations between FOXC1-CLDN7 and FOXC2-CDH2 are not graphically evident. It has however to be pointed out that this pattern of association is hardly evident from the biplot because only the first two components were plotted, and these components account only for half of the total variability associated with EMT genes (see table 1). For such reason, degree of concordance between tables 4-5 and figure 1 for higher components is quite low.

According to relevant association described above, the largest absolute coefficient is the one depicting negative association between ZEB2 and CDH3 (see table 6) that lie almost perfectly on opposite sides of the biplot. Similarly, it is possible to see that also PRKCI and CLDN15 are negatively influenced by the

Table 5. PLS regression loadings for matrix Y; only loadings larger than ±0.5 are reported

Gene	Comp 1	Comp 2	Comp 3	Comp 4	Comp 5	Comp 6	Comp 7	Comp 8
CLDN1					-0.6836			
CLDN4								-0.5730
CLDN5						0.7622		
CLDN7				-0.5378				
CLDN8							-0.7461	
CLDN11								-0.7301
CLDN15	0.8526		-1.0144					
CLDN17								0.6425
INADL							0.5466	
PRKCI	0.5568							
ACTN3							-0.5313	
CDH2			-0.7405	0.6743				-0.5798
CDH3	0.7749		-0.8992				0.7924	
JUP								-0.6440
DLG2								-0.5743
DLG4								-0.6292

Table 6. PLS regression relevant coefficients. A coefficient is reported if for any of the 9 components both genes x and y had a loading larger than ±0.5.

	PREDICTORS					
Gene	SNAI1	SNAI2	ZEB1	ZEB2	FOXC1	FOXC2
CLDN1	-0.423					0.190
CLDN4			-0.320			
CLDN5				0.856		
CLDN7					0.394	-0.498
CLDN11			-0.564			
CLDN15		-1.020		-0.648		
CLDN17			0.613			
PRKCI				-0.420		
CDH2		-0.005	-0.598		-0.507	0.226
CDH3		-0.665		-1.134		
JUP			-0.563			
DLG2			-0.374			
DLG4			-0.620			

specific EMT-related gene ZEB2 (coefficients equal to -0.420 and -0.648 respectively), whereas CLDN5 shows a positive association with it (confirmed by the biplot). CDH2, a polarity gene with patterns of conditional polarity expression similar to CDH3, CLDN15 and PRKCI is relevantly influenced by gene ZEB1, with a coefficient of -0.598, and by FOXC1, with a coefficient of -0.507.

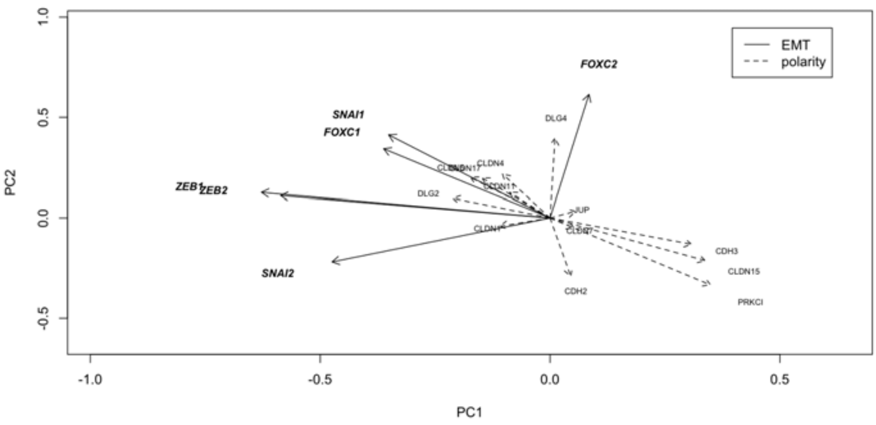


Fig. 2. PCA-based biplot of EMT genes (solid line and italic bold font) and passive projections of polarity genes (dashed lines and plain font) including only genes of table 6

Since biplot of figure 1 is quite overloaded, and for better evaluating concordance between PCA-based biplot and the “filtered” PLS coefficients, a reduced biplot including only those genes showing relevant associations has been drawn (see figure 2). It is of interest to note that the polarity genes that better project over the first principal component of EMT genes, that is CDH3, CLDN15 and PRKCI, are those for which PLS regression “consistently” identifies a negative influence with specific genes EMT-related. On the contrary, if we consider gene ZEB1 it is possible to see that biplot and PLS provide only partially overlapping results. In fact, whereas CLDN4, CLDN11 and DLG2 have a negative PLS coefficient with respect to ZEB1, meaning a negative association possibly related to down-regulation, these genes seem positively associated to ZEB1 in the biplot of figure 1. On the contrary, coefficients for CDH2 and DLG4, genes with a good projection on the second axis, confirm PCA-based visualization results.

Overall, of the 19 coefficients associated to relevant loadings for both predictors and response genes 6 were not confirmed by graphical evaluation via the biplot. Of these, 3 referred to gene ZEB1 (with CLDN4, CLDN11 and DLG2), 2 to gene FOXC2 (with CLDN1 and CDH2) and 1 to gene FOXC1 (with CLDN7).

4 Discussion

Both methods described in section 2 have been used at length in a variety of real-life problems ranging from chemometrics to gene expression [3,7,8], but their complementary use to discover conditional patterns of expression can provide additional useful information when used with genes involved in specific pathways.

The use of PCA-based biplot with the addition of passive projection of additional genes gave us useful insights on the conditional association structure between genes and samples and among genes themselves. The classical PCA-based biplot on EMT genes showed that most of epithelial samples were characterized by negative association with these genes, whereas no clear positive association could be found with both the mixed mesotheliomas (another histotype) and the healthy tissues.

When conditioning to the expression profile of Epithelial to Mesenchymal transition pathway genes, passive projections of cell polarity genes showed that specific EMT-related genes were differently associated with specific sets of polarity-related genes, as discussed in section 3.2.

For quantifying the relationships visualized in figure 1, we resorted to the approach suggested by Datta [12]. The author first suggested a partial least squares approach to circumvent dimensional problems affecting microarray experiments and to identify potential gene relationships requiring further biological investigation.

In this work the method is applied to preselected genes, namely EMT genes, as predictors, and polarity genes, as response. As seen in section 3.3, the associations found after filtering for a cut-off on the loadings of both matrices X (EMT genes) and Y (polarity genes) were mostly confirmed by the passive projections in figure 1, indicating that EMT influences expression profiles for genes polarity-related according to different mesothelioma histotype.

Our dataset contained information only on 8 predictors, for which 8 latent components could be estimated; since our goal was not to reduce dimensionality of the datasets, we retained all these components to explore all of the possible associations between X and Y. As a matter of fact, response genes are relevantly associated with the last component (see table 5), so this choice seems to be the best possible in this specific setting

The use of these methods is particularly useful when attention is focused on set of genes involved in specific pathways for which it is possible to hypothesize a set of dependencies of the kind $X \Rightarrow Y$.

5 Conclusions and Future Work

In the context of microarrays experiment one simultaneously obtains information on thousand of genes involved in different biological pathways. A goal of such experiments is to identify group of genes with similar biological functions, a task which is commonly addressed via cluster analysis. Such methods however can not help in understanding how these genes interact with each other, and how specific patterns of up-and-down regulation for a set of genes are associated with up-and-down regulation in another set.

We applied PCA-related visualization methods and PLS regression to a subset of a microarray experiment considering only genes involved in two specific pathways, EMT and cell polarity, and evaluated whether expression of genes involved in EMT could possibly influence expression profile of cell polarity genes.

PCA-based biplot and passive projections showed the pattern of association between EMT and polarity-related genes; PLS regression mostly confirmed this pattern, providing quantitative information for better understanding the strength of the association.

One of the great advantages of these methods is the simplicity of interpretation: whereas the biplot can be directly interpreted in terms of between-genes correlation and gene-sample association, coefficients of the PLS regression can be thought of as measures of how much predictor gene x influences expression values of gene y.

The main drawback with the graphical procedures presented (PCA-based biplot and passive projection) is that they can actually handle only a limited number of genes and samples, since a too much large dimensionality will result in an onverloaded biplot. This method is thus well suitable to experiment where attention of researcher is focused on specific pathways interest more than on the whole information one can usually obtain in genomic context.

Sparse PCA as described, for instance, by Zou *et al.* [14] and Shen *et al.* [15] will be evaluated as a method for dealing with dimensionality problems relevant to this kind of experiments. Future work will also address the differential pattern of expression according to histotype of subjects in the PLS framework. Moreover, PLS path modeling will be considered after the specification of a detailed dependency network.

References

1. Simon, R.M., Korn, E.L., McShane, L.M., Radmacher, M.D., Wright, G.W., Zhao, Y.: Design and Analysis of DNA Microarray investigations. Springer, New York (2003)
2. Eisen, M.B., Spellman, P.T., Brown, P.O., Botstein, D.: Cluster analysis and display of genome-wide expression patterns. PNAS 95, 14863–14868 (1998)
3. Chapman, S., Schenk, P., Kazan, K., Manners, J.: Using biplots to interpret gene expression in plants. Bioinformatics 18(1), 202–204 (2001)
4. Alter, O., Brown, P.O., Botstein, D.: Singular value decomposition for genome-wide expression data processing and modeling. PNAS 97, 10101–10106 (2000)
5. Gabriel, K.R.: The biplot graphic display of matrices with application to principal component analysis. Biometrika 58(3), 453–467 (1971)
6. Wold, H., Lyttkens, E.: Nonlinear iterative partial least squares (NIPALS) estimation procedures. Bulletin of the International Statistical Institute 43, 29–51 (1969)
7. Wold, S., Sjöström, M., Eriksson, L.: PLS-regression: a basic tool of chemometrics. Chemometr. Intell. Lab. Syst. 58, 109–130 (2001)
8. Boulesteix, A.L., Strimmer, K.: Partial least squares: a versatile tool for the analysis of high-dimensional genomic data. Briefings in Bioinformatics 8(1), 32–44 (2006)
9. Jolliffe, I.T.: Principal Component Analysis, 2nd edn. Springer, New York (2002)
10. Venables, W.N., Ripley, B.D.: Modern Applied Statistics with S. Springer, New York (2002)
11. Kendall, M.G.: A Course in Multivariate Analysis. Griffin's Statistical monographes and courses, London (1957)

12. Datta, S.: Exploring relationships in gene expressions: a partial least squares approach. Gene Expression 9(6), 249–255 (2001)
13. Gordon, G.J., Rockwell, G.N., Jensen, R.V., Rheinwald, J.G., Glickman, J.N., Aronson, J.P., Pottorf, B.J., Nitz, M.D., Richards, W.G., Sugarbaker, D.J., Bueno, R.: Identification of novel candidate oncogenes and tumor suppressors in malignant pleural mesothelioma using large-scale transcriptional profiling. Am. J. Pathol. 166(6), 1827–1840 (2005)
14. Zou, H., Hastie, T., Tibshirani, R.: Sparse Principal Component Analysis. J. Comput. Graph. Stat. 15(2), 265–286 (2006)
15. Shen, H., Huang, J.Z.: Sparse principal component analysis via regularized low rank matrix approximation. J. Multivar. Anal. 99, 1015–1034 (2008)

Qualitative Reasoning on Systematic Gene Perturbation Experiments

Francesco Sambo and Barbara Di Camillo

Department of Information Engineering, University of Padova, Italy
{sambofra,dicamill}@dei.unipd.it

Abstract. Observations of systematic gene perturbation experiments have been proven the most informative for the identification of regulatory relations between genes. For this purpose, we present a novel Qualitative Reasoning approach, based on a qualitative abstraction of DNA-microarray data and on a set of IF-THEN inference rules. Our algorithm exhibits an extremely low rate of false positives, competitive with the state-of-the-art, on both noise-free and noisy simulated data. This, together with the polynomial running time, makes our algorithm an useful tool for systematic gene perturbation experiments, able to identify a subset of the oriented regulatory relations with high reliability and to provide valuable insights on the amount of information conveyed by a set of experiments.

Introduction

Genes of the DNA are the basic blocks which code all the information necessary for an organism to live. A protein coding gene is said to be *expressed* at a particular time instant if its sequence is being transcribed into messenger RNA (mRNA); mRNA is then translated into a protein and some proteins, possibly in combination with each other, have the role of activating or inhibiting the expression of other genes. This self-control mechanism of transcription is known as *gene regulation* and induces a set of causal relations among gene transcripts [4].

Causal relations among genes can be explored by means of the technology of DNA microarray experiments, which allows the simultaneous observation of the rate of transcription of all the genes at particular time instants and under different biological conditions [6]. Identified relations can be represented in a Gene Regulatory Network, a graph in which nodes represent genes and edges represent causal relations between them.

Among all the various types of microarray experiments, steady-state experiments of systematic gene perturbation have been proven the most informative for the purpose of reconstructing Gene Regulatory Networks [8]: a typical experimental design consists in the systematic suppression of each gene (*gene knockout*) followed by a single microarray observation, sampled when the system has reached a steady state. In parallel, a *wild type* experiment is carried out, sampling the steady state of the system when no genes are perturbed.

R. Rizzo and P.J.G. Lisboa (Eds.): CIBB 2010, LNBI 6685, pp. 135–146, 2011.
© Springer-Verlag Berlin Heidelberg 2011

In this paper, we present a novel qualitative reasoning approach to infer regulatory relations between genes from systematic gene perturbation experiments. Our algorithm automates the process of examining the effects of each gene perturbation through a set of inference rules in the form "IF a certain condition is observed in the data, THEN a causal relation between two genes is hypothesized".

The performance of our algorithm is tested on two simulated datasets of systematic gene knock-out experiments, to assess the average behaviour on a rich set of test cases and with complete information on network topologies. The first dataset is generated with the Netsim simulator [2], while the second is extracted from the "Dialogue on Reverse Engineering Assessment Methods" (DREAM4) In Silico Network Challenge [5] [10] [9]: the main goal of the DREAM project is to try to achieve a fair comparison of the strengths and weaknesses of Reverse Engineering methods. On both datasets our approach exhibits an extremely low rate of false positives and provides meaningful insights on the amount of useful information conveyed by steady state perturbations.

The rest of the paper is organized as follows: Section 1 presents the Qualitative Reasoning algorithm, Section 2 describes the two datasets and shows the experimental results and Section 3 draws some conclusions and presents related works and possible future directions.

1 Methods

Figure 1 shows a graphical representation of the measurement of a set of systematic gene knock-out experiments on a simulated network of 20 genes, flanked with the measurement of the wild type experiment (the rightmost column), generated with the NetSim simulator. Rows of the grid correspond to genes and columns to experiments, and the level of gray in each box is proportional to the normalized value of gene expression, from white (no expression) to black (maximum expression). One can observe that, in the majority of the experiments, the expression of most of the genes does not differ much from the wild type: just in a small set of cases a knock-out of a gene has visible effects on a large number of genes. This behaviour is possibly related to the fact that regulatory networks belong to the class of scale-free networks [1], thus exhibiting few highly connected nodes and a large number of loosely connected nodes. Moreover, knocking out genes that reach a low expression value in the wild type experiment has no visible effects on the other genes.

To extract qualitative information contained in a set of knock-out experiments, one can subtract the wild type from all the other experiments and consider the *differentially expressed* genes, *i.e.* the genes for which the absolute value of the difference lies above a threshold θ; for each knock-out experiment, we define these genes as the *observed effects* of the experiment. The result of such an operation on the example dataset is shown in Figure 2, where the effects of the knock-out of each gene are clearly readable in its corresponding column. On top of each column, we reported the number of observed effects for each gene

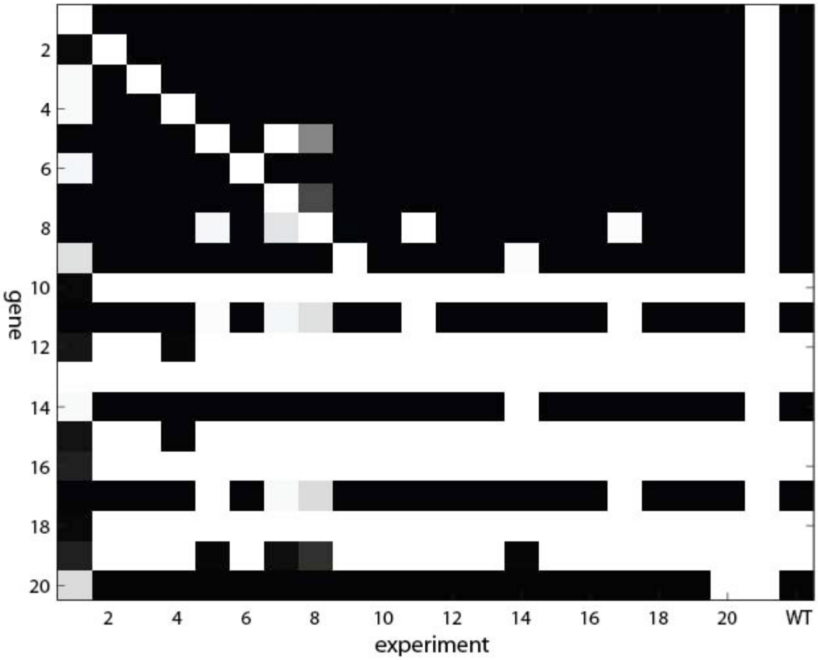


Fig. 1. Graphical representation of a set of 20 gene knock-out experiments, plus the wild type, obtained with the NetSim simulator. Each row corresponds to a gene and each column to an experiment. The level of gray is proportional to the relative expression value of the gene, from white (no expression) to black (maximum expression).

knock-out experiment. Moreover, we define as *not observed* the genes for which the corresponding element on the diagonal exhibits an absolute value smaller than the threshold θ. In Figure 2, the corresponding columns are marked with $\emptyset$. Notice that this is different from having no observed effects, as in the case of genes 2, 3, 6, 9 and 20 in the example. In this work, we chose not to consider the sign of differential expression, *i.e.* we do not distinguish between activatory and inhibitory relations, leaving the task as a possible future extension of our qualitative framework.

With the aim of extracting direct regulatory relations in the form

$$y \Rightarrow x \quad ,$$

where y is one of the regulators of x and x is one of its regulated genes, we take into account for each gene the binary qualitative feature of being or not being an observed effect of a particular knock-out experiment (thus leaving aside the quantitative values of expression). All the observed effects of each gene are mapped to a string representation, as the one in Table 1: each row contains the list of the observed effects of the knock-out of the corresponding gene (or *not observed* if the gene was not observed in the particular experiment). For

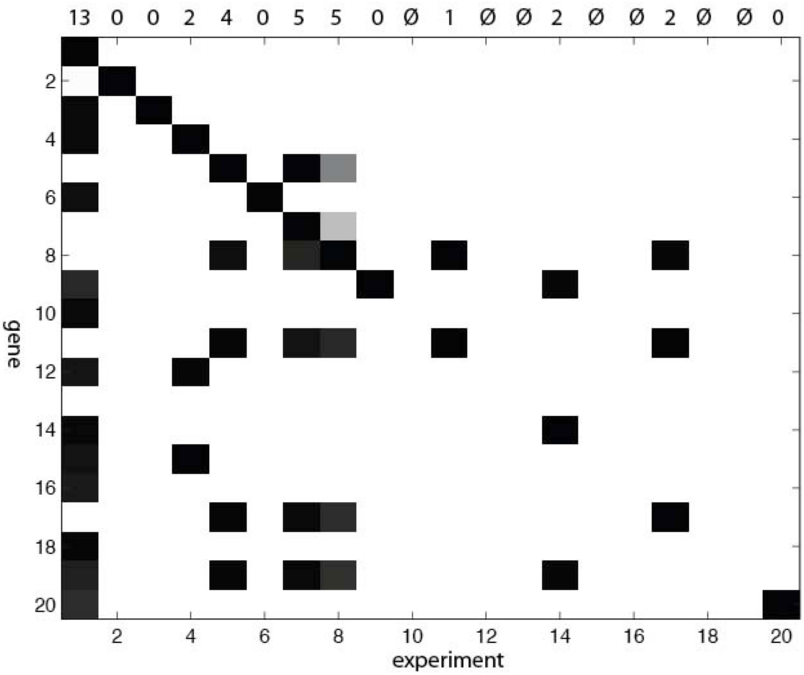


Fig. 2. Graphical representation of the qualitative information contained in the same set of 20 knock-out experiments, obtained by subtracting the wild type from each column and taking absolute values. The number of elements whose value is larger than a fixed threshold θ is reported on top of each column.

convenience, we denote $\mathit{eff}(x)$ the set of observed effects of the knock-out of gene x.

The following considerations can be drawn from the string representation:

- No inference can be carried out on the effects of not observed genes, because the information is not present in the particular set of experiments.
- For knock-out experiments with only one observed effect, a causal relation between the knocked out gene and its observed effect can always be inferred. We name these inferred relations *single effect rules*. Table 1, for example, reveals the causal relation 11 ⇒ 8.
- When more than one observed effect is present for a single knock-out experiment, one has to separate direct effects from indirect effects, *i.e.* effects originated by the propagation of the perturbation through the network: from this idea derive the two following considerations.
- If there exist x and y such that $\mathit{eff}(y) = \{x, \mathit{eff}(x)\}$, the causal relation $y \;\Rightarrow\; x$ can be inferred. The motivation for this rule is the propagation of the perturbation originating from the knock-out of y to all and only the observed effects of the knock-out of x. We name this type of inferred relations *strict inclusion rules*. Two examples from Table 1 are 17 ⇒ 11 and 7 ⇒ 5.

Table 1. String representation of the observed effects for each gene

10:	not observed
12:	not observed
13:	not observed
15:	not observed
16:	not observed
18:	not observed
19:	not observed
2:	
3:	
6:	
9:	
20:	
11:	8
14:	9 19
17:	8 11
4:	12 13 15
5:	8 11 17 19
7:	5 8 11 17 19
8:	5 7 11 17 19
1:	2 3 4 6 9 10 12 13 14 15 16 18 19 20

- If there exist x and y such that $\mathit{eff}(y) = \{x, \mathit{eff}(x), K\}$, with K an additional set of genes, to infer the causal relation $y \Rightarrow x$ one has to exclude that none of the genes in K interposes in the path between y and x (*i.e.* none of them is a direct or indirect cause for x). The latter condition is verified if each $k \in K$ satisfies either of the two following conditions:
  - k is observable and x is not an observable effect of k,
  - there exists a z such that k is an observable effect of z and x is not.

  We name this type of inferred relations *simple inclusion rules*. An example from Table 1 is the rule 5 ⇒ 17: the effect list of gene 5 contains gene 17, the effect list of 17 and gene 19. However, 19 is an observed effect of 14 and 17 is not, thus 19 can not be a cause for 17 and the rule holds.

Strict inclusion and simple inclusion rules have an exception: they cannot be applied to infer the causes of a gene x if there exists at least a gene y such that $\{x, \mathit{eff}(x)\} \equiv \{y, \mathit{eff}(y)\}$. This behaviour is in fact the evidence of the presence in the regulatory network of a non-terminal oriented closed loop to which both x and y belong. For the strict inclusion and simple inclusion rules the two genes are thus indistinguishable in this qualitative framework. An example of genes for which this situation holds in Table 1 is the pair 7 and 8.

A Qualitative Reasoning algorithm can thus be designed to infer single effect rules, simple inclusion rules and strict inclusion rules from a set of systematic gene knock-out experiments; its pseudocode is presented in what follows.

```
QUALITATIVE($\mathbf{A}^{n\times n}, \mathbf{w}^{n\times 1}, \theta$)
 1  Subtract $\mathbf{w}$ from each column of $\mathbf{A}$, and store the absolute value in $\mathbf{D}^{n\times n}$
 2  For each element in $\mathbf{D}$, if $abs(\mathbf{D}[i,j]) < \theta$ then $\mathbf{D}[i,j] = 0$
 3  For each $x$, $\mathit{eff}(x)$ = indices of nonzero elements of the $x$-th column of $\mathbf{D}$.
    // Single effect rules
 4  for $x = 1$ to $n$
 5      if $\mathit{length}(\mathit{eff}(x)) = 1$
 6          $\mathbf{C}[\mathit{eff}(x), x] = 1$
 7  for $l = 1$ to $\underset{x}{\operatorname{argmax}}[\mathit{length}(\mathit{eff}(x)) - 1]$
 8      for all $x \mid \mathit{length}(\mathit{eff}(x)) = l$
            // Strict inclusion rules
 9          if $\exists\, y \mid \mathit{eff}(y) = \{x, \mathit{eff}(x)\}$
10              $\mathbf{C}[x, y] = 1$
                // Simple inclusion rules
11          else if $\exists\, y \mid \mathit{eff}(y) = \{x, \mathit{eff}(x), K\}$
12                  if for each $k \in K$:
13                  $k$ observable and $x \notin \mathit{eff}(k)$
14                  or
15                  $\exists\, z \mid k \in \mathit{eff}(z)$ and $x \notin \mathit{eff}(z)$
16                      $\mathbf{C}[x, y] = 1$
17  return $\mathbf{C}$
```

The algorithm receives as input the squared matrix of knock-out experiments $\mathbf{A}^{n\times n}$, the column vector of the wild type experiment $\mathbf{w}^{n\times 1}$ and the threshold θ; it returns as output the inferred connectivity matrix $\mathbf{C}^{n\times n}$, in which $\mathbf{C}[x,y] = 1$ if the rule $y \Rightarrow x$ was inferred.

Analyzing the computational complexity, one can observe that the preprocessing phase (rows 1–3) take $O\left(n^2\right)$ operations. Searching for single effect rules takes $O\left(n\right)$ operations (rows 4–6); then, the two for loops at rows 7 and 8 scan totally $O\left(n\right)$ elements, searching for simple inclusion rules takes $O\left(n\right)$ and searching for strict inclusion rules can take $O\left(n^3\right)$ in the worst case, *i.e.* if the condition on line 15 has to be verified for every k. Thus, the algorithm has a total worst case complexity of $O\left(n^4\right)$.

The rationale we followed while designing the algorithm was to identify a higly reliable subset of all causal relations. The resulting algorithm is thus rather conservative: it is designed to infer at most one regulatory relation for each line of the string representation, with the gene corresponding to the line as the regulator. Thus, even in the case of all observable genes, no rule can be inferred for nodes wich are leaves in the graph, *i.e.* which have no outgoing edges. Altogether, the maximum possible number of inferred relations is $n - l$, where l is the number of leaves in the graph.

2 Results

To assess the performance of Reverse Engineering algorithms one needs a perfectly known benchmark network. A few regulatory networks are known in the literature with sufficient confidence and, even when some information on a real network is known, it is still impossible to exclude the effect of unknown regulators [8]; moreover, a large set of completely known test instances is needed to robustly test the average behaviour of the algorithm. For these reasons, we chose to rely on simulation to assess the performance of our Qualitative Reasoning algorithm. As perfomance measures, we chose the widely adopted Precision (P) and Recall (R), defined as:

$$P = \frac{tp}{tp + fp}$$

$$R = \frac{tp}{tp + fn}$$

where *tp* is the number of true positives, *i.e.* the number of causal relations correctly identified by the algorithm, *fp* is the number of false positives, *i.e.* the number of relations identified by the algorithm which are not correct, and *fn* is the number of false negatives, *i.e.* the number of relations present in the real network but not identified.

We first run our qualitative inference algorithm on a gene knock-out simulated dataset generated as described in [2] (the NetSim dataset, from now on), composed of four groups of 20 networks each, with network size 10, 20, 50 and 100 nodes, respectively. For each network, we generated noise-free knock-out experiments initializing gene expression at random, fixing to zero the expression of each gene in turn and letting the system evolve to a steady state. For further details on the adopted simulator, please refer to [2].

A separate training set of 4 groups of 5 networks each, of size 10, 20, 50 and 100 nodes, was used to tune the threshold θ, which was then fixed at the 0.5% of the maximum expression value. Boxplots for Precision and Recall on the four test sets of networks are showed in Figure 3.

As it is clear from the figure, Precision is 1 in the vast majority of cases even for large networks, meaning that the number of false positives is extremely low. Average Recall, on the other hand, is low: on average, the method is able to infer approximately the 10% of the real regulatory relations.

We compared the performance of our algorithm with Graphical Gaussian Models (GGMs, [7]), the approach which achieved the best performance on systematic perturbation data in the benchmark paper [8]. GGMs try to estimate the partial correlation between each pair of genes conditioned on all the other genes, combining bootstrap and pseudo-inverse of the correlation matrix, and use it as a proxy of the presence of regulatory relations between gene pairs; correlation is, however, a symmetric measure, thus GGMs provide no information on the direction of the regulation (conversely to our approach). The procedure is implemented in the publicly available R package GeneNet[1].

[1] http://strimmerlab.org/software/genenet/

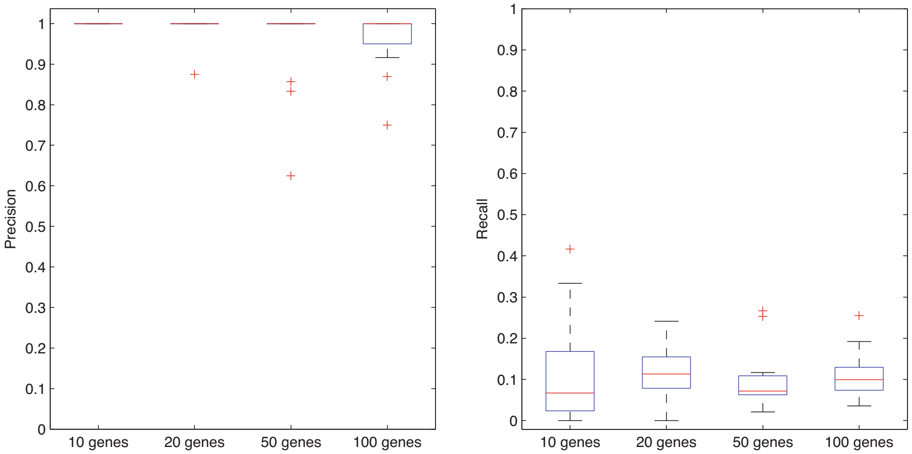


Fig. 3. Boxplots of Precision (left) and Recall (right) of the qualitative inference algorithm, on 20 networks of sizes 10, 20, 50 and 100

The output of the GGMs algorithm is a symmetric matrix reporting confidence levels for regulatory relations between each gene pair. For a comparison with our algorithm, we sorted the returned confidence levels for each network in decreasing order of confidence and computed the Precision of the k topmost regulatory relations, where k is the number of relations identified by our algorithm for the same network; this approach allows us to compare the Precision of the two algorithms at the same level of network complexity (*i.e.* number of edges). Precision of our algorithm is significantly higher than the one of GGMs for networks of sizes 20, 50 and 100, whereas no significant differences are observed for networks of 10 genes (p-values 0.25, 0.016, 0.017 and 0.0023 for networks of sizes 10, 20, 50 and 100, respectively[2]). Our algorithm, thus, when compared to a state-of-the art algorithm exibits equal or higher performance in terms of Precision, while returning also additional information on the direction of regulations.

The two algorithms were then run on a set of simulated knock-out experiments extracted from the DREAM4 In Silico Network Challenge: the current dataset consists of the systematic knock-out of each gene, plus a wild type experiment, for five networks of ten genes. Data, in this case, contain noise both inherent in the dynamical model (a system of stochastic differential equations) and added in a second step to simulate experimental variability.

Given the noisy nature of the dataset and the lack of a separate training set, we fixed a significance level α and computed, for each gene in each experiment, a corresponding confidence threshold θ based on a noise model.

More specifically, we defined the variable

$$\delta_{ij} = x_{ij} - wt_i,$$

[2] Scores were compared using exact Wilcoxon two-sample tests: we considered as significant differences corresponding to a p-value < 0.05.

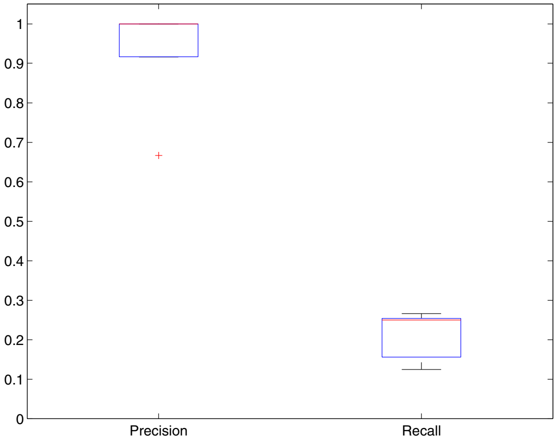


Fig. 4. Boxplots of Precision and Recall of the qualitative inference algorithm, on the 5 networks of size 10 from the DREAM4 In Silico Network Challenge

where x_{ij} is the intensity of the expression of gene x_i measured during the knock-out of gene x_j and wt_i is the intensity of the expression of the same gene measured in the wild type experiment, and we normalized each δ_{ij} by its standard deviation, learned as an intensity dependent parameter [3]. Assuming Gaussian noise, under the Null hypothesis the normalized δ_{ij} is distributed as a standard Gaussian, thus we fixed a significance level α and computed the threshold θ directly from the inverse of the normal cumulative distribution function. The significance level α was set to 0.0005, which corresponds to a significance level of 5%, corrected for the number of tests (10 genes × 10 KO experiments) according to Bonferroni correction for multiple testing.

The results on the DREAM4 data, in terms of Precision and Recall, are shown in Figure 4. As one can observe, the Precision of the qualitative algorithm is rather high even in the presence of noise, with an average of 0.93, and the average Recall, 0.21, is doubled with respect to the one obtained on the previous dataset. No significant differences were observed with the results of GGMs, in terms of Precision at the same level of complexity, but additional information is returned by our algorithm in terms of orientation of the causal relations.

To further assess the robustness of our algorithm to variations of the threshold θ in the noise-free case or, correspondingly, of the significance level α in the noisy case, we observed Precision and Recall of the output of our algorithm at different values of the two parameters for the NetSim and DREAM4 datasets. Figure 5 shows Precision and Recall for the NetSim dataset obtained by directly varying the threshold θ, while Figure 6 shows Precision and Recall for the DREAM4 dataset for different values of the significance level α.

As it is clear from the two figures, Precision and Recall of our algorithm remain in the same range even for variations of the thresholds of several orders of magnitude. Our algorithm can thus be considered robust to che choice of the threshold.

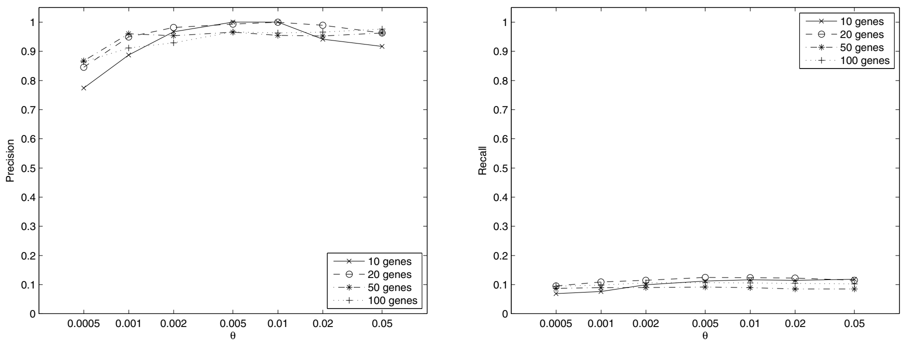


Fig. 5. Average Precision (left) and Recall (right) of the qualitative inference algorithm, on 20 networks of sizes 10, 20, 50 and 100 obtained with the NetSim simulator, for different levels of the threshold θ

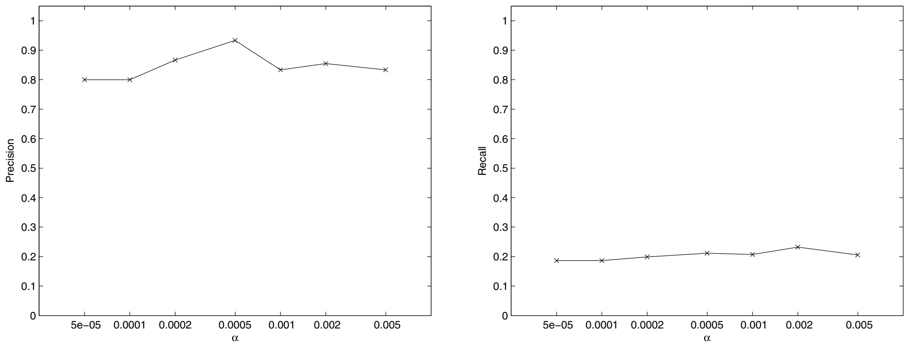


Fig. 6. Average Precision (left) and Recall (right) of the qualitative inference algorithm, on 5 networks of size 10 from the DREAM4 challenge, for different values of the significance level α

The high level of Precision reached by our Qualitative Reasoning algorithm, together with the polynomial running time, makes it a good preprocessing tool for a general inference algorithm, able to provide valuable and reliable information on a subset of the oriented regulatory relations.

3 Discussion

We described in this paper a novel Qualitative Reasoning algorithm for the inference of directed causal relations between genes from steady state experiments of systematic gene perturbation. The algorithm extracts from the data a qualitative description of the observable effects of each perturbation and it is both able to infer three kinds of regulatory rules and to explictly point out which parts of the network are impossible for it to infer, given the outcome of the perturbation experiment. In the example presented in Table 1 it is in fact clear that, being genes 10, 12, 13, 15, 16, 18 and 19 not observed in the particular set of experiments, little information can be gained on their possible regulatory role. Information on unobserved genes is thus valuable and can be exploited when choosing on which genes to focus in possible subsequent experiments.

In the literature, automated processing of steady state perturbation experiments is usually accomplished by means of quantitative methods; an exception is the GenePath software [11], which exploits an IF-THEN rule inference approach to analyze qualitative differences between single and double mutants of the same organism. GenePath, however, is designed to process qualitative variations of phenotypic variables, whereas our approach directly analyzes gene expression data.

As concerns future directions, a first remark can be that the qualitative abstraction of our algorithm is based on a fixed numerical threshold θ, used to select the observed effects of each experiment. A possible future direction would be to make the threshold adaptive to data, relating it to either the expression of the same gene through all the experiments (gene-specific threshold) or to all the genes in each experiment (experiment-specific threshold).

Another possible future direction would be to extend the qualitative framework to consider also the sign of the observed effects of each perturbation, classifying them as overexpressed or underexpressed with respect to the wild type, and to study both how this affects the three rule inference procedures and if new procedures can be defined in the new framework.

Acknowledgments

We would like to thank Professor Silvana Badaloni and Professor Gianna Toffolo of the Department of Information Engineering, University of Padova, for the precious advice and suggestions provided along the research.

References

[1] Albert, R.: Scale-free networks in cell biology. Journal of Cell Science 118, 4947–4957 (2005)

[2] Di Camillo, B., Toffolo, G., Cobelli, C.: A gene network simulator to assess reverse engineering algorithms. Annals of the New York Academy of Sciences 1158(1), 125–142 (2009)

[3] Di Camillo, B., Sanchez-Cabo, F., Toffolo, G., Nair, S.K., Trajanoski, Z., Cobelli, C.: A quantization method based on threshold optimization for microarray short time series. BMC Bioinformatics 6, S11 (2005)

[4] Hunter, L.: Life and its molecules: A brief introduction. AI Magazine - Special issue on AI and Bioinformatics 25(1), 9–22 (2004)

[5] Marbach, D., Schaffter, T., Mattiussi, C., Floreano, D.: Generating Realistic In Silico Gene Networks for Performance Assessment of Reverse Engineering Methods. Journal of Computational Biology 16(2), 229–239 (2009)

[6] Molla, M., Waddell, M., Page, D., Shavlik, J.: Using machine learning to design and interpret gene-expression microarrays. AI Magazine - Special issue on AI and Bioinformatics 25(1), 23–44 (2004)

[7] Schäfer, J., Strimmer, K.: An empirical Bayes approach to inferring large-scale gene association networks. Bioinformatics 21(6), 754–764 (2005)

[8] Soranzo, N., Bianconi, G., Altafini, C.: Comparing association network algorithms for reverse engineering of large-scale gene regulatory networks: synthetic versus real data. Bioinformatics 23(13), 1640–1647 (2007)

[9] Stolovitzky, G., Monroe, D.O.N., Califano, A.: Dialogue on reverse-engineering assessment and methods: The DREAM of high-throughput pathway inference. Annals of the New York Academy of Sciences 1115(1), 1–22 (2007)

[10] Stolovitzky, G., Prill, R.J., Califano, A.: Lessons from the DREAM2 challenges: A community effort to assess biological network inference. Annals of the New York Academy of Sciences 1158(1), 159–195 (2009)

[11] Zupan, B., Bratko, I., Demsar, J., Juvan, P., Curk, T., Borstnik, U., Beck, J.R., Halter, J.A., Kuspa, A., Shaulsky, G.: GenePath: a system for inference of genetic networks and proposal of genetic experiments. Artificial Intelligence in Medicine 29(1-2), 107–130 (2003)

Biclustering by Resampling

Ekaterina Nosova[1,⋆], Roberto Tagliaferri[1],
Francesco Masulli[2], and Stefano Rovetta[2]

[1] Dept. of Mathematics and Informatics, University of Salerno,
84084 Fisciano (Salerno), Italy
[2] DISI, Dept. Computer and Information Sciences, University of Genova,
16146 Genova, Italy

Abstract. The search for similarities in large data sets has a very important role in many scientific fields. It permits to classify several types of data without an explicit information about it. In many cases researchers use analysis methodologies such as clustering to classify data with respect to the patterns and conditions together. But in the last few years new analysis tool such as a biclustering were proposed and applied to the many specific problems. Biclustering algorithms permit not only to classify data with respect to selected conditions, but also to find the conditions that permit to classify data with a better precision. Recently we proposed a biclustering technique based on the Possibilistic Clustering paradigm (PBC algorithm) [1] that is able to find one bicluster at a time. In this paper we propose an improvement to the Possibilistic Biclustering algorithm (PBC Bagging) that permits to find find several biclusters by using the statistical method of Bootstrap aggregation. We applied the algorithm to a synthetic data and to the Yeast dataset, obtaining fast convergence and good quality solutions. A comparison with original PBC method is also presented.

Keywords: data mining, biclustering, clustering, Possibilistic C-Means, bagging, genomic data analysis.

1 Introduction

The simultaneous analysis of the expession of large sets of genes gives a great opportunity of studying genomic information. Data sets are provided, for example, by the DNA Microarray technology, and the results of the experiments carried out on genes under different conditions are the expression levels of their transcribed mRNA stored in DNA chips. Our task is to find a subset of genes that shows similarity under a subset of conditions. Therefore we use the technique of biclustering, that was firstly proposed in 1972 by Hartigan [2]. The method finds the submatrices with the minimum variance. A "perfect bicluster" is a submatrix with zero variance.

⋆ This work is supported by Istituto Nazionale di Alta Matematica Francesco Severi (INdAM) with the scholarship N U 2007/000458 07/09/2007.

R. Rizzo and P.J.G. Lisboa (Eds.): CIBB 2010, LNBI 6685, pp. 147–158, 2011.
© Springer-Verlag Berlin Heidelberg 2011

Cheng and Church [3] gave a more precise definition of bicluster and introduced a measure, the mean squared residue (MSR), that computes the similarity among the expression values within the bicluster. The definition of the Cheng & Church bicluster and the measure of MSR are normally used to date for measuring bicluster quality.

Many further biclustering algorithms are based on the method by Cheng and Church. A different biclustering technique based on Multiobjective Optimization has been employed by Mitra et al. [4]. It used local search strategy for identifying overlapped biclusters in gene expression data.

The Possibilistic Biclustering algorithm, proposed by M. Filippone et al. [1], is based on the Possibilistic Clustering paradigm [5], and finds one bicluster at a time, assigning a membership to the bicluster for each gene and for each condition. The biclustering problem, in which one would maximize the size of the bicluster and minimizing the residual, is faced as the optimization of a proper functional. This algorithm obtains fast convergence and good quality solutions. PBC finds only one bicluster at time. We propose an improved PBC algorithm based on data resampling, specificaly Bootstrap aggregation, for finding all possible biclusters together, including overlapped so lutions.

2 The Biclustering Problem

Let X be the $M \times N$ data matrix and x_{ij} be the i-th input variable of j-th observation (e.g., the expression level of the i-th gene in the j-th condition). A $m \times n$ *bicluster* [3] is a pair $(\mathbf{g}, \mathbf{c})$, where $\mathbf{g} \in \{1, ..., m\}$ is a subset of variables (genes) and $\mathbf{c} \in \{1, ..., n\}$ is a subset of observations (conditions). We are interested in all the largest biclusters with the minimal value of the MSR from the data matrix.

The size (or volume) n of a bicluster is usually defined as the number of cells in the gene expression matrix X belonging to it, that is the product of the cardinalities $n_g = |\mathbf{g}|$ and $n_c = |\mathbf{c}|$:

$$n = n_g \cdot n_c.$$

Let

$$d_{ij}^2 = \frac{(x_{ij} + x_{IJ} - x_{iJ} - x_{Ij})^2}{n}, \tag{1}$$

be the *residual* for data item x_{ij}, where the elements x_{IJ}, x_{iJ} and x_{Ij} are respectively the bicluster mean, the row mean and the column mean of X for the selected genes and conditions:

$$x_{IJ} = \frac{1}{n} \sum_{i \in g} \sum_{j \in c} x_{ij}, \tag{2}$$

$$x_{iJ} = \frac{1}{n_c} \sum_{j \in c} x_{ij}, \tag{3}$$

$$x_{Ij} = \frac{1}{n_g} \sum_{i \in g} x_{ij}. \tag{4}$$

The *mean square residual MSR* measures the bicluster homogeneity:

$$MSR = \frac{1}{n} \sum_{i \in g} \sum_{j \in c} d_{ij}^2.$$

The mean square residual quantifies the mean (square) difference between the actual value of an element x_{ij} and its expected value as predicted from the corresponding row mean, column mean, and bicluster mean.

We want to find large biclusters, optimizing a functional that maximizes the bicluster cardinality n and at the same time minimizes the residual MSR. This is known to be an NP-complete task [6]. The high complexity of this problem has motivated researchers to apply various approximation techniques to generate near optimal solutions. As Filippone et al., we take the approach to combine the criteria in a single objective function.

3 Possibilistic Clustering Paradigm

The possibilistic approach to clustering firstly was proposed by Keller and Krishnapuram [5], [7]. They discussed about the membership function of a cluster (or data point in a fuzzy set) as an absolute, i.e. it is an evaluation of a degree of typicality not depending on the membership values of the same point in other clusters.

Let K be the number of clusters, n_c be the number of conditions in the cluster. Then we consider the membership u_{pq}, $p \in n_c$, $q \in K$ that shows the possibility of every condition p to enter in the cluster q. This approach assumes that the membership function of a data point in a fuzzy set (or cluster) is absolute, i.e. it is an evaluation of a degree of typicality u_{pq} not depending on the membership values of the same point in other clusters. So, following Keller and Krishnapuram we assume:

$$u_{pq} \in [0,1], \forall p, q;$$

$$0 < \sum_{q \in c} u_{pq} < n_c, \forall p;$$

$$\bigvee_{p} u_{pq} > 0, \forall q.$$

The task of the objective function is to find the highest memberships for representative feature points, while unrepresentative points should have low membership in all clusters. In the following function the distance from the features to prototypes is made as low as possible while u_{ij} is as large as possible.

$$J(U,Y) = \sum_{p \in K} \sum_{q \in c} u_{pq} E_{pq}^2 + \sum_{p \in K} \frac{1}{\beta_p} \sum_{p \in c} (u_{pq} \log u_{pq} - u_{pq}),$$

where $E_{pq} = ||k_q - y_p||^2$ is the squared Euclidean distance, and the parameter β_p (that we can term *scale*) depends on the average size of the p-th cluster, and must be assigned before the clustering procedure. Note that $(u_{pq} \log u_{pq} - u_{pq})$ is a monotonically decreasing function in [0,1], similar to $(1 - u_{pq})^m$. Thanks to the regularizing term, points with a high degree of typicality have high u_{pq} values, and points not very representative have low u_{pq} values in all the clusters. Note that if we take $\beta_p \to \infty \; \forall p$ (i.e., the second term of $J_m(U, Y)$ is omitted), we obtain a trivial solution of the minimization of the remaining cost function (i.e., $u_{pq} = 0 \; \forall p, q$), as no probabilistic constraint is assumed.

The pair (U, Y) minimizes J_m, under our constraints only if:

$$u_{pq} = e^{-E_{pq}/\beta_p}, \forall p, q,$$

and

$$y_p = \frac{\sum_{q=1}^{r} x_q u_{pq}}{\sum_{q=1}^{r} u_{pq}}, \forall p.$$

These conditions can be interpreted as formulas for recalculating the membership functions and the cluster centers (Picard iteration technique), as shown, e.g., in [8].

A good initialization of centroids must be performed before applying PCM (using, e.g., Fuzzy C-Means [5], [7], or Capture Effect Neural Network [8]). The PCM works as a refinement algorithm, allowing us to interpret the membership to clusters as cluster typicality degree, moreover PCM shows a high outliers rejection capability as it makes their membership very low.

4 The Possibilistic Approach to Biclustering

In this section following Filippone et al. [8] we represent the concept of biclustering in a fuzzy set theoretical approach. For each bicluster they assign two vectors of membership, one for the rows and one for the columns, denoting them **a** and **b** respectively. Such that if a_i and b_j equal to one(zero) then row i and column j belong(or not) to the bicluster. For an element x_{ij} of X we assign its membership u_{ij} such that:

$$u_{ij} = \text{and}(a_i, b_j).$$

The cardinality of the bicluster is then defined as:

$$n = \sum_i \sum_j u_{ij}.$$

The membership u_{ij} can be obtained like:

$$u_{ij} = a_i b_j, \text{(product)}$$

or

$$u_{ij} = \frac{a_i + b_j}{2}, \text{(average)}.$$

So the equations (1- 4) can be generalized as:

$$d_{ij}^2 = \frac{(x_{ij} + x_{IJ} - x_{iJ} - x_{Ij})^2}{n}, \tag{5}$$

where:

$$x_{IJ} = \frac{\sum_i \sum_j u_{ij} x_{ij}}{\sum_i \sum_j u_{ij}}, \tag{6}$$

$$x_{iJ} = \frac{\sum_j u_{ij} x_{ij}}{\sum_j u_{ij}}, \tag{7}$$

$$x_{Ij} = \frac{\sum_i u_{ij} x_{ij}}{\sum_i u_{ij}}, \tag{8}$$

$$G = \sum_i \sum_j u_{ij} d_{ij}^2. \tag{9}$$

To maximize the bicluster cardinality n and minimize the residual G using the fuzzy possibilistic paradigm Filippone et al. make the following assumptions:

1. one bicluster at a time is considered;
2. the fuzzy memberships a_i and b_j are interpreted as typicality degrees of gene i and condition j with respect to the bicluster;
3. the membership u_{ij} is computed.

All these requirements are fulfilled by minimizing the following functional J_B with respect to **a** and **b**:

$$J_B = \sum_i \sum_j (\frac{a_i + b_j}{2}) d_{ij}^2 + \lambda \sum_i (a_i \ln(a_i) - a_i) + \mu \sum_j (b_j \ln(b_j) - b_j).$$

As in the Possibilistic C-means model, the parameters λ and μ control the size of the bicluster by penalizing too small values of the memberships. Their values can be estimated by simple statistics over the training set, and then possibly hand-tuned, for instanced to incorporate a-priori knowledge.

Setting the derivatives of J_B with respect to the memberships a_i and b_j to zero:

$$\frac{\partial J}{\partial a_i} = \sum_j \frac{d_{ij}^2}{2} + \lambda \ln(a_i) = 0,$$

$$\frac{\partial J}{\partial b_j} = \sum_i \frac{d_{ij}^2}{2} + \mu \ln(b_j) = 0,$$

the following solutions can be obtained:

$$a_i = \exp\left(-\frac{\sum_j d_{ij}^2}{2\lambda} \right)$$

$$b_j = \exp\left(-\frac{\sum_i d_{ij}^2}{2\mu}\right).$$

The Possibilistic Biclustering (PBC) algorithm is the following:

1. Initialize the memberships $\mathbf{a}$ and $\mathbf{b}$
2. Compute d_{ij}^2 for all i, j
3. Update a_i for all i
4. Update b_j for all j
5. if $||\mathbf{a}' - \mathbf{a}|| < \varepsilon$ and $||\mathbf{b}' - \mathbf{b}|| < \varepsilon$ then stop
6. else jump to step 2.

The parameter ε is a threshold controlling the convergence of the algorithm. The memberships initialization can be made randomly or using some a priori information about relevant genes and conditions.

4.1 Bootstrap Aggregating (Bagging)

In this section we follow L. Breiman [9] about the Bootstrap aggregating (Bagging) technique. A learning set L consists of data $(y_n, \mathbf{x}_n), n = 1, ..., N$ where the y's are either class labels or a numerical response. We have a procedure for using this learning set to form a predictor (in our case a bicluster) $\varphi(\mathbf{x}, L)$ - if the input is $\mathbf{x}$ we predict y by $\varphi(\mathbf{x}, L)$. Now, suppose that we have a sequence of learning sets L_k each consisting of N independent observations from the same underlying distribution as L. Our aim is to use the L_k to get a better predictor then the single learning set predictor $\varphi(\mathbf{x}, L)$. The restriction is that we are allowed to work with the sequence of predictors $\varphi(\mathbf{x}, L_k)$.

If y is numerical, an obvious procedure is to replace $\varphi(\mathbf{x}, L)$ by the average of $\varphi(\mathbf{x}, L_k)$ over k. i.e. by $\varphi_A(\mathbf{x}) = E_L\varphi(\mathbf{x}, L)$ where E_L denotes the expectation over L, and the subscript A in φ_A denotes aggregation. If $\varphi(\mathbf{x}, L)$ predicts a class $j \in 1, ..., J$, then one method of aggregating the $\varphi(\mathbf{x}, L_k)$ is by voting.

We have a single learning set L without the luxury of replicates of L. Still, an imitation of the process leading to φ_A can be done. Taking repeated bootstrap samples $L^{(B)}$ from L form a $\varphi(\mathbf{x}, L^{(B)})$. Breiman [9] call this procedure "*bootstrap aggregating*" or bagging.

$L^{(B)}$ forms replicate data sets, each consisting of N cases, drawn at random, but with replacement, from L. Each $(y_n, \mathbf{x}_n)$ may appear repeated some times or not at all in any particular $L^{(B)}$. The $L^{(B)}$ is a replicate data set drawn from the bootstrap distribution approximating the distribution underlying L.

5 Improved Possibilistic Clustering Algorithm

As shown in [1], the PBC algorithm finds the larger bicluster of the data matrix with small MSR, when compared with other methods.

Different runs of the PCB algoritm on the same data matrix find very similar biclusters whith high overlapping.

In order to find further biclusters, in this paper we study the effect of resampling techniques. In particular, we use Bootstrap for generating new versions of a data matrix and after that we apply the PBC model. The new multiple versions of data matrix are obtained by making bootstrap replicates of the biclustering set. In such a way all possible biclusters we can found.

5.1 Applying Bootstrap Aggregating to a PBC Model

Let X be the data matrix $M \times N$ with elements x_{ij}, $i \in M$, $j \in N$. As first step, following the Bootstrap aggregating [9], we create l new data matrices M_{bag}. Every matrix M_{bag} has a random number of column copies from X, such that the dimension of the matrices M_{bag} is $M \times N$.

Then, for every Bagging matrix we apply the PBC algorithm and analyze the result by F, MSR, and the value of enrichment S, that can be seen follow, i.e. the *a-priori* information on the data or the GO term data base information which is useful to identify if some agglomeration of genes in a cluster is significant with respect to a specific annotation [10]. We analyze the biclusters relatively to genes (rows), and consider them as clusters.

5.2 Validating Biclusters

Definition: Let us denote K known a-priori classes (annotations) of the data matrix as C_k, $k = 1, ..., K$ and T biclusters that was found as D_t, $t = 1, ..., T$. The intersection of the a-priori classes and the found biclusters we call *matrix F*, so:

$$F = C \cap D.$$

More precisely, one element q_{tk} of the matrix F shows the number of elements in the intersection of a priori class C_k and found bicluster D_t.

Definition: For each known a-priori class (annotation) i and each found cluster j we define the *enrichment S* as:

$$S_{ij} = \frac{a_{ij}}{a_{ij} + b_{ij}} \frac{A_i + B_i}{A_i}$$

where a_{ij} is the number of positive annotations in the cluster j, b_{ij} is the number of negative annotations in the cluster j, $A_i = \sum_j a_{ij}$ and $B_i = \sum_j b_{ij}$.

6 Results

6.1 The Analysis of the Synthetic Data Matrix

First, we apply our algorithm to the synthetic data matrix X $M \times N$, that consists of 100×50 whose elements values are from 1 to 10 (Fig.1 A) . There are

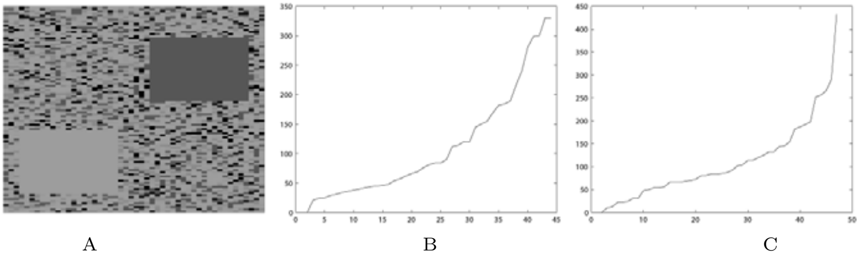


Fig. 1. A) The synthetic data matrix X. B) Number of the elements from A on the ordinate respect to number of the elements from B on the abscissa. C) Number of the elements from B on the ordinate respect to number of the elements from A on the abscissa.

two biclusters A and B of size 30×18 each one. The MSR value of the matrix M is 6.8. We choose the value of the coefficients λ and μ such that:

$$\lambda = \frac{\sum_{i=1}^{N} \lambda_i}{N \times 1.5},$$

$$\mu = \frac{\sum_{i=1}^{N} \mu_i}{N \times 1.3}.$$

The threshold ε is defined as 0.001.

PBC. We apply the PBC method for separating our data. As the result we find one bicluster (49×37) that contains (25×17) elements from the bicluster A and (22×16) elements from B, MSR $= 3.8198$.

PBC Bagging algorithm. We run this algorithm 200 times and find 200 Bagging Matrices M_{bag}; then we apply PBC to these matrices and find the MSR and the value of S for every bicluster. After that we cancel all biclusters that have the MSR value more than 3.39 (half of the MSR of the data matrix X). Then for the remaining biclusters we analyze their matrix F (the first and the second columns of this matrix show how many elements from the biclusters A and B, respectively, enter in the current bicluster). As a result we obtained in many cases the separation to the biclusters as in the first case (PBC).

However, we also obtained separated biclusters A and B. For separating the bicluster A we choose the biclusters with rows that have a value in the first column of F greater than the value of the rows in the second column (size of $A >$ size of B). And *viceversa* for the bicluster B. In Fig.1 B) we show the number of the elements from A on the ordinate respect to number of the elements from B on the abscissa. In Fig.1 C) we show the number of the elements from B on the ordinate respect to number of the elements from A on the abscissa.

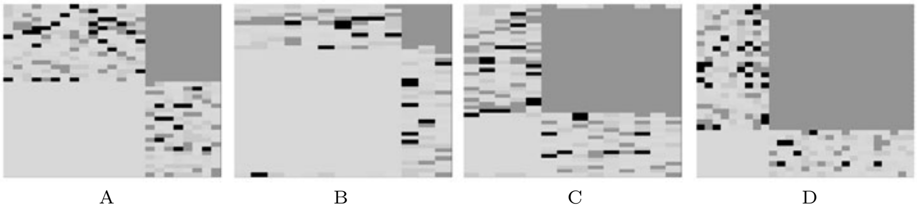


Fig. 2. The Heatmaps of the result of the biclusters A and B separation (cases A and B, C and D respectively)

In the both cases we choose only biclusters that have large size. We can see from the graphics that in the first case the jump of the size values is from 189 to 299 while in the second case the jump is from 182 to 252. So we take all the biclusters with entry size of A greater than 299 in the first case and with entry size of B greater than 252 in the second case. As a result we have:

1. We found two best cases of the separation of the bicluster A (Fig. 2 A and B):
   (a) MSR = 3.2401 size = 920 (40 × 23), 330 elements from A, 156 elements from B;
   (b) MSR = 2.4048 size = 546 (42 × 13), 300 elements from A, 30 elements from B;
2. Two best cases for the separation of the bicluster B (Fig. 2 C and D):
   (a) MSR = 2.9643 size = 644 (46 × 14), 252 elements from B, 75 elements from A;
   (b) MSR = 2.8298 size = 891 (33 × 27), 432 elements from B, 81 elements from A;

6.2 Analysis with the PBC Bagging Method of the Real Data (Yeast)

We consider the real data set Yeast (Fig. 3 A), created by Kenta Nakai, Institute of Molecular and Cellular Biology, Japan, available on: http://archive.ics.uci.edu/ml/datasets/Yeast. This data matrix consists of 8 attributes and 1484 instances, containing the following classes: The matrix has a MSR = 0.0089. There are 10 classes with number of rows of (463, 5, 35, 44, 51, 163, 244, 429, 20, 30 respectively); for the PBC and PBC Bagging analysis we consider the initial conditions:

$$\lambda = \frac{\sum_{i=1}^{N} \lambda_i}{N \times 3},$$

$$\mu = \frac{\sum_{i=1}^{N} \mu_i}{N}.$$

For each of these four cases for PBC Bagging we made 300 runs and built F. We also made an analysis by calculating the enrichment S. For every bicluster we kept the cases with $S \geq 1.1$. We have the follow results (see table 2):

Table 1. Yeast data

class	class definition	number of elements
CYT	(cytosolic or cytoskeletal)	463
NUC	(nuclear)	429
MIT	(mitochondrial)	244
ME3	(membrane protein, no N-terminal signal)	163
ME2	(membrane protein, uncleaved signal)	51
ME1	(membrane protein, cleaved signal)	44
EXC	(extracellular)	37
VAC	(vacuolar)	30
POX	(peroxisomal)	20
ERL	(endoplasmic reticulum lumen)	5

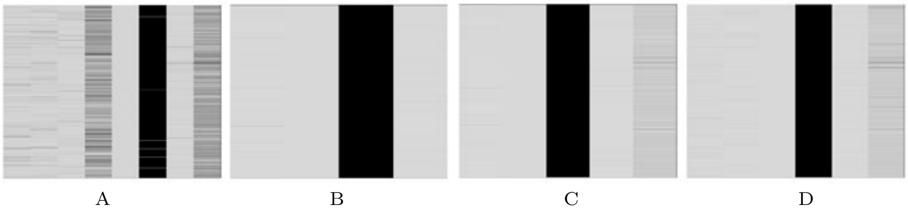


Fig. 3. The Heatmap of data matrix Yeast A) and Heatmaps of some resulting biclusters B)-D)

PBC. MSR = 0.0019, size: 631×6, we found the good separation of the first bicluster.

PBC Bagging. The next three classes were found(results for the average for all the cases):

1. Msr = 0.0024, size: 612×6 - the separation of the bicluster 1.
2. MSR = 0.0029, size: 276×5 - the separation of the Bicluster 6.
3. MSR = 0.0015, size: 269×5 - the separation of the Bicluster 8.

   Together with this results the biclusters that contain some classes together were found. Some of them are:
4. MSR = 0.0028, size: 239×5 - the separation of the Biclusters 1 and 6 together.
5. MSR = 0.0034, size: 622×6 - the separation of the Biclusters 1, 6 and 10 together.

The results for the average of enrichment (e) and the matrix F (in %) can be seen in the table 2. Heatmaps of some biclusters can bee seen in the Fig 3 (C - D).

Table 2. Results of the analysis on Yeast data

Case		1 bic	2 bic	3 bic	4 bic	5 bic	6 bic	7 bic	8 bic	9 bic	10 bic
1.	F	60	0	10	0	15	41	28	44	30	40
	e	1.4	0	0.23	0	0.4	0.96	0.66	1.04	0.7	0.94
2. i.	F	60	0	10	0	10	40	30	40	20	40
	e	1.45	0	0.2	0	0.24	0.96	0.7	0.96	0.5	0.96
2. ii.	F	20	0	10	0	9	35	10	20	10	20
	e	1.07	0	0.53	0	0.5	1.88	0.53	1.07	0.53	1.07
2. iii.	F	20	0	1	0	4	7	7	33	7	8
	e	1.1	0	0.06	0	0.22	0.38	0.38	1.82	0.38	0.44
2. iv.	F	20	0	3	0	5	21	12	17	8	17
	e	1.24	0	0.18	0	0.31	1.3	0.74	1.05	0.49	1.05
2. v.	F	50	0	10	03	15	49	36	45	20	48
	e	1.19	0	0.24	0.007	0.35	1.16	0.85	1.07	0.47	1.15

7 Conclusion

In this paper we presented a new method for the biclustering analysis. Our PBC Bagging algorithm is a very fast algorithm, gives a good separation of the data set with respect to the value of MSR and enrichment and permits to find all the possible biclusters of the desired size (overlapped or not), that can be seen from the results. We decided to calculate the λ and μ values as the mean of the values in the method of Krishnapuram [7], and found a very good separation. Finally, further analysis and biological validation of the obtained results is under study.

References

1. Filippone, M., Masulli, F., Rovetta, S., Mitra, S., Banka, H.: Possibilistic approach to biclustering: An application to oligonucleotide microarray data analysis. In: Priami, C. (ed.) CMSB 2006. LNCS (LNBI), vol. 4210, pp. 312–322. Springer, Heidelberg (2006)
2. Hartigan, J.A.: Direct clustering of a data matrix. Journal of the American Statistical Association 67, 123–129 (1972)
3. Cheng, Y., Church, G.: Biclustering of expression data. In: Proc. Eighth Intl Conf. Intelligent Systems for Molecular Biology (ISMB 2000), pp. 93–103 (2000)
4. Mitra, S., Banka, H.: Multi-objective evolutionary biclustering of gene expression data. Pattern Recognition 39, 2464–2477 (2006)
5. Krishnapuram, R., Keller, J.M.: A possibilistic approach to clustering. IEEE Transactions on Fuzzy Systems 1(2), 98–110 (1993)
6. Peeters, R.: The maximum edge biclique problem is NP-Complete. Discrete Applied Mathematics 131, 651–654 (2003)
7. Krishnapuram, R., Keller, J.: The possibilistic c-means algorithm: insights and recommendations. IEEE Transactions on Fuzzy Systems 4(3), 385–393 (1996)

8. Masulli, F., Schenone, A.: A fuzzy clustering based segmentation system as support to diagnosis in medical imaging. Artificial Intelligence in Medicine 16(2), 129–147 (1999)
9. Breiman, L.: Bagging Predictors. Technical Report No. 421 (1994)
10. Ciaramella, A., Cocozza, S., Iorio, F., Miele, G., Napolitano, F., Pinelli, M., Raiconi, G., Tagliaferri, R.: Clustering, Assessment and Validation: an application to gene expression data. In: Proceedings of International Joint Conference on Neural Networks, pp. 12–17 (2007)

Labeling Negative Examples in Supervised Learning of New Gene Regulatory Connections

Luigi Cerulo[1,2], Vincenzo Paduano[2],
Pietro Zoppoli[2], and Michele Ceccarelli[1,2]

[1] Dept. of Biological and Environmental Studies, University of Sannio
via Port'Arsa, 11 - Benevento, Italy
[2] Biogem s.c.ar.l., Institute of Genetic Research "Gaetano Salvatore"
via Campo Reale - Ariano Irpino (AV), Italy
{lcerulo,ceccarelli}@unisannio.it, {paduano,zoppoli}@biogem.it

Abstract. Supervised learning methods have been recently exploited to learn gene regulatory networks from gene expression data. The basic approach consists into building a binary classifier from feature vectors composed by expression levels of a set of known regulatory connections, available in public databases or known in literature. Such a classifier is then used to predict new unknown connections.

The quality of the training set plays a crucial role in such an inference scheme. In binary classification the training set should be composed of positive and negative examples, but in Biology literature the only collected information is whether two genes interact. Instead, the counterpart information is usually not reported, as Biologists are not aware to state whether two genes are not interacting.

The over presence of topology motifs in currently known gene regulatory networks, such as, feed–forward loops, bi–fan clusters, and single input modules, could drive the selection of reliable negative examples. We introduce, discuss, and evaluate a number of negative selection heuristics that exploits the known gene network topology of *Escherichia coli* and *Saccharomyces cerevisiae*.

Keywords: reverse engineering gene regulatory networks, supervised learning, positive only.

1 Introduction

One of main aims of Molecular Biology is the gain of knowledge about how molecular components interact each other and to understand gene function regulations. Many important biological processes (e.g., cellular differentiation during development, aging, disease aetiology etc.) are very unlikely controlled by a single gene instead by the underlying complex regulatory interactions between thousands of genes within a four-dimensional space. In silico methods revealed promising results allowing the inference of Biological networks from available genomic and

R. Rizzo and P.J.G. Lisboa (Eds.): CIBB 2010, LNBI 6685, pp. 159–173, 2011.

© Springer-Verlag Berlin Heidelberg 2011

post-genomic data. Such networks are usually modeled as directed graphs where nodes represent elements of interactions, eg. genes, proteins, metabolites, and directed edges represent interaction activities between such network components, eg. regulation or inhibition. Approaches proposed in literature falls mainly in the unsupervised category, which are able to extract biological network interactions from experimental data, such as microarray experiments, without any knowledge about the structure of the network to be inferred. Those methods can be distinguished in: i) information theory models, which correlate two genes by means of a correlation coefficient and a threshold, such as ARACNE [1] and CLR [2] that infer the network structure with a statistical score derived from the mutual information and a set of pruning heuristics; ii) boolean network models, which use a binary variable to represent the state of a gene activity and a directed graph, where edges are represented by boolean functions, to represent the interaction between genes (eg. REVEAL [3]); iii) differential and difference equation models, which describe gene expression changes as a function of the expression level of other genes usually with a set of ordinary differential equations (ODE) [4]; and iv) Bayesian models, or more generally graphical models, which make use of Bayes rules and consider gene expressions as random variables [5]. Supervised learning methods have been exploited to reconstruct gene regulatory networks from gene expression data. The reconstruction of a network is modeled as a binary classification problem for each pair of genes. A classifier is trained to recognize the relationships between the activation profiles of gene pairs. The basic principle consists to use the natural inductive reasoning to predict new regulations: if a gene g_1 having expression profile $e(g_1)$ is known to regulate a gene g_2 with expression profile $e(g_2)$, then all other couples of genes g_x and g_y, having expression profiles similar to $e(g_1)$ and $e(g_2)$ are likely to interact. Expression profiles play the role of feature vectors in the machine learning algorithm, while the output is a binary variable representing whether two genes interact or not. Similarly, the prediction of protein–protein interaction [6,7] and metabolic networks [8] make use of a feature vector built upon the sequence representation of proteine and metabolites. A large variety of machine learning algorithms have been proposed in literature and are available as working tools [9]. In the context of gene regulatory networks a first attempt has been made with Bayesian Networks, Linear Regression, Decision Trees, and Support Vector Machines (SVM) [10]. Among all the Support Vector Machine algorithm have attracted the attention of the bio-informatics community. SIRENE [11] is currently the state-of-the-art method for the reconstruction of gene regulatory networks with a Support Vector Machine algorithm, that have reported promising results in the inference of new regulatory connection of *Escherichia coli* genes. Compared to unsupervised methods for gene network inference, supervised methods are potentially more accurate, but for training they need a complete set of known regulatory connections. The need to know some regulations is not a serious restriction as many regulations are progressively discovered. Public regulatory databases are continuously upgraded with new discovered interactions and shared among researchers: RegulonDB[1], KEGG[2], TRRD[3], Transfac[4], IPA[5].

However, the supervised approach raises some open questions. In particular, although known regulatory connections can safely be assumed to be positive training examples, obtaining negative examples is not straightforward, because definite knowledge is typically not available that a given pair of genes do not interact. In fact, the only available information are a partial set of gene regulations, i.e. positive examples, and unlabeled data which could include both positive and negative examples. Recently, in the data mining literature some methods capable of learning a classifier from only positive and unlabeled examples appeared. A class of approaches does not need labeled negative examples [12]. Such approach has been adopted in the context of gene regulatory networks by some author of this paper in a recent paper [13].

In this paper we adopt a method that depends on a starting selection of reliable negatives examples [14,15] which is used to iteratively refine the prediction of a binary classifier. The heuristic adopted to build such a starting negative selection is crucial for the final classification performance. The work stem from an initial attempt made by some of the authors of this paper [16]. In particular we introduce an heuristics that could drive the selection of reliable negative examples by exploiting the over presence of topology motifs in currently known gene regulatory networks, such as, feed–forward loops, bi–fan clusters, and single input modules. Network motifs are small connected subnetworks that a network exhibits in significantly higher occurrences than would be expected for a random connected network. Recently gathered attention especially in biological context [17]. To test our approach we considered the known regulatory networks of *Escherichia coli* and *Saccharomyces cerevisiae* (Yeast) and a set of random microarray experiment artificially generated.

The paper is organized as follows. Section 2 introduces the heuristics to select negative examples from unlabeled gene interaction networks. Section 3 introduces the research questions aimed at evaluating the performance of the proposed heuristic and outlines the process followed to answer such questions. Section 4 describes the context where the process is applied, reports and discusses the results obtained. Finally, Section 5 concludes the paper and outlines directions for future work.

2 Approach

A gene interaction network can be modeled as a directed graph $< G, E >$ where G represents the set of genes, i.e. nodes of the graph, and E represents the set of directed interactions between genes, i.e. edges of the graph. Let $P \subseteq E$ be

[1] http://regulondb.ccg.unam.mx

[2] http://www.genome.jp/kegg/

[3] http://wwwmgs.bionet.nsc.ru/mgs/gnw

[4] http://www.gene-regulation.com

[5] http://www.ingenuity.com

the known gene–gene interactions, $Q = E - P$ the unknown regulatory links, and $N = Complement(E)$ the edges not contained in E. The unknown gene regulatory connections Q can be inferred by a machine learning scheme trained with the set of known regulatory connections. Precisely, P is the set of known positive examples, N is the set of all unknown negative examples and Q is the set of unknown positive examples. The set $N \cup Q$ is also known as the unlabeled set.

A binary classifier should be trained with both positive and negative examples in order to work properly. Anyway, in the context of gene regulatory networks, it is more likely to have knowledge about positive examples (i.e. two interacting genes) rather than the negative ones (i.e. two genes that *do not* interact). Public databases of known regulatory interactions usually report only the first information. An approach to overcome such an issue with ordinary classifier is to select, from the unlabeled set $N \cup Q$ of unknown connections, a sub set of reliable negative examples S which should be as much as possible composed of negative examples, i.e. $S \simeq N$ and $S \cap Q \simeq \oslash$.

The approach proposed in this paper tries to overcome such a limitation by exploiting the known topology of the network model in order to limit the presence of positive examples in the selected set of negatives. In particular, we make use of the over presence of *network motifs* such as, feed-forward loops, bi-fan clusters, and single input modules, to drive the selection of reliable negative examples from the set of unknown interactions. Knowing the over presence of network motifs could enhance the task of negative selection, because they can successfully be used to lower the probability of picking up unknown positives, improving the classification learning process [16].

Network motifs are small connected subnetworks that a network exhibits in significantly higher occurrences than would be expected just by chance in a network with the same number of edges [18]. Recently, they have gathered much attention from the bioinformatics community as a tool to uncover structural patterns of complex biological networks [17]. The analysis of network motifs has led to interesting results, e.g. in the areas of protein-protein interaction prediction [19], hierarchical network decomposition [20] and the analysis of temporal gene expression patterns [21]. An over presence of a network motif means a higher significance in the functionality of that motif in the system represented by the examined network.

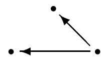
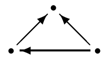
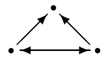
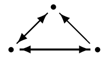
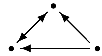
Finding network motifs is a computationally expensive task. We adopted the results of 3 nodes subgraphs reported in [22,23], shown in Table 1. Those motifs have been extracted from the known gene regulatory networks collected in the following databases: RegulonDB[6] for *Escherichia coli* and YoungLab[7] for *Saccharomyces cerevisiae*.

For each pattern the frequency in percentage, the p-value and the Z-score are shown. A positive Z-score means that the network motif is more recurrent than in a random connected network. A null Z-score means that the motif is recurrent

[6] http://regulondb.ccg.unam.mx

[7] http://web.wi.mit.edu/young/regulator_network

Table 1. Patterns and motifs with 3 nodes in the gene networks of *E. coli* and *S. cerevisiae* as reported in [22,23]

	E. coli		S. cerevisiae	
Motif	Z-score	Freq.	Z-score	Freq.
	20.343	97.467%	16.918	93.82%
	13.295	0.318%	10.827	0.298%
	14.401	0.105%	27.202	0.032%
	2.058	<0.001%	4.233	<0.001%
	4.533	0.004%	4.068	<0.001%

as in the random case, while a negative Z-score means a lower recurrence. The table shows only the most recurrent motifs with a p-value < 0.05 in at least one organism. The frequency shows which of them appeared more frequently.

We aim to compare three negative selection heuristics that are based on the known topology of the network. The rationale behind such heuristics is that gene connected as the most frequent motifs are unlikely to have other connections. Those connections could then be considered as negative examples. The following section refines this assertion.

2.1 Candidate Negative Selection Heuristics

Heuristic 0: *Random (RANDOM).* This is the most straightforward heuristic, currently adopted in literature [11]. The selected set, S_{RANDOM}, consists of a random selection without replacement from the unlabeled set $Q \cup N$ of a number of negative examples candidate.

Heuristic 1: *Complement of 3 gene Motifs (MOTIF).* This heuristic is based on the assumption that the most recurrent network motif is an important structure in the system represented by the graph. Let's call the most recurrent 3 nodes motifs over represented in a gene regulatory network as: $\{M_0^{(3)}, M_1^{(3)}, \ldots, M_n^{(3)}\}$. If we find in a regulatory network that 3 genes interacts in a way that their graph representation matches one of those 3 gene motifs $M_i^{(3)}$, *then*, it is unlikely that they are going to form different interactions because of still unknown connections between them. Of course such an assumption could be wrong, but, as shown in the empirical results, the probability that it can occur is low. Thus, we can assume that all the connections not present in $M_i^{(3)}$ are negatives examples.

From such an assumption we define the selected set, S_{MOTIF}, as follows: for each triples of genes, g_1, g_2, and g_3 belonging to the known regulatory network of an organism, let T_{g_1,g_2,g_3} be the sub network composed by those genes. If such a sub network matches one of the most recurrent motif, $\{M_0^{(3)}, M_1^{(3)}, \ldots, M_n^{(3)}\}$, then the selected set of negative examples, S_{MOTIF}, is populated with the connections of $Complement(T_{g_1,g_2,g_3})$.

Heuristic 2: *Transitive Closure (TRANS).* This heuristic has been introduced in [16] but not empirically evaluated. It is built over the assumption that a regulatory network has no or few cycles and that it has a tree like structure. This leads to an heuristic that selects as candidate negatives those given by the union of the transitive closure of the known network and its transpose. Figure 1 summarizes such an heuristic as:

$$S_{TRANS} = TC(P) \cup Transpose(TC(P)) \cup Transpose(P)$$

where $TC(P)$ is the transitive closure of P, i.e. the graph composed by the same nodes of P and the set of edges (g_i,g_j) such that there is a non-null path from g_i to g_j in P; while, $Transpose(X)$ is the graph containing the edges of X reversed.

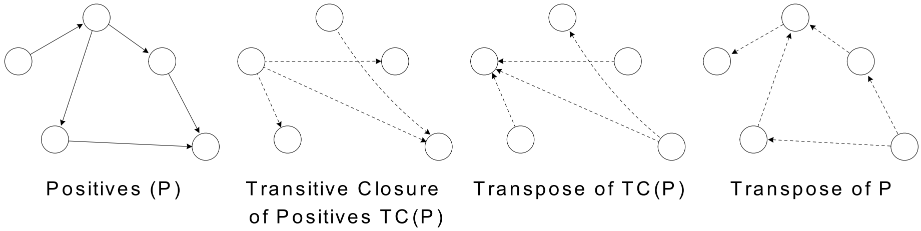


Fig. 1. Transitive closure heuristic example [16]

3 Methods

In this section we introduce the research questions we aim at answering and the methods we followed to pursue such an aim. The main goal is to evaluate, by means of benchmark experiments, the performance of the proposed negative selection heuristics and how they can improve the performance of a Support Vector Machine classifier trained to infer new regulatory connections in synthetic datasets generated over the known gene regulatory networks of *Escherichia Coli* and *Saccharomyces cerevisiae.*

- **RQ1:** *How does the precision/recall of positives, and the precision/recall of negatives of the selected set S, vary with the adopted heuristic and with the percentage of known positives?* In particular, this research question aims to measure the quality of the selected set S built with different heuristics and when the percentage of known positives varies. A high quality of the selected set S depends directly on the precision/recall of negatives and inversely on the precision/recall of positives.

– **RQ2:** *Which is the performance of a classifier trained with the selected set S and the set of known positives?* Specifically, it investigates whether the reliable negative selection heuristics, introduced in Section 2.1, improve the training of a classifier in terms of accuracy of prediction. For such a purpose an SVM (Support Vector Machine) [24] classifier is used.

The quality measures, the datasets used, and the benchmark process to answer the above mentioned research questions are introduced in the following. To perform an assessment a gold standard of the regulatory network is necessary. Simulated regulatory networks are widely used to test gene network inference algorithms as the complete set of gene–gene interactions is available. The process to answer both *RQ1* and *RQ2* consists of the following five steps:

3.1 Random Generation of a Gene–Gene Regulatory Network of G Genes

We generated simulated data with *GeneNetWeaver*[8], a tool used to generate in silico benchmarks in the DREAM3 challenge initiative [25,26]. The *GeneNetWeaver* tool is able to obtain network topologies of a given size G by extracting randomly sub-networks from the gene-to-gene interaction network of *Escherichia Coli* and *Saccharomyces cerevisiae.*

We generated for both *Escherichia Coli* and *Saccharomyces cerevisiae* ten random gene interactions networks composed by 10, 50, and 100 number of genes.

3.2 Generation of Synthetic Microarray Experiments

For each random network of size G, with the *GeneNetWeaver* tool we generated steady state levels for the wild-type and the null–mutant knock-down strains for each gene. This means that for a network of G genes there are $G+1$ experiments (wild-type and knock-down of every gene) leading to a feature vector composed of $2\times(G+1)$ attributes. The data corresponds to noisy measurements mRNA levels which have been normalized such that the maximum value in a given dataset is one. Auto-regulatory interactions were removed, i.e. no self-interactions are considered in the networks. As reported in the DREAM3 documentation, the tool takes great care to generate both network structure and dynamics that are biologically plausible.

3.3 Random Selection of P Non–self Interactions Which Are Assumed to be Known

This leads to a set Q of non–self interactions and a set N of all non–interactions, both assumed to be unknown. The fraction of P with respect to Q is assumed to vary as: $F = \frac{|P|}{|P\cup Q|} \in \{0.1, 0.2, \ldots, 1.0\}$. In a learning scheme, P is the set of

[8] http://gnw.sourceforge.net

labeled, and positive, examples, and $Q \cup N$ is the set of unlabeled examples. The main goal is to select reliable negative examples from $Q \cup N$ in order to train a classifier with both positive and negative examples. For each network of size G, this step is repeated among ten random selection of P positives.

3.4 Selection of Candidate Negative Examples

Let S_H be the set of potentially negative examples, selected with a negative selection heuristic, $H \in \{RANDOM, MOTIF, TRANS\}$, from the unlabeled set $Q \cup N$. To measure the quality of the selected set, and then answer **RQ1**, we used the metrics proposed in [16]:

$$\begin{aligned}
Precision\ of\ Negatives(S_H) &= \frac{|S_H \cap N|}{|S_H|} \\
Precision\ of\ Positives(S_H) &= \frac{|S_H \cap Q|}{|S_H|} \\
Recall\ of\ Negatives(S_H) &= \frac{|S_H \cap N|}{|N|} \\
Recall\ of\ Positives(S_H) &= \frac{|S_H \cap Q_F|}{|Q_H|}
\end{aligned}$$

which in the case of a random selection heuristic becomes:

$$\begin{aligned}
Precision\ of\ Negatives(S_{RND}) &= \frac{|N|}{|N|+|Q|} = \frac{1-\rho}{1-\rho F} \\
Precision\ of\ Positives(S_{RND}) &= \frac{|Q|}{|Q|+|N|} = \frac{\rho(1-F)}{1-\rho F} \\
Recall\ of\ Negatives(S_{RND}) &= \frac{|S_{RND} \cap N|}{|N|} = \frac{|S_{RND}|}{|Q|+|N|} \\
Recall\ of\ Positives(S_{RND}) &= \frac{|S_{RND} \cap Q|}{|Q|} = \frac{|S_{RND}|}{|Q|+|N|}
\end{aligned}$$

where $\rho = \frac{|P|+|Q|}{|P|+|Q|+|N|}$ is the percentage fraction of positive examples in the network. It can be noticed that such precision/recall quantities depend only from ρ, F and the size of S_{rnd}, therefore we consider a random selection heuristic as reference limit. It is important to specify that a random selection from a set assumes that each element of the set have the same probability to be chosen. A new selection heuristic should have a precision/recall of negatives higher and a precision/recall of positives lower that those exhibited by a random selection heuristic.

3.5 Cross Validation of Classification Performance.

The validation consists of a ten-fold cross validation and proceeds as follow. Partition P, Q, and N randomly into ten subsets each of roughly the same size (P_1, Q_1, N_1), ..., (P_{10}, Q_{10}, N_{10}). For each i-th partition a trial is performed with one subset reserved for testing (P_i, Q_i, N_i), while the other nine subsets for building the training set of the classifier. The training set is composed by the set of known positive data, $P_{train} = \bigcup_{k \neq i} P_k$, and the set of candidate negative examples, $S_H \cap \bigcup_{k \neq i} Q_k \cup N_k$, selected with the heuristic $H \in \{RANDOM, MOTIF, TRANS\}$. The i-th trial yields a confusion matrix where TP_i and TN_i are, respectively, the number of positives and negatives correctly predicted by the classifier in the i-th trial; whereas FP_i and FN_i are, the number of false positives and false negatives in the i-th trial. The *Precision* (PR_i) of positives, i.e. Positive Predictive Value, and the *Recall* of positives (RC_i), i.e. Sensitivity, of the i-th trial are computed as:

$$PR_i = \frac{TP_i}{TP_i + FP_i}; \quad RC_i = \frac{TP_i}{TP_i + FN_i}$$

The average indexes are computed among the ten trials as: $PR = \sum^{10} PR_i/10$ and $RC = \sum^{10} RC_i/10$. To measure the effectiveness of a classifier, and then answer **RQ2**, we considered the weighted harmonic mean of precision and recall, i.e. *F-measure*, as a measure that combines Precision and Recall:

$$FM = \frac{2 \cdot PR \cdot RC}{PR + RC}$$

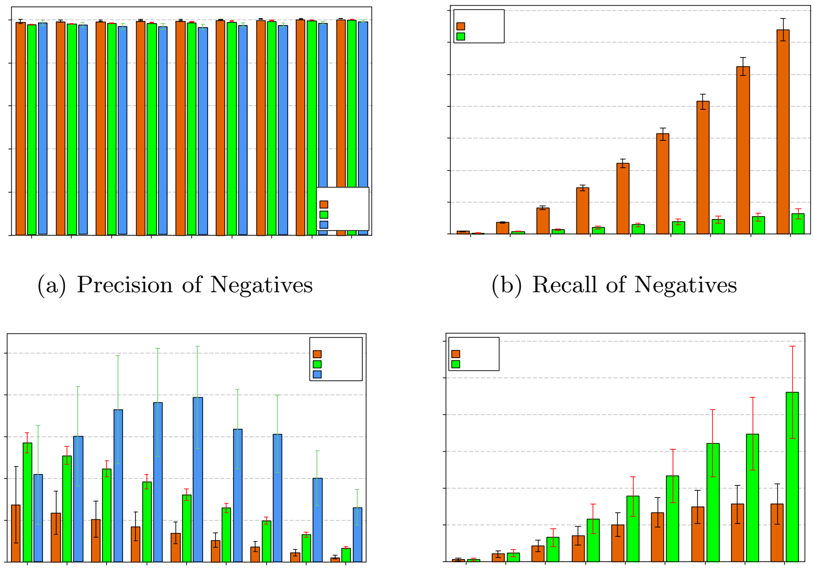
The number of observations for a network of size G and for a selection of P positives is $10 \times 10 = 100$.

3.6 Learning Scheme

We used the SVM implementation provided by LIBSVM, one of the most popular available tool [27]. The basic element of an SVM algorithm is a kernel function $K(x_1, x_2)$, where x_1 and x_2 are feature vectors of two objects to be classified. In our case an object to be classified is a couple of genes, (A,B), represented with a feature vector composed by the concatenation of $e(A)$ and $e(B)$, i.e., $(e(A), e(B)) \in \mathbb{R}^{2n}$, the n-dimensional vectors of expression levels, standardized to zero mean and unit standard deviation, respectively of gene A and B. The idea is to construct an optimal hyperplane between two classes, +1 and -1, such that the distance of the hyperplane to the point closest to it is maximized. The kernel function implicitly map the original data into some high dimensional feature space, in which the optimal hyperplane can be found. A couple of genes, (A, B), classified as +1 means that gene A regulates gene B, instead, classified as -1 means that gene A does not regulate gene B.

4 Discussion

In this section we discuss the empirical results answering *RQ1* and *RQ2*. Due to space limitation only the most significant results are shown, the complete empirical results will be available as supplemental material.



(a) Precision of Negatives

(b) Recall of Negatives

(c) Precision of Positives

(d) Recall of Positives

Fig. 2. Average Precision and Recall of Negatives/Positives in *Escherichia coli* 50 gene simulated networks

The aim of this study is to analyze to what extent the negative selection heuristics introduced in this paper could improve the performance of a classifier adopted to predict new gene–gene regulations within an organism. To allow replicability, raw data are available at the following url:

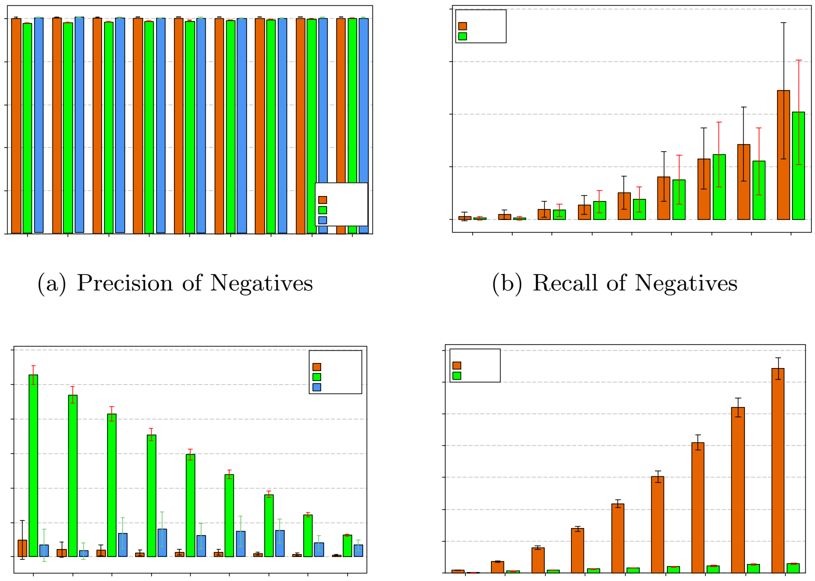
https://www.scoda.unisannio.it/rawdata/motifs0110.tgz.

4.1 Quality of the Selected Set S_H of Negative Examples

Figures 2 and 3 show the results—answering *RQ1*—in the case of simulated regulatory network of 50 genes of respectively *Escherichia coli* and *Saccharomyces cerevisiae*. On the x–axis the percentage of known positives is shown, while on the y–axis the average of Precision or Recall for each heuristic, Complement of Motif (MOTIF), Transitive Closure (TRANS), and Random (RANDOM), is shown with a confidence interval of 95%.

Results regarding the other networks, 10 and 100, were omitted as they exhibit similar behavior and do not depend on the number of gene of the regulatory network. Moreover Positive and Negative Recall of the RANDOM heuristic is not reported as it does not depend on the percentage of known positives (see Section 3.4).

Recall of both negatives and positives increase with the percentage of known positives. This is because the heuristics relies on an increasing number of known positives and then are able to infer more negative examples. Each organism



(a) Precision of Negatives

(b) Recall of Negatives

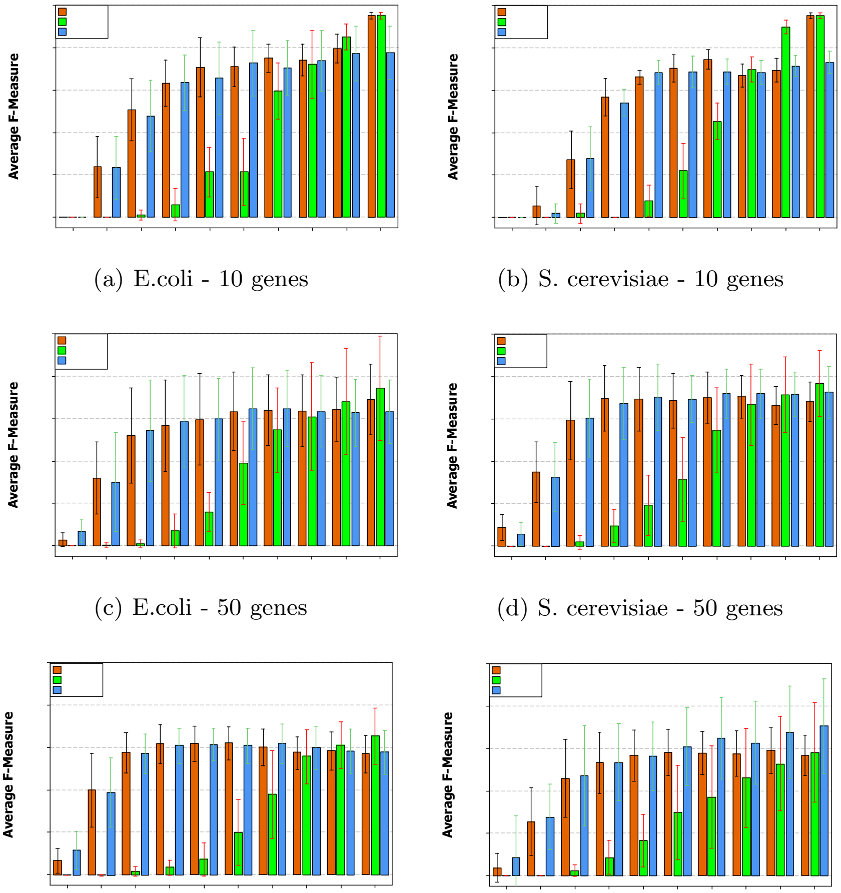
(c) Precision of Positives

(d) Recall of Positives

Fig. 3. Average Precision and Recall of Negatives/Positives in *Saccharomyces cerevisiae* 50 gene simulated networks

has a different increasing Recall trend. In *Escherichia coli* the MOTIF heuristic exhibits a negative recall that is higher than the negative recall of the TRANS heuristic. Inversely, the positive recall of MOTIF is lower that the positive recall of TRANS. The difference at each percentage of known positives is statistical significative with a p-value < 0.01 obtained with a t-test. In *Saccharomyces cerevisiae* the behavior of recall is inverted. Recall of negatives are in absolute low for both MOTIF and TRANS heuristics. Instead, the value of positive recall of the MOTIF heuristic is significantly higher that the recall of positives of the TRANS heuristic.

The trend of precision does not depend on the considered organism. Each organism exhibit similar precision trends of both positives and negatives. All considered heuristics exhibit a very high precision of negatives reaching a value that is very near to 1 at every percentage of known positives. This is because of unbalanced data, as the number of negatives are much more that the number of negatives. Instead, the trend of precision of positives depends on the adopted heuristic: MOTIF and RANDOM have a decreasing trend with the percentage of known positives, TRANS has an increasing trend between 10% and 50% followed by a decreasing trend between 50% and 100%. In *Escherichia coli* the RANDOM precision of positives is significantly higher than the MOTIF precision of positives at every percentage of known positives. Instead, the TRANS heuristic exhibit a higher precision of positives than both RANDOM and MOTIF



(a) E.coli - 10 genes

(b) S. cerevisiae - 10 genes

(c) E.coli - 50 genes

(d) S. cerevisiae - 50 genes

(e) E.coli - 100 genes

(f) S. cerevisiae - 100 genes

Fig. 4. Average F–Measure of an SVM classifier trained with different selection of negatives

when the percentage of known positives is higher that 30%. In *Saccharomyces cerevisiae* the RANDOM precision of positives is significantly higher than both MOTIF and TRANS precision of positives at every percentage of known positives. Instead the precision of positives of both MOTIF and TRANS is very low and does not higher than 0.005.

Such results suggest that the MOTIF heuristic outperforms the other two heuristics in *Escherichia coli*. In such an organism, the MOTIF heuristic exhibits, with respect to RANDOM and TRANS, a high precision and recall of negatives and a low precision and recall of positives, two requisites for a negative selection of good quality. Instead, in *Saccharomyces cerevisiae*, the TRANS heuristic exhibits the best behavior, although the recall of negatives is overall not high.

4.2 Performance of an SVM Classifier Trained with S_H

Figure 4 shows the results—answering *RQ2*—in the case of simulated regulatory network of 10, 50, and 100 genes of *Escherichia coli* and *Saccharomyces cerevisiae*. On the x–axis the percentage of known positives is shown, while on the y–axis the average of F–Measure for each heuristic, Complement of Motif (MOTIF), Transitive Closure (TRANS), and Random (RANDOM), is shown with a confidence interval of 95%.

Each heuristic exhibits a similar trend among different organisms and among different percentage of known positives: for each heuristic the F-Measure increases with the percentage of known positives because of the more quality of the selected set of negatives. Moreover, while the percentage of known positives increases, i.e. the percentage of unlabeled examples decreases, the F-Measure of each heuristic tend to coincide, reaching an almost convergent value at $P = 100\%$. The reason is that when all examples are labeled there is no need for an heuristic to select reliable negative examples.

Instead, when the percentage of known positives is low ($P < 70\%$) the performances of both MOTIF and TRANS are significantly higher than RANDOM (t-test p-value < 0.01). This confirms that selecting reliable negative examples is crucial to build an efficient classifier to predict new gene regulations.

No significant differences between MOTIF and TRANS can be observed, even if the quality of the selected negative set is not the same in both organisms. We suppose that the reason could be the different topology of *Escherichia coli* and *Saccharomyces cerevisiae* gene networks that may compensate different aspects of the heuristic. Further investigation could clarify such an aspect and the combination of two or more heuristics could improve the overall quality of the selected negative set.

5 Conclusion

This paper introduced and examined systematically a number of heuristics to select negative examples from unlabeled data to learn gene regulatory networks from expression data. In particular we found that heuristics based on the known network topology of the gene regulatory network are able to improve the quality of a training set, and then the performance of a classifier. Such an increment of the training set quality can be noticed in terms of precision/recall of positives (false negatives) and precision/recall of negatives (true negatives). We are aware that results presented in this paper are partial and no general conclusions can be drawn. Threats to validity can affect the results reported in Section 4. In particular, our results can be affected by the limitations of the synthetic gene expressions generation tool. Threats to external validity, concerning the possibility to generalize our findings, affect the study although we evaluated the heuristics on two model organisms, *Escherichia coli* and *Saccharomyces cerevisiae*, and on a statistically significant sample of random regulatory sub networks extracted from the current known gene regulatory connection of such organisms. Nevertheless, analyses on further organisms are desirable, as well as the use of different synthetic gene network generation tools and experimental gene expression datasets. Instead, the study can be replicated as the tools are available for downloading, as well as all simulated datasets. The benchmark process is detailed in Section 3 and we made raw data available for replication purposes. We believe that the issues presented in this paper could have an important role in the application of machine learning algorithms in gene regulatory networks discovery.

Acknowledgements

This work was supported by a research project funded by MiUR (Ministero dell'Università e della Ricerca) under grant PRIN2008-20085CH22F.

References

1. Margolin, A.A., Nemenman, I., Basso, K., Wiggins, C., Stolovitzky, G., Dalla Favera, R., Califano, A.: ARACNE: an algorithm for the reconstruction of gene regulatory networks in a mammalian cellular context. BMC Bioinformatics 7(suppl. 1) (2006)
2. Faith, J.J., Hayete, B., Thaden, J.T., Mogno, I., Wierzbowski, J., Cottarel, G., Kasif, S., Collins, J.J., Gardner, T.S.: Large-scale mapping and validation of escherichia coli transcriptional regulation from a compendium of expression profiles. PLoS Biol.
3. Liang, S., Fuhrman, S., Somogyi, R.: Reveal, a general reverse engineering algorithm for inference of genetic network architectures. In: Pac. Symp. Biocomput., pp. 18–29 (1998)
4. Polynikis, A., Hogan, S.J., di Bernardo, M.: Comparing different ODE modelling approaches for gene regulatory networks. Journal of Theoretical Biology (2009)
5. Werhli, A.V., Husmeier, D.: Reconstructing gene regulatory networks with bayesian networks by combining expression data with multiple sources of prior knowledge. Stat. Appl. Genet. Mol. Biol. 6 (2007)
6. Ben-Hur, A., Noble, W.S.: Kernel methods for predicting protein-protein interactions. Bioinformatics 21, i38–i46
7. Bock, J.R., Gough, D.A.: Predicting protein protein interactions from primary structure. Bioinformatics 17, 455–460 (2001)
8. Yamanishi, Y., Bach, F., Vert, J.P.: Glycan classification with tree kernels. Bioinformatics 23, 1211–1216 (2007)
9. Witten, I.H., Frank, E.: Data mining: practical machine learning tools and techniques. Kaufmann series in data management systems. Morgan Kaufmann, San Francisco
10. Grzegorczyk, M., Husmeier, D., Werhli, A.V.: Reverse engineering gene regulatory networks with various machine learning methods. Analysis of Microarray Data

11. Mordelet, F., Vert, J.P.: SIRENE: supervised inference of regulatory networks. Bioinformatics 24, i76–i82 (2008)
12. Elkan, C., Noto, K.: Learning classifiers from only positive and unlabeled data. In: KDD 2008: Proceeding of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 213–220. ACM, New York (2008)
13. Cerulo, L., Elkan, C., Ceccarelli, M.: Learning gene regulatory networks from only positive and unlabeled data. BMC Bioinformatics (2010)
14. Yu, H., Han, J., chuan Chang, K.C.: Pebl: Web page classification without negative examples. IEEE Transactions on Knowledge and Data Engineering 16, 70–81 (2004)
15. Li, X., Liu, B.: Learning to classify texts using positive and unlabeled data. In: IJCAI 2003, Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence, Acapulco, Mexico, August 9-15, pp. 587–594 (2003)
16. Ceccarelli, M., Cerulo, L.: Selection of negative examples in learning gene regulatory networks. In: IEEE International Conference on Bioinformatics and Biomedicine Workshop, BIBMW 2009, pp. 56–61 (2009)
17. Alon, U.: Network motifs: theory and experimental approaches. Nature Reviews Genetics 8, 450–461 (2007)
18. Milo, R., Shen-Orr, S., Itzkovitz, S., Kashtan, N., Chklovskii, D., Alon1, U.: Network motifs: Simple building blocks of complex networks. Science 298 (2002)
19. Albert, I., Albert, R.: Conserved network motifs allow protein protein interaction prediction. Bioinformatics 20, 3346–3352 (2004)
20. Itzkovitz, S., Levitt, R., Kashtan, N., Milo, R., Itzkovitz, M., Alon, U.: Coarse-graining and self-dissimilarity of complex networks. Phys. Rev. E Stat. Nonlin. Soft. Matter Phys. 71 (2005)
21. Kalir, S., McClure, J., Pabbaraju, K., Southward, C., Ronen, M., Leibler, S., Surette, M.G., Alon, U.: Ordering genes in a flagella pathway by analysis of expression kinetics from living bacteria. Science 292, 2080–2083 (2001)
22. Goemann, B., Wingender, E., Potapov, A.P.: An approach to evaluate the topological significance of motifs and other patterns in regulatory networks. BMC System Biology 3 (2009)
23. Shen-Orr, S.S., Milo, R., Mangan, S., Alon, U.: Network motifs in the transcriptional regulation network of escherichia coli. Nature Genetics 31, 64–68 (2002)
24. Lin, H.T., Lin, C.J., Weng, R.C.: A note on platt's probabilistic outputs for support vector machines. Mach. Learn. 68, 267–276 (2007)
25. Marbach, D., Schaffter, T., Mattiussi, C., Floreano, D.: Generating realistic in silico gene networks for performance assessment of reverse engineering methods. Journal of Computational Biology: A Journal of Computational Molecular Cell Biology 16, 229–239 (2009)
26. Stolovitzky, G., Monroe, D., Califano, A.: Dialogue on reverse-engineering assessment and methods: The dream of high-throughput pathway inference. Annals of the New York Academy of Sciences 1115, 1–22 (2007)
27. Chang, C.C., Lin, C.J.: LIBSVM: a library for support vector machines (2001), Software available at , `http://www.csie.ntu.edu.tw/~cjlin/libsvm`
28. Minami, R., Kitazawa, R., Maeda, S., Kitazawa, S.: Analysis of 5'-flanking region of human smad4 (DPC4) gene. Biochimica et Biophysica Acta (BBA) - Gene Structure and Expression 1443, 182–185 (1998)

MOSCFRA: A Multi-objective Genetic Approach for Simultaneous Clustering and Gene Ranking

Kartick Chandra Mondal[1], Anirban Mukhopadhyay[2], Ujjwal Maulik[3], Sanghamitra Bandhyapadhyay[4], and Nicolas Pasquier[1]

[1] I3S Laboratory (CNRS UMR-6070), University of Nice Sophia-Antipolis, Nice-06108, France
keto004@gmail.com, nicolas.pasquier@unice.fr

[2] Department of Computer Science and Engineering, University of Kalyani, Kalyani-741235, India
anirban@klyuniv.ac.in

[3] Department of Computer Science and Engineering, Jadavpur University, Kolkata-700032, India
umaulik@cse.jdvu.ac.in

[4] Machine Intelligence Unit, Indian Statistical Institute, Kolkata-700108, India
sanghami@isical.ac.in

Abstract. Microarray experiments generate a large amount of data which is used to discover the genetic background of diseases and to know the characteristics of genes. Clustering the tissue samples according to their co-expressed behavior and characteristics is an important tool for partitioning the dataset. Finding the clusters of a given dataset is a difficult task. This task of clustering is even more difficult when we try to find the rank of each gene, which is known as *Gene Ranking*, according to their abilities to distinguish different classes of samples. In the literature, many algorithms are available for sample clustering and gene ranking or selection, separately. A few algorithms are also available for simultaneous clustering and feature selection. In this article, we have proposed a new approach for clustering the samples and ranking the genes, simultaneously. A novel encoding technique for the chromosomes is proposed for this purpose and the work is accompleshed using a multi-objective evolutionary technique. Results have been demonstrated for both artificial and real-life gene expression data sets.

Keywords: Multi-objective Evolutionary Algorithm, Gene Ranking, Clustering, Gene Expression Data.

1 Introduction

The microarray technology generates the global and simultaneous view of expression levels for thousands of genes over different time points of different biological experiments. This is an important tool in the research area of Molecular Biology

R. Rizzo and P.J.G. Lisboa (Eds.): CIBB 2010, LNBI 6685, pp. 174–187, 2011.
© Springer-Verlag Berlin Heidelberg 2011

and Bio-Technology [20]. The biological information of a gene is described by the microarray expression pattern, also called as gene expression data. In microarray data, each gene corresponds to each row/column and each tissue sample corresponds to each column/row, respectively. Each row or column is identified as gene or sample expression profile. Each element or the expression in each profile is represented by a real number which denotes the expression level of a specific gene under a specific condition. Analysis of such data finds the relationships among the patterns present in the data, grouping genes/conditions according to their expression pattern and analysis of characteristics for new genes. This data analysis has two parts: forming the gene expression matrix from raw data generated by microarray technology and analysis of these matrices.

Appropriate mining strategies, e.g. clustering [2] and gene selection [3] are needed for analysis of such information. Clustering of co-expressed genes into biologically meaningful groups, helps in inferring the biological role of an unknown gene that is co-expressed with the known gene(s). Clustering is a process for organizing the objects from an object set into set of subsets of objects where the objects of a subset are similar but objects from different subsets are dissimilar in some ways. The clustering process is sometimes also called the *unsupervised learning process.* Clustering helps to partition the input space into K regions, $C_1, C_2, \cdots, C_K$, on the basis of some similarity/dissimilarity metrics, where the value of K may or may not be known previously. One frequently used such measure is called distance functions (dist(x, y) for x = $(x_1, x_2, \cdots, x_d)$ and y = $(y_1, y_2, \cdots, y_d)$). This distance function mainly depends on the type of applications where it is used, i.e., in numerical data, categorical data or in text document. Examples of such kind of useful distance functions are Euclidean distance, Manhattan distance, Mahalanobis distance, Minkowski Distance, Hamming distance and Maximum norm. One important issue in cluster analysis is the evaluation of clustering results to find the partitioning that best fit the underlying data. The process of evaluating cluster is known as *cluster validity* [6]. Several clustering algorithms are proposed in the literature. These algorithms are divided into different types according to their nature of operation (e.g. Hierarchical, Partitional, Density-Based, Grid-Based, etc).

Another important subject of matter is the *gene ranking* [4]. Gene Selection is a combinatorial problem. So, instead of selecting a subset of genes, we can give the weight or rank depending on the relevance, which is called *gene weighting* or *gene ranking* [16,18,19]. Gene ranking is used because of its simplicity, scalability, and good empirical success. Most of the gene ranking methods are based on the wrapper approaches or filter methods. Two well-known heuristic methods for gene weighting are: a) Gradient descent on the input space [21] and b) AdaBoost when each model is trained on one feature only [22].

In this article, we have proposed a multi-objective approach for simultaneous clustering and gene ranking. To the best of our knowledge, the process of simultaneous clustering and gene ranking by using multi-objective optimization is new in this area. The rest of the article is organized as follows: In Section 2, we present an overview on multi-objective Evolutionary paradigm with different concepts of MOO (Multi-objective Optimization). Section 3 presents a detailed discussion of our proposed algorithm with different components used in the algorithm. Section 4 presents the experimental design methods and results obtained during the experiments with a small discussion on them. Section 5 concludes the article and gives some future direction for further improvement of the proposed method.

2 Multi-objective Optimization

Genetic Algorithms (GAs) are very popular meta-heuristic optimization method but could not apply directly for multi-objective problems. Traditional GA are modified to reuse for multi-objective problems by using specialized fitness functions and introducing methods to promote solution diversity. Two general approaches are available for optimizing multiple objective. The first method is to combine every objective function into a single composite function (e.g., utility theory, weighted sum method). The second solution is to move all but one by one objective to the constraint set, a constraining value must be established for each of these former objectives. In all cases, the optimization method would return a single solution rather than a set of solutions that can be examined for trade-offs. For this reason, decision-makers often prefer a set of good solutions considering all the multiple objectives.

Most of the real world engineering problems are generally have multiple conflicting objectives, e.g., minimize cost, maximize performance, etc.. So, another solution for solving such multi-objective problem is to determine an entire *Pareto Optimal Solution Set* or a representative subset. In the Pareto optimal solution set, while moving from one solution to another, there is always a certain amount of sacrifice in some objective(s) to achieve a certain amount of gain in the other(s).

Consider that we want to optimize k objectives that are non-commensurable and equally important. Without loss of generality, we consider that all objectives are of the minimization type.

We also assume that the solution of this problem can be expressed by *decision variable vector* $\{x_1, x_2, \cdots, x_n\}$. The solution space X is generally restricted by a series of *constraints*, such as $g_j(x) = b_j$ for j = 1, $\cdots$, m and bounds on the decision variables. A function $f\colon X \rightarrow Y$ evaluates the quality of a specific solution by assigning it an *objective vector* $(y_1, y_2, \cdots, y_k)$ in the *objective space* Y. Our aim is to find a vector x^* that minimizes a given set of k objective functions $y(x^*) = y_1(x^*), \cdots, y_k(x^*)$.

A formal definition of Pareto optimality from the viewpoint of the minimization problem may be given as follows: A decision vector $\overline{x}^*$ is called Pareto optimal if and only if there is no $\overline{x}$ that dominates $\overline{x}^*$, i.e., there is no $\overline{x}$ such that $\forall i \in \{1, 2, \ldots, k\}$, $y_i(\overline{x}) \leq y_i(\overline{x}^*)$ and $\exists i \in \{1, 2, \ldots, k\}, y_i(\overline{x}) < y_i(\overline{x}^*)$. In words, $\overline{x}^*$ is Pareto optimal if there exists no feasible vector $\overline{x}$ which causes a reduction on some criterion without a simultaneous increase in at least another. In general, Pareto optimum usually admits a set of solutions called *non-dominated* solutions.

In multi-objective problem, our aim is to investigate a set of solutions, where each of which satisfies the objectives at an acceptable level without being dominated by any other solution. A solution is said to be *Pareto optimal* if it is not dominated by any other solution in the solution space. The *Pareto optimal solution set* in the decision space X is denoted as the *(Pareto set)* $X^* \subseteq$ X, and we will denote its image in objective space as *Pareto front* $Y^* = f(X^*) \subseteq$ Y. With many multi-objective optimization problems, knowledge about this set helps the decision maker in choosing the best compromise solution. For most of the multi-objective problems, entire Pareto optimal set identification is practically almost impossible for its size. For many problems, especially for combinatorial optimization problems, proof of solution optimality is computationally infeasible. Therefore, a practical approach is to investigate a set of solutions or *the best-known Pareto set* for multi-objective optimization that represent the Pareto optimal set as well as possible. Therefore, the goal of the optimization problem is to find or *approximate* the Pareto set. The outcome of a MOEA is considered to be a set of mutually non-dominated solutions also called *Pareto set approximation.*

2.1 Non-dominated Sorting Genetic Algorithm-II

For most of the multi-objective problems, entire Pareto optimal set identification is practically impossible for its size. Therefore, the goal of the optimization is to find an *approximate* Pareto set. The outcome of a Multi-Objective Evolutionary Algorithm (MOEA) is considered to be a set of mutually non-dominated solutions. The Non-dominated Sorting Genetic Algorithm-II (NSGA-II) [5], a popular MOEA method, is used here as the underlying optimization strategy. Other very popular optimization methods like PAES [1], AMOSA [15]etc can also be used instead of NSGA-II.

In NSGA-II [5], a random population P_0 is created with N chromosome and known as the initial parent population. According to their non-domination level, they are sorted and give a rank to each one solution under the population equal to their non-domination level. At first, they create a child population Q_t of the same size as parent by using selection, crossover and mutation operations. Then combine the parent and child population and create a population of size R_t and sort according to their non-domination level. Now the next parent population P_{t+1} is created by selecting the chromosome from R_t one by one according to their level. But it is not necessary that L_1 (the population of level 1) to L_i (the last level of the selected population for P_{t+1}) be the exact size of the population. So, here a crowded comparison method in descending order is included for selecting population from level L_i to choose the best solutions needed to fill all population slots. This crowded comparison operator is used to introduce the diversity among the non-dominated solutions (called Diversity Preservation), in selection phase and also in population reduction phase.

3 Clustering

Clustering is a process of grouping the objects from an object set into set of subsets of objects where the objects from a group are similar but objects from different groups are dissimilar in some ways. The number of groups/clusters and the size of each groups are different depending on the *criterion* on which the clustering process is done. Their is no uniform criterion that group the objects into same number of clusters or divided into same size or anything else. These clustering criterion is supplied by the user in such a way that the clustering result is suite their needs. The clustering is an *unsupervised learning process* or *unsupervised classification process* because, there are no previously known classes are present and the kind of relation exist between the data are also unknown.

3.1 Cluster Validity Index

Evaluation of the result found from the experiments that best fits the underlying data is one of the most important task in cluster analysis. This process of evaluation is called the cluster validation. Several cluster validity approaches are proposed in the literature. There are three main approaches are used to investigate cluster validity [6]. The first one is called *external criteria.* That means, evaluation of the results of a clustering algorithm based on a pre-specified structure. Next approach is known as *internal criteria*, where the evaluation of the results of a clustering algorithm is based the quantities that involve the vectors of the data set themselves (e.g. proximity matrix). The last approach for cluster validity is said as *relative criteria.* Here the basic idea is the evaluation of a clustering structure by comparing it with other clustering schemes, resulting by the same algorithm but with different parameter values. Cluster analysis or clustering is a common technique for statistical data analysis, for examples, Data reduction, hypothesis generation, hypothesis testing, prediction based on groups. A number of validity indices have been defined and proposed in the literature ([7], [8], [9], [6], [10], [13], [12]). Here we discussed mainly two indices, DB and XB index [11,10], which we used for our experiments. But one can use other validity indices like I index [13], Dunn index [12] etc.

XB Index: The Xie-Beni index [10], XB, also called the compactness and separation validity function, is a representative index in the category of Indices involving the membership values and the dataset. Consider a fuzzy partition of the data set $X = x_j; j = 1, \cdots, n$ with $v_i(i = 1, \cdots, n_c)$ the centers of each cluster and u_{ij} the membership of data point j to cluster i.

The fuzzy deviation, d_{ij}, of x_j form cluster i is defined as the distance between x_j and the center of cluster weighted by the fuzzy membership of data point j belonging to cluster i. Here, we used the crisp version of XB index where membership values are either 0 or 1.

$$d_{ij} = u_{ij}||x_j - v_i|| \tag{1}$$

Also, for a cluster i, the sum of the squares of deviation of the data point in X, denoted σ_i, is called variation of cluster i. The sum of the variations of all clusters, σ, is called total variation $\pi = (\sum_{i=1}^{K} \sigma_i)$.

Also, the separation of the partitions is defined as the minimum distance between cluster centers, i.e., $D_{min} = \min_{i,j=1 \ to \ K, i \neq j} ||v_i - v_j||$.

Then XB index is defined as

$$XB = \frac{\pi}{(n \times D_{min})} \tag{2}$$

where n is the number of points in the data set. It is clear that small values of XB are expected for compact and well-separated clusters.

DB Index: In Davies-Bouldin (DB) index [11], the similarity measure R_{ij} between the clusters C_i and C_j is defined based on a measure of dispersion of a cluster C_i and a dissimilarity measure between two clusters d_{ij}. The R_{ij} is a non-negative and symmetric. That is, $R_{ij} = (s_i + s_j)/d_{ij}$ and value of each s for each cluster is calculated as $s_i = \frac{1}{|C_i|} \sum_{x \in C_i} ||v_i - x||$.

Then the DB index is defined as

$$DB_n = \frac{1}{n} \sum_{i=1}^{n} R_i \tag{3}$$

$$R_i = max_{i,j=1 \ to \ n, i \neq j} R_{ij} \tag{4}$$

4 Proposed Technique

We propose a novel approach that simultaneously identify the cluster of each sample and rank of each feature (gene) according to their participation to create clusters of samples. Here we identify the clusters of the samples and rank the genes, simultaneously. A novel encoding technique is proposed here for the problem to fit into multi-objective frame work. Since, the Multi-objective Evolutionary Algorithms (MOEA) are known as the global search heuristics primarily used for optimization tasks. We use this process for our simultaneous optimization.

Here, our aim is to propose a method to simultaneously optimize the feature ranking and clustering. To optimize the task of finding the cluster and rank the feature according to their ability to create clusters by maintaining the competing constraints is an NP-complete problem [17]. Due to this high complexity, researchers are motivated to use various approximation techniques to generate near optimal solutions. Since, the MOEA is known as the global search heuristics primarily used for optimization tasks, we use this for our simultaneous optimization. The NSGA-II(Non-dominated Sorting Genetic Algorithm-II) [5] is used here as an important MOEA to optimize the chromosomes under population. Also it is used as a baseline algorithm to compare with other methods. NSGA-II is computationally efficient algorithm for implementing our idea but one can use other MOEA like PAES [1], AMOSA [15]. Another important point, the number of cluster is fixed, so the chromosome length is also fixed. Here we also present the representation and the general framework of MOEA for our simultaneous clustering and gene ranking task.

The Multi-objective Simultaneous Clustering and Feature Ranking Algorithm (MOSCFRA) is summarized as follows:

1. Initialize the chromosome under population as represented bellow.
2. Execute the NSGA-II algorithm with some terminating criteria to optimize the rank as well as cluster center through crossover, mutation, selection, elitism as described bellow.
3. Choose the appropriate solution from the Pareto set solutions for the problem.

The description of each component of our proposed technique are given in the subsequent subsections.

4.1 Chromosome Representation and Initial Population

A gene expression matrix is represented by rows and columns corresponding to samples (experimental biological conditions) and genes. Consider, a gene expression matrix D has d genes and s samples. The samples will be partitioned into K clusters and each cluster has a center which is represented by d dimensions. One solution is represented by one chromosome and each chromosome has $(d + (K \times d))$ bits to represent rank of each gene and K cluster center with d dimensions. The first d bit represents the weight of each gene and are used to encode the rank of the genes. The remaining bits are used for cluster centers. The one population is composed of several such chromosomes. The initial population is generated randomly.

4.2 Fitness Computation

Two validity indices, Xie-Beni(XB) [10] and Davis-Bouldin(DB) [11] are used as two objective functions to validate the generated cluster centers. Both of these objective functions are of minimization type. Small values of XB are expected for compact and well-separated clusters.

Another important idea, *Weighted Distance Method*, is used in our algorithm for computing the validity index. We give a weight to each gene and rank them according to their weight. The weight is also used to calculate the distance between two samples. In our algorithm, we use the Euclidean Distance in weighted form as the distance measure. The equation of Weighted Euclidean Distance is:

$$D(x, y) = \sqrt{\sum_{l=1}^{d} w_l^2 (x_l - y_l)^2} \tag{5}$$

For each chromosome, first we assign the sample in each cluster center present in the chromosome based on nearest center criterion. After assigning the samples, we update the cluster centers according to their sample values by taking the means. The new cluster centers are used to update the chromosome.

4.3 Crossover

In this algorithm, each chromosome in the population has two parts, the gene weight part and the cluster center part. The Uniform Crossover is used for the feature part of the chromosome and Single Point Crossover is used in the cluster center part of the chromosome. In both cases, the same C_p (Crossover Probability) value is used.

After crossover, a pair of parent chromosomes generates a pair of offspring chromosomes. So, the parent population generates the same size of offspring population. This offspring population is used in the mutation process.

4.4 Mutation

Here, a very small mutation probability (M_p) is used. Each time, if mutation is possible the actual value of the mutated bit is replaced by a random value. The range of the random value is between [0,1], since our data sets are normalized. The same technique is used for both part of the chromosome, i.e., gene weight and cluster center part.

4.5 Selection, Elitism and Termination

In our method, we use binary tournament selection with crowded and rank comparison method [5]. After successful completion of the crossover and mutation operation of a generation, the child population is combined with the parent population of that generation. From this combined population, the non-dominated chromosomes are selected and a new population of the same size is created for the next generation. This property of NSGA-II is called the *Elitism*. This technique ensures faster convergence of the process by keeping track of the best solutions generated so far. The NSGA-II has been executed for a fixed number of generations. This fixed number is supplied by the user for terminating the process. After terminating, the process gives a set of non-dominated solutions in the last generation.

4.6 Final Solution Selection

The final solution from the last non-dominated solution set is selected through the CP index and the R index. Both indices are described in the next section. For artificial data, the maximum value of CP index and R index of the solutions are selected but in case of real life micro-array gene expression cancer data, only maximum value of CP index is used. Our approach for simultaneous clustering and gene ranking is unsupervised but the process which is used here for selecting the best solution from the non-dominated set is supervised process. Rank of each genes in a chromosome is evaluated from the first d bits. The highest rank is given to that gene whose weight value is maximum.

5 Experiments and Results

In this section, we present the experimental design procedure and the results of the method with small discussion. For this task, two artificial data sets and two real data sets are used to measure the performance of our proposed method. Two performance measures, CP Index and R Index, are used for this purpose. After that, we compare the performance of the proposed method with several other important methods in this area.

5.1 Experimental Design

Here, we have given the information about the datasets of both real and artificial. Then, the required steps are given for the preprocessing of data sets.

Artificial Datasets
For our experiments, we create two artificial datasets viz., Arda25_30_3 and Arda50_75_5. Arda25_30_3 have 25 genes and 30 samples with 3 classes and Arda50_75_5 have 50 genes and 75 samples with 5 classes. In both the data sets, the genes are artificially generated so that they have different abilities in distinguishing the sample clusters.

Real Life Datasets
From several publicly available real life cancer datasets, two bench mark datasets, viz., Brain tumor and Lung tumor data sets, available at http://algorithmics.molgen.mpg.de/Static/Supplements/CompCancer/datasets.htm, are used for our experiments. The descriptions and their pre-processing are given here.

Brain tumor: This data set contains 42 tissue samples divided in 5 clusters (primitive neuroectodermal tumours (PNETs) (8 samples), atypical teratoid/rhabdoid tumours (Rhab) (10 samples), malignant gliomas (Mglio) (10 samples), medulloblastomas (MD) (10 samples) and normal tissues (Ncer) (4 samples)). There are total 1379 genes in the data set. Depending on the maximum variation of genes across the sample, the numbers of genes are reduced to 100. Therefore, after pre-processing the data size is 42 × 100.

Lung tumor: Using oligonucleotide microarrays, mRNA expression levels corresponding transcript sequences in 186 lung tumor samples and 17 normal lung tissues (NL) has been analyzed. The lung tumors included adenocarcinoma (AD) (139 samples), small-cell lung cancer (SCLC) (6 samples), pulmonary carcinoids (COID) (20 samples) and squamous cell lung carcinomas (SQ) (21 samples). The number of genes in the data set is 1543. Here also the same maximum variations of genes across the samples are used as a preprocessing step. After pre-processing, the size is reduced to 203 × 100.

Both the artificial and real datasets are normalized along the column. So, the value of all the data ranges from 0 to 1.

Parameter Settings

Our experiments are used to measure the quality of our proposed method for identifying the cluster and rank of genes. To compare with the different methods along these lines, we therefore performed the experiments with 100 generations. In each case, twenty trial runs were performed on each expression datasets and the average of the best solution of each run is given in the result. The number of clusters parameter is fixed for particular datasets, the number of clusters for Arda25_30_3 is 3 for instance, and set to 5 for other artificial & real datasets. The crossover rate is 0.8, mutation rate is 0.01 and population size is 50.

Performance Measures

The performance of the algorithm is measured in terms of both clustering and gene ranking ability. These are measured in terms of CP index and a newly defined R index. Only for artificial data sets, these indices calculation are possible, since class label and rank of the features are known. But in the case of real life data sets, since no rank information is available for the genes, the performance of clustering ability is calculated based on the CP index only.

Percentage of CP Index (correctly Classified Pairs) has been used to find the quality of the clustering results. CP index is used to compare a clustering solution with it actual clustering present in the data set. Say for a gene expression data set, the true clustering is C based on domain knowledge and c is a clustering result given by any clustering algorithm. Also assume that, s, d and t be the same, different and total number of pairs that belong to clusters in C and c, respectively. The percentage of CP index is defined as:

$$CP(C,c) = \frac{s+d}{t} \times 100 \tag{6}$$

From the above equation, we can say, higher value of CP means better clustering solution given by the algorithm. So, for CP(C,C) = 100%.

To find the quality of the ranking for a solution, a newly defined index, R index (Rank index) is used. In R index, we compare the generated ranking with true ranking. For this, we first sort the genes according to their ranks in both true and generated rankings. Thereafter, for first g genes, $g = 1, \ldots, d$, the intersection and union of the genes between true set and generated set are calculated and we divide the number of genes in intersection with that in union.

The plot of the corresponding R index is called the R plot. Since the maximum value of R index is 1, the R curve of better solution in the R plot will be nearer to 1.

Competitive Methods

The performance of MOSCFRA is determined by comparing the algorithm with its single objective counter parts that minimizes the objective function DB × XB (SOSCFRA_DX), only DB (SOSCFRA_DB) and only XB (SOSCFRA_XB). All the parameters are exactly same as that of multi-objective method MOSCFRA. Except these, two partitional clustering methods, viz., K-means clustering, fuzzy C-means (FCM) clustering, and five hierarchical clustering methods, viz., single linkage (HICSIL), average linkage (HICAL), complete linkage (HICCOL), centroid linkage (HICCEL), ward linkage (HICWAL) are used.

5.2 Result and Discussion

In table 1 and 2, the average value of the CP index over 20 runs on each artificial data set and real life gene expression data set are given, respectively. In brackets, the standard deviation of CP index is also shown. Moreover, higher value of CP index and lower value of standard deviation in all artificial data sets indicate that each time MOSCFRA outperforms other algorithms in terms of clustering.

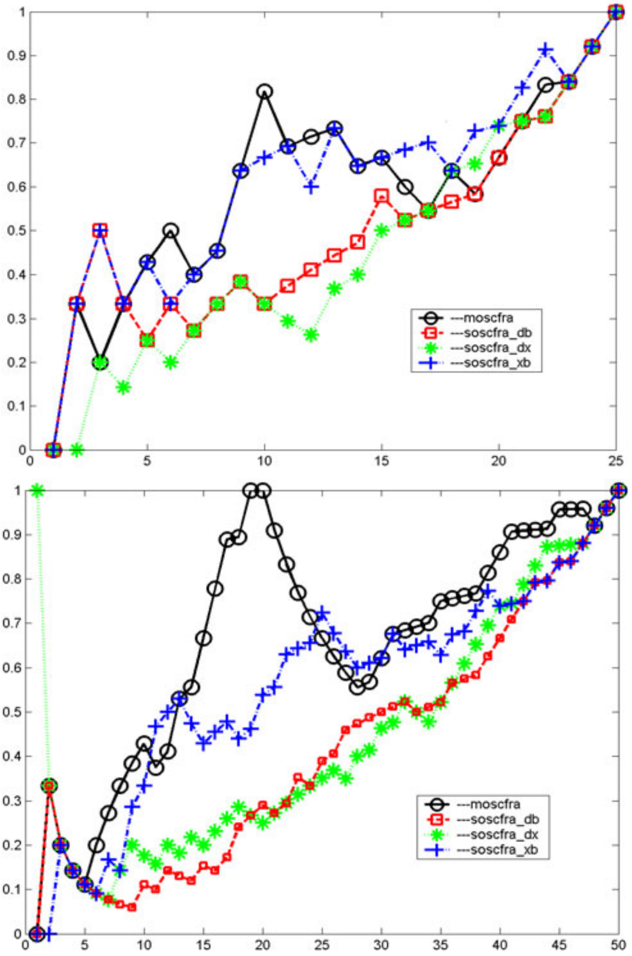


Fig. 1. R plot for the artificial data sets: (above) Arda25_30_3 data, (below) Arda50_75_5 data

The values of performance index generated from the other algorithms are inferior to those generated from MOSCFRA. Because, these methods find cluster from the data set by considering the same weight of the features which affect the clustering results. Through this, we can show the significance of the importance of feature ranking. By comparing all the results generated from all the data sets, it is clear that MOSCFRA technique gives the best clustering performance for these data sets.

Since the actual ranks of the features are available for the artificial data sets and it is absent in case of real data sets, we can compute the R index as described above only for the two artificial data sets. The Figure 1 shows the R plot of the R index for the highest CP index of all the runs on each algorithm (MOSCFRA, SOSCFRA_DX, SOSCFRA_DB, SOSCFRA_XB) for the two artificial data sets, respectively. Since the other algorithms does not generate ranks of the features, they are not shown in the figures. From these figures, we can say that the proposed multiobjective algorithm produces the good ranking result.

Another important thing is that, when the DB index and XB index are merged into our single objective counter part, SOSCFRA_DX, it gives the same result as given in the SOSCFRA_DB. So, from this result, we can say the DB index is affecting the goodness of XB index and also DB index is not a good index in such cases.

Table 1. Experimental Result on Artificial Data Sets

	CP index for Artificial Data Sets.	
Algorithm	Arda25_30_3	Arda50_75_5
MOSCFRA	**100.00(±0.000)**	**86.6036(±3.428)**
SOSCFRA_DX	31.0345(±0.000)	18.9189(±0.000)
SOSCFRA_DB	31.0315(±0.000)	18.9189(±0.000)
SOSCFRA_XB	76.5977(±0.925)	84.8649(±0.680)
K-means	96.0115(±9.800)	84.3532(±2.371)
FCM	95.6322(±0.000)	80.5045(±0.388)
HICSIL	35.8621(±0.000)	25.3333(±0.000)
HICCOL	71.7241(±0.000)	81.6577(±0.000)
HICAL	91.7241(±0.000)	81.5495(±0.000)
HICCEL	77.2414(±0.000)	66.8468(±0.000)
HICWAL	91.7241(±0.000)	85.2973(±0.000)

From the brain tumor genes, the most frequently ranked top ten genes that are responsible for that clustering through our proposed MOSCFRA algorithm are: S81957_at, D38500_at, K02268_at, X64072_s_at, M58297_at, J04132_at, M93 119_at, J04444_at, L36847_at, HG3141-HT3317_f_at.

From the lung tumor genes, the most frequently ranked top ten genes that are responsible for that clustering through our proposed MOSCFRA algorithm are: 39022_at, 939_at, 32251_at, 33373_at, 37849_at, 40195_at, 32034_at, 40647_at, 33273_f_at, 34335_at.

Table 2. Experimental Result on real-life Data Sets

Algorithm	CP index for Real Life Data Sets.	
	Brain Tumor	Lung Tumor
MOSCFRA	**82.0209(±8.515)**	**78.4193(±3.618)**
SOSCFRA_DX	19.6283(±0.000)	49.4659(±0.000)
SOSCFRA_DB	19.6283(±0.000)	49.4659(±0.000)
SOSCFRA_XB	81.9698(±0.878)	76.7605(±0.555)
K-means	73.8850(±11.458)	65.5229(±6.531)
FCM	69.1347(±4.047)	60.4251(±4.052)
HICSIL	30.3136(±0.000)	56.5381(±0.000)
HICCOL	44.0186(±0.000)	71.6919(±0.000)
HICAL	30.3136(±0.000)	57.4111(±0.000)
HICCEL	30.3136(±0.000)	57.4111(±0.000)
HICWAL	67.0151(±0.000)	77.1237(±0.000)

6 Conclusion and Future Scope

In this work, we have described a new algorithmic solution for the problem of simultaneous clustering and gene ranking. Finding the rank corresponding to the weight is an important task in clustering as well as in the data analysis. In this work, we address the problem of unsupervised gene ranking and unsupervised clustering. Here we have used a well-known multiobjective framework (NSGA-II) for simultaneous clustering and gene ranking of gene expression dataset. A novel encoding technique is developed for our problem and XB and DB index are used as optimization criteria which are minimized simultaneously. The performance is demonstrated on two artificial data sets as well as two real-life data sets.

As a scope of future work, other indices like Dunn [12] or I index [13] can also be used for studying the improvement of the result instead of XB and DB. The algorithm can be extended for unknown number of clusters. Also, other important multiobjective algorithms can be applied and more statistical comparison method can be used. Furthermore, choice of objective functions and selection of final solution from Pareto optimal set need closer look. The authors are working in these directions.

References

1. Knowles, J.D., Corne, D.W.: Pareto archived evolution strategy: A new baseline algorithm for Pareto multiobjective optimization. In: Congress on Evolutionary Computation (CEC 1999), vol. 1, pp. 98–105. IEEE Press, Piscataway (1999)
2. Ben-Dor, A., et al.: Clustering gene expression patterns. Journal of Computational Biology 6, 281–297 (1999)
3. Duda, R.O., Hart, P.E.: Pattern Classification and Scene Analysis. John Wiley and Sons, Inc., New York (1973)
4. Guyon, I., Elissee, A.: An introduction to variable and feature selection. Journal of Machine Learning Research 3, 1157–1182 (2003)

5. Deb, K., Pratap, A., Agrawal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Transaction on Evolutionary Computation 6, 182–197 (2002)
6. Theodoridis, S., Koutroubas, K.: Pattern Recognition. Academic Press, London (1999)
7. Halkidi, M., Vazirgiannis, M., Batistakis, Y.: Quality scheme assessment in the clustering process. In: Zighed, D.A., Komorowski, J., Żytkow, J.M. (eds.) PKDD 2000. LNCS (LNAI), vol. 1910, pp. 265–276. Springer, Heidelberg (2000)
8. Rezaee, R., Lelieveldt, B.P.F., Reiber, J.H.C.: A New Cluster Validity Index for the Fuzzy c-Mean. Pattern Recognition Letters 19, 237–246 (1998)
9. Sharma, S.C.: Applied Multivariate Techniques. John Wiley and Sons, Chichester (1996)
10. Xie, X., Beni, G.: A validity measure for fuzzy clustering. IEEE Transaction on P.A.M.I. 13(4), 841–846 (1991)
11. Davies, D., Bouldin, D.: A cluster separation measure. IEEE Transaction on P.A.M.I 1(2), 224–227 (1979)
12. Dunn, J.C.: A Fuzzy Relative of the ISODATA Process and Its Use in Detecting Compact Well-Separated Clusters. Journal of Cybernetics 3(3), 32–57 (1973)
13. Maulik, U., Bandyopadhyay, S.: Performance Evaluation of Some Clustering Algorithms and Validity Indices. IEEE Transactions on Pattern Analysis and Machine Intelligence 24(12) (December 2002)
14. Mukhopadhyay, A., Maulik, U., Bandyopadhyay, S.: Multi-objective Genetic Algorithm based Fuzzy Clustering of Categorical Attributes. IEEE Transactions on Evolutionary Computation 13(5), 991–1005 (2009)
15. Bandyopadhyay, S., Saha, S., Maulik, U., Deb, K.: A Simulated Annealing-Based Multiobjective Optimization Algorithm: AMOSA. IEEE Transactions on Evolutionary Computation 12(3), 269–283 (2008)
16. Zhang, C., Lu, X., Zhang, X.: Significance of gene ranking for classification of microarray samples. IEEE/ACM Transaction on Computational Biology and Bioinformatics 3(3), 312–320 (2006)
17. Garey, M.R., Johnson, D.S.: Computers and Intractability: A Guide to the Theory of NP-Completeness. W.H. Freeman, New York (1979)
18. Jong, K., Mary, J., Cornuéjols, A., Marchiori, E., Sebag, M.: Ensemble feature ranking. In: Boulicaut, J.-F., Esposito, F., Giannotti, F., Pedreschi, D. (eds.) PKDD 2004. LNCS (LNAI), vol. 3202, pp. 267–278. Springer, Heidelberg (2004)
19. Streib, F.E., Dehmer, M., Liu, J., Muhlhuser, M.: A systems approach to gene ranking from dna microarray data of cervical cancer. In: World Academy of Science, Engineering and Technology, vol. 8 (October 2005)
20. Sharan, R.: Click and expander: a system for clustering and visualizing gene expression data. Bioinformatics 19, 1787–1799 (2003)
21. Burges, C., Shaked, T., Renshaw, E., Lazier, A., Deeds, M., Hamilton, N., Hullender, G.: Learning to rank using gradient descent. In: ICML (2005)
22. Freund, Y., Schapire, R.E.: Experiments with a new boosting algorithm. In: Proceedings of the Thirteenth International Conference on Machine Learning, pp. 148–156 (1996)

A Multi-relational Learning Framework to Support Biomedical Applications

Teresa M.A. Basile, Floriana Esposito, and Laura Caponetti

Università degli Studi di Bari - Dipartimento di Informatica- Bari, Italy
{basile,esposito,laura}@di.uniba.it

Abstract. The definition of tools able to extract knowledge from structured biological data in order to support scientists research is increasing as shown by the popularity reached in the field of bioinformatics. In particular we focus our attention on the domain of assisted reproduction techniques with particular interest on the field of intracytoplasmic sperm injection. In this paper we would provide a multi-relational learning framework able to discover hidden relationships between entities involved in this application domain. Our approach is based on a multi-relational partitional clustering algorithm followed by a multi-relational rule induction. Furthermore, the obtained rules can be represented in a easily comprehensible form and can be used as an advisor to the clinicians during their work in order to help them in determining what knowledge sources are relevant for a treatment plan.

1 Introduction

Machine learning has become a rapidly developing and increasingly aspect of many biomedical applications involving clinical information systems and clinical decision support systems. In the field of assisted reproductive technologies, ICSI (IntraCytoplasmic Sperm Injection) fertilization is a medically-assisted reproduction technique, enabling infertile couples to achieve successful pregnancy. In this field crucial points are: the analysis of clinical data of the patient, aimed at adopting an appropriate stimulation protocol to obtain an adequate number of oocytes, and the selection of the best oocytes to fertilize. The main goal is the identification of some factors useful to prognostic a pregnancy.

Generally this analysis is manually performed by the clinicians and is based on the subjective experience. Thus a learning system able to exploit past experiences to suggest possible modifications to an ICSI treatment plan could be useful to aid clinicians in making decisions, for example, about the stimulation protocol to be carried out in order to obtain good quality oocytes. Once the system's knowledge base is populated with a sufficient number of past cases, it can be used to explore and discover interesting relationships among data, thereby achieving a form of knowledge mining.

In this work we present a multi-relational learning approach able to deal with clinical data and relevant features extracted from oocyte images.The aim is to discover new information useful to support the clinicians both in the definition of

R. Rizzo and P.J.G. Lisboa (Eds.): CIBB 2010, LNBI 6685, pp. 188–202, 2011.
© Springer-Verlag Berlin Heidelberg 2011

the stimulation protocol for new unseen patients and in the selection of oocytes from new unseen oocytes. Due to the presence of strong relationships among different stages of the process, multi-relational learning techniques that are able to take into account the relationships existing among all the entities involved in the process seem to be the most suitable approaches in this and similar medical application domains. The approach consists of a multi-relational clustering, based on APAM algorithm [1], followed by a multi-relational rule induction. Furthermore, the obtained rules can be represented in a easily comprehensible form and can be used as an advisor to the clinicians during their work in order to help them in determining what knowledge sources are relevant for a treatment plan.

In the following the ICSI application domain is presented along with its features. Then Section 3 and Section 4 present the multi-relational learning techniques exploited in such domain and the framework for image analysis and knowledge extraction from data. Finally Section 5 reports the preliminary experimental results on real data.

2 Problem Description

Infertility is becoming a frequent problem in the last decades and many assisted reproductive techniques have been designed to overcome it. One of these techniques is the intracytoplasmic sperm injection (ICSI) technique in which a single sperm is directly injected into an oocyte. After the procedure, the oocyte is placed into cell culture and checked on the following day for signs of fertilization. The fertilized oocyte grows in a laboratory for one to five days, then it is placed in the woman's uterus.

Due to ethical and medical reasons a specified number of embryos have to be selected and hence transferred in woman's uterus. As a consequence, even the number of oocytes to fertilize could be under such a restriction and clinicians prefer to appropriately select the most promising oocytes among all the oocytes taken from the woman. Fig. 1 shows a complete overview of the procedure.

The oocytes selection is manually done by non-invasive examination based on simple methods and observation focused on morphology and dynamics of the oocyte. A set of morphological parameters to be examined are present in medical literature such as oocyte/cytoplasm dimension, perivitelline space and zona pellucida thickness, first polar body conformation, and more subtle abnormalities of cytoplasm.

However, these variables are not the unique and independent parameters involved in the process (see Fig. 2). Indeed, in general, before the ICSI procedure, an hormone stimulation protocol on the female patient, consisting of a set of pharmacological treatments, is carried out in order to ensure the development of multiple preovulatory follicles to obtain multiple oocytes to aspire. In this phase, the couples' health conditions and characteristics have to be taken into account as well.

Some works faced the problem of introducing systems to support clinicians in their work. Some approaches work with low level features extracted from oocyte

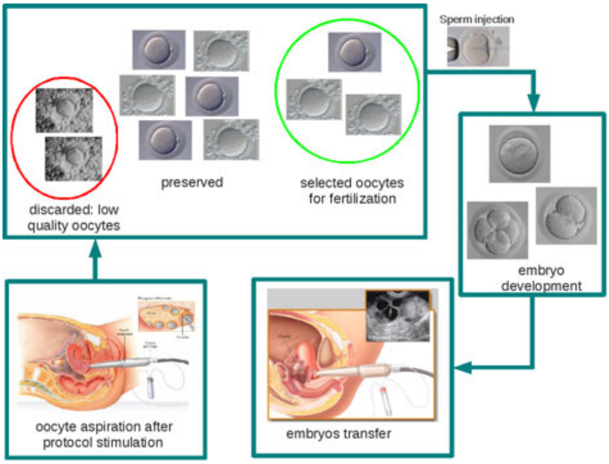

Sperm injection
discarded: low quality oocytes
preserved
selected oocytes for fertilization
embryo development
oocyte aspiration after protocol stimulation
embryos transfer

Fig. 1. ICSI procedure under the restriction of oocyte selection

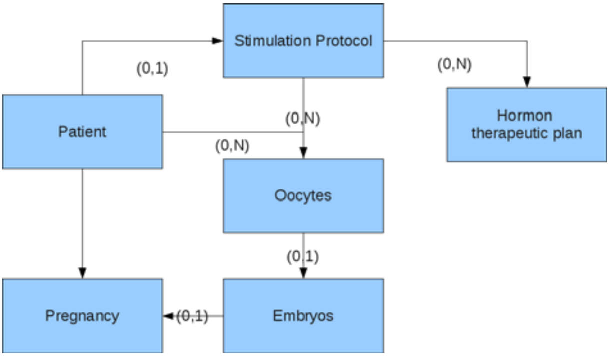


Fig. 2. Entity and relationships involved in the application domain

images to assess their quality, such as [5] that proposes a method to evaluate the oocyte diameter, [7] that presents an approach to evaluate the quantification of oocyte cytoplasm morphology or [8] that defines a quantitative evaluation of birefringence properties of the zona pellucida. Other approaches work with higher level characteristics such as the clinical data of the patients with the aim of grasping structural patterns that define the peculiarities of the patient.

Few approaches are presented in literature that work with both these kinds of information, but they assume that all the information on both features extracted from the images and the clinical data are available [10,16,9]. Furthermore, they exploit an attribute-value description of the data thus losing the relationships existing between oocytes and patients data. Indeed, an important aspect and commonly neglected by these approaches is that each set of variables, i.e. clinical patient data and image features cannot be considered as a stand-alone set since relationships between such sets of data can occur [12]. For example, clinical data of the patient are related both to the oocyte quality and to the implantation success rate; the oocyte quality, intended as its maturity, plays a fundamental role in the embryo development; finally the correct embryo development is crucial for the successive transfer and implantation success [13,14]. For these reasons, multi-relational learning techniques seem to be the most suitable approaches in this application domain.

3 Multi-relational Learning Approach

The learning approach we consider to tackle the knowledge extraction task in the ICSI application domain is based on relational clustering followed by relational rule induction. The representation language used in this work is Datalog [15], a first-order logic language. The first-order alphabet consists of a set of constants $\mathcal{C}$, a set of variables $\mathcal{V}$, a set of function symbols $\mathcal{F}$, and a non-empty set of predicate symbols $\mathcal{P}$. A multi-relational description is made up of a set of predicate symbols $p \in \mathcal{P}$ applied to n terms t_i, ($t_i \in \{\mathcal{C} \cup \mathcal{V}\}$): $p(t_1, \ldots, t_n)$. Multi-relational descriptions are said to be ground whenever they do not contain variables. A Datalog description is a multi-relational description in which only variables and constants are used as predicate arguments.

3.1 Multi-relational Clustering

Clustering is an unsupervised learning technique used to find a partition of a set of objects into clusters so that the objects within each cluster are similar to each other. The similarity between objects can be determined using various distance measures.

Relational clustering works on relational data (i.e., objects with a first-order description as representation language) and uses distance measures that are generally more complex than those used in the case of attribute-value representations. Indeed, the generic Euclidean distance cannot be applied to relational representations of the data as they are not represented by a feature vector of a fixed number of measurements.

Here we use the distance function and the modification of a partitional clustering algorithm, named *Approximate Partition Around Medoids* (APAM) both introduced in [1], and here briefly reported.

As to the distance function an adaptation of the Tanimoto metric [2] to the case of relational descriptions is exploited. Specifically, the Tanimoto metric adaptation to define the distance between two multi-relational descriptions D_1 and D_2 is as follows:

$$d_{T_\cap}(D_1, D_2, \alpha) = \frac{|D_1| + |D_2| - 2s_\cap(D_1, D_2, \alpha)}{|D_1| + |D_2| - s_\cap(D_1, D_2, \alpha)},$$

where $|D_i|$ is the number of components (literals) in the multi-relational description D_i and the number of literals in common between D_1 and D_2 is approximated by:

$$s_{\cap}(D_1, D_2, \alpha) = \frac{\sum_{i=1}^{\alpha} |R_{D_1} \cap R_{D_2 i}|}{\alpha},$$

where $R_{D_1} = ren(1, D_1, C)$ is a fixed renaming of the multi-relational description D_1, $R_{D_2 i} \in ren(\alpha, D_1, C)$ is a renaming of the multi-relational description D_2, $C = max(C_1, C_2)$ (C_i is the set of constants in D_i) and $\alpha > 0$ is the parameter governing the approximation. In other words, $s_{\cap}(D_1, D_2, \alpha)$ is the mean of the number of common literals in D_1 and D_2 for each of the α renamings of D_2. In this setting a renaming of D, indicated by $R(D)$, is a ground description obtained by firstly turning constants into variables in D and then applying a substitution (i.e., a mapping from variables onto a new set of constants) to the result. The set of renamings $S = ren(k, D, C)$ are generated randomly choosing k renamings of D onto the set of constants C:

$$ren(k, D, C) = S_i = S_{i-1} \cup \{R(D)_{\{C\}}\} \quad i = 1, \ldots, k \ and \ S_0 = \emptyset$$

As the partitional clustering algorithms, the following generic schema is considered:

1. randomly choose k representatives for clusters;
2. iteratively improve these initial representatives until the change in the objective function from one iteration to the next drops below a given threshold:
   - (a) assign each object to the cluster it “fits best” in the current clustering
   - (b) compute new cluster representatives using these new assignments

One of the most well-known and commonly used partitioning method is the k-*medoids* clustering algorithm. Traditional k-medoids clustering algorithm seeks to find k medoids among the objects in the data set minimizing, for a given clustering solution $\mathcal{C}$, the following objective function:

$$tightness(\mathcal{C}) = \tfrac{1}{n} \textstyle\sum_{i=1,\ldots,n} d(\mathbf{x}_i, \mu_i)),$$

where μ_i is the medoid of the cluster the object $\mathbf{x}_i$ belongs to and $d(\cdot, \cdot)$ is the distance.

The k-medoids clustering algorithm `PAM` on which `APAM` is based starts with a set of clusters containing the medoids of the complete data set, and greedily inserts new objects into this set of clusters while minimizing the above objective function. Then, it tries to improve the previously obtained clustering by exploring all possible replacements of medoids by non-medoids picking the replacement that enhances the fitness function. If no such fitness improving replacement can be found, the procedure terminates.

`APAM`, the approximate relational clustering variant of `PAM`, uses the following objective function:

$$\mathcal{J}_{tightness}(\mathcal{C}, \alpha) = \tfrac{1}{n} \textstyle\sum_{i=1,\ldots,n} d_{T_{\cap}}(\mathbf{x}_i, \mu_i, \alpha).$$

Similarly to `PAM`, it starts by randomly selecting k medoids and finding the first clustering solution $\mathcal{C}$ by associating each non-medoid instance to the cluster whose medoid is more similar. Then, it iteratively tries to swap a medoid with a non-medoid object, exploring all possible replacements, in order to minimize the value of the objective function $\mathcal{J}_{tightness}(\cdot,\cdot)$. It terminates if no replacement can be found that leads to a clustering with a better (lower) objective value with respect to $\mathcal{J}_{tightness}(\cdot,\cdot)$.

3.2 Multi-relational Rule Induction

Rule induction is a supervised learning technique concerning the extraction of a set of formal rules from a set of labelled observations. One of the rule induction paradigms able to deal with multi-relational representation language is the Inductive Logic Programming (ILP) framework [11]. In this setting, given a background knowledge and a set of labelled (positive and negative) observations represented as a logical database of facts, the aim is to derive a hypothesised logic program (set of rules or theory in the following) which entails all the positive and none of the negative observations. The derivation of the rules is performed by exploring the lattice-based concepts by means of some operators such as refinement, least general generalisation and inverse resolution.

In this work we adopted the ILP system INTHELEX [3] that here we briefly describe, to process the results obtained by the multi-relational clustering technique. It is a learning system for the induction of theories from positive and negative observations which focuses the search for refinements by exploiting the Object Identity bias on the generalization model. It is fully and inherently incremental: this means that, in addition to the possibility of taking as input a previously generated version of the theory, learning can also start from an empty theory and from the first available observation; moreover, at any moment the theory is guaranteed to be correct with respect to all of the observations encountered thus far.

The system can learn simultaneously various concepts, possibly interrelated, and is based on a closed loop architecture, i.e. correctness is checked on any new example and, in case of failure, a revision process is activated on it, in order to restore the correctness. The system deals with theories expressed as sets of Datalog descriptions. It adopts a full memory storage strategy, i.e. it retains all the observations in order to guarantee correctness of the learned theories on all of them. The process of theory refinement, as performed by the system, is now briefly summarized. The system exploits a previous theory (if any) and a memory of all the past (positive and negative) observations that led to the current theory. The new observations are exploited incrementally to modify incorrect hypotheses according to a data-driven strategy. In particular, when a positive observation is not covered, a revision of the theory to restore its completeness is performed as follows:

- replacing a rule in the theory with one of its least general generalizations against the problematic observation;
- adding a new rule to the theory, obtained by properly turning constants into variables in the problematic example;
- adding the problematic observation as a positive exception.

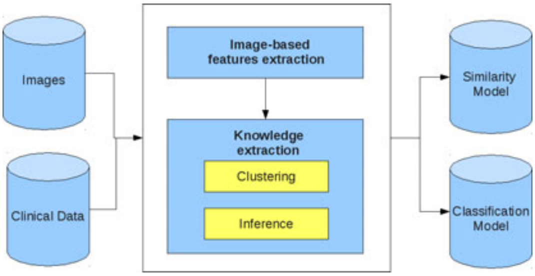


Fig. 3. Schematic representation of the proposed framework

When, on the other hand, a negative observation is covered, the system outputs a revised theory that restores consistency by performing one of the following actions:

- adding *positive* information able to characterize all the past positive observations (and exclude the problematic one) to the rules that concur to the example coverage;
- adding *negative* information to discriminate the problematic observation from all the past positive ones to rule in the theory that covers the problematic observation;
- adding the problematic observation as a negative exception.

4 The Framework

The general framework we propose, depicted in Fig. 3, is made up of a module devoted to image-based features extraction - based on mathematical morphology - and a module for knowledge extraction from both clinical and image features data - based on multi-relational learning techniques.

4.1 Image-Based Features Extraction

The features extraction module is oriented to extract some relevant morpho-structural features from oocyte images, such as the measures of oocyte and cytoplasm diameters. This can be addressed as an image segmentation problem. Since we are interested in extracting the shape of the oocyte from the image, we employ a segmentation method better suited for shape analysis, that is mainly based on mathematical morphology [6].

Basic concept of mathematical morphology is the structuring element: given a two-dimensional binary image $X \subset Z^2$, a structuring element is a particular set $B \subset Z^2$, that gets translated over X and whose relations with X are studied at each location. In the following, we denote B_x the translation of B by x: $B_x = \{b + x \mid b \in B\}$.

The basic operations of mathematical morphology are dilation and erosion. The dilation of an image $X \subset Z^2$ by a structuring element B, denoted by $\oplus$, is

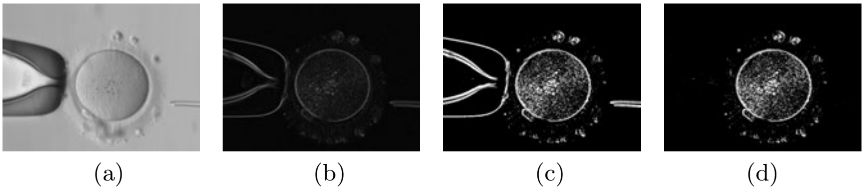


Fig. 4. Preprocessing of an oocyte image: (a) Original. (b) Edge. (c) Binary. (d) After killing borders.

the set of points $x \in Z^2$ such that the translation of B by x has a non-empty intersection with set X: $X \oplus B = \{x \in Z^2 \mid X \cap B_x \neq 0\}$. The erosion of X by a structuring element B, denoted by $\ominus$, is the set of points $x \in Z^2$ such that the translation of B by x is included in X: $X \ominus B = \{x \in Z^2 \mid B_x \subseteq X\}$.

From the erosion and dilation operators, two fundamental morphological operations can be derived as follows: the opening of X by B, denoted by $\circ$, is the union of all the translations of the structuring element that fit inside the image X, i.e. $X \circ B = \bigcup\{B_x \mid B_x \subset X\} = ((X \ominus B) \oplus B)$. The dual operation of the opening is the closing, denoted by $\bullet$, which is defined as: $X \bullet B = ((X \oplus B) \ominus B)$.

On such operators we designed a procedure able to extract the region containing the oocyte, and its diameter, taking out elements which are not of interest (e.g. the holding/injection pipettes, that are visible in many images), along with a good approximation of the cytoplasm diameter. Specifically, the proposed procedure works as follows.

Preprocessing. The preprocessing consists of finding edges in the image, by means of Sobel operator [4], and successively binarize the result (see Fig. 4). After the binarization, elements that are not of interest surrounding the image borders, such as the holding pipette and the injection pipette, have to be taken out. This can be done by firstly selecting a point p in the region of the border and, starting form it, by finding the connected components[1]. This step uses an *extraction of connected components* algorithm [4] based on dilation and intersection of the set of pixels of the binary image.

Oocyte region detection. After preprocessing, the binary image shows segments of high contrast that do not quite delineate the outline of the object of interest (Fig. 4d). Indeed gaps in the segments surrounding the object are evident. These gaps will disappear as soon as the image is dilated twice using circular structuring elements.

The dilated image shows the outline of the object quite nicely, but there are still holes in the interior of the object. The filling of these holes is performed by starting from a point p in the region to fill and iteratively dilating it and intersecting the resulting dilation with the complement of the image to fill [4].

[1] Two pixels are connected in $S \subseteq A$ if there exists a path between them made up of pixels belonging to S. The set of pixels connected to the pixel $p \in S$ is known as connected component of S.

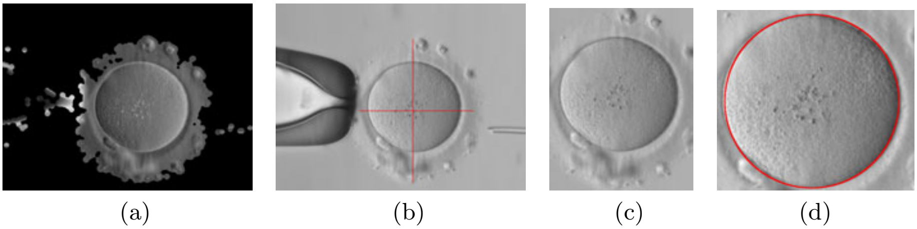


Fig. 5. Oocyte region detection. (a) Detected region (b) Oocyte Diameter (c) Oocyte region extraction (d) Cytoplasm detection

Finally, in order to make the segmented object look natural, the corresponding region is smoothed by an opening-closing operation with a circular structuring element.

Now, by subtracting the obtained image from the original one, the region of the oocyte on a black-background is achieved (see Fig. 5a).

Finally, in order to obtain the smallest rectangle that contains every point of the object, the center of mass of the oocyte region is calculated and, starting from it, the 4-directional Euclidean distances, until a pixel background is encountered, are computed (Fig. 5b). The mean of these values represents the diameter of the oocyte region and the minimum and maximum x and y coordinates of the 4-directional Euclidean distances are the starting points from which to extract the bounding rectangular region containing the oocyte (Fig. 5c).

Cytoplasm region detection. As according to medical literature the cytoplasm dimension is about the 66% of the oocyte dimension [17], this value can be used to approximate the cytoplasm diameter. A more accurate measure of the diameter of the cytoplasm has been obtained by considering that the shape of the cytoplasm can be approximated by a circumference. To this aim the Hough transform is applied to the binary image so as to detect the best circle fitting the shape of the oocyte cytoplasm. This has been done by searching for circles of radius r, varying from $(OD/2 - \delta)$ to $(OD/2 + \delta)$, where δ has been chosen equal the 10% of the oocyte dimension OD. The resulting cytoplasm detection is shown in Fig. 5d.

4.2 Knowledge Extraction

The knowledge extraction step involves the representation of both clinical and image-based data, extracted from oocyte images, and the application of the multi-relational learning approach to build a model able to solve the issues concerning the identification of similarities among situations such as stimulation protocols under specific patients'health conditions and, hence, the predictivity of the goodness of the oocytes to choose as the most promising for the specific task of fertilization.

As to the data representation, the general information on a patient and the clinical data about the couple diseases, before the ICSI procedure starts, is followed by the data describing the ovarian stimulation protocol and by the clinical data of the patient observed after the therapeutic plan has been taken place. Successively, the data about the oocyte aspiration phase are introduced. For each patient a set of n (a value varying form one patient to another) oocytes is obtained and each oocyte is described according to the own features extracted from the images. Hence, once the images are elaborated, the information extracted along with the clinical data are represented in a multi-relational description language as reported in Table 1: for each entity involved in the domain, i.e. patient, ovaric stimulation protocol, hormone pharmacological treatment, oocyte aspired and oocyte components, a set of descriptive attributes are reported along with the existing relationships (italic font in the table).

As to the knowledge extraction phase, we apply two submodules devoted to specific tasks to solve. The first one concerns the application of clustering techniques to identify similarities among patients. Indeed, the aggregation of patients that show a similar behavior could be useful to better understand the conditions under which a pregnancy could be obtained. Once the clustering has been taken place, for each cluster a set of rules are inducted, able to identify relationships between stimulation protocol and health conditions or between number and quality of oocytes obtained. Thus the model can support the clinicians in the stimulation protocol definition for new unseen patients that show a similarity degree with a cluster. On the other hand, the induction mechanism would infer a set of rules to automatically classify unseen oocytes for similar patients in order to support oocyte selection.

Due to the complexity of data in our application domain, the multi-relational techniques previously presented were exploited. In particular, we use (`APAM`) [1] as it is very robust with respect to the existence of outliers (i.e., data points that are very far away from the rest of the data points). This is a fundamental characteristic for our application domain as clinicians can adopt very different stimulation protocols according to their experience and, more importantly, according to the patients'health problems and characteristics. Furthermore, the `APAM` algorithm is based on an approximate evaluation of the clustering membership thus allowing to tackle the uncertainty in the data.

As to concern the inference process, the incremental multi-relational inductive logic system [3] was exploited as its incremental capability makes it able to learn a satisfiable model even with few examples and, more importantly, to revise the learned rules as new examples are provided without restart the learning step from scratch.

5 Evaluation and Discussion

The overall framework was tested on a preliminary set of data collected by the Department of Endocrinology and Molecular and Clinical Oncology of the University Federico II of Naples including clinical data of the patients along with the corresponding light microscope images of the oocytes. The dataset consisted of about 50 patients and 200 oocytes images.

The image processing devoted to the extraction of morpho-structural features from the oocyte images was able to correctly extract the measures of the oocyte

Table 1. Attributes and Relations (italic font) used to describe the entities (patient, protocol, hormone, oocyte, component) involved in the application domain

Predicate	Domain
age(patient,val)	val: integer
bmi(patient,val)	val: real
basal_FSH(patient, val)	val: real
basal_LH(patient, val)	val: real
male_infertility(patient,inf)	inf: none, oligospermia, azoospermia, teratospermia
female_infertility(patient,inf)	inf: tubaric,thyroid,prolactin,uterine,endometrial
stimulation_protocol(patient,IP)	IP: nominal (performed protocol identification)
hormone_stimulation(IP, h)	h: agoGnRH, antagoGnRH,rFH,rLH,HMG, uFSH
dose(h,val)	val: real (provided dose)
timing(h,val)	val: real (duration of the stimulation type)
duration_of_stimulation(IP,val)	val: real (duration of the stimulation protocol)
estradiol_level(IP,val)	val: real (estradiol level at the HCG injection day)
HCG_dose(IP,val)	val: real
aspiration_timing(IP,val)	val: real (HCG injection to oocyte aspiration)
part_of(patient,oocyte_id)	oocyte_id: nominal (oocyte identification)
dimension(oocyte_id, val)	val: real (μm)
part_of(oocyte_id, component)	cytoplasm
dimension(component, val)	val: real (μm)

and cytoplasm diameters in the 90% of the cases with a good approximation with respect to the real diameters manually extracted. This is a good result if we consider that the available images are affected by some noise and low contrast, which make quite difficult the detection of the oocyte. Indeed, just in few cases the method failed to extract correctly the oocyte region, due to the poor quality of the original images. An example of problematic image is given in Fig. 6a where a very low contrast between the oocyte and the background is present, leading to a poor segmentation result, as shown in Fig. 6c. In all other cases, the features extraction step worked quite well, providing a good approximation of the oocyte radius. Indeed, in cases when the procedure correctly was able to extract the oocyte dimension, the standard deviation of the difference between the measured and the real diameter was about 11 μm, i.e. it represents the 0.06% of the real measure.

Oocyte region identification is a very crucial step of the proposed approach since it affects the successive measurements of the oocyte and cytoplasm diameters. For this reason, an automatic evaluation method able to identify such cases has been introduced. It is based on the definition of a threshold value for the oocyte diameter size taken from medical literature. Indeed, it is known that for a good quality oocyte the diameter can range in the interval $[115\mu m - 165\mu m]$. Thus, a value greater/smaller than this threshold indicates that the procedure has failed the detection goal. At this moment, the cytoplasm region detection step based on the Hough transform is always able to detect the cytoplasm region

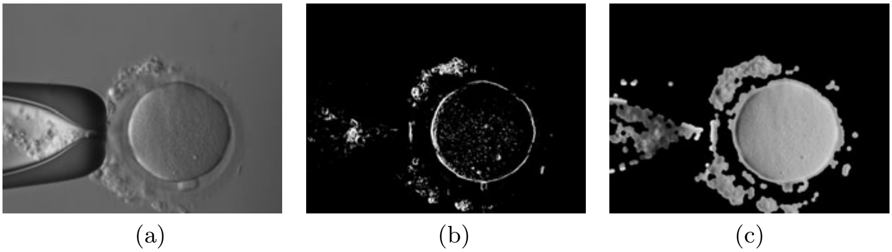


Fig. 6. (a) A problematic image (b) after the pre-processing step and (c) the final oocyte region detection result

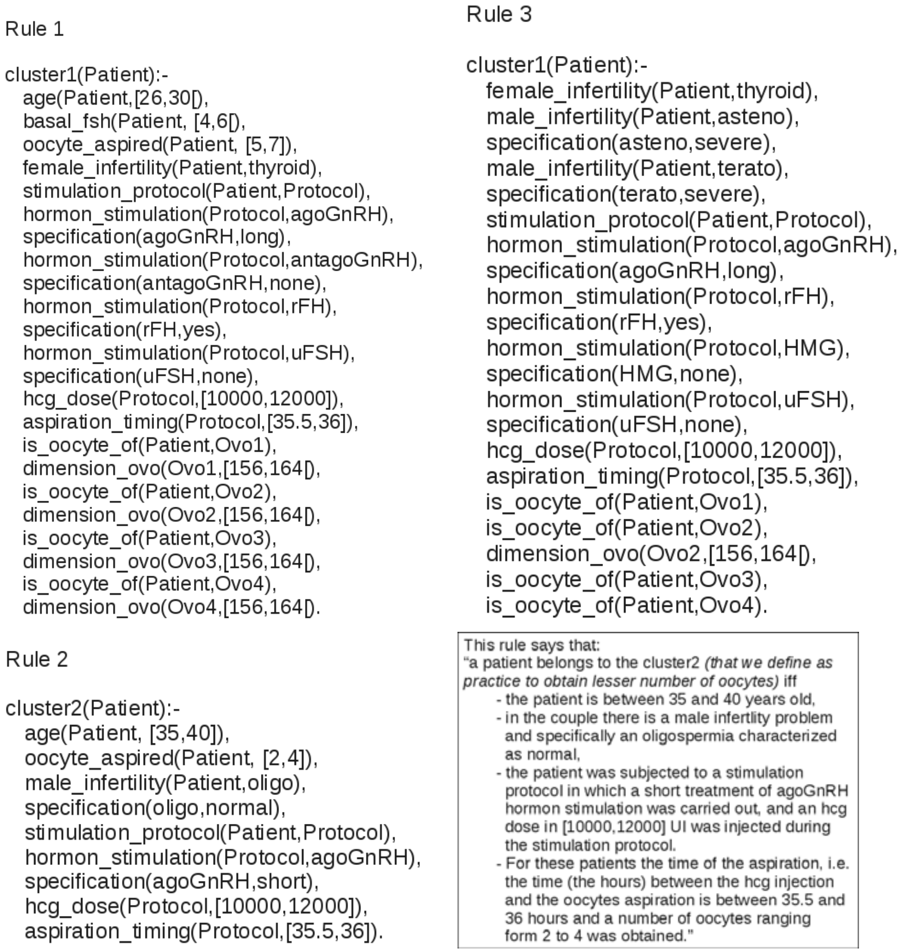
but with a major effort in computational time since it works with no a priori knowledge on the oocyte dimension.

As to concern the application of the multi-relational techniques, the experimental outcomes revealed some interesting features correlating health conditions to the stimulation protocol, that could confirm the medical literature. In particular the clustering was setted so to generate two clusters in order to differentiate good from not good practice.

The data aggregation step results in a first clustering joining couples characterized by few female infertility conditions and many and severe male infertility ones. For such couples mainly long stimulation protocol was carried out resulting in a production of a mean of 6 oocytes for patient, then this can be labelled as *protocol stimulation practice in order to obtain a greater number of oocytes.* This cluster characterization was confirmed by the rule induction step that exactly was able to grasp the concept by inferring rules such the ones reported in Fig. 7. Specifically Rule 1 and Rule 3, say that couples with some female infertility factors and severe male infertility factors, for which the patients were subjected to a *long* stimulation protocol with respect to the other patients, a great number of oocytes was obtained. Furthermore, this rules give further information characterizing the obtained oocytes, i.e. that in such conditions almost all of them have medium oocyte/cytoplasm dimension.

On the other hand, the other cluster result aggregated couples characterized by many female infertility conditions and male infertility conditions of different seriousness. For such couples prevalently short stimulation protocols were carried out and a lesser number of oocytes were obtained (3 in average). This can be labelled as *protocol stimulation practice in order to obtain a lesser number of oocytes.* Even in this case, the rule induction phase was able to learn the concept as reported in Fig. 7, Rule 2. It is worth to note the human understandability of the learning rules as shown in the interpretation reported in Fig. 7 for Rule 2. Hence, the rules reported in Fig. 7 can be exploited as a classifier on new unseen patients and a validation on the fly of obtained rules is planned to be performed as soon as new available data will be available.

An in deep analysis of the oocytes quality in the clustered data was performed according to the further information provided by the clinicians about the oocyte fertilization and growing. This analysis revealed that most of the data about



Rule 1

```
cluster1(Patient):-
   age(Patient,[26,30[),
   basal_fsh(Patient, [4,6[),
   oocyte_aspired(Patient, [5,7]),
   female_infertility(Patient,thyroid),
   stimulation_protocol(Patient,Protocol),
   hormon_stimulation(Protocol,agoGnRH),
   specification(agoGnRH,long),
   hormon_stimulation(Protocol,antagoGnRH),
   specification(antagoGnRH,none),
   hormon_stimulation(Protocol,rFH),
   specification(rFH,yes),
   hormon_stimulation(Protocol,uFSH),
   specification(uFSH,none),
   hcg_dose(Protocol,[10000,12000]),
   aspiration_timing(Protocol,[35.5,36]),
   is_oocyte_of(Patient,Ovo1),
   dimension_ovo(Ovo1,[156,164[),
   is_oocyte_of(Patient,Ovo2),
   dimension_ovo(Ovo2,[156,164[),
   is_oocyte_of(Patient,Ovo3),
   dimension_ovo(Ovo3,[156,164[),
   is_oocyte_of(Patient,Ovo4),
   dimension_ovo(Ovo4,[156,164[).
```

Rule 2

```
cluster2(Patient):-
   age(Patient, [35,40]),
   oocyte_aspired(Patient, [2,4]),
   male_infertility(Patient,oligo),
   specification(oligo,normal),
   stimulation_protocol(Patient,Protocol),
   hormon_stimulation(Protocol,agoGnRH),
   specification(agoGnRH,short),
   hcg_dose(Protocol,[10000,12000]),
   aspiration_timing(Protocol,[35.5,36]).
```

Rule 3

```
cluster1(Patient):-
   female_infertility(Patient,thyroid),
   male_infertility(Patient,asteno),
   specification(asteno,severe),
   male_infertility(Patient,terato),
   specification(terato,severe),
   stimulation_protocol(Patient,Protocol),
   hormon_stimulation(Protocol,agoGnRH),
   specification(agoGnRH,long),
   hormon_stimulation(Protocol,rFH),
   specification(rFH,yes),
   hormon_stimulation(Protocol,HMG),
   specification(HMG,none),
   hormon_stimulation(Protocol,uFSH),
   specification(uFSH,none),
   hcg_dose(Protocol,[10000,12000]),
   aspiration_timing(Protocol,[35.5,36]),
   is_oocyte_of(Patient,Ovo1),
   is_oocyte_of(Patient,Ovo2),
   dimension_ovo(Ovo2,[156,164[),
   is_oocyte_of(Patient,Ovo3),
   is_oocyte_of(Patient,Ovo4).
```

This rule says that:
"a patient belongs to the cluster2 *(that we define as practice to obtain lesser number of oocytes)* iff
- the patient is between 35 and 40 years old,
- in the couple there is a male infertlity problem and specifically an oligospermia characterized as normal,
- the patient was subjected to a stimulation protocol in which a short treatment of agoGnRH hormon stimulation was carried out, and an hcg dose in [10000,12000] UI was injected during the stimulation protocol.
- For these patients the time of the aspiration, i.e. the time (the hours) between the hcg injection and the oocytes aspiration is between 35.5 and 36 hours and a number of oocytes ranging form 2 to 4 was obtained."

Fig. 7. Sample learned rules

oocytes in the first cluster concern poor quality oocytes as they do not grown in the following days after fertilization or they present an high fragmentation rate (> 10) at the embryo stage (fragmentation is a process where portions of the embryo's cells have broken off and are separate from the nucleated portion of the cell. According to the medical literature, little or no fragmentation are preferable as embryos with more than 25% fragmentation have a low implantation potential). On the contrary, the data in the second cluster regard good quality oocytes that in almost all of case grows in embryos and with a low rate of fragmentation (≤ 10) at embryo stage.

This preliminary experiments showed that multi-relational techniques could be able to grasp hidden relationships in data. Better results could be obtained by extending and considering the set of clinical data with more parameters in both the stimulation protocol and in the definition of health conditions, and by extending the image processing module in order to extract more features from the oocyte images and from other images that follow the oocyte development after fertilization, i.e. zygote and embryo images.

6 Conclusion and Future Work

In this paper a tool to support bio-medical applications is presented. The existing approaches work at different level according to the data starting point, i.e. images or clinical data. The proposed framework involves both data with the aim of put together the automatically extracted morpho-structural data of the image and the clinical data with the aim of further elaboration steps devoted to discovery relationships among data.

Future work will concern the extension of the clinical data by considering more parameters in both the stimulation protocol and in the definition of health conditions, and the extension of the image processing module in order to extract more features from the cell images. Finally, an exhaustive experimental phase is planned to be carried out.

Acknowledgment

This work is partially supported by the italian project "`MIUR-FAR 08-09`: The Molecular Biodiversity LABoratory Initiative (`MBLab-DM19410`)."

References

1. Di Mauro, N., Basile, T.M.A., Ferilli, S., Esposito, F.: Approximate relational reasoning by stochastic propositionalization. In: Ras, Z.W., Tsay, L.-S. (eds.) Advances in Intelligent Information Systems. Studies in Computational Intelligence, vol. 265, pp. 81–109. Springer, Heidelberg (2010)
2. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification. Wiley-Interscience Publication, Hoboken (2000)
3. Esposito, F., Ferilli, S., Fanizzi, N., Basile, T.M.A., Di Mauro, N.: Incremental learning and concept drift in INTHELEX. Intell. Data Anal. 8(3), 213–237 (2004)
4. Gonzalez, R.C., Woods, R.E.: Digital Image Processing, 3rd edn. Prentice-Hall, Inc., Upper Saddle River (2006)
5. Griffin, J., Emery, B.R., Huang, I., Peterson, M.C., Carrell, D.T.: Comparative analysis of follicle morphology and oocyte diameter in four mammalian species (mouse, hamster, pig, and human). Journal of Experimental & Clinical Assisted Reproduction 3, 2 (2006)
6. Haralick, R.M., Sternberg, S.R., Zhuang, X.: Image analysis using mathematical morphology. IEEE Trans. Pattern Anal. Machine Intell. 9(4), 532–549 (1987)
7. Losa, G., Peretti, V., Ciotola, F., Cocchia, N., De Vico, G.: The use of fractal analysis for the quantification of oocyte cytoplasm morphology. Fractals in Biology and Medicine, 75–82 (2005)

8. Montag, M., Schimming, T., Kster, M., Zhou, C., Dorn, C., Rsing, B., van der Ven, H., van der Ven, K.: Oocyte zona birefringence intensity is associated with embryonic implantation potential in ICSI cycles. Reprod. Biomed. Online 16(2), 239–244 (2008)
9. Morales, D.A., Bengoetxea, E., Larranaga, P.: XV-Gaussian-Stacking Multiclassifiers for Human Embryo Selection. In: Data Mining and Medical Knowledge Management: Cases and Applications. IGI Global Inc. (2009)
10. Morales, D.A., Bengoetxea, E., Larrañaga, P., García, M., Franco, Y., Fresnada, M., Merino, M.: Bayesian classification for the selection of in vitro human embryos using morphological and clinical data. Computer Methods and Programs in Biomedicine 90(2), 104–116 (2008)
11. Muggleton, S., De Raedt, L.: Inductive logic programming: Theory and methods. Journal of Logic Programming 19/20, 629–679 (1994)
12. Rjinders, P., Jansen, C.: The predictive value of day 3 embryo morphology regarding blastocysts formation, pregnancy and implantation rate after day 5 transfer following in-vitro fertilisation or intracytoplasmic sperm injection. Hum. Reprod. 13, 2869–2873 (1998)
13. Schmutzler, A.G., Rieckmann, O., Sushma, V., Kupka, M., Montag, M., Prietl, G., Krebs, D., Van der Ven, H.: Ideal oocyte morphology depends on estradiol concentration. Hum. Reprod. 13(suppl.), 179 (1998)
14. Scott, L., Alvero, R., Leondires, M., Miller, B.: The morphology of human pronuclear embryos is positively related to blastocysts development and implantation. Hum. Reprod. 15, 2394–2403 (2000)
15. Ullman, J.D.: Principles of Database and Knowledge-Base Systems, vol. I. Computer Science Press (1988)
16. Uyar, A., Nadir Ciray, H., Bener, A., Bahceci, M.: 3P: Personalized pregnancy prediction in IVF treatment process. In: Weerasinghe, D. (ed.) eHealth 2008. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, vol. 1, pp. 58–65. Springer, Heidelberg (2009)
17. Veek, L.L.: An Atlas of Human Gametes and Conceptuses: An Illustrated Reference for Assisted Reproductive Technology. Parthenon (1999)

Data Driven Generation of Fuzzy Systems: An Application to Breast Cancer Detection

Antonio d'Acierno[1,⋆], Giuseppe De Pietro[2], and Massimo Esposito[2]

[1] Institute of Food Sciences - Italian National Research Council
Via Roma 64, Avellino, Italy
dacierno.a@isa.cnr.it

[2] Institute of High Performance Computing and Networking
Italian National Research Council
Via Pietro Castellino 111, Napoli, Italy

Abstract. The detection of diseases often can be formalized as a decision problem that typically has to be solved merging uncertain information; diagnostic tools, intended to aid the physician in interpreting the data, besides attaining the best possible correct classification rate, should furnish some insight into how the problem has been decided. Fuzzy logic is a well known successful attempt to automatize the human capability to reason with imperfect information; fuzzy systems are rule-based so that they can easily provide motivations for their decisions, after having verified some additional conditions.

In this paper we describe a six-steps data driven methodology to automatically build fuzzy systems with a user defined number of rules; almost each step can be approached using several strategies and we thus describe an implementation of the proposed solution. Then, we test our systems on a well known and widely used data set of features of images of breast masses and, having the number of rules varying, we show results both in terms of correct classification rates and in terms of systems' confidence in the obtained decisions. Finally, we select the number of rules that produces the most interpretable and *trustworthy* system; such a system is described in details and tested.

1 Introduction

To increase the change of successful treatments, early detection of almost any disease is a key factor and the *detection* can be often formulated as a binary decision making problem; uncertainty in form of information incompleteness, impreciseness, fragmentariness, not fully reliability, vagueness and contradictoriness often affects these problems [7] so that the ultimate diagnosis can be difficult to obtain even for a medical expert. As a consequence, many computerized diagnostic tools intended to aid the physician in interpreting the data have been developed in the past few decades.

[⋆] Corresponding author.

R. Rizzo and P.J.G. Lisboa (Eds.): CIBB 2010, LNBI 6685, pp. 203–214, 2011.
© Springer-Verlag Berlin Heidelberg 2011

It is widely accepted that a diagnostic tool should possess three characteristics [4]; first, it must attain the best possible performance in terms of correct classification rate (CR) while it would be desirable the system not only provides a diagnosis but also a numerical value (the *confidence* χ) representing the degree to which the system is confident in the solution. It would be also useful if the physician is not faced with a black box that simply outputs answers but the system should provide some insight into how the solution has been derived (*interpretability*). These requirements are often in contrast. Diagnostic tools, however, typically have unequal classification error costs so that straight CR cannot be assumed as a careful measure of the goodness of the classifier; a Receiver Operating Characteristic (ROC) graph [3] has been showed to be a more accurate technique for selecting classifiers based on their performance. We guess that also χ can be used for selecting classifier; in facts, a *good* classifier should be highly confident with correctly classified examples while it should be doubtful with misclassified data points.

Fuzzy logic (a precise logic of imprecision and approximate reasoning [16]) is an attempt at the formalization and mechanization of two remarkable human capabilities: the capability to converse, reason, and make rational decisions in an environment of imperfect information, and the capability to perform a wide variety of physical and mental tasks without any measurement and any computation [16]. Fuzzy logic is a multi-valued logic based on fuzzy set theory [15]; a fuzzy set is a set whose elements have a continuum of grades of membership described using *membership functions*.

A Fuzzy Inference System (FIS) is a system that (tries to) solve a (typically complex and nonlinear) problem by utilizing fuzzy logic methodologies and it is composed (see figure 1) of a fuzzifier (which translates real-valued inputs into fuzzy values), of an inference engine (that applies a fuzzy reasoning mechanism to obtain a fuzzy output), of a defuzzifier (to translate this latter output into a crisp value), and of a knowledge base (containing both rules and membership functions).

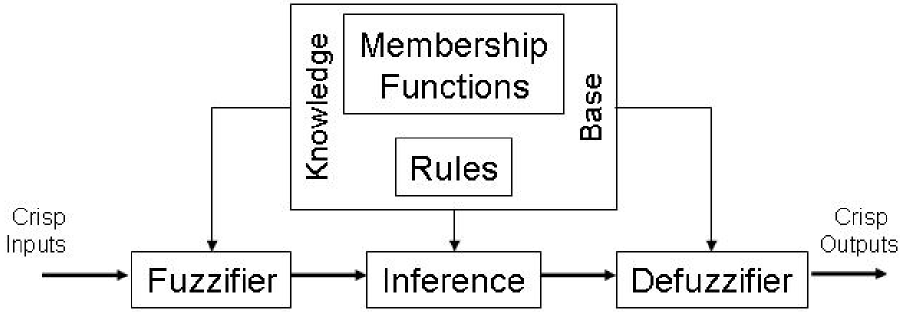


Fig. 1. The basic components of a fuzzy inference system

The inference process is performed by the engine using the rules contained in the rule base, each rule being in the form:

$$\textbf{if} \quad \textit{antecedent} \quad \textbf{then} \quad \textit{consequent}$$

where the antecedent is a fuzzy-logic expression composed of one or more simple fuzzy expressions connected by fuzzy operators (the fuzzy equivalent of the classical *and*, *or* and *not*), and the consequent is an expression that assigns fuzzy values to the output variables (Mamdani systems [8]), i.e.:

$$\textbf{if} \quad \textit{service} \quad \textbf{is} \quad \textit{good} \quad \textbf{then} \quad \textit{tip} \quad \textbf{is} \quad \textit{average}$$

Differently, in Takagi-Sugeno(TS) systems [11], the consequent expresses output variables as a function that maps the input space into the output space, for example:

$$\textbf{if} \quad \textit{service} \quad \textbf{is} \quad \textit{good} \quad \textbf{then} \quad \textit{tip} = f(\textit{service})$$

where f (typically) is a first order linear function that becomes a constant in zero-order TS systems.

Fuzzy modeling is the task of identifying the parameters of a FIS so that a desired behavior is attained. When the available knowledge is complete and the problem space is not very large the system can be constructed directly (*knowledge driven approach*) using knowledge elicited from human experts. Alternatively, an emerging solution is represented by *data driven fuzzy modeling*, that is being more and more applied in a wide variety of fields even if the rule base generated automatically from data may not be fully interpretable especially because of redundancy in the rules and in the fuzzy sets. Three conditions can be defined [5] to obtain an interpretable fuzzy model: *(i)* the fuzzy partition must be readable (the fuzzy sets can be interpreted as linguistic labels), *(ii)* the set of rules must be as small as possible, and *(iii)* the if-part of the rules should be derived from a subset of independent variables rather than from the full set.

In this work we describe a methodology to automatically extract the knowledge base and we show some of the results obtained with reference to a well known and widely used data set of features of images of breast masses. In our approach, the number of rules selected is user-defined and the optimal number of rules for the dataset under test has been defined by exploiting not only the correct classification rate but also a confidence based criterion with the final aim of obtaining an highly understandable system with an interesting overall performance.

The paper is organized as follows. Some relevant related approaches available in the literature are reviewed in section 2 while the proposed method is described in section 3. Experimental conditions and results are discussed in section 4, conclusions and open problems being the concerns of section 5.

2 Related Works

The method proposed in [10] to generate TS fuzzy models firstly assumes that fuzzy sets are described by Gaussian membership functions for which centers and widths have to be estimated; the rules are then generated iteratively until a user defined maximum number of rules (R_{max}) is reached or a performance index (usually MSE) is achieved. The fuzzy antecedents of the first rule are evaluated as the mean and the standard deviation of training data while the consequent is evaluated using least squares techniques; rules are iteratively added selecting as center the vector with the worse error (some conditions are introduced to exclude the change of an outlier data point to be considered as a new rule's center). Parameters are also tuned using an hybrid learning algorithm. Even if the presented results show the goodness of the method for problems with a low number of independent variables, the antecedent of each rule is based on the whole set of variables so making the obtained fuzzy model not really interpretable when problems with many independent variables are considered [5].

In [4], 11 feature selection methods and 3 fuzzy modeling methods are combined and tested using two well known medical binary datasets (namely the Wisconsin breast cancer data [14] and the PIMA Indians diabetes data, both available at UCI Repository) and an industrial dataset (welding flaw data). Results for a single run of a stratified fivefold cross validation are presented in terms of average accuracy, also highlighting the top five combinations. Then, only for the best combination, the results are also reported in terms of area under the ROC curve, sensitivity and specificity (among others parameters). With reference to the WBCD, the best average accuracy obtained is 97.17% using three variables (x_{28}, x_{21}, x_{22}); no details are given for the rules in the first rank system while, for the second rank, we have that for each variable (the same as the first rank system) 5 membership functions have been used and 250 rules have been generated so deriving a not fully understandable system.

Since in rule based systems built starting from numerical data redundancy often exists in form of redundant rules and similar fuzzy sets generating unnecessary structural complexity and decreasing the interpretability of the system, in [1] a simplification method is proposed after rules' extraction (by means of fuzzy clustering associated with a fuzzy partition validity index) and parameters' estimation (by means of a gradient descent algorithm). Results are shown with reference to function approximation, dynamical system identification and mechanical property prediction for hot rolled steels. No result are reported for problems with an high number of variables.

Hierarchical TS fuzzy systems have the advantage that both the number of rules and the number of fuzzy operations involved can be reduced significantly when compared with those requested by single level systems. An automatic way of evolving hierarchical TS fuzzy system using probabilistic incremental program evolution is the concern of [2]; interesting results are shown for some non linear system identification problems (Makey-Glass chaotic time series prediction problem, and the Iris and Wine classification problems).

In [13] it is presented a fuzzy rule based decision support systems for the diagnosis of coronary artery disease (CAD) automatically generated from an initial annotated dataset, by means of a four stage methodology. A set of crisp rules (obtained from a postpruned induction tree based on the well known C4.5 algorithm) are fuzzified using two sigmoidal membership functions (a decreasing one expressing the linguistic term LOW and an increasing one expressing $HIGH$). Rules are then weighted using a likelihood ratio and parameters are optimized using the healed topographical multilevel single linkage algorithm. The reported results clarify that fuzzification and optimization significantly improve the performance of the pruned tree. No information is shown on the obtained rules, so that the overall interpretability of the obtained system cannot be judged.

Genetic algorithms are used in [9] to produce fuzzy systems for an older version of WBCD (444 benign cases and 239 malignant cases, 9 variables) using a fitness function that tries to combine classification performance, the interpretability of the system and a term adding pressure towards systems with low quadratic error. For each variable there are two orthogonal trapezoidal membership functions and the number of rules is assumed to be a user-configurable parameter (limited to be between 1 and 5). After 120 evolutionary runs, the best system has 3 rules for benign cases (one of them is found to be never triggered by any of the input case) and shows a $CR = 98\%$ on the whole data set. Cross validation has been also performed but the choice of learning-set and test-set is performed anew at the outset of every evolutionary run, so deriving not fully generalizable results.

Last, it is worth citing the three stages generic methodology (crisp rules extraction, fuzzification and optimization) proposed in [12] that is able to integrate alternative techniques in each stage. Specific implementations (using decision trees for crisp rules extractions and four optimization strategies) are tested on several well known datasets; on WBCD, the best solution obtains, in a single run of a ten-fold cross validation, a $CR = 95.15$. The number of rules used equals the number of classes, but each rule is not really interpretable since it combines all the crisp rules of a given class.

3 The Proposed Approach

The main components of the implemented system are sketched in figure 2. First, the available data are used to extract crisp rules; since in the current implementation a decision tree is used, each leaf node can be easily translated into a crisp rule parsing the tree from the root to the leaf itself and assuming the tests encountered along the path form the conjunctions of the rule's antecedent, while the class label of the leaf node is clearly assumed to be the rule consequent (zero order TS FIS will be so used).

Here, having in mind to implement a general methodology, we do not adopt any pruning technique, so that we obtain several rules that are likely to over fit the data; for such a reason, we use a well defined stage (the *Selector*) that using some heuristics selects a proper subset of rules for each class (we choose the same number of rules for each class).

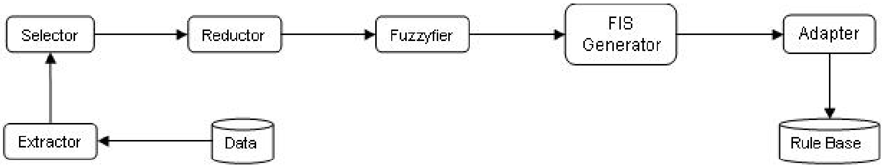


Fig. 2. The proposed approach

The antecedent of the selected i^{th} rule (in the current implementation it clearly corresponds to the path path from the root node to a given leaf) we obtain is in the form of conjunction of conditions:

$$A_i = (x_{1i}\theta_{1i}c_{1i}) \wedge (x_{2i}\theta_{2i}c_{2i}) \wedge \ldots \wedge (x_{ki}\theta_{ki}c_{ki}) \tag{1}$$

where each x_{ji} is the feature used in the node, θ_{ji} is a standard comparison operator $(<, \leq, >, \geq)$ and c_{ji} is a crisp threshold.

Depending on the strategy used to extract the rules, each antecedent A_i is generated with some characteristics. For example, when a decision tree is used, as in the current implementation, we have that, in an antecedent, the same feature could appear several times (see figure 3), and, consequently, we have implemented a *Reductor* that translates each antecedent C_i in an antecedent we have named in *standard form* $\tilde{A}_i$: an antecedent is in standard form if each feature appears at most one time. Given the form of the antecedent (equation 1), and $\forall i$, we act as follows. We collect the terms that refer to the same feature and with the same operator; because of the conjunctions among conditions, these sets can be easily simplified as follows (we assume, without loss of generality and for the sake of simplicity, that just two conditions share the same feature):

$$(x_{mi}\theta_{mi}c_{mi}) \wedge (x_{mi}\theta_{ni}c_{ni}) \rightarrow x_{mi} > max(c_{mi}, c_{ni}) \tag{2}$$

if $\theta_{ni} \in \{>, \geq\}$, or

$$(x_{mi}\theta_{mi}c_{mi}) \wedge (x_{mi}\theta_{ni}c_{ni}) \rightarrow x_{mi} < min(c_{mi}, c_{ni}) \tag{3}$$

if $\theta_{ni} \in \{<, \leq\}$.

After this step we have that the i^{th} condition is already in standard form or some features appear in two conditions (greater than a threshold and less than another threshold, i.e. *between* two thresholds); if this is the case we apply:

$$(x_{ji} > c_{mi}) \wedge (x_{ji} < c_{ni}) \rightarrow x_{mi} \in [c_{mi}, c_{ni}] \tag{4}$$

if $c_{mi} < c_{ni}$, otherwise we simply delete the rule.

Once rules are expressed with antecedents in standard form, they can be easily expressed in fuzzy terms; we have chosen to partition the universe of each feature into three intervals: *low*, *medium* and *high*. Then, each condition for each antecedent is obviously translated as follows:

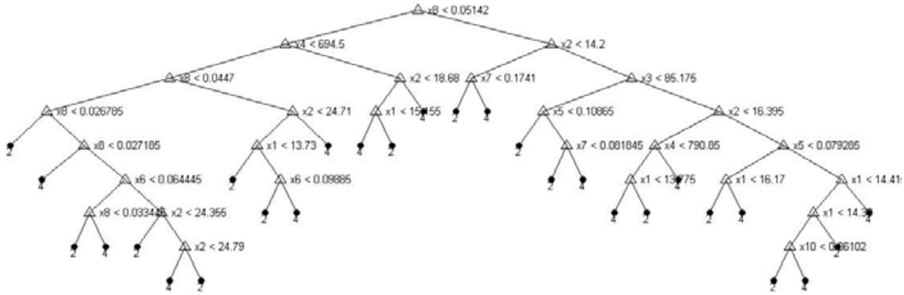


Fig. 3. A decision tree

$$(x_{ji} < c_{mi}) \rightarrow x_{ji} \;\; is \;\; low \tag{5}$$

$$(x_{ji} \in [c_{mi}, c_{ni}]) \rightarrow x_{ji} \;\; is \;\; medium \tag{6}$$

$$(x_{ji} > c_{mi}) \rightarrow x_{ji} \;\; is \;\; high \tag{7}$$

the consequent being the class associated with the leaf node.

In the last step, membership functions have to be adjusted and tuned; a plethora of methods for such a task have been proposed, so here it is just worth to be noted that a back-propagation algorithm is used in the current implementation.

4 Experimental Results

4.1 The Wisconsin Breast Cancer Dataset

Breast cancer is one of the most common cancer among women; the presence of a breast mass is an alert sign, but it does not always indicate a malignant cancer. Fine needle aspiration (FNA) of breast masses is a cost-effective, non-traumatic, and mostly non-invasive diagnostic test that obtains information needed to evaluate malignancy.

The Wisconsin breast cancer diagnosis (WBCD) database [14] is the result of the efforts made at the University of Wisconsin Hospital for accurately diagnosing breast masses based solely on an FNA test. Features computed describe characteristics of the cell nuclei present in the image, and ten visually assessed characteristics of an FNA sample considered relevant for diagnosis were identified (see table 1).

The mean, standard error, and "worst" or largest (mean of the three largest values) of these features were computed for each image, resulting in 30 features. The diagnostics in the WBCD database were furnished by specialists in the field; the used version of the database consists of 357 benign cases and 212 malignant cases. Having in mind to obtain highly understandable systems, we decided to use just the first 10 variables that can be easily estimated simply having a look at the image.

Table 1. The variables of the Wisconsin dataset

1	radius	mean of distances from center to points on the perimeter
2	texture	standard deviation of gray-scale values
3	perimeter	
4	area	
5	smoothness	local variation in radius lengths
6	compactness	perimeter^2 / (area - 1.0)
7	concavity	severity of concave portions of the contour
8	concave points	number of concave portions of the contour
9	symmetry	
10	fractal dimension	"coastline approximation" - 1

4.2 Experimental Parameters

The proposed approach has been implemented using functions available in the standard version of the R2007a 64-bit version of MATLAB.

In the current implementation, to extract crisp rules we use a full decision tree (without pruning) with a Gini's diversity index as split criterion and a split minimum factor equal to 1. Given a user defined number of rules R (assumed to be even), we select, for each class, the $\frac{R}{2}$ most covering leaf nodes. For each selected node, we derive a standard if-then rule that is then fuzzified.

The outcoming FIS is finally trained using the well know ANFIS [6] algorithm with back-propagation and assuming 500 as the maximum number of epochs and 0.1 as the training error goal. For each FIS we compare the defuzzified output with a threshold τ to classify the sample; the chosen τ is the one that maximizes the CR on the learning set.

4.3 Numerical Results

To test the system we use a ten-fold cross validation that is repeated 100 times and we measure the CR on the learning set (LS), on the test set (TS), and on the full set (FS) for both unadapted FISs (UFIS) and adapted ones (AFIS); averaged correct classification rates ($\overline{CR}$) as the number of rules varies are reported in table 2.

First, it should be noted that also unadapted FISs show an interesting performance ($\overline{CR} > 90\%$ on the TS). An interesting feature is that, with some approximation, $\overline{CR}$ on TS (clearly the most interesting one) increases with the number of rules for unadapted systems while it decreases for adapted systems. The best result we obtained is with 4 rules where we have a $\overline{CR} \approx 93\%$ on the TS. We also show the average number of variables used (V) that increases as the number of rules does.

For each FIS we evaluate the ROC graphs (on the test sets) that are then vertically averaged [3] to compute the area under the curve (AUC, equivalent to the probability that the classifier will rank a randomly chosen positive instance higher than a randomly chosen negative instance) as the number of rules varies.

Table 2. Averaged correct classification rate as the number of rules varies. Last column shows the average number of variables used.

	UFIS			AFIS			
R	LS	TS	FS	LS	TS	FS	V
2	92.01	90.63	91.32	93.15	92.22	92.69	**5.21**
4	92.14	90.95	91.54	93.48	**92.66**	93.07	6.23
6	91.77	90.45	91.11	93.22	92.34	92.78	7.19
8	92.12	90.81	91.46	92.62	91.68	92.15	7.70
10	92.19	91.00	91.59	92.22	91.13	91.68	8.10
12	92.27	91.13	91.70	92.09	91.02	91.55	8.26

Table 3. The area under the ROC graph as the number of rules varies

R	2	4	6	8	10	12
AUC	0.965	0.969	0.972	0.970	0.969	0.967

Results (table 3) show that the number of rules does not significantly affects the AUC.

At each iteration, and for each sample in TS, we define the confidence χ as:

$$\chi = \frac{|D - \tau|}{\alpha} \tag{8}$$

D being the defuzzified output of the FIS and α a normalizing factor so that $\chi \in [0, 1]$.

Starting from the confusion matrix on the training set of each system, we measure the cases correctly classified (N_{TP} and N_{TN}) with $\chi > 0.7$ and the number of instances incorrectly classified (N_{FP} and N_{FN}) with $\chi < 0.3$. Table 4 shows the results we obtained (in percentage) as the number of rules varies on the test sets. The most trustworthy system (on average, and also in detecting both false positives cases and true negatives ones) is the system with just two rules while, for example, a system with six rules is highly confident in detecting true positives samples.

Table 4. Number of cases (in percentage) with an *appropriate* χ

R	N_{TP}	N_{FP}	N_{FN}	N_{TN}	AVG
2	64.16	**44.32**	51.21	**70.92**	**57.65**
4	77.09	34.75	58.54	55.45	56.46
6	**81.98**	32.08	**61.53**	31.82	51.85
8	64.78	34.55	49.68	31.85	45.22
10	54.33	37.29	41.25	44.81	44.42
12	50.62	37.35	40.51	46.15	43.66

4.4 The Selected FIS

The best classification rate is obtained with four rules while, with just two rules, we have the most trustworthy (on average) system; here we decide to favor both the interpretability and the overall confidence of the system and so, as a result, we chose the system with two rules. This system, moreover, shows the best confidence in detecting true negative and false positive cases; figure 4 shows the confidences' distribution obtained on the test sets in the experiments.

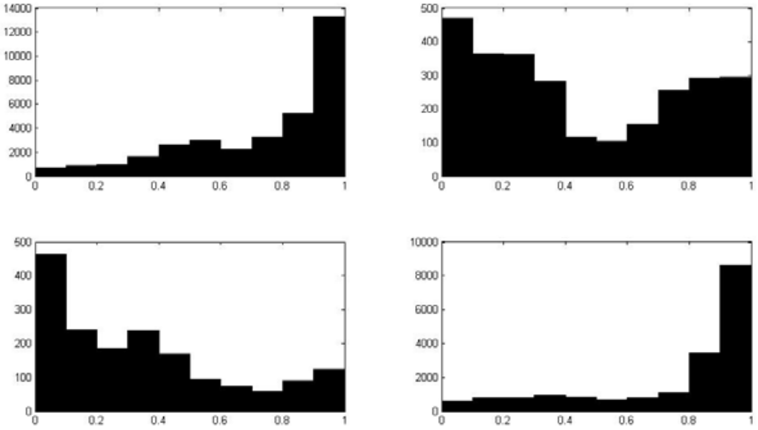


Fig. 4. The distribution of the confidences on the test sets in the experiments by the two rules systems

We then apply our methodology using the full data set and we and obtain the following two rules:

- Rule 1. If (texture is low) and (area is low) and (concave points is low) then (Class is Benign)
- Rule 2. If (texture is high) and (perimeter is high) and (smoothness is high) and (concavity is high) and (concave points is high) then (Class is Malignant)

that use just six variables and at most two linguistic terms (low and high). Membership functions are then adapted; figure 5 (left) shows the root mean square errors obtained, showing that the training process converges quickly and the adapted membership functions are shown in figure 6. The trained system uses a threshold value $\tau = 3.038$ (benign cases are represented with 2 while malignant ones with 4), has the ROC curve shown in figure 5 (right), its area under the curve is 0.972 and the correct classification rate is 93.85 (both on the full data set).

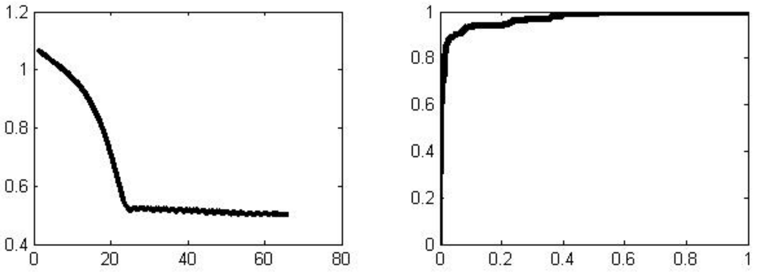


Fig. 5. The training error for the selected system (left) and the ROC curve (right)

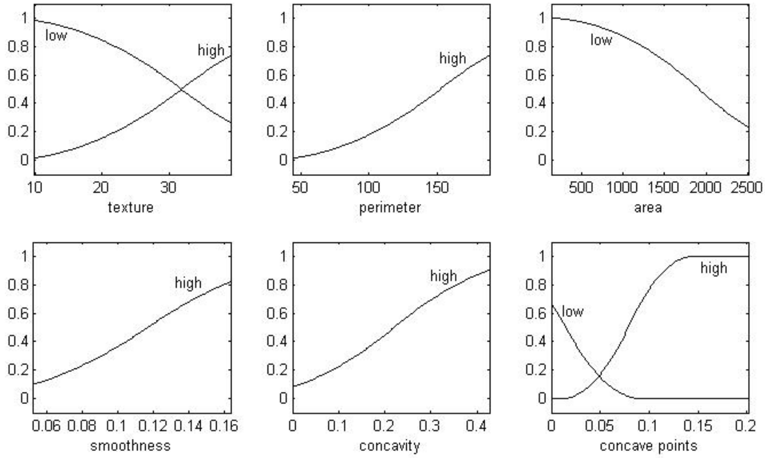


Fig. 6. Themembership functions of the selected system

5 Conclusions and Future Directions

Data driven methodologies in fuzzy modeling are being applied in a wide variety of fields even if attention must be paid when the aim is to obtain interpretable systems. In this paper we describe a six steps general methodology aimed at producing FISs with an user defined number of rules where each step could be carried out in several ways. We test an implementation of the proposed methodology on a well known data set of features of images of breast masses and, having the number of rules varying, we show results both in terms of correct classification rates and in terms of systems confidence in the obtained decisions. Last, using the number of rules able to produce the most interpretable and trustworthy systems, we derive our best system that is described in details.

Concerning our future directions, several questions remain open. First, our methodology needs to be tested with other data sets to be fully considered valuable; we are also planning to test different strategies for extracting rules.

Different techniques to determine the correct thresholds for the FISs (for example choosing the τ that minimizes the mean square error)are also being considered. More promisingly, the optimal threshold could be chosen optimizing an appropriate function that takes into account the unequal classification error costs.

Finally, it is in our opinion worth probing the possibility of using in parallel FISs with different numbers of rules; their predictions could be combined using several strategies based on the confidence showed by each system.

References

1. Chen, M.Y., Linkens, D.A.: Rule-base self-generation and simplification for data-driven fuzzy models. Fuzzy Sets and Systems 142(2), 243–265 (2004)
2. Chen, Y., Yang, B., Abraham, A., Peng, L.: Automatic design of hierarchical takagi-sugeno type fuzzy systems using evolutionary algorithms. IEEE T. Fuzzy Systems 15(3), 385–397 (2007)
3. Fawcett, T.: An introduction to roc analysis. Pattern Recogn. Lett. 27(8), 861–874 (2006)
4. Ghazavi, S.N., Liao, T.W.: Medical data mining by fuzzy modeling with selected features. Artificial Intelligence in Medicine 43(3), 195–206 (2008)
5. Guillaume, S.: Designing fuzzy inference systems from data: An interpretability-oriented review. IEEE Transactions on Fuzzy Systems 9(3), 426–443 (2001)
6. Jang, J.S.R.: Anfis: adaptive-network-based fuzzy inference system. IEEE Transactions on Systems, Man and Cybernetics 23(3), 665–685 (1993), `http://dx.doi.org/10.1109/21.256541`
7. Klir, G., Yuan, B.: Fuzzy Sets and Fuzzy Logic: Theory and Applications. Prentice-Hall, Englewood Cliffs (1995)
8. Mamdani, E.H., Assilian, S.: An experiment in linguistic synthesis with a fuzzy logic controller. International Journal of Man-Machine Studies 7(1), 1–13 (1975)
9. Pena-Reyes, C.A., Sipper, M.: A fuzzy-genetic approach to breast cancer diagnosis. Artificial Intelligence in Medicine 17(2), 131–155 (1999)
10. Rezaee, B., Zarandi, M.F.: Data-driven fuzzy modeling for takagi-sugeno-kang fuzzy system. Information Sciences 180(2), 241–255 (2010)
11. Takagi, T., Sugeno, M.: Fuzzy identification of systems and its applications to modeling and control. IEEE Transactions on Systems, Man, and Cybernetics 15(1), 116–132 (1985), `http://www.hi.cs.meiji.ac.jp/~takagi/paper/TS-MODEL.tar.gz`
12. Tsipouras, M.G., Exarchos, T.P., Fotiadis, D.I.: A methodology for automated fuzzy model generation. Fuzzy Sets Syst. 159(23), 3201–3220 (2008)
13. Tsipouras, M.G., Exarchos, T.P., Fotiadis, D.I., Kotsia, A.P., Vakalis, K.V., Naka, K.K., Michalis, L.K.: Automated diagnosis of coronary artery disease based on data mining and fuzzy modeling. IEEE Transactions on Information Technology in Biomedicine 12(4), 447–458 (2008)
14. Wolberg, W.H., Street, N., Mangasarian, O.L.: UCI machine learning repository (1995), `http://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+(Diagnostic)`
15. Zadeh, L.A.: Fuzzy sets. Information and Control 8(3), 3385–353 (1965)
16. Zadeh, L.A.: Is there a need for fuzzy logic? Inf. Sci. 178(13), 2751–2779 (2008)

A Knowledge Based Decision Support System for Bioinformatics and System Biology

Antonino Fiannaca[2], Salvatore Gaglio[1,2], Massimo La Rosa[2],
Daniele Peri[1,2], Riccardo Rizzo[2], and Alfonso Urso[2]

[1] Department of Computer Science (DINFO), University of Palermo,
Viale delle Scienze, Ed. 6, 90128 Palermo, Italy
[2] ICAR-CNR, National Research Council, Viale delle Scienze,
Ed. 11, 90128 Palermo, Italy
{fiannnaca,larosa,ricrizzo,urso}@pa.icar.cnr.it,
gaglio@unipa.it, peri@dinfo.unipa.it

Abstract. In this paper, we present a new Decision Support System for Bioinformatics and System Biology issues. Our system is based on a Knowledge base, representing the expertise about the application domain, and a Reasoner. The Reasoner, consulting the Knowledge base and according to the user's request, is able to suggest one or more strategies in order to resolve the selected problem. Moreover, the system can build, at different abstraction layers, a workflow for the current problem on the basis of the user's choices, freeing the user from implementation details and assisting him in the correct configuration of the algorithms. Two possible application scenarios will be introduced: the analysis of protein-protein interaction networks and the inference of gene regulatory networks.

Keywords: Decision Support System, Knowledge Base, Meta Reasoning, Workflow Management.

1 Introduction

In recent years, the scientific community has put its focus on bioinformatics and computational approaches to the analysis of biological data. That because of the continuous growing amount of high throughput technologies, which have provided huge quantity of different biological data, like DNA sequence, proteomic sequences and structures, protein-protein interaction data, gene expression data and so on.

In this contest, researchers have begun to develop computational techniques in order to analyse these data, applying well established artificial intelligence approaches and machine learning algorithms and adapting them to the biological evidences. Their purpose is to discover and explain biological phenomena *in silico*, rather than *in vitro*, helping this way the experimentalists in their activities.

Given a biological issue, there are potentially plenty of different tools that could be used, none of them providing the best possible results. Just to make a

R. Rizzo and P.J.G. Lisboa (Eds.): CIBB 2010, LNBI 6685, pp. 215–228, 2011.
© Springer-Verlag Berlin Heidelberg 2011

quick example, for the identification of the tridimensional structure of a protein from its amminoacid sequence, there exist more than 70 softwares [1], called structure predictors, that offer different performances on the basis of the particular analysed protein. It means there is not just one predictor that always gives the best result, but each software has its own strengths and weaknesses.

Our work is focused on the aforementioned situation: we, in fact, are developing a decision support system that can help the experimentalists to choose and run the proper algorithms and services in order to accomplish a given task. In Section 6 we will see how our system can be applied in order to handle two possible scenarios in bioinformatics: analysis of protein-protein interaction networks and inference of gene regulatory networks.

The goal of the system is twofold:

1. To separate the user from the details of the tools or the on-line services used in research work in analysis of data.
2. To build a cognitive path that takes the user from raw data to knowledge and helps him to navigate this path.

The path is based on the knowledge, that is the heuristics and strategies that can be extracted from bioinformatics papers and experiments representing the expertise on the application domain.

The basic idea of our system is, then, to provide to the researcher, or experimentalist, not only the tools able to resolve a problem, but also the knowledge used in order to justify the choice of those specific tools and strategies. This way, we want to highlight not only the workflow, seen as a simple succession of tasks, but also what is the conceptual scheme at the basis of that workflow.

From this point of view, our system can be seen as a crossover between classical decision support systems (DSS) and the most recent workflow management systems (WFMS).

2 Background

Generally speaking, a decision support system (DSS) [2] is a class of interactive computer-based systems that support decision-making activities. Its main features are the chance to deal with semi-structured problems [3]; the extendibility and the adaptability to different domains [4]; combination of models or analytical techniques with data access functions [5].

A typical application of DSS are clinical decision support systems (CDSS) for medical diagnosis: MYCIN [6] is one of the most famous CDSS, developed in 70s for the diagnosis of bacterial diseases; ONCOCIN [7] is a rule-based expert system that aims at giving support about the timing and dosing of chemotherapy; Kon3 [8] is based on a dedicate ontology and rules built upon unstructured databases of medical records and a set of clinical guidelines, used to get recommendations for care process patient.

In recent years Workflow Management Systems (WFMS) [9] have begun to give a valuable support to biologists. WFMS provide an accessible way to build a pipeline of different services using the most common bioinformatics resources, such as online databases and application tools. Then these workflows can be be stored or shared with other users.

Two of the most used WFMS are Taverna [10] and Biowep [11].

Taverna is probably the most known and used WFMS in biological domain. Taverna is able to automatically integrate tools an databases available both locally and on the web in order to build workflows of complex tasks; to run the workflows and to show results in different formats. The system works by means of a Graphical User Interface (GUI) or a script language.

Biowep allows the user to search and run a predefined set of workflows, already tested, validated and annotated. These workflows are annotated on the basis of the application domain, the type of processing, the type of input data and the type of output format.

As already said, our system can be considered as a merger of DSS an WFMS: in fact it offers support, as a DSS, in decision making process through a reasoning component; on the other hand, it allows to build, edit and run a workflow of operations that can be seen at different abstraction levels, according to the complexity of the main request. This kind of workflow is the final output of our system.

3 System Structure

The layered architecture of the proposed system, shown in Figure 1, is inspired by its main goal: separate the researcher from the tools in order to let him focus on the problem. Sometimes researchers do not have a precise idea of the workflow to use and just want to explore many available options. They are not interested in the details of the algorithms or in the configuration of the web services: this is the reason why these objects are buried in the Object layer: the system decides how to use them in order to accomplish the user's goal. The components of the Object layer are not part of the system and can change in time (an algorithm can be substituted by a more efficient one, a web service can be unavailable) so that are represented as cloud.

The Object Layer is accessed by the Controller layer that is the system core. In this layer it is contained the reasoner and the knowledge base of the system: the former decides which operations to perform on the basis of the available knowledge and the user's request. This knowledge is organized by means of an ontology, that encapsulates all the facts representing single pieces of information. The ontology allows to connect the tasks and operations with the problems of a domain (the application domain); and the same tasks with the actual algorithms and services that implements them.

In particular facts are obtained from bioinformatics papers, experimental papers and, in case, from domain experts. At present the core of the facts are extracted from bioinformatics and experimental papers.

The knowledge base also contains a set of rules that describe which are the conditions that should be satisfied in order to run a specific task or algorithm

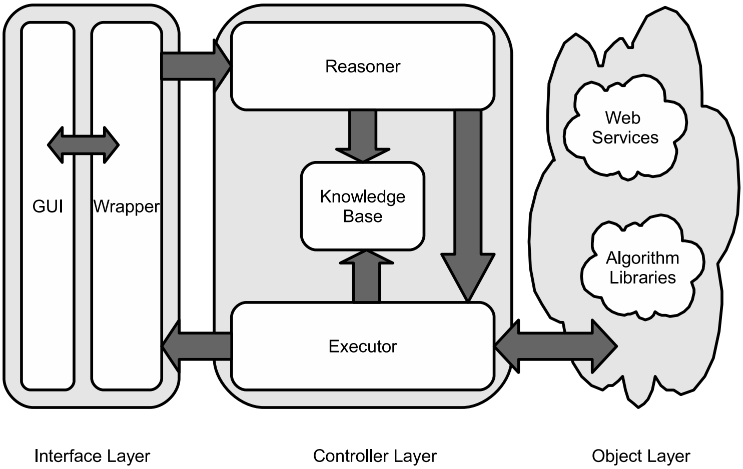


Fig. 1. System Architecture

present in the Object layer: in other words rules code the strategies and heuristics that the system can provide to the user. As facts, the rules can be extracted from different sources, right now the core of the rule set is composed of a large collection of scientific papers.

In the Controller layer there is also the executor module that is the part of the Controller layer that runs the tools in the Object level according to the input data. The executor is controlled by the reasoner and updates the knowledge base with the intermediate results, moreover it will send the final results to the user.

The user looks at the system operation using the GUI and the wrapper that are in the interface layer. The wrapper is the module that manages the communication between the executor in the Controller layer and and the GUI that is the last interface level. The user interacts with the GUI that sends message to the wrapper, the wrapper formats this messages in the right way for the executor module, and sends query to the reasoner. This allows to easily change the GUI without interferences to the other parts of the system.

4 Decision-Making Modules

In order to make the system more efficient and structured, facts and rules of the KB are organized into a set of *decision-making modules.*

A decision-making module, from now on simply *module*, is a collection of specific facts and rules with common features. We can assign to each module a well defined scope and purpose, a specific slice of the decision-making process.

For example, we can have modules suited for taking decisions about preprocessing operations, visualization, clustering and so on that can be used in different application domain.

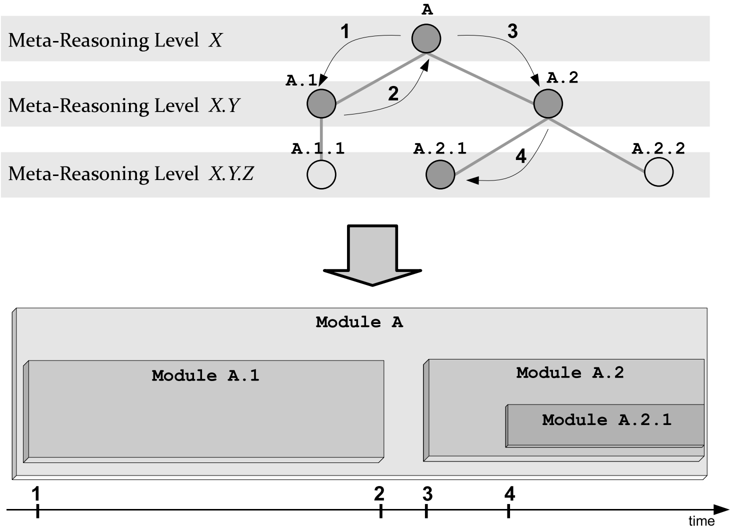


Fig. 2. Meta-Levels Architecture

All the modules are organized in a tree, representing the relationships of specialization or generalization that exist between modules. Modules can have one or more children and the parent module is responsible for the activation of its sub-modules. Each level in the tree represents a different meta-reasoning level. We define it a meta-reasoning because a parent module makes decision on the activation of one of its child module, that in turn has the expertise to take decisions on more specialized activities: so we can say that there is a high level reasoning whose results is another kind of reasoning at a lower level.

The mechanism of modules activation, also called *focusing*, is managed by special rules: when the preconditions of these rules, the **IF** part, are satisfied, their action, the **THEN** part, is to give the focus to a child module. A parent module activates a child module when it needs specialized knowledge, i.e. more specific facts and rules, in order to complete its decision-making activity.

When the module ends its job, the focus is automatically returned to the parent module. The tree representation of modules can be converted in a clearer one using a treemap [12], as in Fig. 2. The treemap allows to immediately visualize the topology of the tree using a set of nested boxes: the parent nodes “englobe” their own children nodes.

In the figure is reported also a four step analysis. In order to solve a specific request, the module A is enabled: at step *1* it call module A.1 to solve a sub-task. At step *2* the module A.1 complete its reasoning and module A takes the control back. At step *3* the module A calls module A.2 to solve another sub-task. At step *4* the module A.2 has not enough knowledge about the sub-task and sends a sub-request to module A.2.1 to resolve a sub-sub-task.

The treemap visualization can be included into a workflow representation, as shown in Fig. 3. The decision making modules, in their treemap organization,

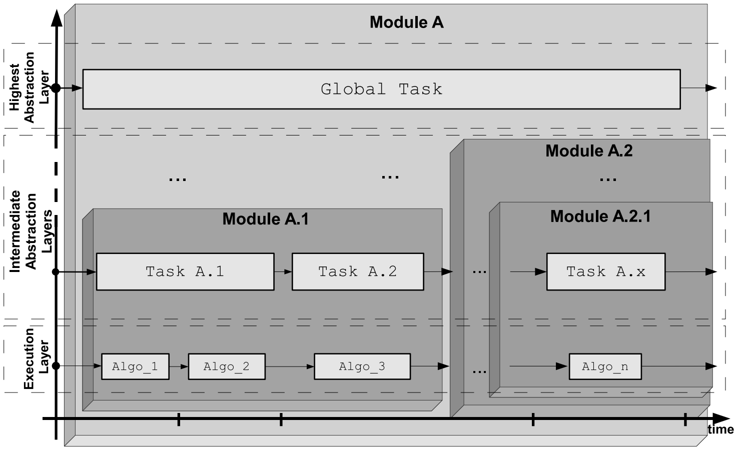


Fig. 3. General Workflow

are in the central part. The horizontal and vertical axis are respectively the abstraction axis and the time axis. The rectangles that intersects the decision-making modules at the various abstraction layers are the executed tools and services, if they are at the bottom layer, or the strategies and heuristics that use them, if they are at higher abstraction layers. The highest abstraction layer is the main goal of the running experiment. Since this scheme includes the temporal dimension, the treemap used for the modules is a bit different from the classic one. The entire tree structure of the modules is not converted into the treemap and projected into the workflow, but only the modules activated during the execution of the experiment are shown.

Module A is the main module, responsible for the supervision of the entire process. Following the time axis, it gives the focus to Module A.1, which decides, through its facts and rules, to launch Task A.1 and Task A.2 done by means of *Algo_1* and *Algo_2* (for Task A.1), and *Algo_3* (for Task A.2). After that, the focus goes back to Module A that pass it to Module A.2 and so on.

This type of multi-layer workflow representation is the actual output of our system.

5 Implementation Details

The different parts of the proposed system, see Section 3, have been implemented using the proper instruments.

The knowledge base and the underlying ontology have been implemented with Protegé [13], that is one of the largest adopted tool for building an ontology and populate it with pieces of information that represent the knowledge of the system.

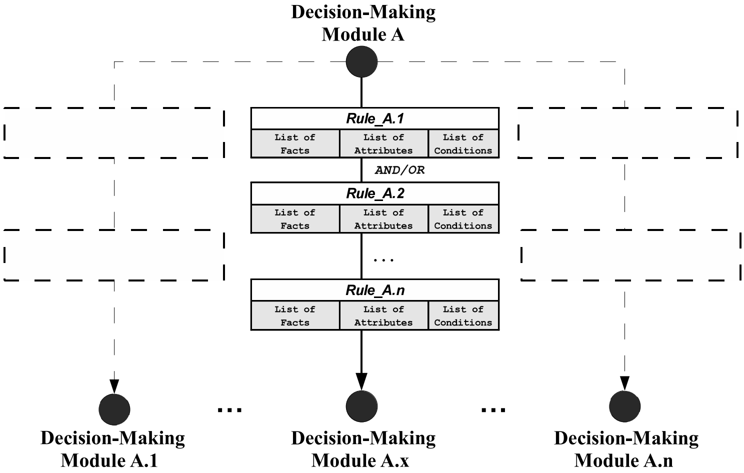


Fig. 4. Focus shifting mechanism for Decision-Making modules

As for the Reasoning part of our system, we made it by means of Jess [14], the Rule Engine for the Java Platform. Using Jess, we were able to implement the different meta-level reasoners through a set of decision-making modules.

The information belonging to the knowledge base is translated in a set of facts, while the reasoning on these facts is done by means of a set of rules. The rules can be combined using logical operators (AND, OR,...). Each rule is in the form **IF** *precondition on facts is true* **THEN** *execute action* and is activated when some constraints on the facts are verified.

Each module is activated, or in other words is given the "focus", by a parent module, and the current module can give the focus to other sub-modules. There can be only an active module at a time, and only the module with the focus can execute the actions of its activated rules.

This mechanism is managed by a stack, with the active module on the top and the other modules below, according to the order of the shift of focus. This way, when a module ends its job, the focus is automatically returned to its direct parent.

Following the same scheme adopted for the meta-level reasoners described in Section 4, each module has its own set of facts and rules, and it takes care of different kind of decisions, according to its complexity level. For example, high level modules, will have rules to decide what are the main phases to solve a request, and then will give the focus to lower level modules that will be responsible, thanks to their facts and rules, to select and suggest a specific strategy, and so on until the lowest level modules whose job is to choose the proper algorithms and services that will be actually run. In Figure 4 is shown a sample view of the decision-making modules and the mechanism of focus shifting described in this Section.

6 Application Scenarios

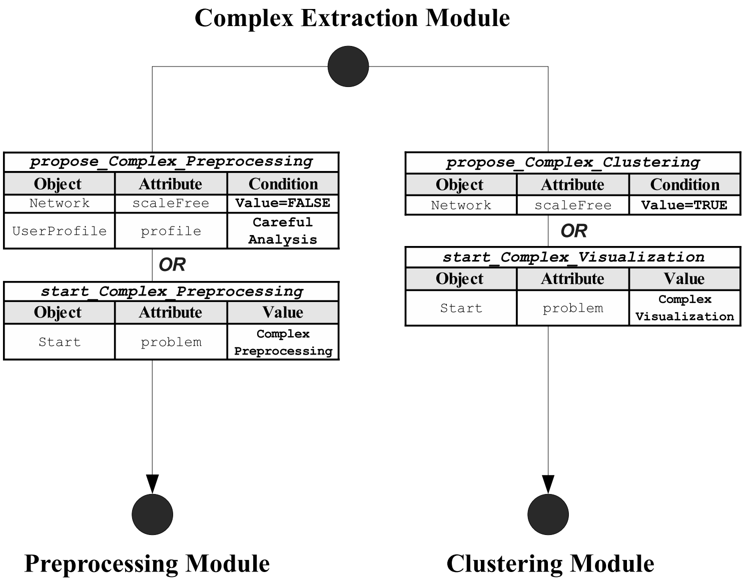
In this Section we present two possible application scenarios. Thanks to the presence of a knowledge base, the system can be easily extended to handle other bioinformatics problems, such as protein structure prediction or protein function prediction, adding the proper expertise in the form of facts and rules.

6.1 PPI Networks

Proteins represent the working molecules of a cell and, as it is well known, they can provide many biological activities by interacting with other proteins.

Analysis of protein-protein interaction (PPI) *in silico* is a hot topic in current bioinformatics researches. A large amounts of PPI data have been identified by using high throughput proteomic technologies, but only a few of them have been verified with small scale experiments (*in vitro*) as real interaction with an emerging function. At biological pathway level, emerging function is not linked to a simple pair of proteins, but arises with protein complex, that is a collection of PPIs.

Protein complexes are implicated as essential components in major biological mechanisms of a cellular process such as DNA transcription, translation, cell cycle control, signal transduction, and so on. Nowadays experimentalists can take advantage of using different online available databases containing a list of PPIs for each species (DIP [15], MIPS [16], etc..), but each DB uses a proper set of



propose_Complex_Preprocessing

Object	Attribute	Condition
Network	scaleFree	Value=FALSE
UserProfile	profile	Careful Analysis

start_Complex_Preprocessing

Object	Attribute	Value
Start	problem	Complex Preprocessing

propose_Complex_Clustering

Object	Attribute	Condition
Network	scaleFree	Value=TRUE

start_Complex_Visualization

Object	Attribute	Value
Start	problem	Complex Visualization

Fig. 5. PPI Modules

features. In addiction, there are a lot of strategies to identify protein complexes (soft-clustering, greedy heuristics, probabilistic approaches, etc..), but each of them has proper pros and cons. For this reason, our system can help the user both to choose the best model representing a specific PPI Network and to run a right technique among available strategies and, if it is necessary, it can combine more than one model.

More in details, we analyse a scenario where the user requires an extraction of protein complexes from a file containing a list of PPIs.

As previously said, the proposed system turns on a module of the meta-reasoning, responsible of this problem: in this scenario, it focuses on the module Complex Extraction. This module have to analyse the input file to build a protein network representation and, then, can suggest next operations. Figure 5 shows that this module is able to trigger two modules at lower meta-levels: "*pre-processing module*" and "*clustering module*". Each of them can be suggested if almost one of the given conditions, highlighted on the connection line, is satisfied. For instance, since both the input file can be modelled as a network without a scale-free topology and the user has selected the profile "*Careful Analysis*", then the system can activate the rule "*propose_complex_preprocessing*"; if the user accepts suggestion, then the preprocessing module will focus and all the rules it contains will be runnable.

Figure 6 shows a workflow generated by the proposed system. In this figure are depicted four abstraction layers: the highest abstraction layer contains the contest of the required problem; the intermediate abstraction layer shows two macro-strategies used to solve the required problem; the lower abstraction layer emphasizes strategies adopted to solve each macro-strategies; the execution layer reports instances of tools used for implementing strategies at lower abstraction layer. Obviously, it exists more than a strategy that solves a macro-strategy, and

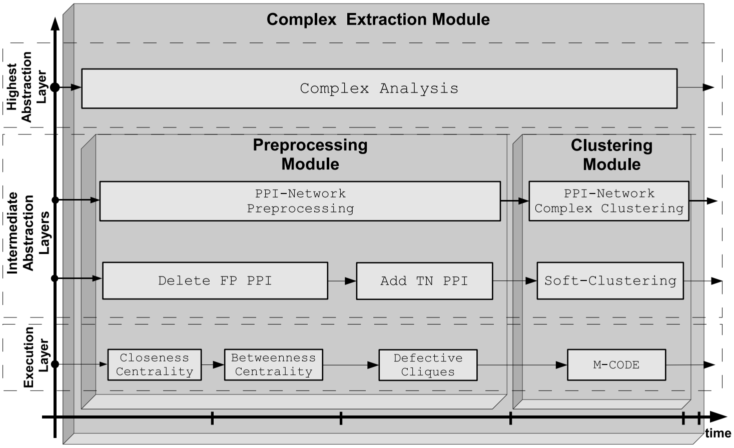


Fig. 6. PPI sample workflow

user could use several strategy in cascade. For instance the macro-strategy "*PPI Network Preprocessing*" was resolved at lower layer deleting some false positive Interactions ("*Delete FP PPI*") and adding calculated true negative interactions ("*Add TN PPI*").

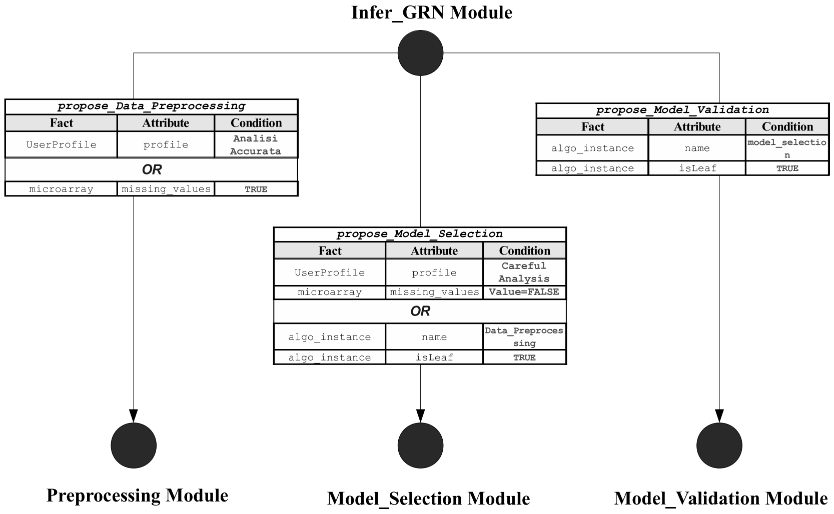
The former substrategy is implemented by *Closeness Centrality* [17] and *Betweenness Centrality* [18] algorithms; the latter substrategy is executed through *Defective Cliques* [19] algorithm.

6.2 GRN Inference

The biological mechanism that, inside the cells, controls the gene expression, i.e the activation or suppression of the transcription of genes into mRNA, is gene regulation [20]. mRNA will then be translated into a protein. Gene regulation is a very complex biological phenomenon and it is not well understood yet. In system biology, gene regulation is studied and modelled by means of Gene Regulatory Networks (GRN): a GRN is a graph whose nodes are biological elements and edges represent regulatory relationships among them. The elements involved in gene regulation are genes, Transcription Factors (TFs), Protein Complex, etc...

From a computational point of view, modelling a GRN is a reverse engineering problem, since from the output of gene regulation, that is gene expression measured through microarray technology, we want to infer the network, with its topology and parameters, that provided those outputs.

Inferring a GRN is an ideal application scenario for our system: looking at the state-of-the-art, in fact, a wide set of algorithms and methods are used for this purpose [21,22,23]. All of these techniques present pros and cons, and

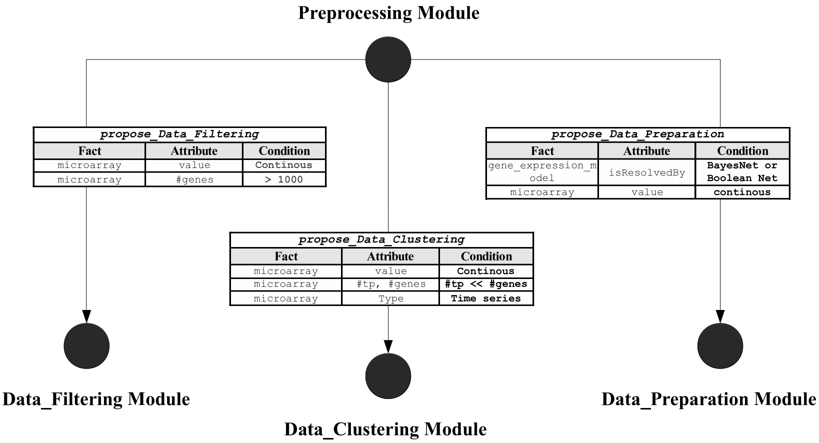


propose_Data_Preprocessing		
Fact	**Attribute**	**Condition**
UserProfile	profile	Analisi Accurata
	OR	
microarray	missing_values	TRUE

propose_Model_Selection		
Fact	**Attribute**	**Condition**
UserProfile	profile	Careful Analysis
microarray	missing_values	Value=FALSE
	OR	
algo_instance	name	Data_Preprocessing
algo_instance	isLeaf	TRUE

propose_Model_Validation		
Fact	**Attribute**	**Condition**
algo_instance	name	model_selection
algo_instance	isLeaf	TRUE

Fig. 7. GRN Modules



propose_Data_Filtering		
Fact	**Attribute**	**Condition**
microarray	value	Continous
microarray	#genes	> 1000

propose_Data_Clustering		
Fact	**Attribute**	**Condition**
microarray	value	**Continous**
microarray	#tp, #genes	**#tp << #genes**
microarray	Type	**Time series**

propose_Data_Preparation		
Fact	**Attribute**	**Condition**
gene_expression_model	isResolvedBy	**BayesNet or Boolean Net**
microarray	value	**continous**

Fig. 8. GRN Submodules for Preprocessing Module

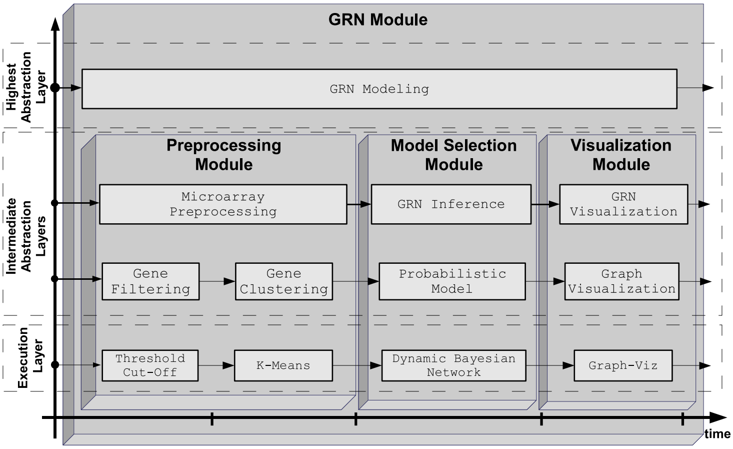


Fig. 9. GRN sample workflow

differ each other according to the type of input data (microarray, gene sequences, protein-protein interactions), the applied algorithm, the desired output, the need of specific data format, the accuracy level of the inferred model, the computational time and resources. Moreover the process of modelling a GRN often needs preprocessing steps, like filtering and clustering, and/or postprocessing steps, like simulation and visualization.

Among the most used methodologies there are static and Dynamic Bayesian Networks [24,25], Factor Graph [26], Boolean Networks [27], correlation methods [28], Ordinary Differential Equations (ODE) [29,30].

Following the same mechanism described in the previous SubSection, in Figs. 7 and 8 are shown the decision-making modules for this scenario and some of the rules that can shift the focus from one module to another one. It is interesting to note that "*preprocessing module*" is responsible for the activation of three other decision-making modules, that are able to take decision for the filtering, clustering and preparation of input data. "Microarray" facts represents, in the knowledge base, the properties of input dataset, that for the scenario of inferring a GRN is a file of gene-expression values.

Figure 9 shows a possible workflow for the current scenario. The preprocessing of microarray input dataset is done by a filtering, using threshold cut-off, and then by a clustering algorithm, with K-Means [31]; then the GRN is modelled using a Probabilistic Model, a Dynamic Bayesian Network, and finally the inferred network is visualized by means of Graph-Viz [32].

7 Conclusions

In this paper we presented a Decision Support System for Bioinformatics and System Biology experiments. The core of our system is a Knowledge Base, containing the expertise of the system about the application domain, and a Reasoner. The Reasoner allows the introduction of a meta-reasoning level for decision-making process. With this decision-making activity, implemented using Jess and a set of decision-making modules, the system is able to suggest and support the user with an advice about the strategies and tools to use in order to resolve the selected problem. Moreover, the system itself manages the proper running of all the selected algorithms, building a workflow for the experiment, assisting the user in the configuration of algorithm's input parameters, when necessary.

Focusing on these two main features, i.e. the decision-making process and the workflow building, our system is an ideal joint between classical decision support system and more recent workflow management system.

References

1. Jauch, R., Yeo, H.C., Kolatkar, P.R., Clarke, N.D.: Assessment of CASP7 structure predictions for template free targets. Proteins: Structure, Function, and Bioinformatics 69, 57–67 (2007)
2. Power, D.J.: Brief History of Decision Support Systems. DSSResources.COM, `http://DSSResources.COM/history/dsshistory.html`
3. Keen, P.G., Scott Morton, M.S.: Decision support systems: an organizational perspective. Addison-Wesley Pub. Co., Reading (1978)
4. Moore, J.H., Chang, M.G.: Design of Decision Support System. Database 12, 8–14 (1980)
5. Parker, B.J.: Decision support systems: the reality that seems hard to accept. Omega 14, 135–143 (1986)
6. Buchanan, B.G., Shortliffe, E.H. (eds.): Rule-Based Expert Systems: The MYCIN Experiments of the Stanford Heuristic Programming Project. AAAI, Menlo Park (1984)

7. Shortliffe, E.H., Scott, A.C., Bischoff, M.B., et al.: ONCOCIN: an expert system for oncology protocol management. In: International Joint Conference on Artificial Intelligence, pp. 876–881 (1981)
8. Ceccarelli, M., Donatiello, A., Vitale, D.: KON3: a Clinical Decision Support System, in oncology environment, based on knowledge management. In: IEEE International Conference on Tools with Artificial Intelligence, vol. 2, pp. 206–210 (2008)
9. Hollinsworth, D.: The Workflow Reference Model. Tech. Rep. TC00-Workflow Management Coalition (1994)
10. Hull, D., Wolstencroft, K., Stevens, R., Goble, C., Pocock, M.R., Li, P., Oinn, T.: Taverna: a tool for building and running workflows of services. Nucleic Acids Res. 34 (2006)
11. Romano, P., Bartocci, E., Bertolini, G., De Paoli, F., Marra, D., Mauri, G., Merelli, E., Milanesi, L.: Biowep: a workflow enactment portal for bioinformatics applications. BMC Bioinformatics 8 (2007)
12. Johnson, B., Shneiderman, B.: Tree-maps: a space-filling approach to the visualization of hierarchical information structures. In: Proceedings of IEEE Conference on Visualization, pp. 284–291 (1991)
13. The Protege Ontology Editor and Knowledge Acquisition System, `http://protege.stanford.edu`
14. Sandia National Laboratories: Jess: The rule engine for the JavaTM platform (2003), `http://herzberg.ca.sandia.gov/jess/`
15. Xenarios, I., Rice, D.W., Salwinski, L., Baron, M.K., Marcotte, E.M., Eisenberg, D.: DIP: the database of interacting proteins. Nucleic Acids Research 28, 289–291 (2000)
16. Pagel, P., Kovac, S., Oesterheld, M., Brauner, B., Dunger-Kaltenbach, I., Frishman, G., Montrone, C., Mark, P., Stümpflen, V., Mewes, H.W., Ruepp, A., Frishman, D.: The MIPS mammalian protein-protein interaction database. Bioinformatics 21, 832–834 (2005)
17. Sabidussi, G.: The centrality index of a graph. Psychometrika 31, 581–603 (1966)
18. Freeman, L.C.: A set of measures of centrality based on betweenness. Sociometry 40, 35–41 (1977)
19. Yu, H., Paccanaro, A., Trifonov, V., Gerstein, M.: Predicting interactions in protein networks by completing defective cliques. Bioinformatics 22, 823–829 (2006)
20. Chen, K., Rajewsky, N.: The evolution of gene regulation by transcription factors and microRNAs. Nat. Rev. Genet. 8, 93–103 (2007)
21. Huang, Y., Tienda-Luna, I., Wang, Y.: Reverse engineering gene regulatory networks. IEEE Signal Processing Magazine 26, 76–97 (2009)
22. Cho, K.H., Choo, S.M., Jung, S.H., Kim, J.R., Choi, H.S., Kim, J.: Reverse engineering of gene regulatory networks. Systems Biology, 149–163 (2007)
23. Hecker, M., Lambeck, S., Toepfer, S., van Someren, E., Guthke, R.: Gene regulatory network inference: Data integration in dynamic models–A review. Biosystems 96, 86–103 (2009)
24. Zou, M., Conzen, S.D.: A new dynamic Bayesian network (DBN) approach for identifying gene regulatory networks from time course microarray data. Bioinformatics 21, 71–79 (2005)
25. Kim, S.Y., Imoto, S., Miyano, S.: Inferring gene networks from time series microarray data using dynamic Bayesian networks. Briefings in Bioinformatics 4, 228–235 (2003)

26. Gat-Viks, I., Tanay, A., Raijman, D., Shamir, R.: A probabilistic methodology for integrating knowledge and experiments on biological networks. J. Comput. Biol. 13, 165–181 (2006)
27. Lahdesmaki, H., Hautaniemi, S., Shmulevich, I., Yli-Harja, O.: Relationships between probabilistic Boolean networks and dynamic Bayesian networks as models of gene regulatory networks. Signal Process. 86, 814–834 (2006)
28. Margolin, A., Nemenman, I., Basso, K., Wiggins, C., Stolovitzky, G., Favera, R., Califano, R.A.: ARACNE: an algorithm for the reconstruction of gene regulatory networks in a mammalian cellular context. BMC Bioinform. 7 (2006)
29. Gardner, T.S., di Bernardo, D., Lorenz, D., Collins, J.J.: Inferring genetic networks and identifying compound mode of action via expression profiling. Science 301, 102–105 (2003)
30. di Bernardo, D., Thompson, M., Gardner, T., Chobot, S., Eastwood, E., Wojtovich, A., Elliott, S., Schaus, S., Collins, J.: Chemogenomic profiling on a genome-wide scale using reverse-engineered gene networks. Nat. Biotechnol. 23, 377–383 (2005)
31. MacQueen, J.B.: Some Methods for classification and Analysis of Multivariate Observations. In: Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability, pp. 281–297 (1967)
32. Gansner, E.R., North, S.C.: An open graph visualization system and its applications to software engineering. Softw. Pract. Exper. (1999)

Dynamic Simulations of Pathways Downstream of ERBB-Family: Exploration of Parameter Space and Effects of Its Variation on Network Behavior

Lorenzo Tortolina[1,4,5,⋆], Nicoletta Castagnino[1,4,5,⋆],
Cristina De Ambrosi[1,2,4,5,⋆], Raffaele Pesenti[3], Franco Patrone[1],
Alberto Ballestrero[1], Eva Moran[1], Alessio Nencioni[1], and Silvio Parodi[1,4,5,⋆⋆]

[1] Department of Internal Medicine, University of Genoa, Italy
[2] Department of Informatics and Information Sciences; University of Genoa, Italy
[3] Department of Applied Mathematics; University of Ca Foscari of Venice, Italy
[4] National Cancer Institute of Genoa (IST), Italy
[5] Research Center for Computational Learning (CRAC), Italy

Abstract. The signaling-network immediately downstream of the ErbB-family is very important in BC and other cancers, especially considering treatment of the excess of function of dominant onco-proteins with onco-protein inhibitors. We studied and implemented dynamic simulations of four downstream pathways. The fragment of the signaling-network we evaluated was described as a Molecular Interaction Map. Our simulations involved 242 modified species and complexes, 279 reversible reactions, 110 catalytic activities. We used Ordinary Differential Equations for our simulations. We started an analysis of sensitivity / robustness of our network, and we systematically introduced fluctuations of total concentrations of independent molecular species. We adopted mostly the strategy of a random sampling of 1000 cases for each instance of increasing numbers of perturbations. Only a small minority of cases showed an important sensitivity, the number of sensitive cases increased moderately for increasing numbers of perturbations. In most cases the effect of introducing virtual mutations and virtual onco-protein inhibitors was more important than the effect of randomly introduced perturbations, this suggests an acceptable robustness of our network. The importance of our work is primarily related to the fact that the complexity of the 39 basic species signaling-network region we analyzed is of difficult intuitive understanding for a "naked" human mind. Dynamic network simulations appear to be an useful support for an *"a posteriori"* mental comprehension by a cancer researcher of the behavior of a network of this degree of complexity. The present report suggests the feasibility of a computational approach even in the presence of a multiple number of uncertainties about parameter values.

[⋆] These authors contributed equally to this work.

[⋆⋆] Scientific coordinator of the team work and corresponding author silvio.parodi@unige.it; Via Pastore 3, 16132, Genova, Italy; Tel.: +39-010-353-38271; Fax: +39-010-353-38272.

R. Rizzo and P.J.G. Lisboa (Eds.): CIBB 2010, LNBI 6685, pp. 229–241, 2011.
© Springer-Verlag Berlin Heidelberg 2011

Keywords: ErbB-family, signaling-networks, dynamic simulations, oncogene mutations, onco-protein inhibitors, systems biology, parameter space.

On the biochemical level [1,2], we may consider a normally differentiated cell as a network of pathways, and we can interpret recent progress in molecular oncology [3,4,5,6] as a description of a cancer cell bearing in the order of two dozen mutated pathways. Potential mutations belonging to the same pathway are hypothesized as being mutually exclusive. Each pathway might contain 10-20 signaling-molecules. In principle, one of them could be mutated / altered through gain or loss of function. The conclusion of these considerations is in agreement with the Vogelstein group's observation that about 20-40 different alterations that are present in an individual tumor are fished out of a pool containing about 200-400 potential oncogenes. In our work, signaling-network molecular pathologies move to the front stage. We have primarily considered Breast Cancer (BC), and a fraction of the G0 -G1 cell cycle transition. The signaling-network to which this parameter-space analysis makes reference was described in detail in [7]. We adopted the approach of reconstructing the molecular anatomy of our network through a Molecular Interaction Map (MIM). We simulated the attainment of a stationary state in our biochemical network through hundreds of ordinary differential equations (ODEs). This approach belongs to a subfield of Systems Biology [8]. The opportunity, even the necessity, for this dynamic simulation approach is the consequence of the fact that the behavior of a 30-40 molecules signaling-network is not easily intuitive a priori for the naked-mind. With some effort the mind of a cancer researcher is however capable of understanding *"a posteriori"* the suggestions coming from the computational approach.

In BC, along the pathway [ErbB-family receptors - PI3K - PTEN - Akt - GSK3β - APC - β-catenin - TCF/LEF] (PI3K - Akt - β-catenin pathway for short), ErbB2 is amplified in 20-30% of cases [9], PI3KCA is mutated in $\approx$ 26% of cases [10], PTEN is hypo-expressed or inactivated in $\approx$ 40% of cases [11], APC is mutated in $\approx$ 4% of cases, and β-catenin is mutated in $\approx$ 2% of cases [10]. As also suggested by the Vogelstein group [5], mutations along a given pathway tend to be mutually exclusive. During cancer progression, not much will happen by adding two adjacent or close mutations within the same pathway to the same cell. At the same time, and as a consequence of the above considerations, the addition of all the mutually exclusive alterations along a given pathway represents the overall pathway alteration frequency for BC. It would appear that in BC, the PI3K - Akt - β-catenin pathway is altered (excess of function) in $\approx$ 100% of cases (by summing each of the individual, mutually exclusive mutation frequencies). It must be pointed out that along a given pathway even the loss of function of the gene product of a recessive oncogene (for instance PTEN) can contribute to the excess of function of the overall pathway. Nuclear β-catenin is a co-transcription factor for the transcription factor TCF/LEF. Cyclin D1 and c-myc (both transcribed by TCF/LEF) are among the genes that are important for the G1 - S transition. At one point or another of the pathway, the PI3K - Akt - β-catenin pathway, is almost always hyper-activated, even in colon cancer and in Non Small Cell Lung Cancer (NSCLC) [10] (and very likely also in many other cancers). Downstream of the ErbB-family of receptors, there is another pathway which is largely, but not completely, independent of the previous one: [ErbB-family receptors - Grb2 - Shc - SOS (an exchange factor for KRAS) - GAP (a GTP-ase for KRAS) - KRAS - BRAF - MEK - ERK - transcription factors activated by the MAP-kinases] (KRAS - ERK for short). It is tempting to believe that even this pathway is almost always hyper-activated in BC and in other cancers. Looking at different tissue types in the COSMIC database [10], KRAS is very frequently mutated in large intestine cancer ($\approx$32%) and also in NSCLC ($\approx$16%). Mutation frequency in BC is about 5% (+ 1% of HRAS mutations and 1% of NRAS mutations). However, in triple negative BC tumors (Estrogen Receptor-; Progesterone Receptor-; ErbB2-), the combined mutation frequency of the three RAS is about 40%. RAS mutations probably play a more important role in this type of aggressive and poorly responsive BC tumor. A smaller pathway (one that we explored less extensively) downstream of the ErbB-family receptors is represented by an activated, mutated, amplified EGFR receptor which can phosphorylate β-catenin in Y-654 and make it independent from E-Cadherin, thus making β-catenin able to migrate to the nucleus and co-operate with the transcription factor TCF/LEF [12]. E-Cadherin is mutated in $\approx$ 22% of BC and is then incapable of binding β-catenin [10], even in the absence of any EGFR stimulation.

The study of molecular-network alterations in cancer, in the presence of onco-protein mutations and onco-protein inhibitors, is a quite modern strategy of crucial importance, and in order to perform this type of research a computational approach is essential. Even for intensively explored network regions, parameter knowledge is often incomplete. To study the degree of tolerance of a network to parameter uncertainty, becomes a prerequisite task. This was the intent of the present study, obviously it is a work in progress.

1 Methods

The MIM that was the object of our simulations is shown in Fig. 1. The syntactic rules for drawing our MIM are described in [13], and are briefly reported in Fig. 1, insert. Table 1SM (Supplementary Material of [7]) shows an Annotation List of the reference sources of the interactions reconstructed in our MIM. Corresponding numbers appear in Fig. 1 and in the Annotation List. Table 2SM of [7] includes a short Glossary. The Glossary not only includes the 39 molecular species we utilized in our simulations, but it also includes 8 additional molecules that are present in the MIM (white cartouches), which, however, are not part of the simulation (Axin; TCF/LEF; pre-enhanceosome; elongation complex RNAPol II; c-myc mRNA; c-myc transcription factors binding sites, promoter proximal region (P2); c-myc DNA coding region). Our MIM describes a network downstream of the ErbB-family receptors that is relevant for BC. Similar networks are also operative in colon cancer, in NSCLC and perhaps in most tumors. Two major and two minor pathways have been described in our MIM (see Introduction). Stationary, temporary equilibrium is assured by a growth factor (EGF), 10 kinases (ErbB1, ErbB2, ErbB3 counted separately), 14 phosphatases (including GAP), 10 signaling / adaptor proteins, and 4 small signaling-molecules, for a total of 39 basic molecular species. Following the suggestion of [14,15] we introduced in our simulation a phenomenon of 'piggyback' binding of SOS and GAP to an activated ErbB receptor in the sub-membrane region where KRAS is also anchored. This was equivalent to local association rate increases of about 250 times (Table 3SM of [7]). Table 3SM shows a list of 279 reversible reactions and 110 catalytic reactions, rate-constants included, which represent the complete set of our dynamic simulations. Table 3SM also reports the concentrations of the 39 basic species. It is accompanied by the references source of the data. Some numerical values have been interpolated by taking into account the constraints imposed by: existing values, molecular anatomy of the network, indirect evidence at the molecular, cellular and clinical level. Narrow ranges of the interpolated values were practically imposed by the rest of the network system. The GDP and GTP species, as well as the cytoskeleton-protein, were considered in large excess (non-consumable).

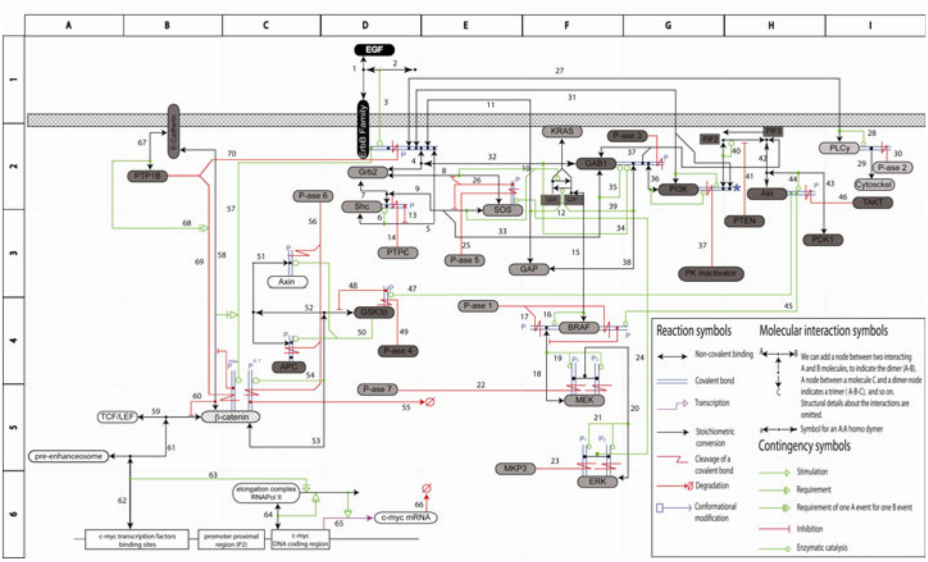

A
B
C
D
E
F
G
H
I
1
2
3
4
5
6
EGF
ErbB Family
E-Cadherin
PTP1B
Grb2
Shc
SOS
PTPC
P-ase 6
P-ase 5
KRAS
P-ase 3
GAB1
PIP2
PIP3
PI3K
Akt
PTEN
PDK1
PK inactivator
PLCγ
P-ase 2
Cytoskel
TAKT
GAP
Axin
GSK3β
P-ase 4
APC
P-ase 1
BRAF
MEK
ERK
MKP3
P-ase 7
β-catenin
TCF/LEF
pre-enhanceosome
elongation complex RNAPol II
c-myc mRNA
c-myc transcription factors binding sites
promoter proximal region (P2)
c-myc DNA coding region
Reaction symbols
Non-covalent binding
Covalent bond
Transcription
Stoichiometric conversion
Cleavage of a covalent bond
Degradation
Conformational modification
Molecular interaction symbols
We can add a node between two interacting A and B molecules, to indicate the dimer (A-B). A node between a molecule C and a dimer-node indicates a trimer (A-B-C), and so on. Structural details about the interactions are omitted.
Symbol for an A:A homo dymer
Contingency symbols
Stimulation
Requirement
Requirement of one A event for one B event
Inhibition
Enzymatic catalysis

Fig. 1. A heuristic MIM downstream of [ligand - ErbB-family receptors], which is relevant in BC. Pathways that were taken into consideration include: [PI3K - PTEN - Akt - GSK3β - APC - β-catenin]; [Grb2 - Shc - SOS (an exchange factor for KRAS) - GAP (a GTP-ase for KRAS) - KRAS - BRAF - MEK - ERK]. An activated ErbB-family receptor induces a P-Y654-β-catenin, making it independent of E-Cadherin, [Cadherin/Catenin adhesive complex]. The small [ErbB-family - PLCγ - P-ase2 - cytoskeleton-protein] pathway is also shown.

1.1 Simulations Using ODEs

In our simulations, we started from a situation out of equilibrium. The total concentration, relative to a given basic species and all its complexes and post-translational modifications, is initially entirely attributed to the corresponding free molecule. We bring the reactions to a stationary equilibrium, which causes redistribution of each total concentration among all its components. We verified that if we start from different out of equilibrium conditions we can have different provisional transitory peaks or curves, but then we converge, typically within a virtual time of 7-10 hours (residual differences in the order of 5%), toward the same stationary equilibrium, provided that total concentrations and rates remain unchanged. In this paper we do not provide a detailed analysis, but when the system is already in a stationary state and we vary only the EGF concentration, for instance from a physiological EGF concentration (.1 nM) to a pharmacological EGF concentration (10 nM), a new stationary state is reached within a virtual time of $\approx$ 20 minutes. The parameters that are the focus of our present analysis are the stationary states reached in the presence of variations of the concentrations of the 34 basic species. To simulate the signaling-network we considered in this paper, we mathematically formalized the reaction scheme of Table 3SM of [7], in terms of the reactions' kinetic laws [16]. The kinetic laws of a reaction describe the velocity at which the reactants are transformed into the products of the reaction. More specifically, we assumed that all reactions followed a mass action kinetic law (a consequence of the IInd law of thermodynamics). According to this kinetic law, the velocity of the reaction is directly proportional to the concentration of the reactants multiplied by the reaction rate. As an example, given the reversible reaction

$$[\text{A}] + [\text{B}] \leftrightarrow [\text{A-B}]$$

the velocity of the [A-B] formation reaction is:

$$k_1[\text{A}][\text{B}] \text{ - } k_{-1}[\text{A-B}];$$

where each [X] indicates the concentration of a given reactant, k_1 and k_{-1} are forward (association) and backward (dissociation) rates, respectively, of the reversible reaction. At equilibrium

- ○ $k_1 \cdot [\text{A}][\text{B}] = k_{-1} \cdot [\text{A-B}]$
- ○ $([\text{A}][\text{B}])/[\text{A-B}] = k_{-1}/k_1 = K_d$ (equilibrium constant K)

We can also have an irreversible catalytic reaction of the type:

[XP-Phosphatase] → [X] + [Phosphatase] + P (P goes into the phosphates pool)

$$\text{v} = k_{cat}[\text{XP-Phosphatase}]$$

where k_{cat} is a catalytic rate (a turnover number).

In turn, knowledge of the kinetic laws of the reactions has allowed us to describe the rate of change of each complex concentration by means of an ordinary differential equation in which the velocities of the reactions that produce or consume the reactant are algebraically summed. The collection of this type of differential equations for all 242 complexes + 39 basic species included in the signaling-network fully describes the dynamic behavior of our biologic system. Unfortunately, the non linear nature of the above differential equations has prevented us from determining the analytical expressions for the system evolution over time. Nevertheless, we have been able to numerically simulate the system evolution with the help of dedicated software, such as the SimBiology toolbox of Matlab (http://www.mathworks.com/products/simbiology/?BB=1). This kind of numerical approach has been pursued by different authors, among them [2,15,17,18,19,20], and all the other authors whose models are available in the BioModels Database [21].

1.2 Sensitivity / Robustness of Our Network to Perturbations

We implemented perturbations of the concentrations introduced at the very beginning of our simulations (see Table 3SM of [7]), concerning 34 consumable basic molecular species. EGF, GDP, GTP, PIP3 and cytoskeleton-protein were not considered. All PLCγ-P can convert to a [cytoskeleton-protein:PLCγ-P] complex (cytoskeleton-protein was implicitly considered in large excess). We introduced combinations of 10x and 10/ perturbations in one, or two, or three or four of the n = 34 consumable total molecular concentrations. Perturbed species were always not coincident with the perturbing species. Only perturbations on basic species are reported herein, while perturbation effects on modified species and complexes are not reported. For the moment we have not reported perturbation of rates either. We examined concentration levels of the 34 basic molecular species individually, always in the presence of the physiologic .1 nM EGF concentration. We sorted all of them for one or two combinations of perturbations or a random subset of the combinations of three or four perturbing species, and we examined the effects induced on the (n - 1), or (n - 2), or (n - 3), or (n - 4) remaining species. Because one perturbing species can have effects on the remaining (n - 1) perturbed species, for the 34 consumable basic species considered in our case we have 34x33 = 1,122 combinations. In our computational approach each of the 34 simulations (in terms of perturbing species) gives information about all the remaining 33 perturbed species. The computing time on our pc (Dell Optiplex 960, Intel Core 2 Duo processors @3.00 GHz, 4.00GB of Ram) was 22x34 seconds, approximately ≡ 12.5 minutes. The computational time becomes 25 minutes considering both 34x and 34/ perturbations. Considering again both 34x and 34/ perturbations, in the case of two perturbing species the number of perturbing combinations is (34x2)x(33x2)/2 = 2,244. The computing time on our pc was 22x2,244 seconds, approximately ≡ 13.7 hours, a still acceptable computing time. The total number of combinations of perturbed species was 2,244x32 = 71,808. In the case of three perturbing species the number of perturbing combinations is 68x66x64/6 = 47,872. The computing time on our pc would have been 22x47,872 seconds, approximately ≡ 293 hours ≡ 12.2 days. We considered this computing time too long for our pc and we preferred to sort out randomly 1,000 of 47,872 combinations. This implies a computer time of 22x1,000 seconds, approximately equivalent to 6.1 - 6.2 hours, a still acceptable length of computing time. Notice that sorting randomly 1,000 (perturbing x perturbed) combinations we generate a subset of 1,000x31 = 31,000 perturbed species. From the previous subset, we decided to sort out randomly only 1,000 (perturbing x perturbed) combinations. In the case of four perturbing species the number of perturbing combinations is 68x66x64x62/24 = 742,016. The computing time on our pc would have been 22x742,016 seconds, approximately ≡ 4,534.5 hours ≡ 189 days. We considered a fortiori this computing time too long for our pc and we preferred to sort randomly 1,000 (perturbing x perturbed) combinations out of a total of 742,016 combinations. Notice that in this case sorting randomly 1,000 perturbing combinations we generate a subset of 1,000x30 = 30,000 perturbed species. We decided again to sort out randomly only 1,000 of them. The strategy of randomly sorting 1,000 samples from an entire population of possible perturbations can be applied to even larger sets of combinations of perturbing species. We could miss singularities / outliers, but we will still capture the general trend of the perturbation. A perturbed species will present a molecular concentration deviated from the concentration presented at the end of a simulation performed in physiological conditions (a virtual time of 20-30 minutes to come close to a stationary state).

Deviations from the physiological conditions were measured according to the following formula:

$$\left| \frac{(\text{value of perturbation in the species concentration})_{[\text{EGF .1 nM}]}}{(\text{physiological value})_{[\text{EGF .1 nM}]}} \right| \quad (1)$$

We clustered these results into 8 classes: (e^0 < Ratio < e^1), (e^1 < Ratio < e^2), (e^2 < Ratio < e^3), (e^3 < Ratio < e^4), (e^4 < Ratio < e^5), (e^5 < Ratio < e^6), (e^6 < Ratio < e^7), (Ratio > e^7). The 8 classes cover the following intervals: 1 - 2.72 - 7.39 - 20.09 - 54.60 - 148.41 - 403.43 - 1,096.63 - > 1,096.63.

The molecules depicted in our MIM in Fig. 1 may be considered nodes connected by edges [22]. We counted the number of edges as follows:

[total number of edges] = Σ (for each of the 1,2,3,4 perturbing species,
distance in terms of edges from the perturbed species)

We obtained significant indications about sensitivity / robustness of the network described in our MIM. In this article we only report the case of the four perturbing species combination (1,000 randomly selected perturbed species), see Fig. 2.

2 Results

In our work we reconstructed the four pathways mentioned in the Introduction section and that are drawn in our MIM. All four pathways are joined because they are downstream of the ErbB-family of receptors. Based on a careful search through literature data [2,15,17,18,19,20], we inserted quantitative parameters related to concentrations, association rates, dissociation rates and catalytic turnover numbers (Table 3SM of [7]). Interpolation of some values is explained in the Materials and Methods Section of [7]. Additional comments are given in Table 3SM of [7]. We simulated the attainment of a stationary state by representing all our reactions through ordinary differential equations (ODEs). Our network reaches a stationary state because it describes a relatively short period (in the order of 20 - 30 min) of the G0 - G1 transition. During this time interval we only introduced post-translational modifications and formation of new complexes, but not protein neo-synthesis or degradation. Our network not only can simulate a "physiological" condition, but we can also introduce several oncogenic virtual mutations as well as the activity of virtual inhibitors of onco-proteins affected by an excess of function.

2.1 Molecular Interaction Map Supporting Our Simulations

The MIM which represents the molecular network anatomy supporting our dynamic simulations is shown in Fig. 1. The molecular anatomy of the network was reconstructed from a large number of literature reports describing a given protein-protein interaction that became an edge between two nodes of the MIM. These reports are quoted in the annotation list of our previous work [7]. MIMs are based on a system of symbols and syntactic conventions (Fig. 1, insert): reactions operate on molecular species; contingencies operate on reactions or on other contingencies. A complete description of how to draw a MIM, with examples, is provided in [13]. As a suggestion for the cancer researcher reader, becoming familiar with the network described in our MIM will make it easier to understand the comments explaining the results of our simulations.

2.2 Sensitivity / Robustness of Our Simulations

We calculated deviations from the standard behavior, obtained according to best parameters assignment (see Table 3SM of [7]). We examined the behavior of our network at the EGF concentration .1 nM. For each of the 34 perturbing basic molecular species, we observed the effects on the (n-1) perturbed basic species. The fluctuations introduced around the standard basic species concentration (sbsc) were as follows: sbsc x 10; sbsc/10. Deviations were measured as ratios (absolute values) with respect to the corresponding physiological concentration value.

Deviations from the corresponding physiological point were measured and classified as a function of the number of edges separating the perturbing from the perturbed species. For more than one perturbation we computed the global number edges as follows:

[total number of edges] = Σ (for each of the 1,2,3,4 perturbing species, distance in terms of edges from the perturbed species)

Table 1. 1) One perturbation 10x, 2) One perturbation 10/, 3) Two perturbations; 4) Three perturbations, 5) Four Perturbations. In cases 3), 4) and 5) only 1000 samples were randomly selected.

	e^0<R<e^1	e^1<R<e^2	e^2<R<e^3	e^3 <R<e^4	e^4<R<e^5	e^5<R<e^6	e^6<R<e^7	R>e^7
1	1085	25	7	3	0	1	1	0
2	1101	15	4	1	1	0	0	0
3	965	20	6	4	0	4	0	1
4	947	34	15	3	0	1	0	0
5	924	42	25	4	0	3	2	0

We report the following results:

1. One perturbation: effects on the 34x33 (perturbing x perturbed) combinations, for each of the 34 perturbing species deviating 10x. Results are reported in Tab. 1.
2. One perturbation: effects on the 34x33 (perturbing x perturbed) combinations, for each of the 34 perturbing species deviating 10/. Results are reported in Tab. 1.
3. Two perturbations: the number of perturbing combinations is (34x2)x(33x2)/2 = 2,244. The computing time on our pc was 22x2,244 seconds, approximately ≡ 13.7 hours, a still acceptable computing time. The complete set of perturbed species comprised 71,808 perturbations but we randomly sorted only 1,000 cases of perturbations from the entire population. Results are reported in Tab. 1. At point 3, 4 and 5 we considered both 10x and 10/ perturbations.
4. Three perturbations: for reasons of computing time we randomly sorted 1,000 cases of perturbations from the entire population of 47,872 cases. 1,000 randomly sorted perturbing combinations generate in this case 31,000 combinations of perturbed combinations. As for point 3., we randomly sorted only 1,000 of them. Results are reported in Tab. 1.
5. Four perturbations: for reasons of computing time we randomly sorted 1,000 cases of perturbations from the entire population of 742,016 cases. 1,000 randomly sorted perturbing combinations generate in this case 30,000 combinations of perturbed combinations. As for point 3., we randomly sorted only 1,000 of them. Results are reported in Tab. 1 and Fig. 2.
6. From the whole of the two-perturbation set we sorted out a subset of 20 most deviating perturbations (ratio > e^7). We combined them with 31x2 (10x + 10/) different perturbations, in order to explore if there was a third perturbation increasing significantly the deviation from the physiological condition. In alternative, the two initial perturbations could be the most important ones for the final deviation. Results are reported in Tab. 2 and in Fig. 3, A.
7. From the whole of the two-perturbation set it was possible to sort out a subset of 20 least deviating perturbations. We combined them with (31x2) different perturbations, in order to explore the opposite condition in respect to point 6. Results are reported in Tab. 2 and in Fig. 3, B.

If, for most perturbing species, we consider (see Fig. 17 and 18 of [7]) the one edge distance major fluctuations as being the most typical, we see that they are, in general, direct protein-protein interactions. In this case, if one of the two

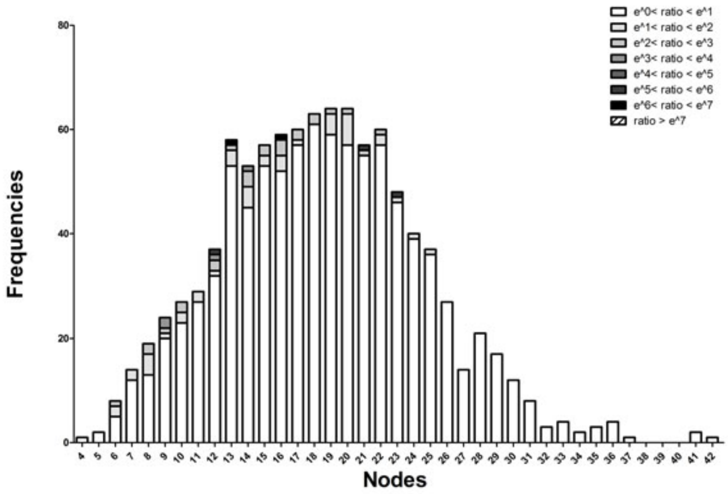


Fig. 2. Four perturbations. Notice that in this sorting the interval "ratio $> e^7$" is an empty class.

Table 2. Two-perturbation subset: 20 most deviating or least deviating perturbations, combined with a third (31x + 31/) perturbation

	e^0<R<e^1	e^1<R<e^2	e^2<R<e^3	e^3 <R<e^4	e^4<R<e^5	e^5<R<e^6	e^6<R<e^7	R>e^7
6	84	17	13	5	30	21	46	1024
7	1177	11	44	6	0	2	0	0

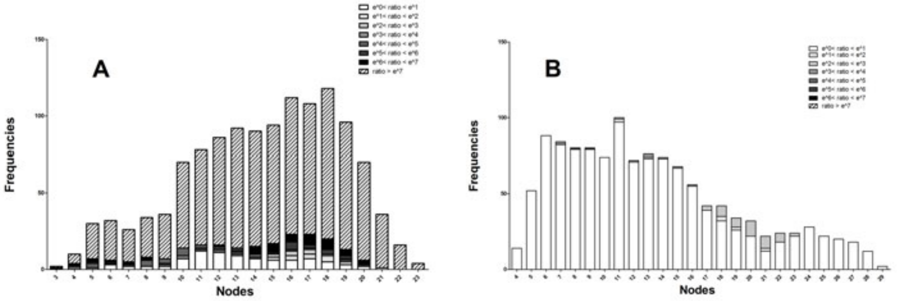


Fig. 3. 20 most (A) or least (B) deviating perturbations from the two-perturbation set, combined with (n-3) 10x+10/ perturbations (31x20x2=1240 perturbations)

components has a 10x concentration and their affinity is relatively high ($K_d \approx 10^{-8}$ M), the protein in excess will eat up the less abundant protein, leaving very little of the free species of the invariant-concentration protein concerned. As an example, we carried out an in-depth examination to see what happens to the strongest deviations and we confirmed our hypothesis. The situation is more intuitive in the cases in which there is a one edge distance. When two molecules are separated by two, and especially by three edges, the situation is much less intuitive, and requires careful study of the network, and above all, simulations must be carried out. With regards to the largest class with minimum deviation ($e^0 <$ Ratio $< e^1$), always for one perturbing species, the modal value of the number of edges is not 1 (as in the most deviating classes) but 4, and there is a long tail up to 13 edges. This upper value reflects the molecular anatomy (number of molecule-molecule interactions) of our MIM.

3 Discussion

A preliminary overall comparison of the results of this exploration of parameter space, with the variations induced by mutations and onco-protein inhibitors [7], suggests that these are in general larger than most of the random perturbations. We analyzed our network from a stability point of view. Tab. 1 and Fig. 2 show that only rare combinations of perturbing and perturbed species are sensitive to a perturbation. Furthermore, these larger perturbations are closely linked to direct reversible protein-protein interactions, where the perturbing protein which is now in vast excess is "eating up" the less concentrated perturbed protein (or the other way around). These disproportions can sometimes reverberate their effect at a distance of two - three edges, because all the interactions are connected, but progressively more weakly as the number of edges increases.

Both Tab. 2 and Fig. 3 clearly show the strong bi-directional effect of a previous two-perturbation condition on the subsequent addition of a new perturbation.

These explorations of network stability will have to be extended to complexes and rates. In a previous work we noticed that the perturbation of rates has smaller effects than the perturbation of concentrations [23]. Parameter space can also be explored more systematically using different strategies [2], therefore, we are considering strategies of the type used by these authors for a future study. Computations shared in a grid may be useful to decrease computational time and allow a more extensive exploration of parameter space. The overall robustness of our network increases the relevance of the large and reasonable changes we observed in the presence of mutations and inhibitors.

Acknowledgments

This work was supported by: MIUR PRIN 2007-prot. 2005061408_005; national project Compagnia di San Paolo 2489 IT/CV, 2006.1997, Genoa Operative Unit (O.U.); Regione Liguria, Cap. 5296 F.S.R. Nr. Prov. 23, Prot. Gen. 616, 2007 and dgr n. 1487 (NP/16837) Nov. 06 2009; Deliber. GR n. 280 (09/02/2010); Istituto Superiore di Oncologia, O.U.s of an "Istituto Oncologico Veneto"project, and a "Regione Campania" project; project Compagnia di San Paolo 4998 IT/CV, 2007.0087, Genoa University; Project LIMONTE-SP1P 2009 (with Castagnola and Zardi); Project CARIGE, Genoa University 2008 (with Patrone, Ballestrero et al.); Project CARIGE Genoa University Prot. n. 513 (29/07/2010); Fellowships to S.P.; E.M. and A.N. are supported by "Associazione Italiana per la Ricerca sul Cancro".

References

1. Aldridge, B.B., Burke, J.M., Lauffenburger, D.A., Sorger, P.K.: Physicochemical modeling of cell signalling pathways. Nat. Cell Biol. 8(11), 1195–1203 (2006)
2. Chen, W.W., Schoeberl, B., Jasper, P.J., Niepel, M., Nielsen, U.B., Lauffenburger, D.A., Sorger, P.K.: Input-output behavior of ErbB signaling pathways as revealed by a mass action model trained against dynamic data. Mol. Syst. Biol. 5, e239 (2009)
3. Sjöblom, T., Jones, S., Wood, L.D., Parsons, D.W., Lin, J., Barber, T.D., Mandelker, D., Leary, R.J., Ptak, J., Silliman, N., Szabo, S., Buckhaults, P., Farrell, C., Meeh, P., Markowitz, S.D., Willis, J., Dawson, D., Willson, J.K., Gazdar, A.F., Hartigan, J., Wu, L., Liu, C., Parmigiani, G., Park, B.H., Bachman, K.E., Papadopoulos, N., Vogelstein, B., Kinzler, K.W., Velculescu, V.E.: The consensus coding sequences of human breast and colorectal cancers. Science 314(5797), 268–274 (2006)
4. Wood, L.D., Parsons, D.W., Jones, S., Lin, J., Sjöblom, T., Leary, R.J., Shen, D., Boca, S.M., Barber, T., Ptak, J., Silliman, N.S., Szabo, S., Dezso, Z., Ustyanksky, V., Nikolskaya, T., Nikolsky, Y., Karchin, R., Wilson, P.A., Kaminker, J.S., Zhang, Z., Croshaw, R., Willis, J., Dawson, D., Shipitsin, M., Willson, J.K., Sukumar, S., Polyak, K., Park, B.H., Pethiyagoda, C.L., Pant, P.V., Ballinger, D.G., Sparks, A.B., Hartigan, J., Smith, D.R., Suh, E., Papadopoulos, N., Buckhaults, P., Markowitz, S.D., Parmigiani, G., Kinzler, K.W., Velculescu, V.E., Vogelstein, B.: The genomic landscapes of human breast and colorectal cancers. Science 318, 1108–1113 (2007)
5. Lin, J., Gan, C.M., Zhang, X., Jones, S., Sjöblom, T., Wood, L.D., Parsons, D.W., Papadopoulos, N., Kinzler, K.W.: A multidimensional analysis of genes mutated in breast and colorectal cancers. Genome Res. 17(9), 1304–1318 (2007)
6. Jones, S., Zhang, X., Parsons, D.W., Lin, J.C., Leary, R.J., Angenendt, P., Mankoo, P., Carter, H., Kamiyama, H., Jimeno, A., Hong, S.M., Fu, B., Lin, M.T., Calhoun, E.S., Kamiyama, M., Walter, K., Nikolskaya, T., Nikolsky, Y., Hartigan, J., Smith, D.R., Hidalgo, M., Leach, S.D., Klein, A.P., Jaffee, E.M., Goggins, M., Maitra, A., Iacobuzio-Donahue, C., Eshleman, J.R., Kern, S.E., Hruban, R.H., Karchin, R., Papadopoulos, N., Parmigiani, G., Vogelstein, B., Velculescu, V.E., Kinzler, K.W.: Core signaling pathways in human pancreatic cancers revealed by global genomic analyses. Science 321(5897), 1801–1806 (2008)
7. Castagnino, N., Tortolina, L., Balbi, A., Pesenti, R., Montagna, R., Ballestrero, A., Soncini, D., Moran, E., Nencioni, A., Parodi, S.: Dynamic simulations of pathways downstream of ErbB-family, including mutations and treatments. Concordance with experimental results. Current Cancer Drug Targets 10, 737–757 (2010)
8. Alon, U.: An Introduction to Systems Biology: Design Principles of Biological Circuits, 1st edn. Chapman and Hall/CRC Mathematical and Computational Biology Series. Taylor and Francis Group, London (2006)
9. Mukherji, M., Brill, M., Ficarro, L.M., Hampton, S.B., Schultz, G.M.,, P.G.: A phosphoproteomic analysis of the ErbB2 receptor tyrosine kinase signaling pathways. Biochemistry 45(51), 15529–15540 (2006)

10. Catalogue of Somatic Mutation In Cancer (COSMIC), `http://www.sanger.ac.uk/genetics/CGP/cosmic/`
11. Leslie, N.R., Downes, C.P.: PTEN function: how normal cells control it and tumour cells lose it. Biochem. J. 382(1), 1–11 (2004)
12. Orsulic, S., Huber, O., Aberle, H., Arnold, S., Kemler, R.: E-cadherin binding prevents β-catenin nuclear localization and β-catenin/LEF-1-mediated transactivation. J. Cell. Sci. 112(8), 1237–1245 (1999)
13. Kohn, K.W., Aladjem, M.I., Kim, S., Weinstein, J.N., Pommier, Y.: Depicting combinatorial complexity with the molecular interaction map notation. Mol. Syst. Biol. 2, e51 (2006)
14. Kholodenko, B.N., Hoek, J.B., Westerhoff, H.V.: Why cytoplasmic signalling proteins should be recruited to cell membranes. Trends Cell Biol. 10(5), 173–178 (2000)
15. Wolf, J., Dronov, S., Tobin, F., Goryanin, I.: The impact of the regulatory design on the response of epidermal growth factor receptor-mediated signal transduction towards oncogenic mutations. FEBS J. 274(21), 5505–5517 (2007)
16. Tyson, J.J., Novak, B., Odell, G.M., Chen, K., Thron, C.D.: Chemical kinetic theory: understanding cell-cycle regulation. Trends Biochem. Sci. 21(3), 89–96 (1996)
17. Kholodenko, B.N., Demin, O.V., Moehren, G., Hoek, J.B.: Quantification of short term signaling by the epidermal growth factor receptor. J. Biol. Chem. 274(42), 30169–30181 (1999)
18. Markevich, N.I., Moehren, G., Demin, O.V., Kiyatkin, A., Hoek, J.B., Kholodenko, B.N.: Signal processing at the Ras circuit: what shapes Ras activation patterns? Syst. Biol (Stevenage) 1(1), 104–113 (2004)
19. Kiyatkin, A., Aksamitiene, E., Markevich, N.I., Borisov, N.M., Hoek, J.B., Kholodenko, B.N.: Scaffolding protein Grb2-associated binder 1 sustains epidermal growth factor-induced mitogenic and survival signaling by multiple positive feedback loops. J. Biol. Chem. 281(29), 19925–19938 (2006)
20. Birtwistle, M.R., Hatakeyama, M., Yumoto, N., Ogunnaike, B.A., Hoek, J.B., Kholodenko, B.N.: Ligand-dependent responses of the ErbB signaling network: experimental and modeling analyses. Mol. Syst. Biol. 3, e144 (2007)
21. BioModels Database, `http://www.ebi.ac.uk/biomodels-main/`
22. Barabási, A.L., Oltvai, Z.N.: Network biology: understanding the cell's functional organization. Nat. Rev. Genet. 5(2), 101–113 (2004)
23. Castagnino, N., Tortolina, L., Montagna, R., Pesenti, R., Balbi, A., Parodi, S.: Simulations of the EGFR - KRAS - MAPK signalling network in colon cancer. Virtual mutations and virtual treatments with inhibitors have more important effects than a 10 times range of normal parameters and rates fluctuations. In: Masulli, F., Peterson, L.E., Tagliaferri, R. (eds.) CIBB 2009. LNCS, vol. 6160, pp. 151–164. Springer, Heidelberg (2010)

Robustness Analysis of a Linear Dynamical Model of the *Drosophila* Gene Expression

Alexandre Haye, Jaroslav Albert, and Marianne Rooman

Unité de Bioinformatique génomique et structurale,
Université Libre de Bruxelles, 1050 Brussels, Belgium
{ahaye,jalbert,mrooman}@ulb.ac.be

Abstract. The evolution of the gene expression levels of Drosophila melanogaster, from the embryonic to adult development phases, has been studied on the basis of a microarray time series involving the expression levels of more than 4000 genes over 67 time-points, and has been modeled by a system of linear differential equations with constant coefficients. Here we investigate the robustness of this model against perturbations of its parameters and of the initial data values. We found that the model is not robust at all for fully connected networks, but that the robustness significantly increases after parameter reduction. This puts some limits to the biological relevance of linear models for gene expression evolution.

Keywords: gene regulatory networks, mathematical modeling, model robustness.

1 Introduction

Some of us have recently developed a linear model capable of reproducing the time evolution of gene expression across the embryonic-to-adult development phases of *Drosophila melanogaster* [1]. The transcript concentration profiles that were modeled were taken from a 67-point DNA microarray time series involving 4028 genes [2]. To reduce the dimensionality of the problem, the genes having similar expression profiles were grouped into 17 clusters [3]. This simple linear model was shown to be able to reproduce the experimental data with very good accuracy. Remarkably, the parameter reduction allowed the elimination of up to 80–85% of these connections while keeping fairly good fit between data and simulation. This result supports the low-connectivity hypothesis of gene expression networks, with about three connections per cluster, without introducing *a priori* hypotheses such as an upper limit on the number of connections per gene [4, 5, 6]. This is in accordance with experimental evidence about gene regulatory networks [7].

However, for our model to have some biological relevance, its performance in mimicking data is not sufficient. It has also to be reasonably robust against parameter perturbations and changes in initial conditions. In particular, a model of gene regulatory network in which a slight perturbation of some connections, *i.e.* of some interactions between genes and gene products, would lead to totally different expression

R. Rizzo and P.J.G. Lisboa (Eds.): CIBB 2010, LNBI 6685, pp. 242–252, 2011.
© Springer-Verlag Berlin Heidelberg 2011

profiles, with possibly unstable behaviors, is irrelevant. Similarly, a slight perturbation of some expression levels, due to the stochastic nature of these interactions, should not affect the profiles too strongly. Linear models are of course expected to yield divergent behaviors, but the question is whether these occur for biologically expected perturbations and over biologically relevant times. With this in mind, we tested the robustness of the linear model of *Drosophila* gene expression by perturbing its parameters and initial values and compared the estimated expression curves before and after perturbation. This procedure was performed before and after parameter reduction, so as to analyze the effect of the connectivity on the robustness.

2 Background: The Model

We briefly recall our model of *Drosophila* gene expression. Details can be found in [1]. The expression profiles $x_i(t)=\log_2(I_i(t)/I_{iR})$ for the 4028 *Drosophila* genes labeled by *i* are expressed in terms of the intensities $I_i(t)$ and I_{iR} measured by DNA microarray techniques [2] where $I_i(t)$ was measured at v=67 time-points denoted by τ_k. The four developmental phases, *i.e.* embryonic, larva, pupal, and adult phases, contained 31, 10, 18, and 8 time-points respectively. I_{iR} is a reference intensity independent of the development stage. These 4028 genes were grouped into 17 clusters containing genes with similar expression profiles [3]. Each cluster was represented by the average expression profile $x_c(t)$, where the index *c* labels the clusters.

To reproduce these profiles, a linear model with constant coefficients was chosen, in which the time derivative of the gene expression level xc only depends on the evolution of the gene expression levels x_d of all clusters *d* [8]. Defining the vector $\mathbf{x}=(x_1,x_2,...,x_n)^T$, where n=17 is the number of clusters, the model takes the form

$$\frac{d\,\mathbf{x}}{d\,t}=\mathbf{M}\,\mathbf{x} \quad , \tag{1}$$

where *t* is the (real) time and **M** a *n*x*n* matrix with constant elements.

To estimate the n^2 elements of **M,** a two step procedure was used. The first step involved an estimation of the time derivatives d$x_c(t)$/d*t* followed by a least square estimation of **M**, denoted $\hat{\mathbf{M}}^{\mathrm{LS}}$. To have an objective quantification of the quality of the modeling, we defined the cost function or score $S(\hat{\mathbf{x}})$, which corresponds to the standard deviation of experimental and modeled profiles $\mathbf{x}(t)$ and $\hat{\mathbf{x}}(t)$, weighted by the inverse of the variance $\sigma_c(\tau_k)^2$ of the experimental data:

$$S(\hat{\mathbf{x}})=\sqrt{\frac{1}{n\,v}\sum_{c=1}^{n}\sum_{k=1}^{v}\frac{\left(x_c(\tau_k)-\hat{x}_c(\tau_k)\right)^2}{\sigma_c(\tau_k)^2}} \tag{2}$$

The weighting by $\sigma_c(\tau_k)^2$ ensures that the lower the disparity of the data at a certain time-point τ_k, the larger the weight in the cost function and thus, the more important the quality of reproduction at that point.

In the second step, a nonlinear parameter estimation was performed, using as initial parameter values those obtained by the linear identification procedure. The algorithm used is a simplex search method [9]. This procedure yielded the $\hat{\mathbf{M}}^{\mathrm{Opt}}$ matrix and the initial values $\hat{\mathbf{x}}^{\mathrm{Opt}}(\tau_0)$ that minimize a cost function $S^{\mathrm{Opt}}(\hat{\mathbf{x}})$.

The estimated matrix $\hat{\mathbf{M}}^{\mathrm{Opt}}$ that encodes the mutual influence of the gene clusters has no vanishing parameters, whereas other parameter sets, with some vanishing parameters, could possibly model the expression profiles almost as well. In order to find such sets, a parameter reduction of the model was performed, with the aim of determining the connections that are necessary to keep a good profile modeling, and which can be viewed as biologically relevant.

The reduction procedure used is iterative and draws a unique trajectory among the parameters to eliminate. At each iteration, the parameter that, when cancelled, induces the smallest $S(\hat{\mathbf{x}}^{\mathrm{LS}})$ was permanently set to zero, yielding the reduced parameter matrix $\hat{\mathbf{M}}_N^{\mathrm{LS}}$, where *N* is the number of eliminated parameters. A nonlinear parameter optimization was then performed so as to minimize $S(\hat{\mathbf{x}}^{\mathrm{Opt}})$ while maintaining the eliminated parameters in $\hat{\mathbf{M}}_N^{\mathrm{LS}}$ equal to zero; this yields the parameter matrix $\hat{\mathbf{M}}^{\mathrm{Opt}}$. The value $S(\hat{\mathbf{x}}^{\mathrm{Opt}})$ was shown to remain roughly constant or to increase very slowly until the number of eliminated parameters *N* reaches 227. At this point, $S(\hat{\mathbf{x}}^{\mathrm{Opt}})$ =0.44 and 62 parameters remained, which amounts to an average of 3.65 connection per gene cluster. Further parameter reduction led to a significant jump in $S(\hat{\mathbf{x}}^{\mathrm{Opt}})$.

3 Methods

3.1 Robustness of the Model against Perturbations of the Parameters

We analyzed the robustness of the full and reduced gene expression networks defined by $\hat{\mathbf{M}}^{\mathrm{Opt}}$ and $\hat{\mathbf{M}}_N^{\mathrm{Opt}}$ against two types of perturbations: individual (P_1) and collective (P_{All}). In the first case the elements of $\hat{\mathbf{M}}^{\mathrm{Opt}}$ or $\hat{\mathbf{M}}_N^{\mathrm{Opt}}$ were modified one at a time by adding or subtracting a given percentage of its original value, which we chose to be P_1=±1% or P_1=±5%. The expression profile $\hat{\mathbf{x}}_N^{P_1=p}(t)$, estimated with these perturbed parameters, and the associated cost function $S(\hat{\mathbf{x}}_N^{P_1=p})$ were then computed. For each perturbation $P_1=p$ of a given parameter $\mathrm{M_{cd}}$, the cluster and time-point τ_k for which the deviation between the unperturbed and perturbed estimated profiles $\hat{\mathbf{x}}^{\mathrm{Opt}}(\tau_k)$ and $\hat{\mathbf{x}}_N^{P_1=p}(\tau_k)$ is maximum was identified. Finally, the minimum ($Min[S(\hat{\mathbf{x}}_N^{P_1=p})]$) and maximum ($Max[S(\hat{\mathbf{x}}_N^{P_1=p})]$) values of these deviations, obtained when perturbing each of the parameters $\mathrm{M_{cd}}$ individually, were considered for interpretation.

In the second type of perturbation, all parameters $\mathrm{M_{cd}}$ of the network were modified at the same time. This was done by adding to each $\mathrm{M_{cd}}$ a (different) random percentage of its value, where these random percentages are contained in [-*p*, +*p*], with *p*=1% or *p*=10%. The expression profiles $\hat{\mathbf{x}}_N^{P_{All}=p}(t)$ and the scores $S(\hat{\mathbf{x}}_N^{P_{All}=p})$

estimated with these perturbed parameters were then computed. This procedure was repeated 50 times for different random perturbations. The average value of $S(\hat{\mathbf{x}}_N^{P_{All}=p})$ was then computed along with the standard deviation. Given that the score is always positive and deviates from a normal distribution, a left and a right standard deviation was computed, σ_L and σ_R. These were defined by considering either all scores that are smaller than the mean (σ_L) or all scores that are larger than the mean (σ_R).

3.2 Robustness of the Model against Perturbation of the Initial Conditions

Another type of perturbation we considered involves the modification of the estimated initial conditions $\hat{\mathbf{x}}^{\mathrm{Opt}}(\tau_0)$ or $\hat{\mathbf{x}}_N^{\mathrm{Opt}}(\tau_0)$, rather than the network parameters $\hat{\mathbf{M}}^{\mathrm{Opt}}$ or $\hat{\mathbf{M}}_N^{\mathrm{Opt}}$. Here also we consider two types of perturbations, the individual and collective parameter perturbations, noted P_{in1} and P_{inAll}, which are defined in exactly the same way as P_1 and P_{All}.

4 Results and Discussion

4.1 Robustness against Network Perturbations

Single Parameter Perturbations

The evolution of the scores before and after single parameter perturbation, *i.e.* $S(\hat{\mathbf{x}}_{\mathrm{N}}^{\mathrm{Opt}})$ and $S(\hat{\mathbf{x}}_{\mathrm{N}}^{\mathrm{P_1=p}})$ with $p=\pm1\%$ and $p=\pm5\%$, is given as a function of the number of eliminated parameters (N) in Figure 1. More precisely, for each value of N, the smallest ($Min[S(\hat{\mathbf{x}}_N^{P_1=p})]$)) and largest ($Max[S(\hat{\mathbf{x}}_N^{P_1=p})]$) scores among those obtained by perturbing one of the parameters (see Methods section) are depicted. For clarity, we used the $\log_{10}$ of the scores instead of the scores themselves in all the figures.

In Figure 1, the green dashed lines are identical to the blue dotted line; this means that, whatever the number of parameters in the network, there is always at least one parameter that can be perturbed without changing significantly the mean score. In contrast, it can be seen that when $N<215$ (that is, when up to 74% of the parameters are set to zero), the model is always very sensitive to the perturbation of at least one parameter. The model remains sensitive up to $N=241$ (when 83% of the parameters vanish) for larger perturbations of ±5%.

As a consequence, the linear model is always robust against the perturbation of some particular connections, but only becomes robust against all single parameter perturbations when the network is reduced to about 3 connections per gene cluster.

Collective Parameter Perturbations

The behavior of the score $S(\hat{\mathbf{x}}^{\mathrm{Opt}})$ as a function of N, when all parameters are modified at the same time (P_{All}), is shown in Fig. 2, for $p=1\%$ or $p=10\%$. The mean value $<S(\hat{\mathbf{x}}_N^{P_{All}=p})>$ computed over 50 random perturbations is indicated as well as the confidence interval [$<S(\hat{\mathbf{x}}_N^{P_{All}=p})>-\sigma_L$, $S(\hat{\mathbf{x}}_N^{P_{All}=p})>+\sigma_R$] (see Methods).

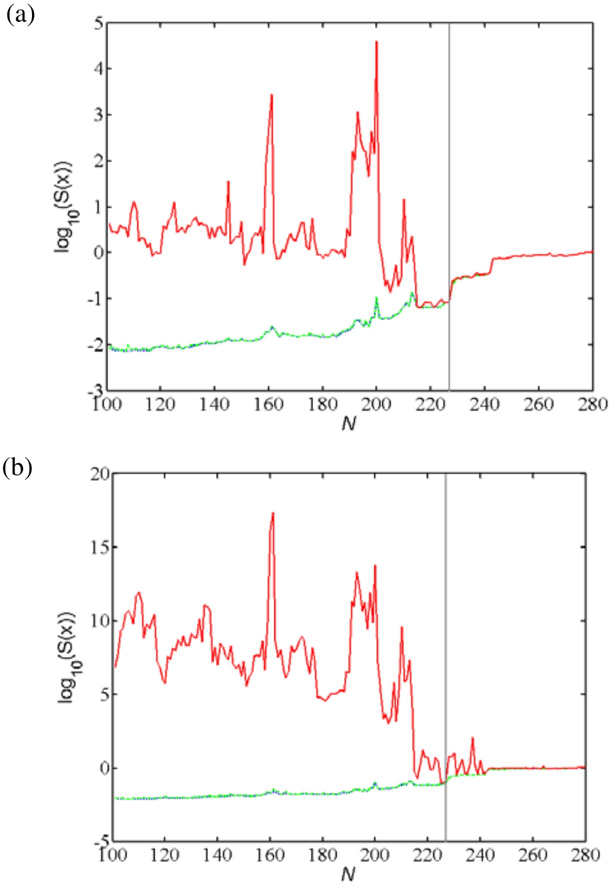


Fig. 1. Log$_{10}$ of the scores S before and after single parameter perturbations P_1 = ±1% (a) and P_1 =± 5% (b), as a function of the parameter reduction N. The vertical grey line indicates the optimal reduction N=227. Blue dotted line: log$_{10}$ of the score $S(\hat{\mathbf{x}}_{\mathrm{N}}^{\mathrm{Opt}})$ without perturbation; green dashed line and red solid line: log$_{10}$ of the scores $Min[S(\hat{\mathbf{x}}_{\mathrm{N}}^{\mathrm{P}_1=\mathrm{p}})]$ and $Max[S(\hat{\mathbf{x}}_{N}^{\mathrm{P}_{\mathrm{in1}}=\mathrm{p}})]$, respectively.

The first observation is that on the left hand side of the Fig. 2b, that is, when the model still contains more than 74 parameters (N<215), the score after parameter perturbation is significantly higher than before perturbation: on average these scores are approximately equal to 10^{60} and 0.5, respectively. This implies that, in this range of N, our linear model is totally unstable with respect to even small perturbations of all parameters at the same time. Nevertheless, when the number of parameters decreases below 48 (N=241), the mean score before and after perturbation become close. However this threshold is much below the N=227 threshold of optimal reduction, above which the estimated and experimental profiles are considered to be insufficiently similar [1].

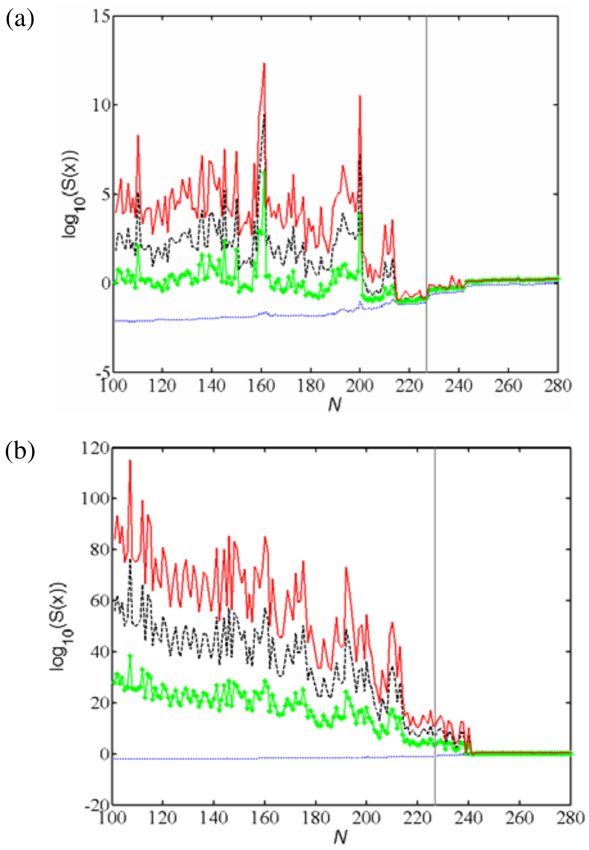


Fig. 2. Log$_{10}$ of the scores S before and after collective parameter perturbations P_{All} = 1% (a) and P_{All} = 10% (b), as a function of the parameter reduction N. The vertical grey line corresponds to the optimal reduction N=227. Blue dotted line: log$_{10}$ of the score $S(\hat{\mathbf{x}}^{\mathrm{Opt}})$ before perturbation; black dashed line: log$_{10}$ of the mean values $< S(\hat{\mathbf{x}}_N^{P_1=p}) >$ of the scores after 50 different random perturbations; green asterisk-line and red solid line: log$_{10}$ of $\left(< S(\hat{\mathbf{x}}_N^{P_1=p}) > -\sigma_{\mathrm{L}}\right)$ and $\left(< S(\hat{\mathbf{x}}_N^{P_1=p}) > +\sigma_{\mathrm{R}}\right)$, respectively.

Robustness of Particular Reduced Solutions

Let us now focus on the optimal reduced parameter network, with N=227, obtained previously as described in Section 2 [1]. In this case, 62 parameters remain in the network, and the mean connectivity is thus 3 to 4 connections per cluster. Typical estimated curves $\hat{\mathbf{x}}_N^{P_1=p}(t)$ and $\hat{\mathbf{x}}_N^{P_{All}=p}(t)$ obtained by perturbing all parameters individually and together are shown in Fig. 3 and are compared to the reduced unperturbed profile $\hat{\mathbf{x}}_N^{\mathrm{Opt}}(t)$ and to the data.

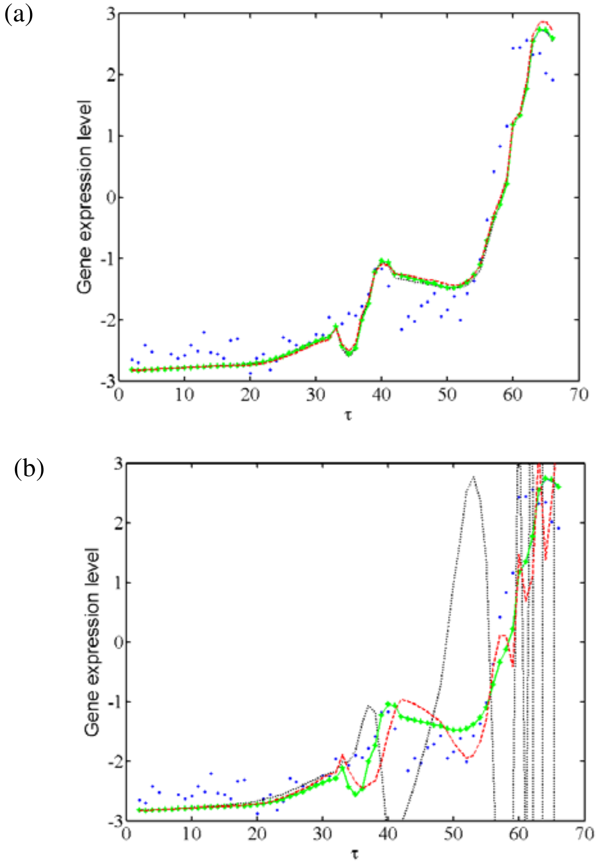


Fig. 3. Real and estimated expression profiles for cluster 11 after optimal reduction and before and after single-parameter and collective perturbations. Blue points: experimental data $\mathbf{x}(\tau_k)$; blue solid line: unperturbed estimated profiles $\hat{\mathbf{x}}_N^{\mathrm{Opt}}(\tau_k)$; green asterisk-line and red dashed line: two reduced profiles $\hat{\mathbf{x}}_N^{P_1=p}(\tau_k)$ corresponding to the single-parameter perturbation P_1 leading to $Min[S(\hat{\mathbf{x}}_{\mathrm{N}}^{\mathrm{P_1=p}})]$ and $Max[S(\hat{\mathbf{x}}_N^{\mathrm{P_1=p}})]$, respectively; black dotted line : estimated profile $\hat{\mathbf{x}}_N^{P_{All}=p}(t)$ of one particular set of collective perturbations P_{All}. (a) P_1=±1% and P_{All}=1%. Note that the curves in blue (solid), green (asterisk), black (dotted) and red (dashed) almost coincide. (b) P_1=±5% and P_{All}=10%.

As shown in Fig. 3a, the estimated expression profiles after parameter reduction up to N=227 are largely insensitive to both single-parameter and collective perturbations, when these perturbations remain small, *i.e.* P_1=±1% and P_{All}=1%. The results are similar for all tested random perturbations with P_{All}=1% and for the other clusters. However, as seen in Fig. 3b, when the parameters of the network are more strongly perturbed, *i.e.* when P_1=±5% and even more when P_{All}=10%**,** we observe large changes in the estimated profiles with a tendency to unstable behavior. Note that this does not happen for all tested random perturbations with P_{All}=10%, because some parameters are less sensitive to noise and moreover in some random realizations many perturbation values can be close to zero.

This figure shows that the expression profiles obtained after optimal parameter reduction are robust against parameter perturbations as long as these perturbations remain small. In contrast, the profiles resulting from less (or not) reduced networks are not robust at all (Figs. 1-2).

4.2 Robustness against Perturbation of the Initial Conditions

Perturbation of Individual Initial Gene Expression Values

The behavior of the score before and after perturbation of one of the initial conditions, with P_{in1}=1% or P_{in1}=5%, is shown in Fig. 5. The linear model appears to be not very

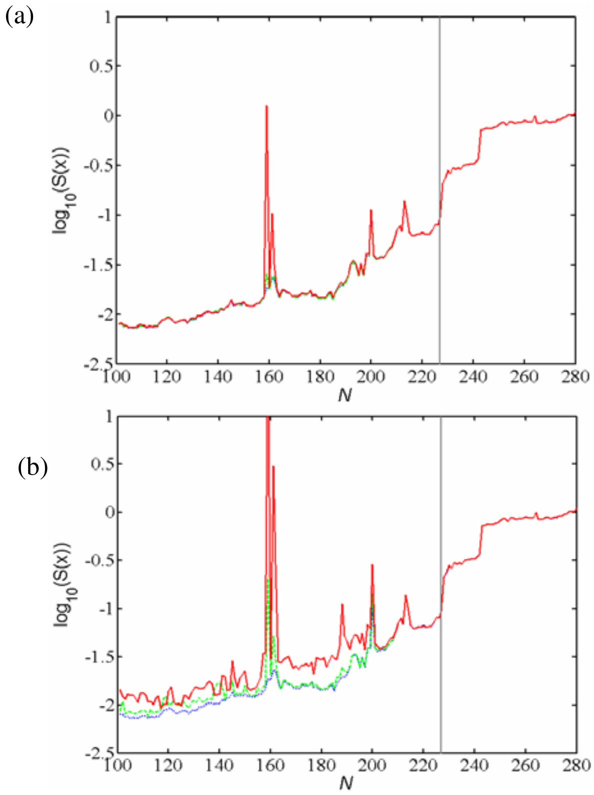


Fig. 4. Log$_{10}$ of the scores S before and after single parameter perturbations P_{in1} = ±1% (a) and P_{in1} =± 5% (b), as a function of the parameter reduction N. The vertical grey line indicates the optimal reduction N=227. Blue dotted line: log$_{10}$ of the score $S(\hat{\mathbf{x}}^{\mathrm{Opt}})$ before perturbation; green dashed line and red solid lines: log$_{10}$ of the scores $Min[S(\hat{\mathbf{x}}_{\mathrm{N}}^{\mathrm{P_l=p}})]$ and $Max[S(\hat{\mathbf{x}}_{N}^{\mathrm{P_{in1}=p}})]$, respectively.

sensitive to the perturbation of any of the initial conditions. Moreover, the model's robustness against such perturbations appears to be much less dependent on the number of parameters in the network than on the parameters of the network themselves. Note that the peak around *N*=160 may correspond to poorly optimized unperturbed solutions, which become even less optimal upon perturbation.

Perturbation of all Initial Gene Expression Values

Figure 5 shows the behavior, as a function of *N*, of the mean scores before and after collective perturbations P_{inAll}=1% and P_{inAll}=10% of the initial values. Clearly, the small random collective perturbations P_{inAll}=1% almost do not change the score, which remains below the unperturbed score at the optimal reduction level of *N*=227.

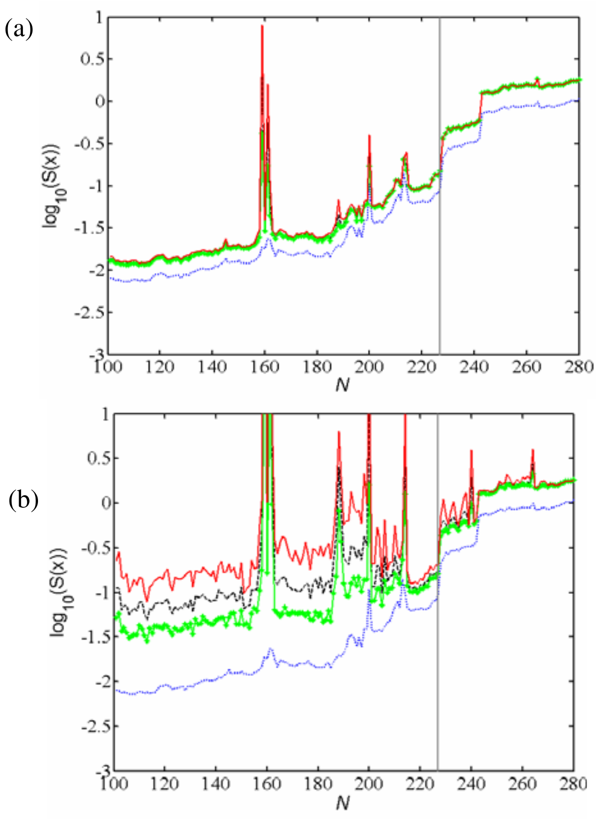


Fig. 5. Log10 of the scores S before and after collective perturbations of the initial values PinAll = 1% (a) and PinAll = 10% (b), as a function of the parameter reduction N. The vertical grey line corresponds to the optimal reduction N=227. Blue dotted line: log10 of the score $S(\hat{\mathbf{x}}^{\mathrm{Opt}})$ before perturbation; black dashed line: log10 of the mean values $<S(\hat{\mathbf{x}}_N^{P_1=p})>$ of the scores after 50 different random perturbations; green asterisk-line and red solid line: log10 of ($<S(\hat{\mathbf{x}}_N^{P_1=p})>$ - σ_L) and ($<S(\hat{\mathbf{x}}_N^{P_1=p})>$ + σ_R), respectively.

Moreover, the standard deviations are very small and, whatever the random perturbation, the new score increases by roughly the same amount. When P_{inAll} =5%, the scores as well as their standard deviations increase somewhat more, but they remain close to the score of the N=227 reduction level.

Robustness of the Particular Reduced Solutions

We focus on the optimal reduced parameter set with N= 227. As seen in Fig. 6, even when the perturbations are larger (P_{in1}=±5% or P_{inAll}=10%), the data is well reproduced by the model with the perturbed initial conditions. The results for the other clusters are similar. This shows that with this particular parameter set, our linear model is robust against perturbations of the initial conditions.

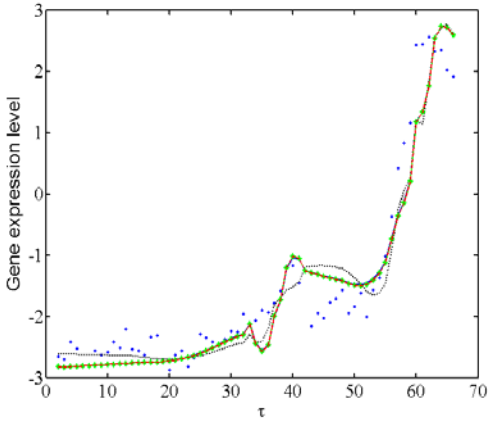


Fig. 6. Estimated expression profiles for cluster 11 after optimal reduction and before and after perturbations P_{in1}=±5% and P_{inAll}=10%. Blue points: experimental data $\mathbf{x}(t)$; blue solid line : unperturbed estimated profile $\hat{\mathbf{x}}^{\mathrm{Opt}}(\tau_k)$; green asterisk-line and red dashed line : two profiles $\hat{\mathbf{x}}_N^{P_1=p}(\tau_k)$ corresponding to the individual perturbation P_1 = ±5% of the parameter leading to $Min[S(\hat{\mathbf{x}}_{\mathrm{N}}^{\mathrm{P_1=p}})]$ and $Max[S(\hat{\mathbf{x}}_N^{\mathrm{P_1=p}})]$, respectively; black dotted line: estimated profile $\hat{\mathbf{x}}_N^{P_{All}=p}(t)$ of one particular set of collective perturbations P_{All} = 10%. Note that the curves in blue (solid), green (asterisk) and red (dashed) coincide.

5 Conclusion and Outlook

The present analysis of the robustness of the gene expression model for *Drosophila* development proposed in [1] highlights the strengths and limitations of linear models.

A first conclusion is that the fully connected linear model is very sensitive to even small perturbations of the parameters. This tendency remains true for reduced networks with vanishing parameters, as long as the number of connections is higher than 3 to 4 per gene cluster. At this point, the model starts being robust against some perturbations, but not against all. It only becomes really robust when the model stops reproducing the data correctly, that is when about 2.5 connections per cluster remain.

It has to be noted that the model is much more robust against perturbations of the initial conditions than it is with respect to the network parameters.

We may thus conclude that linear models appear to gain robustness as the number of connections decreases, but never become sufficiently robust while maintaining a good modeling capacity. Moreover, the estimated expression profiles sometimes tend to diverge after perturbation, which removes all biological relevance to the model. Therefore, nonlinear models will have to be considered to reach the biologically required robustness characteristics. In particular, nonlinear models that describe the transcription activation or repression as well as the gene product degradation [10], and present attractive stationary points during the *Drosophila* lifetime, will be considered.

Acknowledgments

This work was supported by the Belgian State Science Policy Office through an Interuniversity Attraction Poles Programme (DYSCO), the Belgian Fund for Scientific Research (FNRS) (FRFC project). AH benefits from a FRIA grant, JA is Postdoctoral Researcher and MR is Research Director at the FNRS.

References

1. Haye, A., Dehouck, Y., Kwasigroch, J.M., Bogaerts, P., Rooman, M.: Modeling the temporal evolution of the Drosophila gene expression from DNA microarray time series. Physical Biology 016004, 9 (2009) doi:10.1088/1478-3975/6/1/016004
2. Arbeitman, M.N., Furlong, E.E., Imam, F., Johnson, E., Null, B.H., Baker, B.S., Krasnow, M.A., Scott, M.P., Davis, R.W., White, K.P.: White Gene expression during the life cycle of Drosophila melanogaster. Science 297, 2270–2275 (2002)
3. Ma, P., Castillo-Davis, C.I., Zhong, W., Liu, J.S.: A Data-Driven Clustering Method for Time Course Gene Expression Data. Nucleic Acids Res. 34, 1261–1269 (2006)
4. Gardner, T.S., Faith, J.: Reverse-engineering transcription control networks. Physics of Life Reviews 2, 65–88 (2005)
5. Yeung, M.K., Tegnér, J., Collins, J.J.: Reverse engineering gene networks using singular value decomposition and robust regression. Proc. Natl. Acad. Sci. USA 99(9), 6163–6168 (2002)
6. Ciliberti, S., Martin, O.C., Wagner, A.: Innovation and robustness in complex regulatory gene networks. Proc. Natl. Acad. Sci. USA 104(34), 13591–13596 (2007)
7. Thieffry, D., Huerta, A.M., Perez-Rueda, E., Collado-Vides, J.: From specific gene regulation to genomic networks: a global analysis of transcriptional regulation in Escherichia coli. Bio. Essays 20, 433–440 (1998)
8. Chen, T., He, H.L., Church, G.M.: Modeling gene expression with differential equations. In: Proc. Pacific. Sympos., pp. 29–40 (1999)
9. Lagarias, J.C., Reeds, J.A., Wright, M.H., Wright, P.E.: Convergence Properties of the Nelder-Mead Simplex Method in Low Dimensions. SIAM Journal of Optimization 9, 112–147 (1998)
10. Albert, J., Rooman, M.: Dynamic modeling of gene expression in prokaryotes: application to glucose-lactose diauxie. Escherichia coli. Accepted to Systems and Synthetic Biology (2010)

Intelligent Clinical Decision Support Systems for Non-invasive Bladder Cancer Diagnosis

Alexandru G. Floares[1], Carmen Floares[2], Oana Vermesan[1], Tiberiu Popa[1], Michael Williams[3], Sulaimon Ajibode[3], Liu Chang-Gong[3], Diao Lixia[3], Wang Jing[3], Traila Nicola[5], David Jackson[4], Colin Dinney[3], and Liana Adam[3]

[1] OncoPredict & SAIA & IOCN, Artificial Intelligence Department,
Str. Vlahuta, Bloc Lama C/45, Cluj-Napoca, Romania
alexandru.floares@ieee.org

[2] OncoPredict & IOCN, Department of Medical Oncology, Cancer Institute,
Str. Republicii, Nr. 34-36, Cluj - Napoca, Romania
carmen.floares@iocn.ro

[3] UT-MD Anderson Cancer Center, Houston, Tx;
Departments of Urology and Bioinformatics

[4] Life Biosystems, Heidelberg, Germany

[5] UMF, Timisoara, Romania

Abstract. The aim of this study was to find the set of biomarkers based on plasma microRNAs which can predict in a noninvasive way the diagnosis of bladder cancer. We presented here a methodology and the related concepts to develop intelligent molecular biomarkers using knowledge discovery in data and artificial intelligence methods. To the best of our knowledge, this is the first time when plasma miRNAs are combined using artificial intelligence and the prediction accuracy of the developed systems for medical decision support is the best published by now, some of them having even 100%.

Keywords: bladder cancer diagnosis, noninvasive, microRNA, artificial intelligence.

1 Introduction

Bladder cancer was the fourth most common cancer diagnosis in men in the United States in 2009 and it affected nearly 71,000 people. Over fourteen thousand people succumbed to the disease during the same time frame [8]. From a traditional clinical perspective, bladder cancer is classified as either superficial (stages Ta or T1) or invasive (stages T2, T3 and T4). The invasive disease often requires multi-modal therapies combining surgery as well as chemotherapeutic approaches, while most cases with a superficial disease have long-recurrence free episodes. To the present day, there is no screening method recommended for individuals at average risk. The diagnostic is determined by microscopic examination of cells from urine or bladder tissue and examination of the bladder wall with a cystoscope. The 5-year relative survival rate is 80% for all the stages.

R. Rizzo and P.J.G. Lisboa (Eds.): CIBB 2010, LNBI 6685, pp. 253–262, 2011.
© Springer-Verlag Berlin Heidelberg 2011

Starting from bladder cancer discovered while the tumor is in situ (97% 5-year survival) it decreases down to 6% for distant stage disease (6% 5-year survival), making it very important to discover the cancer in time. [1]

As Rabiya S. Tuma pointed out [14], personalized medicine is the ultimate goal of modern cancer treatment, and its success depends on the availability of tumor biomarkers that can be used to guide treatment. Molecular biomarkers represent alterations in gene sequences, expression levels, protein structure or function which can be used to detect cancers at an early stage, determine prognosis, and monitor disease progression or therapeutic response [13].

They are invaluable tools for both cancer research and clinical practice, yet few biomarkers are in clinical use despite decades of intense effort. This is because ”It is a hard statistical problem, it is a hard clinical problem, and it is a hard biological problem.”, as Marc Buyse emphasized (cited in [14]).

In our opinion, there are some general paradigm shifts in medicine affecting biomarkers field too. One is from the search for a single molecule, functioning like an ideal biomarker, to the search for panels of biomarkers. This is a natural consequence of the (gen)omics enterprize. Another one is from a reductionistic to a systemic view, placing these molecules on functional networks and pathways. There is also a general trend to favor *non-invasive* biomarkers, usually from serum, urine, and other body fluids.

Usually, in high-throughput experiments one investigates thousands of molecules in parallel. Statistical and bioinformatics tools must be used to select and rank a subset of molecules, hundreds or preferably tens, capable to discriminate between two or more medical situations. Most studies end up with such lists of ranked molecules and p-value is the most common ranking criterion. One can also place these molecular alterations on networks and pathways, entering in the realm of systems biology. These systems have a completely or partially known structure but no dynamics. While this is a real scientific progress of the last decade, it does not help too much our understanding of the complex molecular dynamical systems, nor is very helpful in clinical practice. Important questions are:1) Can artificial/computational intelligence help our understanding of complex molecular dynamical systems? 2) Can artificial/computational intelligence help developing clinically useful tools?

Our answer to the first question is yes. We developed RODES [6], a class of algorithms based on artificial intelligence to automatically extract mathematical models, in the form of systems of differential equations (dynamical systems), from high-throughput time-series data. It is based on a combination of knowledge discovery in data and knowledge mining, making use of genetic programming (GP), when all the variables of the system are available, and neural networks control when some variables are missing.

In this paper we will focus mainly on the second question. More precisely, the question is: Can we use artificial intelligence to transform, an interesting but not very useful list of ranked genes, in an intelligent system, based on the most relevant subset(s) of these genes, capable to predict a diagnosis or any other important clinical outcome, and supporting in this way clinical decisions?

The answer to this challenging biomedical informatics question is yes too, and we will illustrate this with our investigations on non-invasive bladder cancer diagnosis (BCa).

There is no reported study where free, circulating miRNAs were measured in bladder cancer. Tumor-derived miRNAs are not the same as circulating miRNAs, as the latter may also include other tissue/cells that release these microRNAs, representing the result between tumor and the host interactions. The idea of using circulating miRNAs is not new, however, this is the first study using such a comprehensive array of miRNA probes and is the first to be used in bladder cancer as a diagnostic screening tool.

The goal of this paper was to develop intelligent molecular biomarkers, for the non-invasive diagnosis of bladder cancer, based on plasma microRNAs (or miRNAs), via a knowledge discovery in data approach, using computational intelligence methods.

To the best of our knowledge, this is the first time when plasma miRNAs are combined, using artificial intelligence, to predict in a non-invasive way, the bladder cancer diagnosis. The prediction accuracy of the systems was between 91.67% and 100%.

2 Methods

In [5] we proposed a general methodology for developing i-Biomarkers, a basic taxonomy, and some relationships with other intelligent (the prefix ”i-” comes from intelligent) clinical decision support systems (i-CDDSs), we have developed. These are illustrated here presenting the main steps for developing non-invasive bladder cancer diagnosis i-Biomarkers. i-Biomarkers are a subset of i-CDSS, a concept first introduced by us in [7]. Stated briefly, these are clinical decision systems [2] based on artificial intelligence. Generally, i-CDSSs are the result of a knowledge discovery in data approach:

1. Extracting and integration information from various biomedical data sources, after a careful preprocessing consisting mainly in:
   (a) cleaning features and patients,
   (b) various treating of missing data,
   (c) background correction
   (d) normalization
   (e) various transformations
   (f) ranking features
   (g) selecting features
   (h) balancing data, etc.
2. Testing various classifiers.
3. Testing various ensemble methods.

The dataset used to develop the non-invasive BCa i-biomarkers was acquired using customized microRNA array [9]. It was clean and without missing data. The first steps consist in exploratory data analysis and data pre-processing. Background correction aims to adjust the intensity readings for technical variability between arrays due to subtle differences in handling labeling, hybridization and scanning. It is essential to use background correction in order to obtain good sensitivity from the data. The background correction was done by subtracting "B635 Median" from "F635 Median" and the resulted values represent the gene expression data. The F635 Median values represent the median feature pixel intensity for the 635 nm channel and the B635 values represent the median feature background intensity for the same channel.

The next step was to transform the raw data into log2 scale and normalize it with the quantile normalization method. The quality of the data was assessed using density plots (not shown), boxplots and unsupervised hierarchical clustering (based on Pearson correlation and the Ward linkage rule), and principal components analysis (PCA). The replicates of each probe were averaged after normalization and the non-human microRNA probes were filtered.

After preprocessing and filtering out all non-human microRNA we tried to find the miRNAs that are differentially expressed between two sample groups. A two-sample t-test was used for this purpose. For all these pre-processing steps we used the freely available bioinformatics collection of software packages Bioconductor (http://www.bioconductor.org).

We initially performed a set of experiments choosing the first N miRNAs, with $N = 5$, 10, 15, 20, 25, etc., in the decreasing order of their p-values. The results (not shown) were not as good as we expected and we decided to continue the feature selection.

For ranking the features, we used a simple but effective method which considers one feature at a time, to see how well each feature alone predicts the target variable. For each feature, the value of its importance is calculated as (1 - p), where p is the p value of the corresponding statistical test of association between the candidate feature and the target variable. The target variable was categorical with two categories for all investigated problems and all inputs were continuous.

A number of artificial intelligence methods were tested for developing intelligent clinical decision support systems for diagnosis prediction:

1. Artificial Neural Networks (ANN) [3]
2. Support Vector Machines (SVM) [12]
3. Shrinkage Discriminant Analysis (SHRINKLDA)
4. Penalized Logistic Regression (PLR) [15]
5. K nearest neighbors (KNN)
6. Random Forest [4]
7. Partial Least Squares combined with Linear Discriminant Analysis (PLSLDA)
8. Fisher's Linear Discriminant Analysis (FDA)

For creating the training sets, the chosen method was leave one out cross validation due to the small number of cases. For each training set was performed feature selection using t-test, f-test, Wilcox test, Welch test, Random Forest Variable Importance Measure, Lasso. The classification methods presented above were applied using a number of 25 genes. The results are presented in the following section. (see Table 2)

Neural Networks were chosen in our study in order to predict the diagnosis BCa or Normal, for each patient. Due to the fact that the output is categorical we used neural networks for classification. The type of neural network with which we performed the experiments was Multilayer Perceptron (MLP), because the results obtained with other types of networks were not satisfactory and they tended to be slower than MLP. The preparatory steps for conducting the experiments consisted in determining: a) the size of the training and testing set; b) the error function; c) the number of hidden units; d) the activation functions for the hidden and the output neurons; e) the minimum and maximum values for weight decay.

For the hidden neurons were used as activation functions identity, logistic, tanh and exponential and the same ones were chosen for the output neurons. The error functions used were sum of squares and cross entropy. The minimum and the maximum number of hidden units were chosen differently for each experiment because we conducted several experiments which had different number of inputs. The larger the number of hidden units in a neural network model the stronger the model is, the more capable the network is to model complex relationships between the inputs and the target variables. Due to the small number of cases the number of hidden units was not large. The optimal number of hidden units is minimum 1/10 of the number of training cases and maximum 1/5, but we have varied this interval [10]. The use of decay weights for hidden layer and output layer was preferred in order to prevent overfitting, thereby potentially improving generalization performance of the network. Weight decay or weight elimination are often used in MLP training and aim to minimize a cost function which penalizes large weights. These techniques tend to result in networks with smaller weights. The minimum chosen weight decay was 0.0001 and the maximum 0.001.

3 Results

3.1 Patients Data

This study is based on a lot of 38 individuals, which has 20 (52%) patients with bladder cancer, 5 females and 15 males, 10 (26%) with invasive BCa, and 10 (26%) with superficial BCa, and 18 (48%) individuals without any known cancer. The BCa patients were 7 (35%) with stage Ta, 3 (15%) with stage T1, 7 (35%) with stage T2, 1 (5%) with stage T3 and 2 (10%) with stage T4, 2 (5%) with grade 1, 5 (25%) with grade 2, and 14 (70%) with grade 3. The molecular biology data consists in 38 samples of microRNAs isolated from plasma of these individuals. There where measured 19200 miRNAs from blood plasma.

3.2 Preprocessing Results

The results of the preprocessing steps performed to find the differentially expressed miRNAs are summarized in the following figures and tables:

1) Figure 1 shows the distributions of p-values from the feature-by-feature two sample T-test, 2) Table 1 summarizes the numbers of significant features using different false discovery rates (FDR) cutoff values from the feature-by-feature two sample T-test, and 3) Figure 2 shows a heatmap of the most differentially expressed features selected at an FDR level of 0.20.

Because of the multiple testing involved in this approach (feature-by-feature), the individual p-values are not particularly meaningful. However, when we look across the entire set of tests, the distribution of p-values (under the null hypothesis that no miRNAs were differentially expressed) should be uniform (indicated by the black line in Figure 1). If, on the other hand, some features are differentially expressed, we would expect an overabundance of small p-values. We can capture this situation by modeling the distribution of the p-values as a beta-uniform mixture (BUM) described by Pounds and Morris in [11].

Table 1. Summary of significant miRNAs by different FDR cutoff values

FDR	#sign	P-value cutoff
0.20	18	1.74e-03
0.25	23	2.60e-03
0.30	30	3.70e-03
0.35	31	5.09e-03
0.40	34	6.85e-03
0.45	45	9.11e-03
0.50	55	1.20e-02

Using Bioconductor we obtained a list of features that are differentially expressed among samples by performing feature-by-feature two-sample t-tests. The list contained 2247 genes ordered by their p-value and we retained the first 252. We have applied feature selection on the remaining genes in order to obtain the most significant genes that influence the output, normal or BCa. We have chosen for further testing the variables with the p value, based on Pearsons Chi-square, larger then 0.97 (a value we chose arbitrarily to keep a good proportion of miRNAs vs patients: 63 miRNAs measured for 38 patients).

The methods presented in the previous section were applied after the preprocessing step and the differential analysis. The best results were provided by the Support Vector Machines (SVM) which misclassified only one case and the accuracy was 97.4%. This model was applied on the training sets after using leaving one out method and performing gene selection using Wilcox test. Other settings that were used are the linear kernel due to that fact that we have a small sample (38 cases) and a relative large number of predictors (25 genes) and the cost parameter to allow some flexibility in separating the categories (BCa and normal), having values between 0.1 and 100.

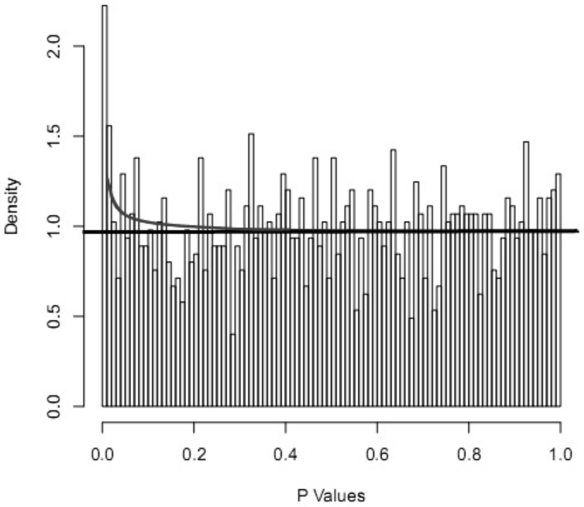


Fig. 1. Histogram showing the distribution of p-values from the feature-by-feature two sample t-test

The next experiments involved the usage of neural networks. Two approaches were used. We began the experiments with a small number of genes, 5 in the beginning, and this number was increased with each experiment. The other approach was to do the experiments with all the selected genes and decrease their number with each performed experiment. We obtained neural networks with 100% accuracy. To perform the experiments the data were split in a training (67 %) and a testing dataset (33 %). As error functions, we tested the cross entropy and the sum of squares. The weight decays in both the hidden and the output layer varied between 0.0001 and 0.001. The list of genes we used in the neural networks experiments are presented in Table 3. The output of the experiments had two categorical values BCa and Normal. We trained a large number of neural networks and we retained the best 20 ones. Training more neural networks allowed us to explore various combinations of options.

Due to the fact that a different number of inputs was used at each experiment, the amount of hidden units varied. The minimum and the maximum number of hidden units with which we obtained the best results can be seen in Table 4.

For the first experiment we used a set of 5 genes. We obtained 100% accuracy for training and 91.67% accuracy for testing, which means than one case was incorrectly classified. The case was BCa and it was classified as Normal. We removed the value of this output and we used the K-Nearest Neighbors (KNN) to see which value it will take. The previous value was BCa and after applying this method was Normal. The experiment was done once again without this case and the results had 100% accuracy.

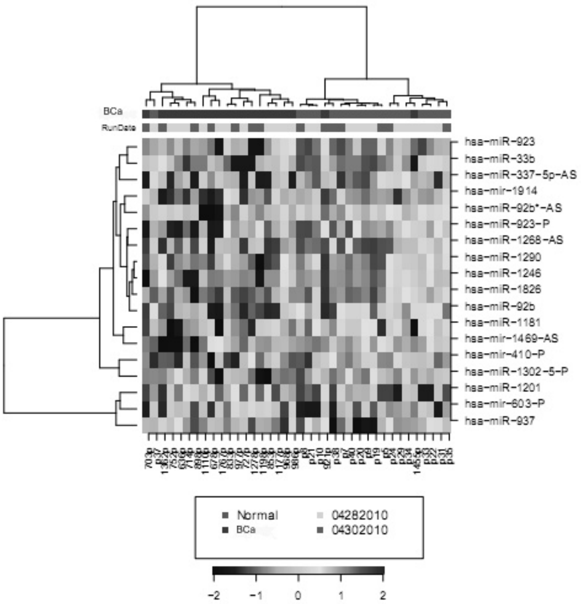


Fig. 2. Heatmap of the most differentially expressed miRNAs selected at an FDR level of 0.2

Table 2. Best six algorithms compared

Model	Misclassification rate	No. misclassifications
NN	0	0
SVM	0.026	1
SHRINKLDA	0.079	3
KNN	0.105	4
PLR	0.105	4
PLSLDA	0.105	4
RF	0.105	4

The second experiment had as inputs the first 10 genes resulted from the feature selection process. The results were the same as in the experiment with the first 5 genes and the same case was incorrectly classified. We found one neural network that classified correctly that case. We eliminated again this case and the results had 100% accuracy.

The number of genes was increased to 15 and we have proceeded like in the previously presented experiments. There were neural networks with 100% accuracy for the training sample and with 91,67% for the testing sample, but we confronted with some cases of overfitting for the training sample, even after we removed the case which was always incorrectly classified.

Table 3. The list of genes used in neural networks experiments

Genes in neural networks experiments		
hsa-miR-923-P	hsa-miR-33b	hsa-miR-1826
hsa-miR-92b	hsa-miR-1246	hsa-miR-337-5p-AS
hsa-miR-1268-AS	hsa-miR-1290	hsa-miR-92b*-AS
hsa-mir-1914	hsa-miR-1181	hsa-mir-603-P
hsa-miR-937	hsa-miR-1302-5-P	hsa-mir-410-P
hsa-mir-1469-AS	hsa-miR-923	hsa-miR-1201
hsa-miR-16-AS	hsa-miR-487a	hsa-miR-219-1-3p
hsa-miR-1290-AS	hsa-miR-96*	hsa-miR-638
	hsa-miR-629*	

Table 4. Number of hidden units

5 genes	10 genes	15 genes	20 genes	25 genes
3 - 11	4 -13	5 - 16	6 - 18	6 - 20

The last two experiments with 20 and 25 genes as inputs had the best results. All the retained neural networks had 100% accuracy and even the case incorrectly classified in the previous experiments was classified as being BCa and not Normal. Increasing the number of hidden units lead to a correct classification of all the cases.

The Neural Network based clinical decision support system has 100% accuracy. As it was previously stated, they are molecular i-Biomarkers which can be used as i-CDSS for noninvasive bladder cancer diagnosis. This is an important step toward bladder biopsy replacement in bladder cancer suspicion or at least a mean for reducing the number of necessary biopsy.

4 Conclusions

We proposed intelligent clinical decision support systems for a noninvasive diagnosis in bladder cancer. The systems are based on artificial intelligence methods and a set of newly discovered biomarkers that can distinguish between the two medical outcomes. This is an important step toward bladder biopsy replacement when diagnosing bladder cancer.

References

1. ACS: American Cancer Society. Cancer Facts and Figures 2010. American Cancer Society, Atlanta (2010), `http://www.cancer.org/acs/groups/content/@nho/documents/document/acspc-024113.pdf`
2. Berner, E.S.: Clinical Decision Support Systems: Theory and Practice. Springer, New York (1998)

3. Bishop, C.M.: Neural Networks for Pattern Recognition. Oxford University Press, Inc., New York (1995)
4. Breiman, L.: Random forests. Machine Learning 45(1), 5–32 (2001)
5. Floares, A., Balacescu, O., Floares, C., Balacescu, L., Popa, T., Vermesan, O.: Mining knowledge and data to discover intelligent molecular biomarkers: prostate cancer i-biomarkers. In: Proceedings of the 4th International Workshop on Soft Computing Applications, July 15–17 (2010)
6. Floares, A.G.: Toward Personalized Therapy Using Artificial Intelligence Tools to Understand and Control Drug Gene Networks. In: New Trends in Technologies. INTECH (2010), `http://sciyo.com/articles/show/title/toward-personalized-therapy-using-artificial-intelligence-tools-to-understand-and-control-drug-gene-`
7. Floares, A.G.: Using computational intelligence to develop intelligent clinical decision support systems. In: Masulli, F., Peterson, L.E., Tagliaferri, R. (eds.) CIBB 2009. LNCS, vol. 6160, pp. 266–275. Springer, Heidelberg (2010)
8. Jemal, A., Siegel, R., Ward, E., Hao, Y., Xu, J., Thun, M.: Cancer statistics. CA Cancer J. Clin. 4(59), 225–249 (2009)
9. Liu, C.G., Calin, G.A., Volinia, S., Croce, C.M.: Microrna expression profiling using microarrays. Nature Protocols 3, 563–578 (2008)
10. Nisbet, R., Elder, J., Miner, G.: Handbook of Statistical Analysis and Data Mining Applications. Academic Press, London (2009)
11. Pounds, S., Morris, S.W.: Estimating the occurrence of false positives and false negatives in microarray studies by approximating and partitioning the empirical distribution of p-values. Bioinformatics 19(10), 1236–1242 (2003), `http://bioinformatics.oxfordjournals.org/cgi/content/abstract/19/10/1236`
12. Schlkopf, B., Smola, A.J.: Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond (Adaptive Computation and Machine Learning), 1st edn. The MIT Press, Cambridge (2001)
13. Sidransky, D.: Emerging molecular markers of cancer. Nat. Rev. Cancer 2(3), 210–219 (2002)
14. Tuma, R.S.: Biomarker developers face big hurdles. J. Natl. Cancer Inst. 100(7), 456–461 (2008), `http://jnci.oxfordjournals.org`
15. Zhu, J., Hastie, T.: Classification of gene microarrays by penalized logistic regression. Biostat. 5(3), 427–443 (2004)

Automatic Unsupervised Segmentation of Retinal Vessels Using Self-Organizing Maps and K-Means Clustering

Carmen Alina Lupaşcu and Domenico Tegolo

Dipartimento di Matematica e Informatica
Università degli Studi di Palermo
Palermo, Italy
lupascu@math.unipa.it, domenico.tegolo@unipa.it

Abstract. In this paper an automatic unsupervised method for the segmentation of retinal vessels is proposed. A Self-Organizing Map is trained on a portion of the same image that is tested and K-means clustering algorithm is used to divide the map units in 2 classes. The entire image is again input for the Self-Organizing Map, and the class of each pixel will be the class of the best matching unit on the Self-Organizing Map. Finally, the vessel network is post-processed using a hill climbing strategy on the connected components of the segmented image.

The experimental evaluation on the publicly available DRIVE database shows accurate extraction of vessels network and a good agreement between our segmentation and the ground truth. The mean accuracy, 0.9459 with a standard deviation of 0.0094, is outperforming the manual segmentation rates obtained by other widely used unsupervised methods. A good kappa value of 0.6562 is inline with state-of-the-art supervised and unsupervised approaches.

Keywords: Retinal Vessels, Self-Organizing Map, K-means.

1 Introduction

Automatic analysis of retinal vasculature is important in the diagnosis of many eye pathologies. Once the vessel tree is extracted from retinal images, it is useful not only for diagnosis purposes, but also in the registration of retinal images. Branching and crossover points in the vasculature structure are used as landmarks for image registration. Image registration is needed to integrate information from several images, but also to observe the progression of diseases over time. Finally, automatically generated vessel maps have been used to guide the identification of retinal landmarks like the optic disc and the fovea.

1.1 Related Work

Many different approaches for automated vessel segmentation have been proposed. Some of them are rule-based methods (those based on vessel tracking ([4]), those based on matched filter responses ([3], [6]) and other ones are based on mathematical morphology ([14], [22])). The methods listed above are unsupervised.

R. Rizzo and P.J.G. Lisboa (Eds.): CIBB 2010, LNBI 6685, pp. 263–274, 2011.
© Springer-Verlag Berlin Heidelberg 2011

In addition to the rule-based methods, supervised methods have also been used for vessel segmentation. Sinthanayothin et al. in [16] classify pixels using a multilayer perceptron neural net, for which the inputs were derived from a principal component analysis (PCA) of the image and edge detection of the first component of PCA. In [15] a simple feature vector is extracted for each pixel from the green plane and then a K-nearest neighbor (kNN) is used to distinguish vessel and non-vessel pixels. Another supervised method, called primitive-based method, was proposed in [19]. This algorithm is based on the extraction of image ridges (expected to coincide with vessel centerlines) used as primitives for describing linear segments, named line elements. Consequently, each pixel is assigned to the nearest line element to form image patches and then classified using a set of features from the corresponding line and image patch. The feature vectors are classified using a kNN-classifier. The method presented by Soares et al. in [17] produces also segmentation using a supervised classification. Each image pixel is classified as vessel or non vessel based on the pixel feature vector, which is composed of the pixel intensity and two-dimensional Gabor wavelet transform responses taken at multiple scales. A Gaussian mixture model classifier is then applied to obtain a final segmentation.

Pixel classification based on supervised methods requires hand-labeled ground truth images for training. Supervised learning assumes that the training samples are classified by an expert as either vessel or non-vessel. The operation of classification of pixel as either vessel or non-vessel is time consuming, as the supervised training process itself. The lack of experts and the time consuming processes involved in the automatic supervised methods described in the literature determined us to search for an automatic unsupervised method for classification of the pixels from a retinal image as vessel or non-vessel. We found that self-organizing maps combined with K-means for clustering map units give good results in clustering pixels (especially when the number of classes is small, like in our case, the number of classes equals to 2) and is also very fast. This method is attractive also because training is performed on a portion of the same image we want to segment, hence there is no need to develop a separate training set like in other supervised or unsupervised methods.

1.2 Dataset

The database we use is one public database: the DRIVE database (Digital Retinal Images for Vessel Extraction). The photographs for the DRIVE database were obtained from a diabetic retinopathy screening program in The Netherlands. Each image has been JPG compressed. The images were acquired using a Canon CR5 non-mydriatic 3CCD camera with a 45 degree field of view (FOV). Each image was captured using 8 bits per color plane at 768 by 584 pixels. The FOV of each image is circular with a diameter of approximately 540 pixels. For this database, the images have been cropped around the FOV. For each image, a mask image is provided that delineates the FOV. The data set includes 40 584x565 fundus images. We use only the 20 images from the test set for testing our methodology. All images are available for download at `http://www.isi.uu.nl/Research/Databases/DRIVE/download.php` (the web site of Image Sciences Institute).

2 Methodology

Recently, we have developed a new supervised method for retinal vessel segmentation called FABC. The method was based on computing feature vectors for every pixel in the image and training an AdaBoost classifier with manually labeled images. In ([12]), the feature vector is a collection of measurements at different scales taken from the output of filters (the Gaussian and its derivatives up to the 2 order, matched filters and two-dimensional Gabor wavelet transform responses), from the identification of edge and ridge pixels and from other local information which are extracted after computing the Hessian of the image for each pixel. The basic idea is to encode in the feature vector as well local information (pixel's intensity, Hessian-based measures), spatial properties (the gray-level profile of the cross-section of a vessel can be approximated by a Gaussian curve) and structural information (vessels are geometrical structures which can be seen as tubular). We used an AdaBoost classifier to divide pixels into two classes, i.e., vessels and non vessel pixels.

2.1 Pixel Features

Features are extracted from the green plane of the retinal images, because in the green plane the contrast between vessel and background is higher than in the blue or red plane. The feature vector consisted of the output of filters (features 1 and 2 in the list below) plus vesselness and ridgeness measures based on eigen-decomposition of the Hessian computed at each image pixel (features 3, 4 and 5) and a two-dimensional Gabor wavelet transform response taken at multiple scales (feature 6 below). Moreover the feature vector was augmented with the principal curvatures, the mean curvature, the values of principal directions, but also the value of the gradient and the intensity value within the green plane of each pixel (features 7 and 8). We give below the list of components of the feature vector (the computation method of each component is described better in [12]).

1. The Gaussian and its derivatives up to order 2.
2. The green channel intensity of each pixel.
3. A multiscale matched filter for vessels using a Gaussian vessel profile. ([18])
4. Frangi's vesselness measure. ([5])
5. Lindeberg's ridge strengths. ([10])
6. Staal's ridges. ([19])
7. Two-dimensional Gabor wavelet transform response taken at multiple scales. ([17])
8. Values of the principal curvatures, of the mean curvature, of the principal directions and of the gradient of the image.

The scales used in order that vessels with various dimensions could be detected were $4{:}\sqrt{2}$, 2, $2 * \sqrt{2}$ and 4, hence the total number of features is 41.

In order to establish which features play the most important role in the vessel/non-vessel classification task, we performed a comparative study on feature selection methods applied as a preprocessing step to the AdaBoost classification. In [11] we presented five feature selection heuristics designed to evaluate the usefulness of features through feature subsets. Experiments showed that the features that seemed to play the most important discriminatory role, i.e., the ones that were selected by *all* the heuristics, were the 2nd-order derivative of the Gaussian in the y direction at scale $2\sqrt{2}$, the maximum response of a multiscale matched filter using a Gaussian vessel profile, and the feature containing information about Staal's ridges. We use only these 3 features for the feature vectors used for clustering with self-organizing maps and K-means.

2.2 Self-Organizing Maps and K-Means Clustering

A Self-Organizing Map (SOM) is a neural network that is trained, using unsupervised learning, to build a map of the input space of the training samples. A new input vector will be automatically classified using the map built in the training phase. SOM was developed by Teuvo Kohonen in 1980 [8]. It consists of m neurons organized on a regular low-dimensional grid. Each neuron i is a d-dimensional weight vector $(w_{1i}, ..., w_{di})$ called *prototype vector or codebook vector*, where d is equal to the dimension of the input vectors. Usually, before the training phase, the prototype vectors are linearly initialized. It has been suggested by Kohonen et al. [9] to use rectangular (but non quadratic) maps and the number of neurons of the map is computed as 5 times the square root of the number of training samples. The SOM is trained on a part of the image (we choose the training sample as half of the FOV pixels, selected randomly), hence we have around 106800 training pixels. The number of map units is about $5\sqrt{106800} = 1634$ neurons. After the number of map units has been determined, the map size is determined by setting the ratio between column number and row number of map units equal to the ratio of two biggest eigenvalues of the training data. The product of the column and row numbers must be as close to the number of map units as possible. ([20]) Following these rules, a 86×19 map has been used for training.

The SOM training algorithm is based on competitive learning which is a particular case of neural network unsupervised learning. At each training iteration, a sample vector x is randomly chosen from the training set. Euclidean distances between x and all the prototype vectors are computed, in order to find the best matching neuron unit (BMU). The BMU is selected as the unit that is the nearest to the input vector at an iteration t, using

$$\|x(t) - w_c(t)\| = \min_i \|x(t) - w_i(t)\|, \tag{1}$$

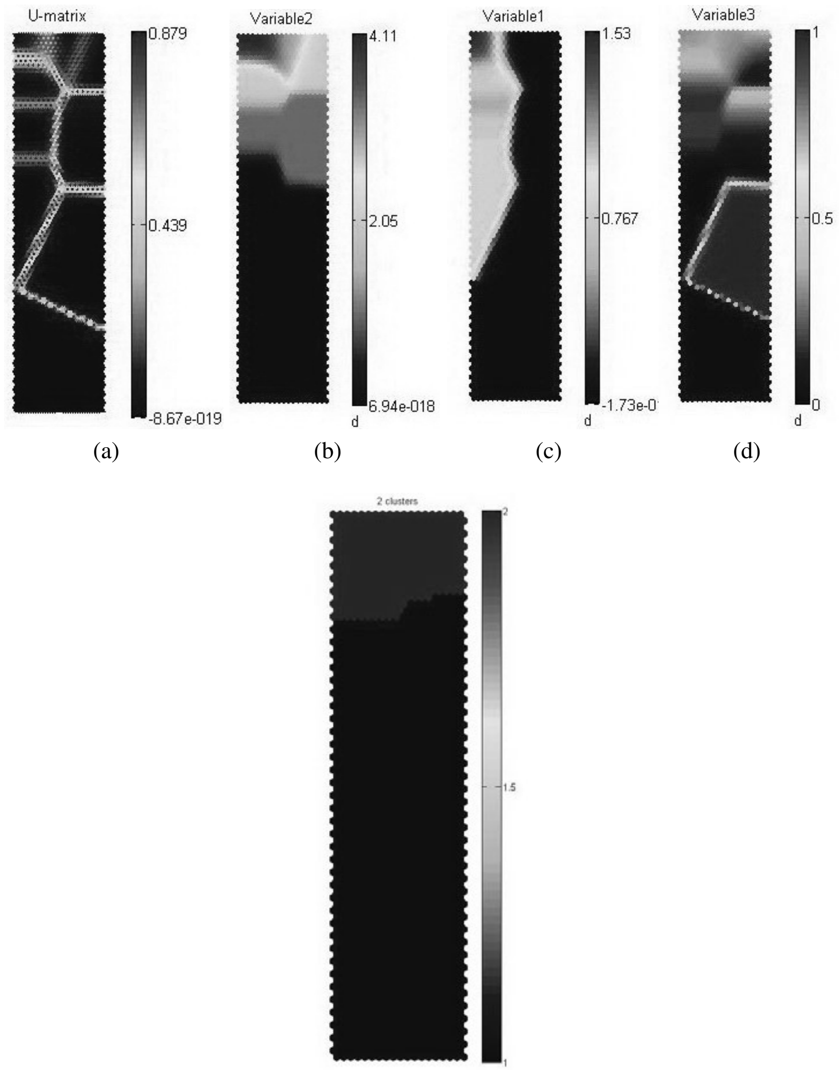
where w_c is the weight of the winner neuron. After finding the BMU, the prototype vectors of the BMU and its neighbors are moved closer to the input vector using the following update rule for a neuron i:

$$w_i(t+1) = w_i(t) + \alpha(t)[x(t) - w_i(t)], \text{ for } i \in N_c \tag{2}$$

$$w_i(t+1) = w_i(t), \text{ for } i \notin N_c \tag{3}$$

where $\alpha(t)$ is the learning rate and N_c is the neighborhood of the winning neuron.

After training the map, clusters may be visualized using the U-matrix (Unified distance matrix), which represents the distance of the neurons to their adjacent. To inspect



(a) (b) (c) (d)

(e)

Fig. 1. Clustering of map units using K-means with K=2. a),b),c),d) U-matrix and component planes. e) Clustered map.

the clustering structure of the map, components plane may be used also. Each component plane shows the values of one variable in each map unit. On the U-matrix lighter regions indicate cluster boundary and darker regions indicate cluster itself. Separating clusters by visually inspecting the U-matrix is very difficult as we may see in Figure 1. For this reason K-means clustering algorithm is usually used.

K-means clustering tries to find a partition that minimizes the sums of squared errors about the cluster means as described by the equation:

$$\sum_{k=1}^{n} \sum_{x \in q_i} \|x - c_i\|^2, \tag{4}$$

where n is the number of clusters, c_i is the centroid of the i-th cluster q_i. It was proved that SOM and K-means algorithms have the same results when the radius of the neighborhood function in the SOM equals zero [2].

After the neurons from the SOM were classified, the class of a new input pixel will be the class of its BMU.

We postprocess the segmented images, trying to eliminate small connected components in order to remove noisy pixels and to improve in this way the segmentation accuracy and the agreement between our segmentation and the ground truth. A hill climbing strategy was used in order to determine the connected components to be removed from the segmentation.

If $CC = \{c_1, ..., c_n\}$ is the set of n connected components ordered in ascending order by the number of pixels in the connected component, the algorithm starts with the set $Toberemoved = \{c_1, c_2\}$. The mean of the cardinalities of the connected components included in the set $Toberemoved$ is computed, as well as their standard deviation. Connected component c_3 is added to the set $Toberemoved$ if

$$|c_3| < mean(Toberemoved) + 3std(Toberemoved).$$

The algorithm stops when a successive connected component from the set CC can not be added to the set of connected components to be removed.

3 Experimental Results

Performance is given mainly as accuracy and kappa value.

The sensitivity (SE) is computed also, by dividing the number of pixels correctly classified as vessel pixels (TP) by the total number of vessel pixels in the gold standard segmentation,

$$sensitivity = \frac{TP}{P} = \frac{TP}{TP + FN},$$

where FN is the number of pixels incorrectly classified as non-vessel pixels. The specificity (SP) is computed as the number of pixels correctly classified as non-vessel pixels (TN) divided by the total number of non-vessel pixels in the gold standard:

$$specificity = \frac{TN}{N} = \frac{TN}{FP + TN}.$$

Here, FP is the number of pixels incorrectly classified as vessel pixels.

An important quality parameter is the accuracy. The accuracy (ACC) for one image is the fraction of pixels correctly classified

$$ACC = \frac{TP + TN}{P + N} = \frac{TP + TN}{TP + FN + FP + TN}.$$

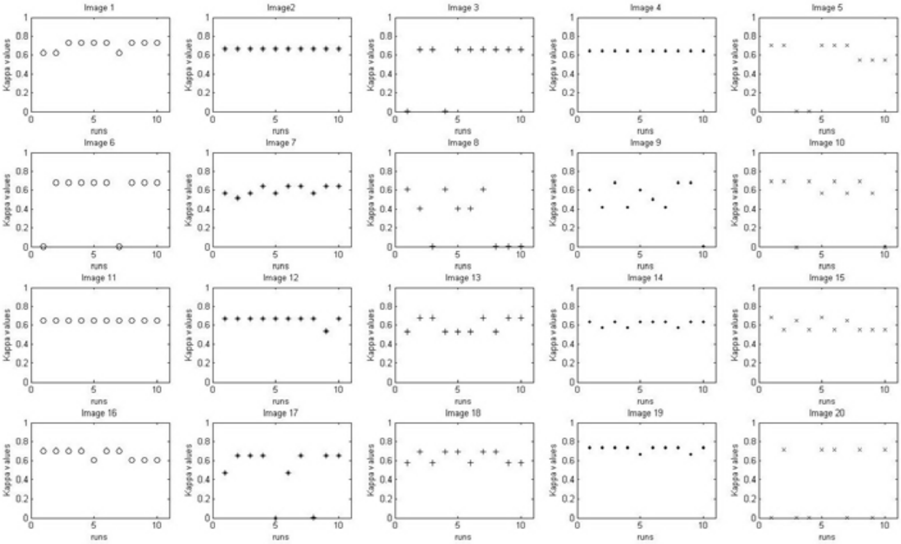


Fig. 2. Evolution of kappa values during ten runs for each of the 20 test images

Table 1. Results for the 20 test images from the DRIVE database. SE indicates sensitivity, SP indicates specificity, ACC indicates the accuracy and $Kappa$ indicates the kappa value.

Image	SE	SP	ACC	Kappa
1	82.71	96.47	0.9525	0.7291
2	54.77	99.60	0.9505	0.6671
3	71.08	96.16	0.9367	0.6551
4	51.42	99.66	0.9526	0.6410
5	70.43	97.63	0.9509	0.7012
6	66.77	97.72	0.9472	0.6813
7	74.80	95.38	0.9351	0.6411
8	27.53	99.81	0.9366	0.4015
9	55.00	98.32	0.9482	0.6046
10	74.64	97.03	0.9519	0.6915
11	53.80	99.49	0.9541	0.6537
12	76.08	96.20	0.9447	0.6731
13	71.46	96.75	0.9429	0.6771
14	80.56	94.75	0.9361	0.6347
15	84.30	92.48	0.9190	0.5563
16	77.74	96.61	0.9491	0.7053
17	71.34	96.59	0.9446	0.6541
18	80.54	96.41	0.9516	0.6980
19	87.85	96.17	0.9548	0.7381
20	79.65	97.21	0.9592	0.7194

Table 2. Overview of the performance of different methods. $Kappa$ indicates the kappa value and ACC indicates the accuracy.

Segmentation method	Drive set	
	$Kappa$	ACC
FABC (training and test confined to the dedicated sets from the database) [12]	0.7200	0.9597 (0.0054)
FABC (leave-one-out tests) [12]	-	0.9575
Human observer	0.7589	0.9473 (0.0048)
Soares et al. [17]	-	0.9466
SOM and K-means	0.6562	0.9459(0.0094)
Staal et al. [19]	0.7345	0.9442 (0.0065)
Niemeijer et al. [15]	0.7145	0.9416 (0.0065)
Zana et al. [22]	0.6971	0.9377 (0.0077)
Al-Diri et al. [1]	0.6716	0.9258 (0.0126)
Jiang et al. [7]	0.6399	0.9212 (0.0076)
Martinez et al. [13]	0.6389	0.9181 (0.0240)
Chaudhuri et al. [3]	0.3357	0.8773 (0.0232)
All background	0	0.8727 (0.0123)

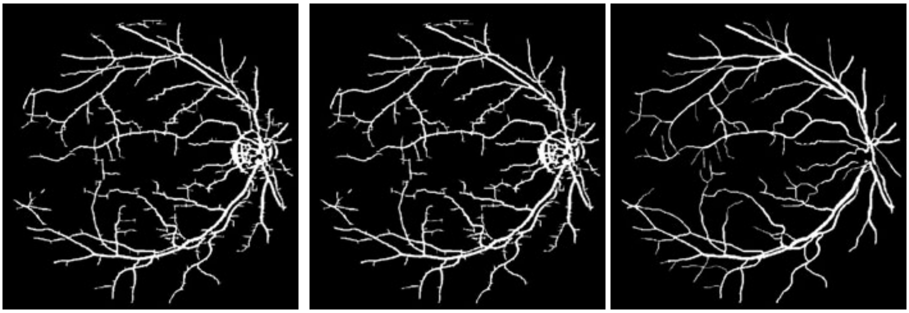
We compute also the kappa value (a measure for observer agreement, where the two observers are the gold standard and the segmentation method)

$$kappa = \frac{P(A) - P(E)}{1 - P(E)},$$

where $P(A) = \frac{TP+TN}{P+N}$ is the proportion of times the 2 observers agree, while $P(E) = \frac{TP+FP}{P+N} * \frac{TP+FN}{P+N} + (1 - \frac{TP+FP}{P+N})(1 - \frac{TP+FN}{P+N})$ is the proportion of times the 2 observers are expected to agree by chance alone.

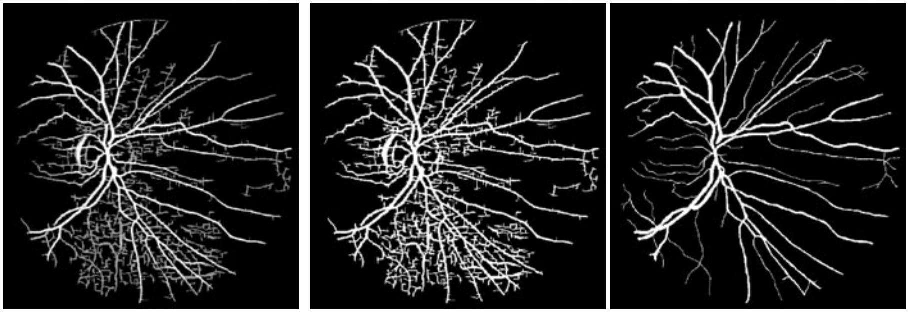
As the training samples are selected randomly, we performed ten runs for each of the 20 test images from the database (Figure 2). We noticed that kappa values are variable, because of the randomness of the training samples. Hence, we produced a soft classification by summing the ten images resulted in the ten runs. After that a hard classification is obtained by thresholding (at half of the maximum gray level).

In Table 1 we see the results for the 20 test images from the DRIVE database.



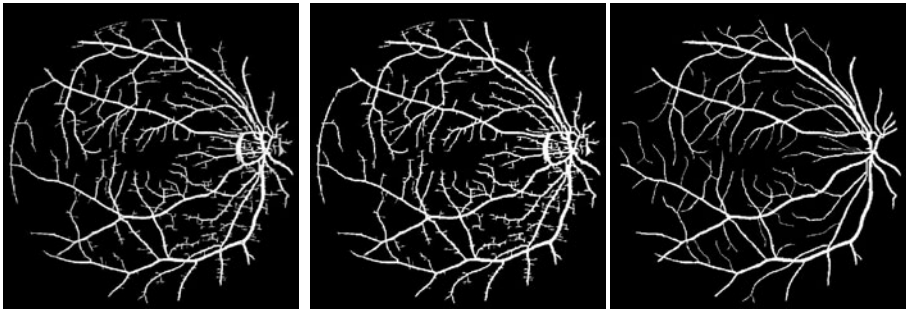
(a) soft classification (b) our segmentation (c) ground truth

Fig. 3. Best segmentation in terms of ACC (Image $20\_test.tif$ from DRIVE database)



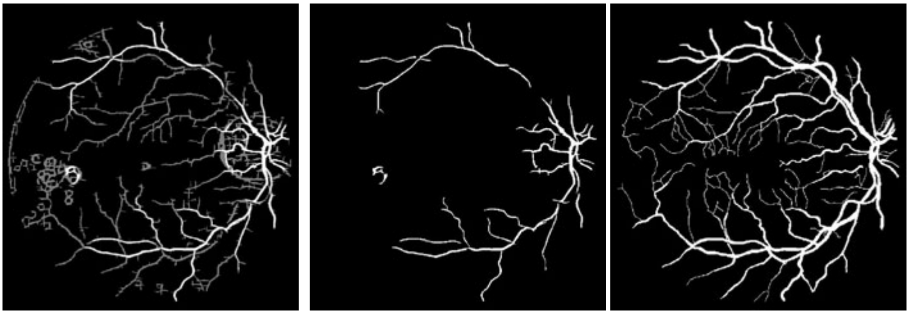
(a) soft classification (b) our segmentation (c) ground truth

Fig. 4. Worst segmentation in terms of ACC (Image $15\_test.tif$ from DRIVE database)



(a) soft classification (b) our segmentation (c) ground truth

Fig. 5. Best segmentation in terms of Kappa value (Image $19\_test.tif$ from DRIVE database)



(a) soft classification (b) our segmentation (c) ground truth

Fig. 6. Worst segmentation in terms of Kappa value (Image $08\_test.tif$ from DRIVE database)

4 Conclusions

We have presented an automatic unsupervised method for retinal vessel segmentation based on Self-Organizing Maps and K-means clustering. The choice to use Self-Organizing Maps instead of a simple K-means is based upon the fact that a SOM provides a direct means to visualize relations among different clusters (represented by the prototype vectors in the input space and by the map's neurons in the output space). Moreover, a prototype vector of the SOM is adjusted according to not only the data points that are associated with it, but also to data points that are assigned to other prototype vectors, hence the results of clustering the prototype vectors are more accurate than clustering directly the data points.

Overall the 20 test images, the mean accuracy is 0.9459 with a standard deviation of 0.0094. The mean Kappa value is 0.6562. As we may see also from [21] and Table 2, the mean ACC of our proposed method outperforms the mean ACC of any of unsupervised methods used for comparison. In Figures 3 and 4 we show the best and the worst segmentation in terms of accuracy produced by our method, together with their respective ground truths. In the same time, in Figure 5 and 6 we may see the best and the worst result in terms of agreement between our automatic segmentation and the observer's manual segmentation. The worst result is obtained on a pathological image and as in [21], the proposed method enhances all region-of-interest, i.e. both vessel network and pathological findings in the soft classification. This effect is desired in computer-aided diagnosis tools.

The method we used have the advantage that it uses knowledge about the vessel network morphology like the most accurate supervised methods, but is completely unsupervised as we do not have any a priori knowledge about the labels of the pixels we want to classify as vessel or non-vessel. Another advantage of the proposed method is its fast computational time, compared to supervised methods which are computationally more expensive.

Although the vessel network produced before the post-processing in most cases was acceptable, post-processing methods removed efficiently false positives and improved the accuracy.

In the future we aim to extend this work and to conduct a qualitative analysis of the improvements that the post-processing step brought to the initial vessel map. In the same time, using sampling techniques, we would like to analyze how does the size of the sample used for training the SOM influences the results. We would like to study also the influence on the results of the choice of some parameters of the SOM map (like the number of iterations, the size of the initial radius of the neighborhood and the choice of distance measure). Finally, we would like to compare the performances of the SOM combined with K-means with the performances of simple K-means and fuzzy C-means with the goal to prove that SOM combined with K-means is a better choice.

References

1. Al-Diri, B., Hunter, A., Steel, D.: An active contour model for segmenting and measuring retinal vessels. IEEE Transactions on Medical Imaging 28(9), 1488–1497 (2009)
2. Bodt, E.d., Verleysen, M., Cottrell, M.: Kohonen maps versus vector quantization for data analysis. In ESANN 1997, Bruges (1997)
3. Chaudhuri, S., Chatterjee, S., Katz, N., Nelson, M., Goldbaum, M.: Detection of blood vessels in retinal images using two-dimensional matched filters. IEEE Transactions on Medical Imaging 8(3), 263–269 (1989)
4. Chutatape, O., Zheng, L., Krishnan, S.M.: Retinal blood vessel detection and tracking by matched gaussian and kalman filters. In: Proceeding of IEEE Int. Conf. Emg. and Bio. Society, vol. 20(6), pp. 3144–3149 (1998)
5. Frangi, A.F., Niessen, W.J., Vincken, K.L., Viergever, M.A.: Multiscale vessel enhancement filtering. In: Wells, W.M., Colchester, A.C.F., Delp, S.L. (eds.) MICCAI 1998. LNCS, vol. 1496, pp. 130–137. Springer, Heidelberg (1998)
6. Hoover, A., Kouznetsova, V., Goldbaum, M.: Locating blood vessels in retinal images by piece-wise threshold probing of a matched filter response. IEEE Transactions on Medical Imaging 19(3), 203–210 (2000)
7. Jiang, X., Mojon, D.: Adaptive local thresholding by verification-based multithreshold probing with application to vessel detection in retinal images. IEEE Transactions on Pattern Analysis and Machine Intelligence 25(1), 131–137 (2003)
8. Kohonen, T.: Self-organization and associative memory. Springer, Berlin (1989)
9. Kohonen, T., Hynninen, J., Kangas, J., Laaksonen, J.: Som pak: The self-organizing map program package. Report A31, Helsinki University of Technology, Laboratory of Computer and Information Science (January 1996)
10. Lindeberg, T.: Edge detection and ridge detection with automatic scale selection. Int. J. Comp. Vis. 30, 117–156 (1998)
11. Lupaşcu, C.A., Tegolo, D., Trucco, E.: A comparative study on feature selection for retinal vessel segmentation using FABC. In: Jiang, X., Petkov, N. (eds.) CAIP 2009. LNCS, vol. 5702, pp. 655–662. Springer, Heidelberg (2009)
12. Lupascu, C.A., Trucco, E., Tegolo, D.: Fabc: Retinal vessel segmentation using adaboost. IEEE Transactions on Information Technology in Biomedicine 14(5), 1267–1274 (2010)
13. Martínez-Pérez, M.E., Hughes, A.D., Stanton, A.V., Thom, S.A., Bharath, A.A., Parker, K.H.: Retinal blood vessel segmentation by means of scale-space analysis and region growing. In: Taylor, C., Colchester, A. (eds.) MICCAI 1999. LNCS, vol. 1679, pp. 90–97. Springer, Heidelberg (1999)
14. Mendonça, A.M., Campilho, A.: Segmentation of retinal blood vessels by combining the detection of centerlines and morphological reconstruction. IEEE Transactions on Medical Imaging 25(9), 1200–1213 (2006)
15. Niemeijer, M., Staal, J., van Ginneken, B., Loog, M., Abràmoff, M.D.: Comparative study of retinal vessel segmentation methods on a new publicly available database. SPIE Medical Imaging 5370, 648–656 (2004)
16. Sinthanayothin, C., Boyce, F.J., Cook, L.H., Williamson, H.T.: Automated localisation of the optic disc, fovea, and retinal blood vessels from digital colour fundus images. Br. J. Ophthalmol. 83, 902–910 (1999)
17. Soares, V.B.J., Leandro, J.G.J., Cesar, R.M.J., Jelinek, F.H., Cree, M.J.: Retinal vessel segmentation using the 2-d gabor wavelet and supervised classification. IEEE Transactions on Medical Imaging 25(9), 1214–1222 (2006)

18. Sofka, M., Stewart, C.V.: Retinal vessel centerline extraction using multiscale matched filters, confidence and edge measures. IEEE Transactions on Medical Imaging 25(12), 1531–1546 (2006)
19. Staal, J., Abràmoff, M.D., Niemeijer, M., Viergever, M.A., van Ginneken, B.: Ridge-based vessel segmentation in color images of the retina. IEEE Transactions on Medical Imaging 23(4), 501–509 (2004)
20. Vesanto, J.: Som-based data visualization methods. Intelligent Data Analysis Journal (1999)
21. Vlachos, M., Dermatas, E.: Multi-scale retinal vessel segmentation using line tracking. Computerized Medical Imaging and Graphics 34, 213–227 (2010)
22. Zana, F., Klein, J.C.: Segmentation of vessel-like patterns using mathematical morphology and curvature evaluation. IEEE Transactions on Image Processing 10(7), 1010–1019 (2001)

Classification of Clinical Gene-Sample-Time Microarray Expression Data via Tensor Decomposition Methods

Yifeng Li* and Alioune Ngom

School of Computer Science, University of Windsor,
Windsor, Ontario, CanadaN9B 3P4
{li11112c,angom}@uwindsor.ca
http://cs.uwindsor.ca/uwinbio

Abstract. With the recent advances in microarray technology, the expression levels of genes with respect to samples can be monitored over a series of time points. Such three-dimensional microarray data, termed gene-sample-time (GST) microarray data, are gene expression matrices measured as a time-series. They have not yet received considerable attention, and analysis methods need to be devised specifically to tackle the complexity of GST datasets. We propose methods that are based on tensor decomposition for the sample classification. We use tensor decomposition in order to extract discriminative features as well as multilinearly reducing high dimensionality. We then classify the test samples in the reduced spaces. We have tested and compared our approaches on a real GST dataset. We show that our methods are at least comparable in prediction accuracy to recent methods devised for GST data. Most importantly, our methods run much faster than current approaches.

Keywords: Gene-Sample-Time Data, Tensor Decomposition, HOSVD, HOOI, HONMF.

1 Introduction

DNA microarray technology can monitor thousands of genes in parallel, dramatically accelerating molecular biology experiments and providing a huge amount of data to find co-regulated genes, functions of genes, genetic networks, and marker genes, for instance. There are two types of microarray data: gene-sample data sets, which compile the expression levels of various genes over a set of biological samples; and gene-time data sets, which record the expression levels of various genes over a series of time-points. Both types of data are represented by a two-dimensional (2D) gene expression matrix, where genes correspond to rows in the matrix and each matrix entry contains the expression level of a given gene for some sample or at certain time-point. The gene-sample data are static data, while the gene-time data are dynamic data. The gene-sample data are typically analyzed in clinical research, while the gene-time data are usually obtained to investigate the gene regulations. Since genes regulations and expressions are temporally different, and a snap-shot is insufficient to capture the activities of genes, which may lead some false discovery when using this sort of static data.

* Corresponding author.

R. Rizzo and P.J.G. Lisboa (Eds.): CIBB 2010, LNBI 6685, pp. 275–286, 2011.
© Springer-Verlag Berlin Heidelberg 2011

Within the last few years in medical research, the expression levels of genes with respect to biological samples have been monitored over a series of time-points [1]. At each time point, the genes activities of each sample are are captured as a snap-short. This corresponds to a three-dimensional (3D) data set, termed gene-sample-time (GST) microarray data [2]; which can be viewed as a collection of gene-sample data over a series of time-points, or a collection of gene-time data across some samples.

GST data can be used to develop models to diagnose diseases much more precisely than with static microarray data, or to monitor dose or drug treatment responses of patients over time in pharmacogenomics studies [3], or to determine genes or samples patterns, or to find regulatory pathways [2]. There are many problems associated with the analysis of GST data. Genes or samples may contain missing values at some time points. The expression measurements may contain noise due to technical issues in the measuring process. The expressions of large number of genes are measured from a small number of samples across a small set of time points. Unlike in two-dimensional microarrays, a gene or sample in a GST array is a matrix rather than a vector, and therefore GST data require special methods for its analysis. Computational analysis of GST data are therefore much more difficult than their two-dimensional counterparts. All these problems, among many other problems, substantially affect the effectiveness and efficiency of analysis algorithms devised for GST data.

In multilinear algebra, a tensor of order d [4] is d-dimensional array, and tensor algebra is the extension of vector and matrix algebra to order d tensors. In this sense, a vector an order-1 tensor; and a matrix is of order 2. Since a GST microarray data is naturally an order-3 tensor, therefore, known theories and operations from tensor algebra can be of benefit and used directly to analyze such data rather than performing matrix operations on a *matricized* representation of the GST data (as it is currently done in literature, see [5]). Fig. 1 shows an example of such GST data. As far as we know, this paper is the first attempt at using tensor methods for analyzing GST data. Our approach is to perform multilinear dimensionality reduction methods in order to extract a small sets of discriminative features from the initial GST data, and then perform sample classification in the reduced space. We use tensor decomposition approaches in the dimensionality reduction phase of the GST data classification.

Tensor decomposition is an extension of matrix factorization to tensor data and attempt to find a smaller representation describing the initial tensor data. Matrix factorization methods such as, *singular value decomposition* (SVD) [6], *independent component analysis* (ICA) [7], *non-negative matrix factorization* (NMF) [8] have been extended to tensor data, such as *higher-order SVD* (HOSVD) [9], *multilinear ICA* (MICA) [10], and *higher-order NMF* (HONMF) [11]. Uncorrelated features can be generated by SVD, while independent features with non-Gaussianity can be extracted by ICA. NMF aims to extract non-negative and independent features. These three matrix decomposition methods are well-known linear dimensionality reduction methods that have been applied successfully to the analysis of two-dimensional (2D) microarray data [12, 13, 14]. Inspired by these methods for 2D data, in this paper, we devise approaches based on tensor decomposition methods to classify GST microarray sample data.

This paper is organized as follows. In section 2, we survey currently known methods for GST data classification. Tensor-decomposition-based analysis in other fields is also

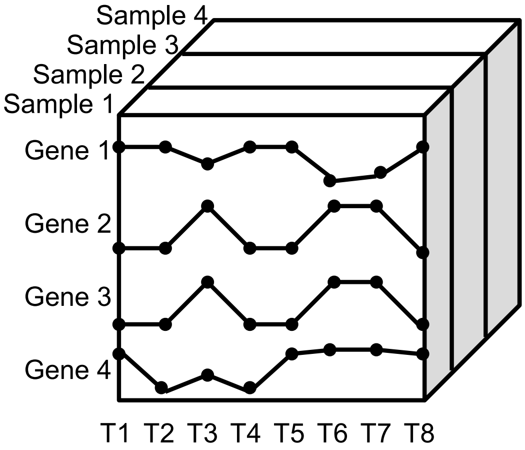


Fig. 1. An Example of a GST Dataset. This example shows that the GST data can be represented by a tensor. This example contains 4 genes, 4 samples, and 8 time points. The number of genes in real data is much larger than the number of samples time points.

reviewed. Our approaches are described in section 3. Section 4 shows our computational experiment results.

2 Related Works

Ref. [15] proposed an *integrated Bayesian inference system* (IBIS) to select triplets of genes for classifying INFβ samples (a GST microarray data) but using only the first time point, and thus did not benefit from (nor consider) the full GST data. Ref. [3] used *support vector machines* based on *dynamical systems kernels* (denoted by dsSVM in this paper) to classify INFβ samples. Since each GST data sample is represented by a matrix, it is not appropriate to use the kernels which take vectorial inputs, such as, the *radial basis functions* (rbf). Dynamical systems kernels accept matrix inputs and take into account the temporal information. Ref. [16] devised *generative hidden Markov models* (GenHMMs) and *discriminative HMMs* (DiscHMMs) approaches for classifying INFβ samples. Samples from the same class are used to train a GenHMM whereas samples from all classes are used to train a DiscHMM, for each class, then a test sample is assigned to a class based on maximum conditional likelihood. Baum-Welch algorithm is used to estimate the parameters of the models. For DiscHMMs, backward gene selection method is first performed to find a small number of discriminative genes before training the models.

[17] propose a robust constrained mixture estimation approach to classify the INFβ data. This approach combines the constrained clustering method with a mixture estimation classification framework. Subdivision of classes and mislabeled samples can be investigated by this approach. During training, negative constraints were restricted on pairs of samples. The constrained mixture model, with linear HMMs, as components, is optimized by an EM algorithm. The supervised version of this approach (*HMMConst*)

only uses training set in the estimation of parameters, while the semi-supervised version (HMMConstAll) uses all data. The emission probability for each state is modeled by mixtures of multivariate Gaussians for patient expression values, noise, and missing values, respectively. In order to select genes contribute to classification, a HMMs based gene ranking method is used. Each component of the mixture model is assigned to a class. When testing, a test sample is assigned to a class according to the maximum entry in their posterior distribution.

Ref. [18] applied HOSVD on an order-5 tensor data for face recognition. In the training phase, a basis tensor for a certain view, illumination, and expression is obtained through HOSVD and then matricized to a basis matrix in order to obtain vector of each training sample. In the testing, the coefficient vector of a testing sample is obtained through a linear projection approach using the basis matrix. A *1-nearest neighbor* (1-NN) classifier is used to determine the class labels of the testing samples. It is not clear in the original paper that if L_1 or L_2 distance is used by the classifier. Ref. [19] used HOSVD to analyze the integration of DNA microarray data from different studies. They create a tensor data of order 3 by combining three gene time-series microarray datasets from yeast cell-cycles, and then decompose the tensor by HOSVD. The resulting core tensor obtained from the decomposition contains the significant features representing important biological experimental phenomena. Ref. [20] devised two different approaches based on HOSVD decomposition to classify a dataset of handwritten digits represented as a tensor data of order 3. HOSVD is used to extract small feature sets that explained the original data but the methods differ in how the core basis tensors are obtained (i.e., either from each class separately, or from the whole data) and in how the class of a test sample is predicted (i.e, either by regression or by projection). Ref. [10] generalized ICA to MICA, and used it for extracting features to be used in face recognition. Initially, facial images are vectorized then represented as tensor data of order 3. MICA is employed to decompose this tensor into factors containing important facial features. A test sample is then multilinearly (rather than linearly, as in [18]) projected into the space spanned by the obtained core basis tensor and a nearest-neighbor classifier using cosine similarity measure employed to predict the class of the test sample. Ref. [21] also applied MICA decomposition to classify integrated tumor gene expression data from different studies. Their working order-3 tensor is a combination of three gene-sample tumors datasets. Two core basis tensors are obtained via MICA decomposition, separately over training samples and test samples. A SVM classifier is then trained on the matricized version of the core tensor obtained from the training sample and validated using the core tensor generated from the test data.

3 Methods

Hereafter, we use the following notations unless otherwise noted:

- A matrix is denoted by a bold capital letter, e.g. $\boldsymbol{A}$.
- A (column) vector is denoted by a bold lowercase letter, e.g. $\boldsymbol{a}$.
- A bold lowercase letter with a subscript $\boldsymbol{a}_i$ denotes the i-th column vector in matrix $\boldsymbol{A}$.

- The italic lowercase letter with two subscripts a_{ij} is the (i, j)-th scalar element of matrix $\boldsymbol{A}$.
- A boldface Euler script, e.g. $\mathcal{X}$, denotes an order-3 tensor. That is $\mathcal{X} \in \mathbb{R}^{I \times J \times K}$.
- $\boldsymbol{X}_{(1)p}$ denotes the p-th frontal slice of $\mathcal{X}$, of size $I \times J$.
- $\boldsymbol{X}_{(n)}$ denotes the matrix obtained through the mode-n matricization of the tensor $\mathcal{X}$. Columns of $\boldsymbol{X}_{(n)}$ are the mode-n fibers of tensor $\mathcal{X}$. A mode-n fiber is a vector defined through fixing every index but the nth index. This is the extension of matrix row and column in tensor algebra. $\boldsymbol{X}_{(1)}$ therefore denotes the matrix of size $I \times JK$, unfolded in mode-1 of $\mathcal{X}$, that is $\boldsymbol{X}_{(1)} = [\boldsymbol{X}_{(1)1}, \boldsymbol{X}_{(1)2}, \cdots, \boldsymbol{X}_{(1)K}]$. See Fig. 2 as an example.
- The (i, j, k)-th scalar element of $\mathcal{X}$ is denoted by x_{ijk}.

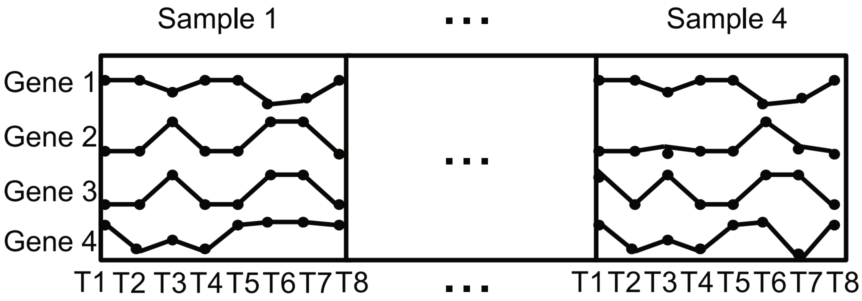


Fig. 2. The Mode-1 Matricization of the Order-3 Tensor in Fig. 1. The tensor is of size 4 × 8 × 4. The unfolded matrix, in mode 1, is of size 4 × 32.

Also, $\boldsymbol{A} \otimes \boldsymbol{B}$ denotes the Kronecker tensor product [4] of matrices $\boldsymbol{A}$ and $\boldsymbol{B}$.

The mode n product of a tensor $\mathcal{X}$ and a matrix $\boldsymbol{A}$, written as $\mathcal{X} \times_n \boldsymbol{A}$, is:

$$\mathcal{X} \times_n \boldsymbol{A} = \sum_{i_n=1}^{I_n} x_{i_1 i_2 \cdots i_N} a_{j i_n} \ , \tag{1}$$

where $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ and $\boldsymbol{A} \in \mathbb{R}^{J \times I_n}$. This results in a tensor $\mathcal{Y} \in \mathbb{R}^{I_1 \times \cdots I_{n-1} \times J \times I_{n+1} \times \cdots \times I_N}$.

$\mathcal{X}$ can be matricized into matrices in different modes. For example, $\boldsymbol{X}_{(1)} = [\boldsymbol{X}_{(1)1}, \boldsymbol{X}_{(1)2}, \cdots, \boldsymbol{X}_{(1)K}]$ is matricized in the first mode (see Fig. 2).

Tensor decomposition methods mainly include PARAFAC and Tucker decompositions [4]. Tucker3 is the most well-known among the Tucker decompositions and factorizes a tensor $\mathcal{X}$ into a core tensor $\mathcal{C}$ and 3 mode matrices $\boldsymbol{G}$, $\boldsymbol{T}$, and $\boldsymbol{S}$ as follows:

$$\mathcal{X} \approx \mathcal{C} \times_1 \boldsymbol{G} \times_2 \boldsymbol{T} \times_3 \boldsymbol{S} = \sum_{p=1}^{P} \sum_{q=1}^{Q} \sum_{r=1}^{R} c_{pqr} \boldsymbol{g}_p \circ \boldsymbol{t}_q \circ \boldsymbol{s}_r = [\![\mathcal{C}; \boldsymbol{G}, \boldsymbol{T}, \boldsymbol{S}]\!] \ . \tag{2}$$

The decomposition is illustrated in Fig. 3. In light of Eq. 2, it is clear that an element of core tensor $\mathcal{C}$ indicates the degree of interaction among the corresponding mode vectors

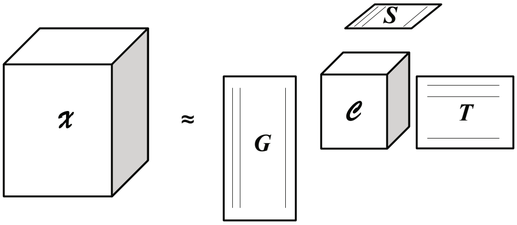


Fig. 3. Tucker3 Decomposition

from different mode matrices. For instance, c_{pqr} reflects the interaction between $\boldsymbol{g}_p$, $\boldsymbol{t}_q$, and $\boldsymbol{s}_r$.

Generally speaking, there are no constraints on the core tensor and mode matrices in Tucker3 decomposition. However, constraints such as orthogonality, non-negativity, and non-Gaussianity can be enforced by the decomposition algorithm. For instance, HOSVD enforces the orthogonality constraints on the mode matrices and is among the most popular Tucker3 algorithms. It calculates the left singular matrices for different matrices in different modes as factors. The core tensor is obtained through $\mathcal{C} = \mathcal{X} \times_1 \boldsymbol{G}^T \times_2 \boldsymbol{T}^T \times_3 \boldsymbol{S}^T$. Interested readers can refer to [9] for more details. *Higher-order orthogonal iterations* (HOOI) is an alternating least squares (ALS) algorithm initialized by HOSVD which give better decomposition than HOSVD itself (see [22] and [4] for details). HOOI also generates orthogonal mode matrices. HONMF imposes non-negativity constraints on the core tensor and mode matrices. Multiplicative updates rules corresponding to core and mode matrices are extended by [11] and inspired by NMF [8]. The core tensor and mode matrices are alternatively updated until the convergence criteria are met. Ref. [8] have observed that good interpretation and learning performance can be benefited by adding non-negativity and sparsity constraints on matrix factorization. Even though the sparsity can be imposed and controlled, sparsity is sometimes a by-product of non-negativity constrained matrix (maybe tensor also) factorization without explicit sparsity constraint.

Next, we describe our unsupervised dimension reduction approaches based on HOSVD, HOOI, and HONMF. *Linear dimension reduction* (LDR) [23] techniques have received considerable attention for decades. A few new features can be extracted by LDR methods to capture useful information for specific analysis. Each of the new feature is a linear combination of the original features. A transformation matrix projects the original samples into a new space, termed *feature space*. A sample in the feature space is a *representation* of the corresponding original sample. Take NMF for example, a non-negative training set $\boldsymbol{X}^{\text{train}}$ with m genes and p samples of a gene-sample data can be decomposed into non-negative basis matrix $\boldsymbol{A}^{\text{train}}$ and non-negative coefficient matrix $\boldsymbol{Y}^{\text{train}}$, that is

$$\boldsymbol{X}^{\text{train}}_{m\times p} \approx \boldsymbol{A}^{\text{train}}_{m\times r}\boldsymbol{Y}^{\text{train}}_{r\times p}, \quad \boldsymbol{X}^{\text{train}}, \boldsymbol{A}^{\text{train}}, \boldsymbol{Y}^{\text{train}} \geq 0 \,. \tag{3}$$

each column of $\boldsymbol{Y}^{\text{train}}$ is a representation of the corresponding original sample in the feature space spanned by columns of $\boldsymbol{A}^{\text{train}}$. In the feature space, a new feature is a linear combination of the original p genes. A sample in the original sample can be mapped into the feature space by the transformation matrix $(\boldsymbol{A}^{\text{train}})^T$. The feature space has much less dimensions than the original space, which solves the curse of dimensionality. LDR methods extends into *multilinear dimension reduction* (MLDR) methods in tensor algebra. Let $\mathcal{X}$ be a training set, from a GST dataset, with I genes, J time points, and K samples. Through The Tucker3 model defined in Eq. 2, we can obtain

$$\mathcal{X} \approx \mathcal{B} \times_3 \boldsymbol{S} = [\![\mathcal{B}; \boldsymbol{I_G}, \boldsymbol{I_T}, \boldsymbol{S}]\!] \ , \tag{4}$$

where $\mathcal{B} = \mathcal{C} \times_1 \boldsymbol{G} \times_2 \boldsymbol{T}$, $\boldsymbol{I}_G$ and $\boldsymbol{I}_T$ are identity matrices of sizes $I \times I$ and $J \times J$, respectively.

Making use of multilinear operations, we have

$$\begin{aligned} \boldsymbol{X}_{(1)} &\approx \boldsymbol{I}_G \boldsymbol{B}_{(1)} (S \otimes \boldsymbol{I}_T)^T \\ &= \boldsymbol{I}_G [\boldsymbol{B}_1, \boldsymbol{B}_2, \cdots, \boldsymbol{B}_R][\boldsymbol{s}_1 \otimes \boldsymbol{I}_T, \boldsymbol{s}_2 \otimes \boldsymbol{I}_T, \cdots, \boldsymbol{s}_R \otimes \boldsymbol{I}_T]^T \\ &= \sum_{r=1}^{R} \boldsymbol{I}_G \boldsymbol{B}_r (\boldsymbol{s}_r \otimes \boldsymbol{I}_T)^T \ , \end{aligned} \tag{5}$$

where $\boldsymbol{B}_r = \mathcal{B}(:,:,r)$ is the r-th frontal slice of $\mathcal{B}(:,:,r)$, and $\boldsymbol{s}_r$ is the r-th column vector of S. Via tensorization, we have

$$\mathcal{X} \approx \sum_{r=1}^{R} \boldsymbol{B}_r \times_3 \boldsymbol{s}_r \ , \tag{6}$$

which approximates the GST data, $\mathcal{X}$, by the summation of R tensors. More clearly,

$$\begin{aligned} \boldsymbol{X}_{(1)} &\approx \boldsymbol{I}_G \boldsymbol{B}_{(1)} (S \otimes \boldsymbol{I}_T)^T \\ &= [\boldsymbol{B}_1, \boldsymbol{B}_2, \cdots, \boldsymbol{B}_R] \begin{bmatrix} \boldsymbol{s}_{11}\boldsymbol{I}_T & \cdots & \boldsymbol{s}_{k1}\boldsymbol{I}_T \\ \vdots & \vdots & \vdots \\ \boldsymbol{s}_{1R}\boldsymbol{I}_T & \cdots & \boldsymbol{s}_{kR}\boldsymbol{I}_T \end{bmatrix} . \end{aligned} \tag{7}$$

Thus k-th frontal slice of $\mathcal{X}$, that is, the k-th sample, can be fitted by the summation of the frontal slices of $\mathcal{B}$:

$$\boldsymbol{X}_{(1)k} \approx \sum_{i=1}^{R} \boldsymbol{B}_r \boldsymbol{s}_{kr} \ , \tag{8}$$

where the coefficients are in the k-th row of $\boldsymbol{S}$.

Thus, $\mathcal{B}$ is the basis matrix for the samples and $\boldsymbol{S}$ is the encoding matrix. We can define the matrix space spanned by $\mathcal{B}$ feature space, and $\boldsymbol{s}_k$ the representation of the k-th sample in the feature space. In the sense of feature extraction, these matrix slices of $\mathcal{B}$ are the*features*. This reduces the original sample slice to a vector $\boldsymbol{s}_k$ in the feature space. Additionally, it is noted that $\mathcal{C} \times_2 \boldsymbol{T} \times_3 \boldsymbol{S}$ and $\mathcal{C} \times_1 \boldsymbol{G} \times_3 \boldsymbol{S}$ are the basis matrices for the genes and time points, respectively. If the training set is decomposed by HONMF, the extracted non-negative features would be interpretable, and a sample will be an additive summation of the features.

In the test phase, each test sample $\boldsymbol{Y}_l$ is projected into the feature space. $\boldsymbol{Y}_l$ is a linear combination of the basis matrices in $\mathcal{B}$:

$$\boldsymbol{Y}_l = \sum_{r=1}^{R} \boldsymbol{B}_r \alpha_r \ , \tag{9}$$

where $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, \cdots, \alpha_R]^T$ is the representation of $\boldsymbol{Y}_l$ in the feature space. Finding α is equivalent to solve the following generalized least squares problem:

$$\min_{\boldsymbol{\alpha}} \|\boldsymbol{Y}_l - \sum_{r=1}^{R} \boldsymbol{B}_r \alpha_r\|_F^2 \ , \tag{10}$$

where $\| \bullet \|_F$ is Frobenius norm of a matrix. The general solution to this problem is $\alpha_r = \frac{<\boldsymbol{Y}_l, \boldsymbol{B}_r>}{<\boldsymbol{B}_r, \boldsymbol{B}_r>}$ [24], where $< \bullet, \bullet >$ is the inner product of two matrices. For different test samples, we put $\boldsymbol{\alpha}$'s in the corresponding rows of a coefficient matrix $\boldsymbol{A}$. The above HOSVD, HOOI, and HONMF based unsupervised MLDR methods are referred to as uHOSVDls, uHOOIls, and uHONMFls.

Alternatively, given the test samples $\mathcal{Y}$, we can fix $\mathcal{C}$, $\boldsymbol{G}$, and $\boldsymbol{T}$ to calculate the coefficient matrix $\boldsymbol{A}$ of $\mathcal{Y}$. We need to find $\boldsymbol{A}$ that satisfies

$$\mathcal{Y} \approx \mathcal{C} \times_1 \boldsymbol{G} \times_2 \boldsymbol{T} \times_3 \boldsymbol{A} \ . \tag{11}$$

For HOSVD and HOOI, the mode matrices are orthogonal and $\boldsymbol{A}$ is the R leading left singular vectors of $\boldsymbol{Z}_{(3)}$. $\boldsymbol{Z}_{(3)}$ is matricized from $\mathcal{Z}$ which is calculated by the following equation:

$$\mathcal{Z} = \mathcal{Y} \times_1 \boldsymbol{G}^T \times_2 \boldsymbol{T}^T \ . \tag{12}$$

For HONMF, the constraint on the mode matrices is non-negativity rather than orthogonality. Instead of solving the non-negativity constrained equation similar to Eq. 12, $\boldsymbol{A}$ can be rapidly obtained using the update rules of the HONMF algorithm. We can iteratively only update $\boldsymbol{A}$, while keeping $\mathcal{C}$, $\boldsymbol{G}$, and $\boldsymbol{T}$ constant. If this method is used for HONMF and Eq. 12 is used for HOSVD and HOOI, then the resulting algorithms are denoted by uHONMFtf, uHOSVDtf, and uHOOItf, respectively.

Once $\boldsymbol{A}$ is obtained, we do not need to learn on the training samples and classify the test samples represented by the matrices. Instead, any classifier can be trained on $\boldsymbol{S}$ and classify the rows of $\boldsymbol{A}$. That is the classification is conducted in the feature space.

The decomposition methods described above are unsupervised dimensionality reduction techniques but they can be modified to perform in a supervised manner, i.e. such that class information is taken into account during decomposition. Let m be the number of distinct class labels in the data. The idea is to first partition the training set into m subsets $\mathcal{X}^1$, $\mathcal{X}^2$, $\cdots$, $\mathcal{X}^m$, where each subset $\mathcal{X}^i$ contains only samples of class i. Next, m core tensors $\mathcal{B}^1$, $\mathcal{B}^2$, $\cdots$, $\mathcal{B}^m$ are obtained through decomposition using Eq. 4. The resulting basis matrices are then normalized using the Frobenius norm. For a normalized test sample, we fit it using these basis tensors, respectively, through Eq. 10. This sample is assigned to the class where the minimal fitting residual is obtained. For simplicity, we denote the supervised version of HOSVD, HOOI, and HONMF based classification methods by sHOSVD, sHOOI, sHONMF. This supervised decomposition approach is described in [20] for hand written recognition using HOSVD.

4 Experiments

We used our approaches to predict good or bad responders to *Interferon beta* (INFβ) treatments. INFβ is a protein used for treating patients afflicted with multiple-sclerosis (MS), among other diseases. Some MS patients who received INFβ therapy do not respond well to the drug and the reasons are still not clear [1]. Medical researchers are seeking reasons at the levels biological molecules. Baranzini *et al.* [15], among others researchers, applied Bayesian learning methods on a clinical microarray dataset to determine pairs or triplets of genes that can discriminate between bad and good INFβ responders. This dataset is online available as the supplemental material of [15]. The initial dataset is a GST data sampled from 53 MS patients who were initially treated with equal dose of INFβ over a time period. This initial dataset contains the expression measurements for 76 genes at 7 time points (0, 3, 6, 9, 12, 18 and 24 months) for each patient, with 31 patients responding well and the remaining 22 responding bad to the treatment. This dataset contains genes with missing expression measurements at some time points. Those genes and corresponding samples were removed from our analysis, and hence, the resulting ”complete” data contains 53 genes and 27 samples (18 good responders and 9 bad responders).

We implemented our tensor-based approaches using Matlab, applied them to the INFβ data, and compared them with rbfSVM, dsSVM, GenHMMs, and DiscHMMs approaches (described in the related work section). rbfSVM is the method that vectorized samples are classified by SVM classifier using rbf kernel function. Our implementation is based on *The N-way toolbox for MATLAB* [25] and *Algorithms for SN-TUCKER (Higher-Order Non-Negative Matrix Factorization)* [11] [26]. We used k-nearest neighbor classifier with Euclidean distance in the classification phase of our unsupervised methods. Training with 9-fold cross-validation is employed. All our methods are performed for 20 runs, and the means and standard deviations of specificity, sensitivity, and accuracy are reported in Table 1. Specificity is the prediction accuracy of the good responders, while sensitivity is that of the bad responders. The parameter of rbfSVM is the value of λ in the rbf function. The first parameter of dsSVM is the number of hidden states, and the second one is the parameter of the dynamical systems kernel function. The parameter for GenHMMs and DiscHMMs is the number of selected genes; absence of such parameter means gene selection is not used. The parameter of the tensor decomposition based approaches are rank-(P, Q, R) and grid search is performed to find the values of P, Q, R that give best classification performance.

As show in Table 1, uHONMFtf obtains the highest mean prediction accuracy (0.8148). This is significantly better than dsSVM, GenHMMs, and DiscHMMs without and with gene selection (0.7593 and 0.7611, respectively). uHOSVDls, uHOOIls, and uHOOItf obtains similar accuracies which are competitive with dsSVM, GenHMMs, and DiscHMMs. This means that the tensor-decomposition based unsupervised methods can capture the discriminative information. The poor result of rbfSVM implies that vectorizing GST samples makes the temporal information lost, and therefore the powerful pattern recognition methods for 2D data may not appropriate for tensor data. uHONMFtf outperforms the HOSVD and HOOI based methods due to non-negativity. The reasons why uHONMFls and uHOOItf do not performed well needs further investigation. The supervised sHOSVD, sHOOI and sHONMF did not achieve good results.

Table 1. Comparison of Classification Performance on Complete INF-Beta Data

Methods	Param.	Specificity	Sensitivity	Accuracy
rbfSVM	1	**1.000±0.000**	0.000±0.000	0.667±0.000
dsSVM	1,5	0.972±0.082	0.422±0.013	0.789±0.023
GenHMMs	-	0.8611±0.036	0.5556±0.000	0.7593±0.044
DiscHMMs	-	0.8611±0.036	0.5556±0.000	0.7593±0.044
GenHMMs	7	0.8611±0.063	0.5611±0.008	0.7611±0.047
DiscHMMs	7	0.8611±0.063	0.5611±0.008	0.7611±0.047
uHOSVDls	7,3,3	0.8389±0.039	0.5944±0.020	0.7574±0.050
uHOOIls	4,3,10	0.9000±0.031	0.5000±0.012	0.7667±0.035
uHONMFls	3,5,3	0.8972±0.079	0.3056±0.034	0.7000±0.052
uHOSVDtf	4,2,3	0.7639±0.053	0.5500±0.041	0.6926±0.046
uHOOItf	3,7,3	0.8111±0.048	0.6611±0.055	0.7611±0.050
uHONMFtf	3,5,3	0.7889±0.029	**0.8667±0.154**	**0.8148±0.040**
sHOSVD	4,3,8	0.8306±0.054	0.6333±0.012	0.7648±0.044
sHOOI	3,4,4	0.7611±0.045	0.6667±0.000	0.7296±0.039
sHONMF	3,4,6	0.9583±0.110	0.0056±0.069	0.6407±0.075

This may be because the same parameter is used when learning from the training subsets of good responders and bad responders, separately. Different classes should have different structures, which requires different parameters, however, it is expensive to search the best parameters corresponding to different classes.

The multi-dimensional reduction techniques are able to dramatically reduce the dimension of the original dataset and transform the sample matrices into new ”equivalent” short vectors which are used for classification. In uHONMFtf for example, a 53 by 7 test sample can be represented by a vector of size 1 by 3 in the new feature space; thus reducing the data by 99.19% while preserving discriminative information.

In order to compare our proposed methods with dsSVM, GenHMMs and DiscHMMs in time complexity, the execution times (in seconds) are recorded for each method. Table 2 shows the time results. The number of selected genes is parameterized to 7 for DiscHMMs. The tensor decomposition based approaches use the same parameter $(3, 5, 3)$. It can be seen that the tensor based methods, in particular HOSVD and HOOI methods, are much faster than the HMMs based method while giving at least comparable classification results. The uHOSVDls and uHOOIls methods run much faster than the dsSVM methods.

Table 2. Comparison of Running Time on Complete INF-Beta Data

Methods	dsSVM	DiscHMMs	uHOSVDls	uHOOIls	uHONMFtf
Time (s.)	93.474	2.117×10^3	1.321	1.057	1.662×10^3

5 Conclusion

Methods devised specifically for the analysis of GST data will be very useful in the near future, as many recent clinical and biological data are given in the form of tensor data of order 3 or more. In this regards, we have implemented a number of tensor-based methods for classifying sample GST data from INFβ dataset. We have shown that our approaches are faster and still comparable in classification performances to three recent methods developed for analyzing the same dataset. Our methods should be also suitable for other types of microarray data which are represented by tensors. More research need to be done, however, to test the scalability proposed approach on more real data, to improve the classification performances of the tensor-based methods, and in particular to devise methods that can deal with missing values. We also plan to investigate gene selection methods such as gene-pairs or gene-triplets search algorithms for bio-marker selection. Beside classification, bi-clustering and tri-clustering approaches for GST data will be studied for determining pattern of genes or samples given certain doses (in dose-response GST data) or time intervals (in drug-response GST data).

Acknowledgments. The authors would like to thank Dr. Rasmus Bro who gave us valuable suggestions in the analysis of the INFβ dataset, and Dr. Morten Mørup who revised the HONMF MATLAB code for our application. This research has been supported by IEEE CIS Walter Karplus Summer Research Grant 2010, and Canadian NSERC Grants #RGPIN228117-2006.

References

1. Weinstock-Guttman, B., Badgett, D., Patrick, K., Hartrich, L., Santos, R., Hall, D., Baier, M., Feichter, J., Ramanathan, M.: Genomic Effects of IFN-β in Multiple Sclerosis Patients. The Journal of Immunology 171(5), 2694–2702 (2003)
2. Zhang, A.: Advanced Analysis of Gene Expression Microarray Data. World Scientific Press, Singapore (2009)
3. Borgwardt, K.M., Vishwanathan, S.V.N., Kriegel, H.P.: Class Prediction from Time Series Gene Expression Profiles Using Dynamical Systems Kernels. In: Proc. Pacific Symposium on Biocomputing, pp. 547–558. World Scientific Press, Singapore (2006)
4. Kolda, T.G., Bader, B.W.: Tensor Decompositions and Applications. SIAM Review 51(3), 455–500 (2009)
5. Li, Y., Ngom, A., Rueda, L.: Missing Value Imputation Methods for Gene-Sample-Time Microarray Data Analysis. In: Proc. IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology, pp. 183–189. IEEE Press, New York (2010)
6. Golub, G.H., Van Loan, C.F.: Matrix Computations, 3rd edn. The Johns Hopkins University Press, Maryland (1996)
7. Hyărinen, A., Karhunen, J., Oja, E.: Independent Component Analysis. Wiley, New York (2001)
8. Lee, D.D., Seung, S.: Learning the Parts of Objects by Non-Negative Matrix Factorization. Science 401, 788–791 (1999)
9. Lathauwer, L.D., Moor, B.D., Vandewalle, J.: A Multilinear Singular Value Decomposition. SIAM Journal on Matrix Analysis and Applications 21(4), 1253–1278 (2000)
10. Vasilescu, M.A.O., Terzopoulos, D.: Mulitlinear Independent Component Analysis. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition, vol. 1, pp. 547–553. IEEE Press, New York (2005)
11. Mørup, M., Hansen, L.K., Arnfred, S.M.: Algorithms for Sparse Nonnegative Tucker Decompositions. Neural Computation 20(8), 2112–2131 (2008)

12. Alter, O., Brown, P.O., Botstein, D.: Singular Value Decomposition for Genome-Wide Expression Data Processing and Modeling. Proceedings of the National Academy of Sciences 97(18), 10101–10106 (2000)
13. Huang, D., Zheng, C.: Independent Component Analysis-Based Ppenalized Discriminant Method for Tumor Classification Using Gene Expression Data. Bioinformatics 22, 1855–1862 (2006)
14. Zheng, C., Zhang, P., Zhang, L., Liu, X., Han, J.: Gene Expression Data Classification Based on Non-Negative Matrix Factorization. In: Proc. International Joint Conference on Neural Networks, pp. 3542–3547. IEEE Press, New York (2009)
15. Baranzini, S.E., Mousavi, P., Rio, J., Caillier, S.J., Stillman, A., Villoslada, P.: Transcription-Based Prediction of Response to INFβ Using Supervised Computational Methods. PLOS Biology 3(1), e2 (2005)
16. Lin, T., Kaminski, N., Bar-Joseph, Z.: Alignment and Classification of Time Series Gene Expression in Clinical Studies. Bioinformatics 24 (ISMB 2008), i147–i155 (2008)
17. Costa, I.G., Schönhuth, A., Hafemeister, C., Schliep, A.: Constrained Mixture Estimation for Analysis and Robust Classification of Clinical Time Series. Bioinformatics 25 (ISMB 2009), i6–i14 (2009)
18. Vasilescu, M.A.O., Terzopoulos, D.: Multilinear Image Analysis for Facial Recognition. In: Proc. the International Conference on Pattern Recognition, vol. 3, pp. 511–514. IEEE Press, New York (2002)
19. Omberg, L., Golub, G.H., Alter, O.: A Tensor Higher-Order Sigular Value Decomposition for Integrative Analysis of DNA Microarray from Different Studies. Proceedings of the National Academy of Sciences 104(47), 18371–18376 (2007)
20. Savas, B., Eldén, L.: Handwritten Digit Classification Using Higher Order Singular Value Decomposition. Pattern Recongtion 40, 993–1003 (2007)
21. Du, M., Zhang, S., Wang, H.: Tumor Classification Using Higher-Order Gene Expression Profiles Based on Multilinear ICA. Advances in Bioinformatics (2009), doi:10.1115/2009/926450 (2009)
22. Andersson, C.A., Bro, R.: Improving the Speed of Multi-Way Algorithms: Part I. Tucker3. Chemometrics and Intelligent Laboratory Systems 42, 93–103 (1998)
23. Rueda, L., Herrera, M.: Linear Dismensionality Reduction by Maximizing the Chernoff Distance in the Transformed Space. Pattern Recognition 41, 3138–3152 (2008)
24. Savas, B.: Analyses and Tests of Handwritten Digit Recognition Algorithms. Master thesis, Dept. Mathathmatics Scientific Computing, Linköping University, Linköping, Sweden (2003)
25. Andersson, C.A., Bro, R.: The N-Way Toolbox for MATLAB. Chemometrics and Intelligent Laboratory Systems 52, 1–4 (2000)
26. Mørup, M.: Algorithms for SN-TUCKER, `http://www2.imm.dtu.dk/pubdb/views/edoc_download.php/4718/zip/imm4718.zip`

Author Index