Methods in Molecular Biology 1231

Springer Protocols

Alessio Mengoni
Marco Galardini
Marco Fondi *Editors*

Bacterial Pangenomics

Methods and Protocols

Humana Press

METHODS IN MOLECULAR BIOLOGY

Series Editor
John M. Walker
School of Life Sciences
University of Hertfordshire
Hatfield, Hertfordshire, AL10 9AB, UK

For further volumes:
http://www.springer.com/series/7651

Bacterial Pangenomics

Methods and Protocols

Edited by

Alessio Mengoni

Department of Biology, University of Florence, Florence, Italy

Marco Galardini

EMBL-EBI, Cambridge, UK

Marco Fondi

Department of Biology, University of Florence, Florence, Italy

Humana Press

Editors
Alessio Mengoni
Department of Biology
University of Florence
Florence, Italy

Marco Fondi
Department of Biology
University of Florence
Florence, Italy

Marco Galardini
EMBL-EBI
Cambridge, UK

ISSN 1064-3745 ISSN 1940-6029 (electronic)
ISBN 978-1-4939-1719-8 ISBN 978-1-4939-1720-4 (eBook)
DOI 10.1007/978-1-4939-1720-4
Springer New York Heidelberg Dordrecht London

Library of Congress Control Number: 2014951682

Printed on acid-free paper

Humana Press is a brand of Springer
Springer is part of Springer Science+Business Media (www.springer.com)

Preface

From a pioneering field a decade ago, now bacterial genomics is a mature research interdisciplinary field, which is approached by ecologists, geneticists, bacteriologists, molecular biologists, and evolutionary biologists working in medical, industrial, and basic science. The high diffusion of bacterial genomics in many different fields has been helped by the low costs of genome and transcriptome sequencing performed by the so-called Next Generation Sequencing (NGS) technologies. Now, the cost of a draft bacterial genome sequence is as low as few hundreds of Euro (or Dollars). This low cost is allowing many laboratories to perform genome sequencing of virtually every "interesting" bacterial strain they have in hand. In parallel, bioinformatic analysis of the data has grown and the specialized bioinformatician is an obliged professional figure in every laboratory that is interested in genome sequencing.

One of the most striking differences of bacterial genomics with respect to the genomics of eukaryotic multicellular organisms is the concept of pangenome, which was introduced in the late 2005 by researchers working on bacterial pathogenic species. The pangenome is defined as a genomic approximation to describe a species' genome in terms of the sum of core (conserved in all strains) and dispensable (variable among strains) genes. For bacterial species, the pangenome concept is particularly relevant since closely related strains usually show large differences in gene content between them. Consequently, when speaking about bacterial genomics, often people are referring to comparative analysis of bacterial genomes and then to what we can call "bacterial pangenomics." Understanding which genetic components of this large pangenomic variability are functionally, clinically, or evolutionary relevant is a challenging task; in fact, a large fraction of the dispensable genome is found to have a poor functional characterization. The availability of powerful and precise analysis tools is therefore of paramount importance.

Thanks to the large diffusion of bacterial genome analysis (or bacterial pangenomic studies), the present book is intended to provide the most recent methodologies about the study of bacterial pangenomes. Three major areas are covered, namely the experimental methods for approaching bacterial pangenomics ("Preparing the bacterial pangenome"), the bioinformatic pipelines for analysis and annotation of sequence data ("Defining the pangenome"), and finally the methods for inferring functional and evolutionary features from the pangenome ("Interpreting the pangenome"). In each of these sections, researchers from both academia and private leading companies of NGS and bioinformatic analysis (as Beijing Genome Institute, Life Technologies, Era7 Bioinformatics) are providing the most up-to-date protocols and procedures for bacterial genome analysis, from assessment of genome size and structure to the analysis of raw sequence data and their annotation and biological interpretation in terms of gene activity and metabarcoding diversity and genome evolution.

The aim of the present book is then to serve as a "field guide" both for qualified investigators on bacterial genomics who want to update their technical knowledge and for less-experienced researchers who want to start working with bacterial genomics and pangenomics.

Additionally, the book could serve to graduate students as a manual of methods used in bacterial pangenomics and as a supplemental textbook in classes of genomics and bioinformatics.

Florence, Italy — *Alessio Mengoni*
Florence, Italy — *Marco Fondi*
Cambridge, UK — *Marco Galardini*

Contents

Contributors

Rosa Alduina • *Department of Biological, Chemical and Pharmaceutical Sciences and Technologies, University of Palermo, Palermo, Italy*

Alexey Alekhin • *Oh no sequences! Research group, Era7 Bioinformatics, Granada, Spain*

M. Ancora • *Istituto Zooprofilattico Sperimentale dell'Abruzzo e Molise "G. Caporale", National and OIE Reference Laboratory for Brucellosis, Teramo, Italy*

Giovanni Bacci • *Department of Biology, University of Florence, Florence, Italy; Consiglio per la Ricerca e la Sperimentazione in Agricoltura, Centro di Ricerca per lo Studio delle Relazioni tra Pianta e Suolo (CRA-RPS), Rome, Italy*

Dariusz Bartosik • *Institute of Microbiology, Department of Bacterial Genetics, University of Warsaw, Warsaw, Poland*

Marco Bazzicalupo • *Department of Biology, University of Florence, Florence, Italy*

Emanuele Bosi • *Department of Biology, University of Florence, Florence, Italy*

C. Cammà • *Istituto Zooprofilattico Sperimentale dell'Abruzzo e Molise "G. Caporale", National and OIE Reference Laboratory for Brucellosis, Teramo, Italy*

Duccio Cavalieri • *Department of Computational Biology, Fondazione Edmund Mach, San Michele all'Adige, Trento, Italy*

Alice Checcucci • *Department of Biology, University of Florence, Florence, Italy*

Lei Cheng • *BGI, Shenzen, China*

F. Del Chierico • *Unit of Parasitology and Unit of Metagenomics, Bambino Gesù Children's Hospital, IRCCS, Rome, Italy*

Salvatore Conti • *Thermo Fisher Scientific, Monza, Italy*

Bruno Contreras-Moreira • *Centro de Ciencias Genómicas, Universidad Nacional Autónoma de México, Cuernavaca, Morelos, Mexico; Estación Experimental de Aula Dei, Consejo Superior de Investigaciones Científicas (EEAD-CSIC), Zaragoza, Spain; Fundación ARAID, Zaragoza, Spain*

Francesca Decorosi • *Dipartimento di Scienze delle Produzioni Agroalimentari e dell'Ambiente (DISPAA), University of Florence, Florence, Italy*

Lukasz Dziewit • *Department of Bacterial Genetics, Institute of Microbiology, University of Warsaw, Warsaw, Poland*

Kristof Engelen • *Department of Computational Biology, Fondazione Edmund Mach, San Michele all'Adige, Trento, Italy*

Renato Fani • *Department of Biology, University of Florence, Florence, Italy*

Elio Fantini • *Italian National Agency for New technologies, Energy and Sustainable development, Rome, Italy*

Alessia Fiore • *Italian National Agency for New technologies, Energy and Sustainable development, Rome, Italy*

Marco Fondi • *Department of Biology, University of Florence, Florence, Italy*

Marco Galardini • *EMBL-EBI, Cambridge, UK*

Giulio Gianese • *Italian National Agency for New technologies, Energy and Sustainable development, Rome, Italy*

Luciana Giovannetti • *Dipartimento di Scienze delle Produzioni Agroalimentari e dell'Ambiente (DISPAA), University of Florence, Florence, Italy*

Giovanni Giuliano • *Italian National Agency for New technologies, Energy and Sustainable development, Rome, Italy*
Zhenfei Hu • *BGI, Shenzen, China*
Evdokim Kovach • *Oh no sequences! Research group, Era7 Bioinformatics, Granada, Spain*
Kris Laukens • *Department of Mathematics and Computer Science, University of Antwerp, Antwerp, Belgium; Biomedical Informatics Research Center Antwerp (biomina), Antwerp University Hospital, University of Antwerp, Edegem, Belgium*
Pietro Liò • *Computer Laboratory, University of Cambridge, Cambridge, UK*
Marina Manrique • *Oh no sequences! Research group, Era7 Bioinformatics, Granada, Spain*
M. Marcacci • *Istituto Zooprofilattico Sperimentale dell'Abruzzo e Molise "G. Caporale", National and OIE Reference Laboratory for Brucellosis, Teramo, Italy*
Emmanuela Marchi • *Dipartimento di Scienze delle Produzioni Agroalimentari e dell'Ambiente (DISPAA), University of Florence, Florence, Italy*
Alessio Mengoni • *Department of Biology, University of Florence, Florence, Italy*
Pieter Meysman • *Department of Mathematics and Computer Science, University of Antwerp, Antwerp, Belgium; Biomedical Informatics Research Center Antwerp (biomina), Antwerp University Hospital, University of Antwerp, Edegem, Belgium*
Stefano Mocali • *Consiglio per la Ricerca e la sperimentazione in Agricoltura, Centro di Ricerca per l'Agrobiologia e la Pedologia (CRA-ABP), Florence, Italy*
Marco Moretto • *Department of Computational Biology, Fondazione Edmund Mach, San Michele all'Adige, Trento, Italy*
Valerio Orlandini • *Department of Biology, University of Florence, Florence, Italy; Department of Protein Biochemistry, National Research Council, Napoli, Italy*
Eduardo Pareja • *Oh no sequences! Research group, Era7 Bioinformatics, Granada, Spain*
Pablo Pareja-Tobes • *Oh no sequences! Research group, Era7 Bioinformatics, Granada, Spain*
Eduardo Pareja-Tobes • *Oh no sequences! Research group, Era7 Bioinformatics, Granada, Spain*
Annalisa Pisciotta • *Department of Biological, Chemical and Pharmaceutical Sciences and Technologies, University of Palermo, Palermo, Italy*
L. Putignani • *Unit of Parasitology and Unit of Metagenomics, Bambino Gesù Children's Hospital, IRCCS, Rome, Italy*
Ermanno Rizzi • *Institute for Biomedical Technologies (ITB), National Research Council (CNR), Segrate, MI, Italy*
Paolo Sonego • *Department of Computational Biology, Fondazione Edmund Mach, San Michele all'Adige, Trento, Italy*
Raquel Tobes • *Oh no sequences! Research group, Era7 Bioinformatics, Granada, Spain*
Pablo Vinuesa • *Centro de Ciencias Genómicas, Universidad Nacional Autónoma de México, Cuernavaca, Morelos, Mexico*
Roberto Viola • *Department of Computational Biology, Fondazione Edmund Mach, San Michele all'Adige, Trento, Italy*
Carlo Viti • *Dipartimento di Scienze delle Produzioni Agroalimentari e dell'Ambiente (DISPAA), University of Florence, Florence, Italy*
Hai Wang • *BGI, Shenzen, China*
Zhuofei Xu • *Section of Microbiology, Department of Biology, University of Copenhagen, Copenhagen, Denmark*
Rui Zhou • *Section of Microbiology, Department of Biology, University of Copenhagen, Copenhagen, Denmark*

Chapter 1

Pulsed Field Gel Electrophoresis and Genome Size Estimates

Rosa Alduina and Annalisa Pisciotta

Abstract

Pulsed field gel electrophoresis (PFGE) is a quick and reliable procedure to resolve DNA molecules larger than 30 kb by applying an electric field that periodically changes direction. This technique can be used to estimate genome size of a microorganism, to reveal if a genome is circular or linear, to indicate the presence of megaplasmids, and to show if a strain contains only one or more chromosomes.

Key words Genome size, Genome topology, Multi-replicons, Megaplasmids

1 Introduction

Pulsed field gel electrophoresis (PFGE) is an electrophoretic technique to resolve DNA fragments from 30 kb to various Mb by applying an electric field that periodically changes direction, overcoming the size limitations, due to running DNA molecules in a conventional gel electrophoresis, where a static electric field is applied. The concept that large DNA molecules could be separated by using alternating electric fields was introduced in 1982 [1]. The pulsed electrophoresis effect has been utilized by a variety of instruments (FIGE, TAFE, CHEF, OFAGE, PACE, and rotating electrode gel) to increase the size resolution of both large and small DNA molecules [2–5]. Contour-clamped homogeneous electric field (CHEF) is the most widely used apparatus that produces homogeneous electric fields so that all lanes of a gel run straight and allow separation of molecules up to 10,000 kb.

General applications of PFGE can be the separation of whole chromosomes, the resolution of megaplasmids, and the determination of genome and plasmid size and topology. Here, methods to resolve and size high-molecular-weight DNA fragments are described.

Alessio Mengoni et al. (eds.), *Bacterial Pangenomics: Methods and Protocols*, Methods in Molecular Biology, vol. 1231, DOI 10.1007/978-1-4939-1720-4_1,

Table 1
Microbial genome size and GC % content

Organism	Size (Mb)	GC %	Relevant feature
Archaea[a]			
Nanoarchaeum equitans Kin4-M	0.49	31.6	Smallest genome
Methanosarcina acetivorans C2A	5.75	42.7	Largest genome
Methanosphaera stadtmanae DSM 3091	1.77	27.6	Lowest GC% content
Salinarchaeum sp. Harcht-Bsk1	3.26	66.6	Highest GC% content
Bacteria[b]			
Candidatus Tremblaya princeps PCIT	0.139	58.8	Smallest genome
Sorangium cellulosum So0157-2	14.78	72.1	Largest genome
Candidatus Zinderia insecticola CARI	0.21	13.5	Lowest GC% content
Anaeromyxobacter dehalogenans 2CP-C	5.01	74.9	Highest GC% content

Data are from http://www.ncbi.nlm.nih.gov/genome/browse/. Only complete sequences were taken onto account
[a]165 sequences
[b]2,640 sequences

1.1 Determination of Genome Size

Bacteria exhibit a large variability concerning genome size; among all completely sequenced 2,805 archaeal and bacterial genomes (NCBI Complete Microbial Genomes http://www.ncbi.nlm.nih.gov/genomes/lproks.cgi) thus far, the 139 kb of *Candidatus Tremblaya princeps* represents the smallest genome [6] and the 14.78 Mb of *Sorangium cellulosum So0157-2* is the largest one [7], followed by 13.7 Mb of *Ktedonobacter racemifer* SOSP1-21 T [8], 13.03 Mb of *Sorangium cellulosum* So ce56 [9], and most actinomycetes that usually have a genome larger than 8 Mb, i.e., *Streptomyces bingchenggensis*, 11.9 Mbp [10], *Catenulispora acidiphila*, 10.5 Mbp [11], and *Streptosporangium roseum*, 10.4 Mbp [12]. Table 1 shows the limits, so far known, of genome size (0.036–14.78 Mb) and GC% content (13.5–74.9 %) of Archaea and Bacteria.

Pulsed field gel electrophoresis (PFGE) and/or complete genome sequencing are the predominant applied methods to determine bacterial genome size. Different methods of complete genome sequencing will be presented in the following chapters.

If the electrophoretic method is used, cells are grown at the exponential phase in a liquid broth, embedded in agarose plugs, and lysed; after washing steps, genomic DNA, protected in the agarose, is digested with an appropriate restriction enzyme, and fractionated by PFGE.

The choice of the suitable restriction enzyme is a challenging issue and, mainly, depends upon the base composition (%G+C content) of the DNA of the microorganism of interest. Indeed, it

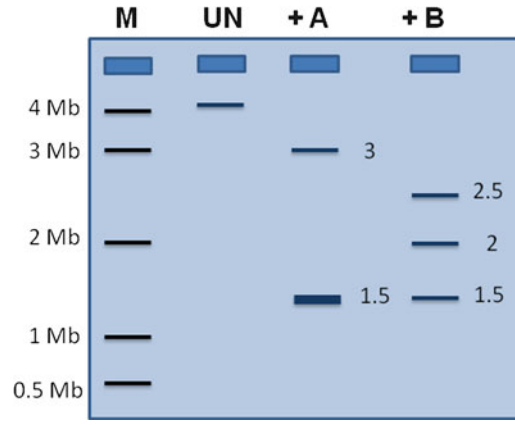

Fig. 1 Size estimation of a bacterial genome. PFGE analysis of undigested (indicated by *UN*), A-digested (+*A*), and B-digested (+*B*) genomic DNA. Note the higher band intensity of 1.5 Mb band in lane + A. *M* molecular marker. DNA separation can be obtained in a Gene Navigator® system (Amersham Pharmacia Biotech) using 1 % agarose gel in TBE 0.5×, pulse time 600″ × 24 h, 160 V, 12 °C

is advisable to use enzymes, which recognize relatively few sites on the genome and give a resolvable and informative number of DNA fragments on the PFGE gel. After staining of the gel, the size of the bacterial chromosome is consistently calculated from the sums of restriction fragment lengths. To get a more accurate determination of the genome size the use of different restriction enzymes is worthwhile. Figure 1 shows a schematic example, in which two enzymes A and B were used to determine the genome size of a microorganism. Enzyme A gave two bands, of 3 and 1.5 Mb, for a total of 4.5 Mb, while enzyme B three bands of 2.5, 2, and 1.5 Mb, for a total of 6 Mb. This size discrepancy is due to the fact that the enzyme A gave a 1.5 Mb band corresponding to two DNA fragments, evident by the higher intensity of this band. Thus, the genome size can be supposed to be 6 Mb.

In the case of GC-rich bacteria, like actinomycetes, enzymes that recognize specific base sequences rich in A and T nucleotides might be suitable for generating a distribution of DNA fragments that would be useful for analysis of genomic DNA, i.e., *AseI* (ATTAAT), *DraI* (TTTAAA), and *SspI* (AATATT); on the contrary, in the case of low GC bacteria, enzymes cutting sequences rich in G and C nucleotides are preferred, like *SmaI* (CCCGGG) and *NotI* (GCGGCCGC).

Another tough issue is to get a good resolution of all fragments in one track that requires optimal adjustment of the pulse time conditions and that sometimes cannot be obtained only in a run, but different runs, changing key parameters, will be needed to run. It is convenient to perform different runs optimizing electrophoretic conditions for separation in the low-, intermediate-, and

Table 2
Examples of running parameters to discriminate different DNA fragments

DNA size range (kb)	Pulse time	Run time (hours)	Voltage	Buffer	% Agarose
0.5–200	20″ + 4″	14 + 4 = 18	160	0.5× TBE	1
50–1,000	90″	30	200	0.5× TBE	0.8
150–2,200	70″ + 120″	15 + 11 = 26	200	0.5× TBE	0.8
200–3,000	200″ + 20″	20 + 4 = 24	160	0.5× TBE	1
200–5,000	500″	24	160	0.5× TBE	1

high-molecular-weight range. On the basis of the expected size of DNA fragments, different PFGE conditions can be applied. In Table 2 examples of run parameters (pulse time, run time, set voltage, gel strength, buffer), that we used for separation of large-molecular-weight DNA, are indicated. These parameters were used with Gene Navigator® system from Amersham Biosciences.

1.2 Construction of a Physical Map

Besides the utility of PFGE to determine microbial genome size, PFGE and restriction endonuclease digestion were used to construct physical maps, when genetic linkage maps could not be determined. After appropriate restriction of an intact genome and PFGE discrimination of restricted DNA, it is necessary to deduce the linkages between DNA fragments and various approaches can be applied. The most commonly used method is the hybridization of complete single or double digestions with gene probes, containing the rare-cutter site used to generate the digested sample (Fig. 2). A probe containing the restriction site will hybridize with two discrete DNA bands that correspond to adjacent DNA fragments along the chromosome. The example in Fig. 2 shows the hybridization signals of a blot of digested genomic DNA with two probes (p1 and p2), revealing that DNA fragments of 3 and 2 Mb are close, since both are positive to probe p1, while DNA fragments of 2 and 1 Mb are adjacent since both gave a hybridization signal with probe p2.

1.3 Genome Topology

By far the majority of bacterial genomes exist as a single circular chromosome, like most studied model bacteria, like *Escherichia coli* and *Bacillus subtilis.* Relatively recently, linear and/or multiple replicons were found also in many bacteria, i.e., linear chromosomes have been found in Streptomycetes, i.e., *Streptomyces coelicolor* [13], *S. lividans* [14], and *S. hygroscopicus* [15], while a mixture of linear and circular replicons were identified in *Borrelia burgdorferi* [16], *Agrobacterium tumefaciens* [17], *Rhodococcus fascians* [18], and related species.

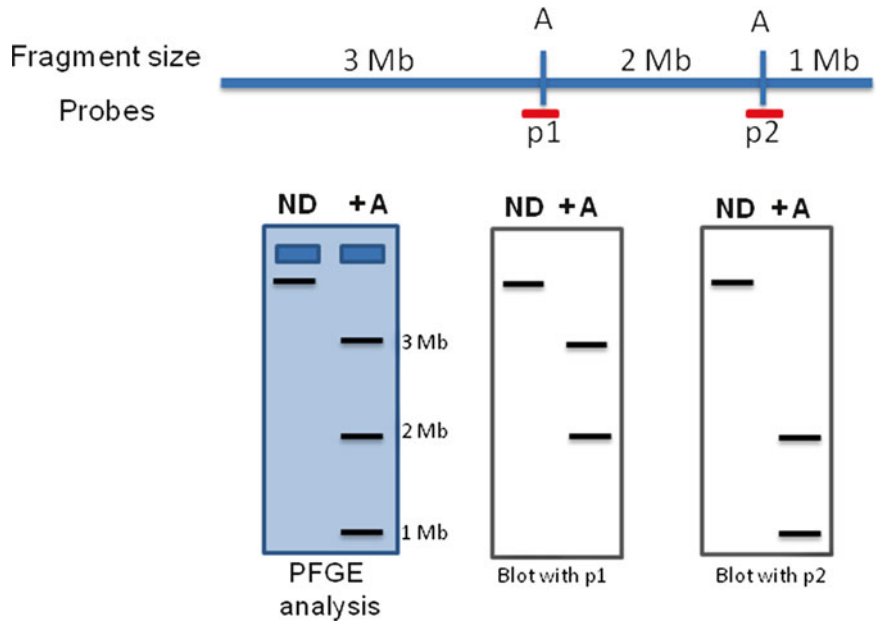

Fig. 2 Southern hybridization of complete digestion with known probes can be used to link adjacent clones. Probes p1 and p2, indicated by *red lines*, are necessary to recognize linked DNA fragments. *A* indicates the sites for the rare cutting restriction enzyme A. *M* molecular marker (color figure online)

However, linear chromosomes are kept as circular ones inside the cell because of covalently bound terminal proteins. Both naturally circular and protein-covalently bound linear chromosomes remain trapped in the slot and will not enter the gel and thus nothing other than the well is stained with ethidium bromide. To discriminate between these two different topologies, a straightforward procedure including proteinase K (PK) treatment can be applied.

PK treatment of a circular chromosome will not change its mobility into the gel, while, in the case of linear chromosomes kept circular by covalently bound terminal proteins, PK will cause the dissociation of the proteins, rendering the chromosome linear and able to enter the gel. To evaluate genome topology, genomic DNA, embedded in an agarose plug, is prepared using a procedure including proteinase K treatment. In parallel, two controls are usually performed: a plug is treated without PK, but with sodium dodecyl sulfate (SDS) to remove non-covalently bound proteins from DNA. Without PK treatment, the lysis might be incomplete or some binding proteins might still be present, thus retarding the mobility of the free chromosome and rendering it unable to enter the gel. The other control is performed incubating the plug of the same stock preparation with a restriction enzyme to generate several bands and to rule out the possibility of not having enough

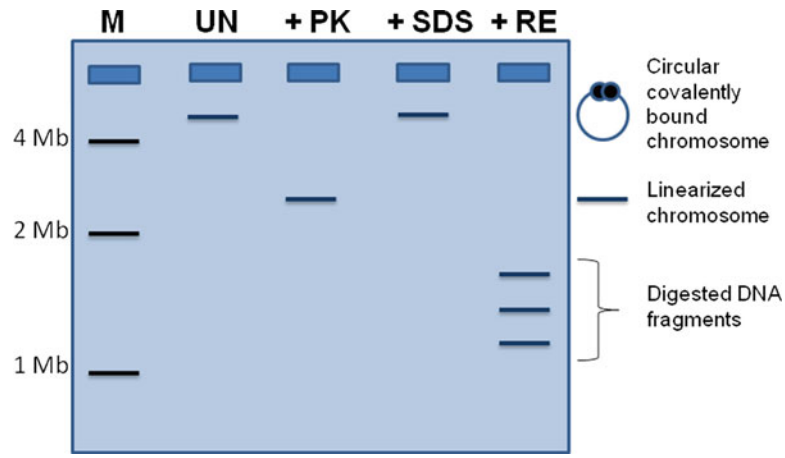

Fig. 3 Chromosome topology determined by incubating genomic DNA with PK, SDS, and a restriction enzyme. PK treatment (+PK) will make the chromosome linear, if it was kept circular by covalently bound proteins. Treatment with SDS buffer (+SDS) will remove non-covalently bound proteins that could interfere with DNA mobility. Digestion with a known restriction enzyme (+RE) will assure that enough DNA is present inside the plug. Not in scale. *M* molecular marker

DNA in the plug. Figure 3 summarizes the expected mobility of a linear chromosome kept circular inside the cell after incubation with PK, SDS, and restriction enzyme.

1.4 Multichromosomes or Megaplasmids?

In the last years it was demonstrated that bacteria can contain more than one chromosome (*Rhizobium*, *Burkholderia*, *Vibrio cholera*, *Borrelia burgdorferi*) and/or megaplasmids greater than 100 kb in size (*Streptomyces*, *Rhizobium*, *Agrobacterium*).

Megaplasmids have been described in a variety of microorganisms and many are responsible for distinctive and significant bacterial traits, including virulence, root nodulation, nitrogen fixation, antibiotic and heavy metal resistance, conjugation, and plant tumor induction.

A challenging test to distinguish if the smaller replicon(s) is a plasmid or a chromosome may be to consider whether the bacterium can grow without the second replicon. If yes, it is a plasmid that is commonly considered as accessory genetic material, not necessary for bacterial growth. Anyway, the elimination of the second replicon can be hard to obtain. Thus, a more straightforward method is to investigate if the second replicon contains genes encoding functions essential for bacterial metabolism that is indicative of a chromosome. Probes made from both 16S rRNA PCR products or metabolic genes can be used in hybridization experiments. The presence of 16S rDNA or metabolically essential genes, particularly if in a unique copy, is a strong proof that the replicon is a chromosome.

To determine the size of a plasmid, linear forms are preferred, in that they migrate at rates that allow size determination by comparison with linear markers. For the size determination of linear plasmid, a PK treatment will eliminate terminal covalently bounded proteins. Differently, circular megaplasmids with their closed-circular supercoiled forms move very slowly in PFGE and relaxed or nicked open-circular forms remain trapped in the sample wells. In addition, their migration depends upon running conditions and their size cannot be easily calculated. For an accurate determination of their sizes, one could perform plasmid purification away from the chromosomal DNA, selection of an appropriate restriction enzyme for digestion, and summation of the sizes of the resulting fragments after gel electrophoresis, but serious technical challenges are encountered when working with very large extrachromosomal DNA molecules.

To size circular megaplasmids, S1 nuclease treatment of DNA embedded in agarose plugs to convert the plasmids into unit-length linear molecules can be carried out and PFGE of the S1-treated plug can be performed [19]; indeed S1 nuclease first nicks the supercoiled plasmid DNA, and then it cuts the intact strand opposite to one of the nick, where the DNA actually is single stranded, resulting in a molecule of linearized plasmid DNA. Usually, treatment with SDS buffer of DNA embedded in agarose plugs is performed as control, to remove non-covalently bound proteins that could interfere with DNA mobility. Expected results are shown in Fig. 4.

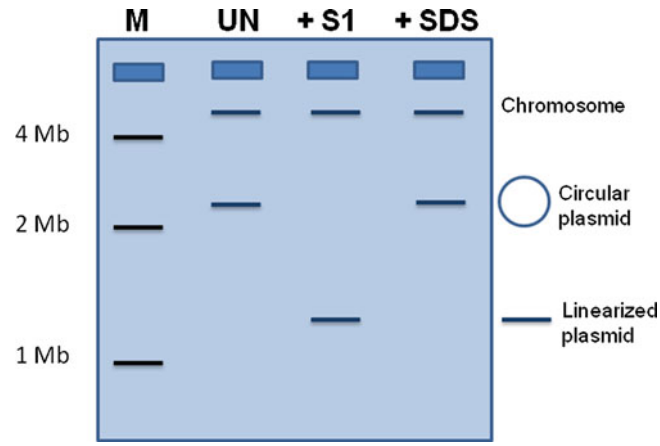

Fig. 4 Determination of size of a circular plasmid after treatment of DNA embedded in an agarose plug with S1 nuclease (+S1). Treatment with SDS buffer of DNA (+SDS) is performed as negative control. *UN* undigested DNA; *M* molecular marker

2 Materials

2.1 Solutions and Buffers for Plug Preparation

STE buffer: 10 mM Tris pH 8.0, 50 mM NaCl, 100 mM EDTA.

Lysis solution: 10 mM Tris pH 8.0, 50 mM NaCl, 100 mM EDTA, 0.2 % Na-deoxycholate (Sigma), 0.5 % sarkosyl (Sigma).

ESP buffer: 0.5 M EDTA pH 8.0, 1 % sarkosyl, 1 mg/ml proteinase K (added fresh).

TE buffer: 10 mM Tris pH 8.0, 100 mM EDTA.

NDS buffer: 0.5 M EDTA pH 8.0, 1 % sodium lauroyl sarcosine.

PMSF: 40 mg phenyl methyl sulfonate in 1 ml isopropanol.

SDS buffer: 2 % SDS in 0.5 M EDTA, pH 8.0.

2.2 Running Buffer and Agarose Gel

10× TBE: 108 g Tris base, 55 g boric acid, 9.3 g disodium EDTA $2H_2O$, water to 1 l.

10× TAE: 242 g Tris base, 57.1 ml glacial acetic acid, 100 ml 0.5 M EDTA pH 8.0, water to 1 l.

Staining solution: 0.5× TBE containing 1 μg/ml EtBr.

Destaining solution: 0.5× TBE or distilled water.

3 Methods

To perform PFGE analysis, particular care in preparing high-molecular-weight DNA is necessary. Large-molecular-weight DNA has to be handled with extreme care, so it is normally prepared by embedding the cells in agarose prior to solubilization and enzymatic digestion of the non-DNA components. Individual cells are embedded in agarose, which protects the DNA against breakage while allowing the free flow of solutions necessary for lysis and digestion. High concentrations of EDTA are used to inhibit nuclease activity in the presence of Proteinase K that will digest cellular proteins. Material released by this digestion diffuses out of the agarose during the washes while the DNA remains trapped. DNA prepared in agarose is stable and remains available as a substrate for enzymatic restriction.

3.1 Preparation of DNA Embedded in Agarose Plugs

1. Grow cells in 10 ml rich medium to mid or late log phase (*see* **Note 1**).
2. Harvest cells by spinning at 400–4,000 × *g* for 10 min at 4 °C (*see* **Note 2**).
   - Heat the pellet for 20 min at 75 °C in case of virulent strains.
3. Wash twice cell pellet by resuspending in 2–4 ml of 10 % glycerol, decant the supernatant off very carefully, and recentrifuge at 400–4,000 × *g* for 10–30 min (*see* **Note 3**).

4. Resuspend the cell pellet in one-fifth the original culture volume (2 ml) of STE buffer (*see* **Note 4**).
5. Prepare molten 1.6 % low-melting-point agarose made in 1× TE, pH 8 (*see* **Note 5**) and keep it in a warm bath at 45 °C to avoid premature gelification.
6. Mix 600 µl cells with 600 µl molten low-melting-point agarose (*see* **Note 6**).
7. Pipette well to mix, and then add 100 µl of the suspension cells/agarose to disposable plug moulds. Let the agarose harden on ice for 10–20 min (*see* **Note 7**).
8. Push plugs into 10 ml of lysis solution (*see* **Note 8**).
9. Incubate at 37 °C with gentle agitation for 2–4 h.
10. Remove the lysis solution and transfer the plugs to new tubes containing 10 ml of ESP buffer.
11. Incubate for 1–2 days at 50 °C with gentle agitation (*see* **Note 9**).
12. Add 50 µl of 0.1 M PMSF, mix gently, and place the tube on ice for 1 h. PMSF destroys residual Proteinase K in the plugs (*see* **Note 10**).
13. Wash plugs three to four times in 20 ml of TE buffer at 4 °C for 30 min.
14. Store 10–12 plugs in 10 ml of NDS buffer (*see* **Note 11**).

3.2 Genomic Treatment with Enzymes

1. Wash the number of the plugs you need in TE buffer (2 ml per plug) at 4 °C for 1 h to overnight. The last wash can be done with sterile water.
2. Put one plug per a 1 ml microcentrifuge tube.
3. Digest the plugs in 1× buffer with 20–30 U of enzyme for 4 h to overnight (*see* **Note 12**). If S1-PFGE is carried out, treat total DNA embedded in agarose gel plug with 20 U of S1 nuclease and separate the DNA by pulsed field gel electrophoresis.
4. Stop the reaction by adding 1 ml of 50 mM EDTA (pH 8.0) or by directly loading the samples in PFGE apparatus.

3.3 Gel Preparation

Gels are cast and prepared using the same conditions and reagents used for conventional electrophoresis, but they are usually prepared without ethidium bromide and are stained after the run; this is due to the large volume of buffer that is used and that should be discarded later, and to the fact that intercalation of ethidium bromide slows DNA migration.

1. Add the desired amount of agarose to the correct amount of electrophoresis buffer (*see* **Note 13**). 0.5× TBE buffer (Tris–borate–EDTA) and 1× TAE buffer (Tris–acetate–EDTA) are the two buffers most frequently used for PFGE (*see* **Note 14**).
2. Heat the flask to boiling in a microwave oven. Avoid boilover (*see* **Note 15**).

3. Cool agarose to 40–50 °C before pouring (*see* **Note 16**).
4. Prepare the gel casting mould with the appropriate comb.
5. Pour delicately the agarose solution into the rubber casting frame, supplied with the apparatus (*see* **Note 17**). Leave a few ml of agarose solution for sealing the wells in the next step.
6. Remove very delicately the comb.

3.4 Gel Loading

Samples prepared in agarose plugs are loaded before the gel is placed in the chamber and the wells are sealed with the left agarose, prepared and used for the gel, to avoid their escape from the wells and floating in the running buffer.

1. Prepare a working area by placing some parafilm over the bench and providing a clean scalpel and a clean needle (*see* **Note 18**).
2. Decide the order of the samples. Do not forget an appropriate size marker (*see* **Note 19**).
3. Let the plug sliding from the microcentrifuge tube to some parafilm, take the plug with the scalpel, removing the excess of liquid, and let the plug sliding from the scalpel to the well; if necessary, softly push the plug into the well (*see* **Note 20**).
4. Seal the wells with the agarose left and wait till it hardens.
5. Remove the rubber casting frame (*see* **Note 21**) and transfer the gel, solidified into the plastic tray, to the gel chamber, filled with the cold running buffer (*see* **Note 22**).
6. Insert the electrode in the right position and close the lid of the electrophoretic chamber (*see* **Note 23**).
7. Connect the electrodes, balance the electrophoresis chamber, switch on the pump (*see* **Note 24**), and start the run (*see* **Note 25**).
8. After the run, take the tray containing the gel and put carefully on the bench. Push the gel to one side and let the gel slide to a glass plate bigger than the gel.
9. Put the glass plate with the gel in a staining solution (*see* **Note 26**) and incubate for 30 min to ON (*see* **Note 27**).
10. Destain for 1 h in 0.5× TBE (*see* **Note 28**).
11. Pump old buffer out from electrophoresis chamber. Rinse with ca. 2.5 l MilliQ water.

4 Notes

1. Standard cell OD ensures that each sample contains approximately the same amount of DNA. OD_{600} of 0.6–1 gives usually good DNA quality. For Gram positive, glycine to a final

concentration of 0.2 M is usually added to the growth medium to facilitate following cell wall degradation. Use the appropriate volume of culture on the basis of the plugs you need. Volumes from 5 to 25 ml are suggested.

2. If DNA quality is not good enough, a smearing will be visible in the absence of incubation with a restriction enzyme. Try to harvest cells earlier. Some bacteria, like actinomycetes, produce a lot of nucleases; to get good-quality DNA, it is suggested to preheat the cells at 65 °C or to reduce nucleases by using a phenol/chloroform treatment.
3. Centrifugation gravity depends upon the kind of bacterial cells.
4. It is more convenient to use plugs at three different DNA concentrations. If the samples are too concentrated, DNA will be difficult to be completely digested and analyzed. If this is the case, try to use half plug.
5. Use high-quality pulsed field gel electrophoresis (PFGE)-grade agarose. Make sure that agarose is completely melted; use a microwave and pulse it in short bursts, but do not boil it over. Discard expired agarose.
6. Cell pellets can be also kept at 45 °C. Other percentages of agarose and other ratios of agarose/cells can be used. Usually, final 0.8–1 % of agarose allows to easily handle plugs. A lower agarose concentration can cause breakage of the plug.
7. Other methods of pouring agar plugs, such as using plastic syringes as moulds, can be used. When non-disposable moulds are used, before pouring agarose suspension, close them on the bottom with paper tape. After gel solidification, remove delicately the tape. Wash the moulds in 0.2 % SDS for 1–2 h and rinse with water.
8. Use 50 ml conical screw-cap tubes and put 10–12 plugs per 10 ml of lysis solution. In case of Gram-positive bacteria, lysozyme (1 mg/ml) is added. For *Staphylococcus aureus*, the incubation with 50 μg/ml of lysostaphin is preferred. Consider that more resistant cell walls need stronger treatment, for example 1 % Triton X-100 or 1 % SDS, to render the bacteria more susceptible to lysis. RNase A (DNase free) can be added at 10 mg/ml.
9. NDS treatment for 48 h is suggested for actinomycetes.
10. PMSF treatment can be avoided, but this could inhibit downstream restriction analysis.
11. If the plugs will be used soon, let two plugs in TE buffer, so their analysis will require fewer washes before restriction.
12. A total volume of 160 μl of liquid keeps the plug submerged. Consider that the volume of the plug is 100 μl, so that the total volume is 260 μl. Pay attention if the enzyme has star effect, i.e., *DraI*, or if it works better at less than 37 °C.

13. 0.8–1 % gel is usually used. Use only glassware, combs, and gel forms that are clean.
14. 0.5× TBE is the most commonly used buffer; it does not need to be changed, even over multi-day runs; 1× TAE buffer is more useful when separating megabase-sized DNA fragments (>3 Mb). Use high-quality water to make the 0.5× TBE buffer used for the gel and running buffer. Some bacteria have a fragile DNA that undergoes DNA degradation in the presence of Tris-containing buffer. In this case, Hepes-containing buffer can be used.
15. Adjust the volume with the buffer after boiling. Make sure that the agarose is uniformly melted by swirling the flask. Pay attention: Overheated solutions can boil and over suddenly when swirled.
16. Too hot temperature can cause leaking of the agarose solution and can weaken or distort the casting mould. We pour the gel when we can keep the flask by hands.
17. Remove air bubbles, lint, dust, and visible particulates from the gel before it solidifies. Pay attention on how to prepare the gel cast. Every cast has only a way to be mounted. If you are wrong, the run will not start.
18. We use the tip of a disposable inoculating loop and needle.
19. Different markers exist in the market. It is advisable to load the border lanes with the same marker.
20. Avoid bubbles in loading plugs; a clean needle can be helpful.
21. We remove the rubber cast after loading the samples, so that the gel is more stable.
22. Cold buffer restricts premature cell lysis and subsequent DNA degradation. If running buffer is stored in cold room, allow to stand at room temperature for ca. 1 h before adding to electrophoresis chamber. 2.8 l of buffer is usually enough to cover the gel. Switch on the chiller. Temperature of 12 °C is usually used.
23. If electrodes are wet or wrongly positioned or the buffer is insufficient in the electrophoresis chamber, the run will not start. Try to disassemble and reassemble the electrodes and the lid and make sure that the buffer covers the gel. Otherwise add more buffer.
24. Ensure that the pump is working. Otherwise, your run will be unsuccessful.
25. Examples of run conditions are shown in Table 2, but many factors, such as voltage, switch interval, running time, agarose concentration of the gel, running temperature, running buffer, and angle of the alternating electric field, affect DNA migration in PFGE gels, so that different experimental attempts can be necessary.

26. The running buffer can be used for preparation of the staining solution.
27. We recycle the staining solution 3-4 times to reduce the ethidium bromide-containing waste.
28. Destaining can be done more quickly with distilled water.

References

1. Schwartz DC, Saffran W, Welsh J, Haas R, Goldenberg M, Cantor CR (1983) New techniques for purifying large DNAs and studying their properties and packaging. Cold Spring Harbor Symp Quant Biol 47:189–195.
2. Vollrath D, Davis RW (1987) Isolation of DNA molecules greater than 5 megabases by contour-clamped homogeneous electric fields. Nucleic Acids Res 15:7865–7876
3. Lai E, Birren BW, Clark SM, Simon MI, Hood L (1989) Pulsed-field gel electrophoresis. Biotech 7:34–42
4. Lai E, Birren BW (1995) Use of secondary pulsed field gel electrophoresis in separation of large DNA. Anal Biochem 1(224):68–74
5. Basim H (2001) Pulsed-field gel electrophoresis (PFGE) technique and its use in molecular biology. Turk J Biol 25:405–418
6. López-Madrigal S, Latorre A, Porcar M, Moya A, Gil R (2011) Complete genome sequence of "Candidatus Tremblaya princeps" strain PCVAL, an intriguing translational machine below the living-cell status. J Bacteriol 193:5587–5588
7. Han K, Li ZF, Peng R, Zhu LP, Zhou T, Wang LG, Li SG, Zhang XB, Hu W, Wu ZH, Qin N, Li YZ (2013) Extraordinary expansion of a *Sorangium cellulosum* genome from an alkaline milieu. Sci Rep 3:2101
8. Chang YJ, Land M, Hauser L, Chertkov O, Del Rio TG, Nolan M, Copeland A, Tice H, Cheng JF, Lucas S, Han C, Goodwin L, Pitluck S, Ivanova N, Ovchinikova G, Pati A, Chen A, Palaniappan K, Mavromatis K, Liolios K, Brettin T, Fiebig A, Rohde M, Abt B, Göker M, Detter JC, Woyke T, Bristow J, Eisen JA, Markowitz V (2011) Non-contiguous finished genome sequence and contextual data of the filamentous soil bacterium *Ktedonobacter racemifer* type strain (SOSP1-21 T). Stand Genomic Sci 5:97–111
9. Schneiker S, Perlova O, Kaiser O, Gerth K, Alici A, Altmeyer MO, Bartels D, Bekel T, Beyer S, Bode E, Bode HB, Bolten CJ, Choudhuri JV, Doss S, Elnakady YA, Frank B, Gaigalat L, Goesmann A, Groeger C, Gross F, Jelsbak L, Jelsbak L, Kalinowski J, Kegler C, Knauber T, Konietzny S, Kopp M, Krause L, Krug D, Linke B, Mahmud T, Martinez-Arias R, McHardy AC, Merai M, Meyer F, Mormann S, Muñoz-Dorado J, Perez J, Pradella S, Rachid S, Raddatz G, Rosenau F, Rückert C, Sasse F, Scharfe M, Schuster SC, Suen G, Treuner-Lange A, Velicer GJ, Vorhölter FJ, Weissman KJ, Welch RD, Wenzel SC, Whitworth DE, Wilhelm S, Wittmann C, Blöcker H, Pühler A, Müller R (2007) Complete genome sequence of the myxobacterium *Sorangium cellulosum*. Nat Biotechnol 25:1281–1289
10. Wang XJ, Yan YJ, Zhang B, An J, Wang JJ, Tian J, Jiang L, Chen YH, Huang SX, Yin M, Zhang J, Gao AL, Liu CX, Zhu ZX, Xiang WS (2010) Genome sequence of the Milbemycin-producing bacterium *Streptomyces bingchenggensis*. J Bacteriol 192:4526–4527
11. Copeland A, Lapidus A, Glavina Del Rio T, Nolan M, Lucas S, Chen F, Tice H, Cheng JF, Bruce D, Goodwin L, Pitluck S, Mikhailova N, Pati A, Ivanova N, Mavromatis K, Chen A, Palaniappan K, Chain P, Land M, Hauser L, Chang YJ, Jeffries CD, Chertkov O, Brettin T, Detter JC, Han C, Ali Z, Tindall BJ, Göker M, Bristow J, Eisen JA, Markowitz V, Hugenholtz P, Kyrpides NC, Klenk HP (2009) Complete genome sequence of *Catenulispora acidiphila* type strain (ID 139908 T). Stand Genomic Sci 1:119–125
12. Nolan M, Sikorski J, Jando M, Lucas S, Lapidus A, Glavina Del Rio T, Chen F, Tice H, Pitluck S, Cheng JF, Chertkov O, Sims D, Meincke L, Brettin T, Han C, Detter JC, Bruce D, Goodwin L, Land M, Hauser L, Chang YJ, Jeffries CD, Ivanova N, Mavromatis K, Mikhailova N, Chen A, Palaniappan K, Chain P, Rohde M, Göker M, Bristow J, Eisen JA, Markowitz V, Hugenholtz P, Kyrpides NC, Klenk HP (2010) Complete genome sequence of *Streptosporangium roseum* type strain (NI 9100 T). Stand Genomic Sci 2:29–37
13. Kieser HM, Kieser T, Hopwood DA (1992) A combined genetic and physical map of the *Streptomyces coelicolor* A3(2) chromosome. J Bacteriol 174:5496–5507
14. Leblond P, Redenbach M, Cullum J (1993) Physical map of the *Streptomyces lividans* 66 genome and comparison with that of the

related strain *Streptomyces coelicolor* A3(2). J Bacteriol 175:3422–3429

15. Pang X, Zhou X, Sun Y, Deng Z (2002) Physical map of the linear chromosome of Streptomyces hygroscopicus 10-22 deduced by analysis of overlapping large chromosomal deletions. J Bacteriol 184:1958–1965
16. Fraser CM, Casjens S, Huang WM, Sutton GG, Clayton R, Lathigra R, White O, Ketchum KA, Dodson R, Hickey EK, Gwinn M, Dougherty B, Tomb JF, Fleischmann RD, Richardson D, Peterson J, Kerlavage AR, Quackenbush J, Salzberg S, Hanson M, van Vugt R, Palmer N, Adams MD, Gocayne J, Weidman J, Utterback T, Watthey L, McDonald L, Artiach P, Bowman C, Garland S, Fuji C, Cotton MD, Horst K, Roberts K, Hatch B, Smith HO, Venter JC (1997) Genomic sequence of a Lyme disease spirochaete, *Borrelia burgdorferi*. Nature 11:580–586
17. Allardet-Servent A, Michaux-Charachon S, Jumas-Bilak E, Karayan L, Ramuz M (1993) Presence of one linear and one circular chromosome in the *Agrobacterium tumefaciens* C58 genome. J Bacteriol 175:7869–7874
18. Pisabarro A, Correia A, Martín JF (1998) Pulsed-field gel electrophoresis analysis of the genome of Rhodococcus fascians: genome size and linear and circular replicon composition in virulent and avirulent strains. Curr Microbiol 36:302–308
19. Barton BM, Harding GP, Zuccarelli AJ (1995) A general method for detecting and sizing large plasmids. Anal Biochem 226:235–240

Chapter 2

Comparative Analyses of Extrachromosomal Bacterial Replicons, Identification of Chromids, and Experimental Evaluation of Their Indispensability

Lukasz Dziewit and Dariusz Bartosik

Abstract

Bacterial genomic information can be divided between various replicons, including chromosomes, plasmids, and chromids (essential plasmid-like replicons with properties of both chromosomes and plasmids). Comparative analyses of bacterial plasmids, including homology searches, phylogenetic and phylogenomic analyses, as well as network construction for the characterization of their relationships, are good starting points for the identification of chromids. Chromids possess several chromosome-like genetic features (e.g., codon usage, GC content), but most significantly, they carry housekeeping genes, which make them indispensable for cell viability. However, it is important to confirm in silico predictions experimentally. The essential nature of a predicted chromid is usually verified by the application of a target-oriented replicon curing technique, based on the incompatibility phenomenon. Further tests examining growth in various media are used to distinguish secondary chromids from plasmids, and mutational analysis (e.g., using the yeast FLP/FRT recombination system) is employed to identify essential genes carried by particular chromids.

Key words Extrachromosomal bacterial replicon, Plasmid, Chromid, Comparative genomics, Target-oriented replicon curing technique, Growth assay, Mutational analysis

1 Introduction

The sequencing of bacterial genomes has revealed that many have multipartite structures. They often contain numerous extrachromosomal replicons, including well-characterized plasmids and also chromids, a newly distinguished group of indispensable replicons, sharing features of both plasmids and chromosomes [1].

The main characteristics of chromids are (a) their considerable size, (b, c) the presence of plasmid-type replication systems and adaptive genes typical for plasmids that are useful in particular ecological niches, (d, e) a G+C content and codon usage similar to those of the host chromosome, and most importantly, (f) the presence of housekeeping genes of chromosomal origin.

Alessio Mengoni et al. (eds.), *Bacterial Pangenomics: Methods and Protocols*, Methods in Molecular Biology, vol. 1231, DOI 10.1007/978-1-4939-1720-4_2,

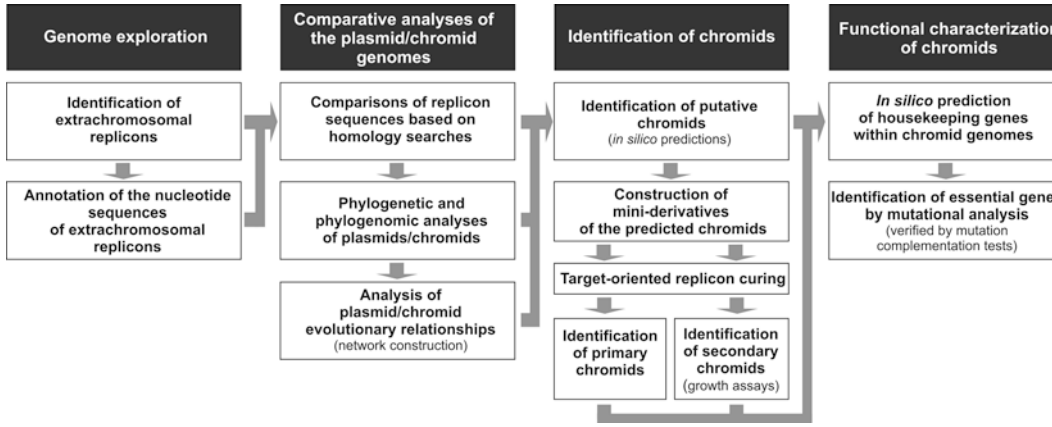

Fig. 1 Scheme of the protocol for the identification and analyses of chromids

The chromosome-like features indicate the long-term co-evolution of chromids and chromosomes, and the presence of essential housekeeping genes explains the indispensability of these replicons. However, due to the low density of housekeeping genes in chromid genomes, their identification is not simple (e.g., [2]).

Two types of chromids can be distinguished by functional analysis: (a) "primary" chromids, which (similarly to chromosomes) are necessary for host viability, and (b) "secondary" chromids, that are required for survival in the natural environment, but are dispensable under optimal laboratory conditions (e.g., they may carry genetic information enabling growth of the host strain in minimal media) [2].

Here, we propose a protocol for the identification and characterization of both types of chromids (Fig. 1). The starting point is comparative in silico analyses of extrachromosomal bacterial replicons, including homology searches, phylogenetic and phylogenomic analyses, as well as network construction for the characterization of relationships. The identification of putative chromids can also be based on the detection of the aforementioned chromosome-like features, and the presence of essential housekeeping genes, which may be identified using several bioinformatic tools.

The experimental verification of in silico predictions primarily involves the target-oriented replicon curing technique, which is based on the construction of a shuttle plasmid carrying the replication system of the analyzed extrachromosomal replicon. The obtained plasmid is introduced into the host strain to remove (by incompatibility) the natural replicon in question (plasmid or chromid). In this way, "indispensable" primary chromids, that are necessary for cell viability, can be distinguished from "dispensable" replicons, i.e., plasmids and secondary chromids. Secondary chromids may then be characterized by growth tests in various minimal and rich media. Further investigations are focused on defining the genes responsible for the essential nature

of primary and secondary chromids. For this purpose, mutational analysis (e.g., generation of deletions using the yeast FLP/FRT recombination system [3]) is performed and the essential genes are identified by complementation.

2 Materials

Bacterial strain and plasmids used in particular experiments are presented in Table 1.

2.1 Triparental Mating

1. Donor strain, e.g., *E. coli* DH5α, containing a shuttle plasmid carrying the replication system of the analyzed replicon.
2. Recipient strain of choice. This strain should be kanamycin-sensitive if using vector pABW1, and carry a marker such as rifampin resistance to assist the selection of transconjugants.
3. *E. coli* DH5α carrying helper plasmid pRK2013.
4. Luria Bertani (LB) broth and agar media.
5. Appropriate antibiotics for supplementation of LB media.

2.2 DNA Manipulations, Visualization, and Introducing Plasmid DNA into E. coli Cells

1. Polymerase for PCR (e.g., *Phusion* High-Fidelity DNA polymerase) and appropriate reagents (dNTPs, buffer, oligonucleotide primers).
2. Restriction enzymes and appropriate buffers.
3. T4 DNA ligase and reaction buffer.
4. Reagents for the standard alkaline lysis DNA extraction procedure [4].

Table 1
Bacterial strains and plasmids

Strain and plasmids	Genotype	Reference
Strain		
Escherichia coli DH5α	F$^-$, Φ80d *lacZ*ΔM15 (*lacZY A-orgF*) U169 *deoR recA1 endA1 hsdR17 phoA supE44* λ^- *thi1 gyrA96 relA1*	[39]
Plasmids		
pABW1	Km^r, *oriV* pMB1, *oriT* RK2	[38]
pBBR1	Broad host range plasmid originated from *Bordetella bronchiseptica*	[40]
pJQFRT	Gm^r, *sacB*, *oriV* p15A, *oriT*, FRT	[3]
pKFRT/FLP	Km^r, *tetR*, *flp*, *oriV* ColE1, *oriT*, FRT	[3]
pRK2013	Km^r, helper plasmid carrying RK2 *tra* genes	[41]

Km kanamycin, *Gm* gentamicin

5. Reagents for the chemical transformation of bacterial cells [5].
6. Reagents for the in-gel cell lysis and DNA electrophoresis procedure [6, 7].

2.3 General Growth Assays

1. Luria Bertani (LB) broth or any other rich medium.
2. Minimal salts medium, e.g., AC [8, 9] [Na_2HPO_4—6.15 g/l, KH_2PO_4—1.5 g/l, NaOH—9.2 g/l, NH_4Cl—0.4 g/l, $MgSO_4 \times 7H_2O$—0.19 g/l supplemented with 10 ml/l Tuovinen's salts (mixture of microelements) and an appropriate carbon source, e.g., 0.2 % arabinose] or any other minimal medium.

2.4 Mutational Analysis Using the FLP/FRT Recombination System

1. Luria Bertani (LB) broth or any other rich medium.
2. Donor strain *E. coli* DH5α containing an appropriate pJQFRT or pKFRT/FLP derivative.
3. Recipient strain of choice. This strain should be kanamycin- and gentamicin-sensitive, and carry a marker such as rifampin resistance to assist the selection of transconjugants.
4. *E. coli* DH5α carrying helper plasmid pRK2013.
5. Antibiotics: kanamycin, gentamicin, anhydrotetracycline.
6. Sucrose.
7. Polymerase for PCR (e.g., *Taq* DNA polymerase) and appropriate reagents.
8. Restriction enzymes and appropriate buffers.
9. T4 DNA ligase and reaction buffer.
10. Reagents for the standard alkaline lysis DNA extraction procedure [4].
11. Oligonucleotide primers: FRT-leftF—5′-AATCCATCTTGTTCAATCATGC-3′ and FRT-SP6R—5′-TACGATTTAGGTGACACTATA-3′ [3].

3 Methods

This section describes in detail a step-by-step protocol (Fig. 1) for the identification and characterization of bacterial chromids. The bioinformatic tools specified are those that we use routinely in our research, but other programs performing similar functions are available. Bacteria of the class *Alphaproteobacteria*, which commonly contain chromids (e.g., [10, 2]), are used as a model when describing some of the techniques.

3.1 Comparative Genomic Analyses of Extrachromosomal Bacterial Replicons

1. Annotation of the nucleotide sequences of extrachromosomal bacterial replicons can sometimes be problematic, especially when the replicons originate from a poorly studied group of bacteria lacking appropriate well-annotated reference plasmid or chromid genomes.

   Fully manual sequence annotation using the Artemis tool [11] (*see* **Note 1**), for example, is the most accurate annotation method, but it can be extremely time consuming. Alternatively, there are several good pipelines for the automatic annotation of bacterial genomes, which can also be applied to the sequences of plasmids and chromids, e.g., RAST Annotation Server [12] or GenDB [13]. However, automatic annotation should always be manually verified by the means of BLAST programs [14] and the PRIAM tool [15].

2. Comparative genomic analyses of plasmids/chromids provide important information concerning the diversity and plasticity of these replicons. Such analyses, based on nucleotide or protein sequence homology searches, usually employ the BLAST or BLAT algorithms [14]. BLAT is a BLAST-like pairwise sequence alignment algorithm designed to reduce the time required to align multiple sequences. BLAT analyses may be performed with the GeneOrder4.0 tool [16].

   Following comparative sequence analyses, the next step is the annotation and high quality visualization of the obtained data. There are numerous tools that may be used for the preparation of publication quality figures, but we have experience with the following programs: Easyfig, a Python application for creating linear comparison figures of genomes with an easy-to-use graphical interface [17]; MAUVE, a tool for the identification and alignment of conserved genomic DNA in the presence of various rearrangements [18]; ACT, the Artemis Comparison Tool, which allows interactive visualization of genome comparisons generated by NCBI-BLASTN, NCBI-TBLASTX, or MUMmer [11]; and M-GCAT, a tool that efficiently constructs multiple genome comparison frameworks, especially in closely related species [19].

   There are also bioinformatic tools that permit more "quantitative" comparisons of plasmid/chromid genome sequences. To identify the core genes of a set of plasmids or chromids, a Venn diagram may be created showing all possible relationships between a finite collection of sets. The Venn diagram schemes can be drawn using programs such as the VennDiagram R-package [20] or more automatically by application of the EDGAR tool [21].

3. Phylogenetic and phylogenomic analyses of extrachromosomal bacterial replicons enable their classification into groups and provide information about their reciprocal relationships.

Traditionally, the classification of plasmids has been based on their incompatibility behavior (closely related plasmids are incompatible, i.e., they cannot stably coexist in the same bacterial cell). PCR-based typing is often used for the identification of plasmids containing particular replication systems, but such methods are inappropriate for novel or deviant plasmid types, due to their great sequence diversity [22]. In this case, phylogeny-based classification schemes are more appropriate.

For the characterization of extrachromosomal bacterial replicons, various gene/protein sequences can be used as the basis for the construction of phylogenetic trees. The most commonly used amino acid sequences are those of replication initiation proteins and relaxases involved in conjugal transfer (e.g., [23, 22]). Phylogenetic analyses are conveniently performed using the "user-friendly" MEGA package (current version MEGA6) [24] (*see* **Note 2**).

In some cases, a phylogenomic approach may be used. This employs the sequences of a set of genes (proteins) instead of just a single one. Although the phylogenomic approach better reflects relationships between replicons, its use is limited to the analyses of closely related genes encoding orthologous proteins. Therefore, this approach is usually used when analyzing whole bacterial genomes. The EDGAR tool is a fully automated bioinformatic pipeline enabling such analyses [21]. It can differentiate core genes and singletons, and permits the reconstruction of the phylogenetic trees of replicons (or genomes). The core genes from all the analyzed genomes are used to produce such trees. Multiple alignments created using MUSCLE [25] are automatically "cleaned" of badly aligned regions using GBLOCKS [26]. The remaining parts of all the alignments are concatenated and the resulting multiple alignment is used to generate the phylogenetic tree using PHYLIP [27].

4. The evolutionary relationships of plasmids/chromids can also be analyzed by the application of gene-sharing networks. A bioinformatic tool suitable for this purpose is Blast2Network, which creates networks representing all the sequence identities/similarities existing among the proteins encoded within the analyzed plasmids or chromids. Each node in these networks represents a particular protein, whereas links indicate the existence of sequence identity/similarity between proteins [28, 29]. Visualization of the network clustering and gene sharing amongst plasmids/chromids can be achieved using the program Circos [30], or its online representation Circoletto [31].

3.2 In Silico Identification of Chromids

1. The presence of chromosome-like genetic features (codon usage and GC content) can be used for the identification of chromids in the course of in silico analyses of bacterial genomes

(*see* **Note 3**). Putative chromids usually possess a GC content similar to that of the host chromosome, with a cut-off value not more than ±2 % (usually about 0.5 %) [1]. The second parameter used for chromid identification is their relative synonymous codon usage (RSCU), which is a measure of codon bias calculated as the ratio of the observed frequency of a particular codon to the frequency expected for a synonymous codon group with uniform codon usage [32]. The RSCU may be calculated using the freely available CAIcal SERVER [33].

$$\mathrm{RSCU}_i = \frac{X_i}{\frac{1}{n}\sum_{i=1}^{n} X_i}$$

n = number of synonymous codons ($1 \le n \le 6$) for the studied amino acid, X_i = number of occurrences of codon i.

2. The presence of many orthologous gene pairs placed within a chromosome and co-residing extrachromosomal replicon may be strong evidence for chromid identification. It has also been shown that a large number of genes are conserved within chromids of bacteria of the same genus [1]. Tools for the identification of orthologous gene pairs include OrthoMCL [34], which utilizes an all-against-all BLASTP algorithm, and the aforementioned GeneOrder4.0 tool [16].
3. In silico predictions of essential housekeeping genes within chromids can be used to establish a replicon's nature. Such genes may be predicted by applying several bioinformatic tools including DEG (Database of Essential Genes) [35] (*see* **Note 4**) and KEGG (Kyoto Encyclopedia of Genes and Genomes) [36]. These analyses require advanced microbiological knowledge to be able to evaluate which enzyme is essential for host viability. After the identification of crucial genes within a studied chromid, it is necessary to verify whether there is an additional copy of the particular gene in the genome or if a bypass pathway for the chromid-encoded metabolic process is present.

3.3 Identification of Primary Chromids Using the Target-Oriented Replicon Curing Technique

The replicon curing technique is based on the incompatibility phenomenon. This method requires the construction of a mini-derivative of the studied replicon, i.e., a shuttle plasmid containing the replication system. Target-oriented replicon curing generates appropriate replicon-less derivatives in order to distinguish between plasmids and primary chromids. Under certain conditions, the incompatibility phenomenon also enables the removal of secondary chromids from a cell, so further functional analysis is required to distinguish these from plasmids [2].

The presented approach assumes that the nucleotide sequence of the analyzed replicon is known.

1. Perform complex in silico sequence analyses of the predicted chromid to distinguish its replication (REP) and partitioning (PAR) modules containing incompatibility determinants.
2. Design oligonucleotide primers and amplify DNA fragments carrying the REP and PAR modules from a chromid DNA-containing template by PCR (*see* **Note 5**).
3. Purify the amplified DNA fragment and digest it with appropriate restriction enzymes to facilitate cloning.
4. Linearize a narrow host range vector using the same restriction enzymes used to cleave the PCR product (*see* **Note 6**).
5. Ligate the linear vector with the PCR product using T4 DNA ligase.
6. Prepare chemically competent *Escherichia coli* DH5α cells using a standard procedure (e.g., [5]) and transform with the ligation mixture.
7. Identify colonies containing recombinant constructs (mini-derivatives of the analyzed replicon) using appropriate selection, e.g., blue–white screening when using pABW1 for *Alphaproteobacteria*.
8. Sequence the cloned replication system fragment to verify that it does not contain any mutations introduced during PCR amplification (*see* **Note 7**).
9. Introduce the mini-derivative shuttle plasmid into the recipient strain via triparental mating [37]. Overnight cultures of the donor strain *E. coli* DH5α carrying the (mobilizable) mini-derivative, the recipient strain (host from which the given REP system originated), and *E. coli* DH5α carrying the helper plasmid pRK2013 are grown. The cells are harvested by centrifugation and then washed twice to remove antibiotics. Cell suspensions of the donor, host, and helper strains are then mixed in a 1:2:1 ratio and 100 μl of this mixture is spread onto a plate of LB agar medium. After overnight incubation at an appropriate temperature, bacteria are washed from the plate and suitable dilutions of the cell suspension are plated on selective medium containing rifampin (selection for the recipient strain) and kanamycin (selection for the shuttle plasmid) (*see* **Notes 8–10**).
10. Verify the presence of an autonomous form of the introduced shuttle plasmid within the obtained transconjugants (*see* **Note 11**).
11. Confirm the presence of the analyzed plasmid/chromid using the in-gel cell lysis/DNA electrophoresis procedure [6, 7] (allows the visualization of mega-sized replicons).

12. If the analyzed replicon cannot be removed, it is highly probable that it is a primary chromid, necessary for cell viability. If it is readily removed from the host cells, it is either a "dispensable" plasmid or a secondary chromid and further investigations are required to distinguish between these two possibilities.

3.4 General Growth Assays for the Identification of Secondary Chromids

Secondary chromids are only "facultatively" essential, so additional analyses are required to differentiate them from plasmids. They may contain genetic information that enables the host strain to grow in minimal media, i.e., in conditions similar to those in the natural environment. A simple growth assay may be performed to test whether a plasmid/chromid-less strain is able to grow in minimal media.

1. Obtain a strain deprived of the analyzed replicon by incompatibility (as described in Subheading 3.3).
2. Inoculate rich liquid medium, in which growth is optimal (e.g., LB medium), with the replicon-deficient strain and the wild-type strain as a control, and grow overnight (*see* **Note 12**).
3. Harvest the cells by centrifugation and wash twice with minimal medium appropriate for the tested strain, e.g., AC minimal salts medium for *Alphaproteobacteria*.
4. Prepare portions of minimal medium supplemented with different carbon compounds appropriate for the host strain and then inoculate with the cell suspensions to an initial optical density at 600 nm (OD_{600}) of 0.05 and grow cultures.
5. Monitor the growth rate by measuring the OD_{600} of the cultures at 12- or 24-h intervals (for *Alphaproteobacteria*) for a further 72 h.
6. Determine viable cell counts by plating dilutions of samples taken every 12 or 24 h on plates of LB agar or another rich medium.
7. The growth of strains deprived of a secondary chromid is significantly reduced or even completely inhibited in minimal medium (*see* **Note 13**).

3.5 Identification of Genes Responsible for the Indispensability of Chromids: Mutational Analysis

Various approaches involving mutational analysis may be employed to identify essential genes of primary and secondary chromids. To verify in silico predictions of the indispensability of a single gene, a simple gene-replacement technique resulting in an antibiotic resistance-marked mutation can be applied. However, this approach is not suitable when larger DNA regions of chromids have to be examined for the presence of housekeeping genes. In this case, deletion analysis using the yeast FLP/FRT recombination system [3] can be used.

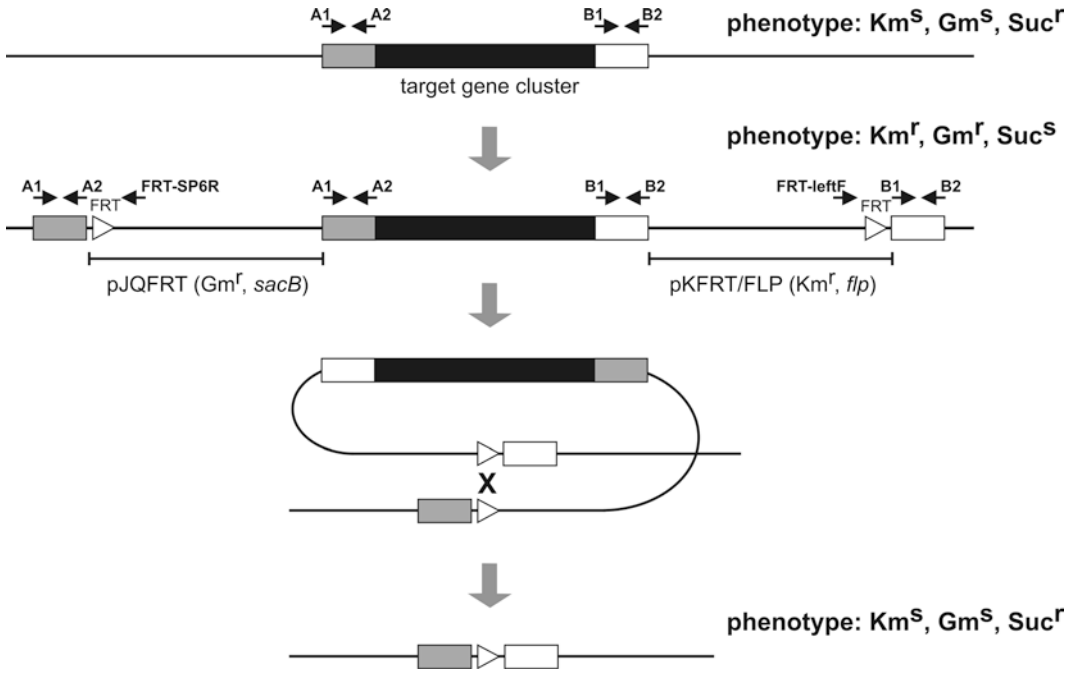

Fig. 2 Scheme of mutational analysis using the FLP/FRT recombination system. *Gray* and *white rectangles* represent upstream and downstream regions flanking the target gene cluster (*black rectangle*), respectively. *White arrow heads* represent FRT recombination sites. *Black arrows* indicate the location of PCR primers. *X* indicates the site of homologous recombination. *Km* kanamycin, *Gm* gentamicin, *Suc* sucrose

This unmarked mutagenesis technique is useful for the excision of target gene clusters that can exceed 100 kb. The procedure involves the integration (via homologous recombination) of plasmids on both flanks of the target gene cluster. The two plasmids contain different antibiotic resistance markers (Gm and Km). One carries the *sacB* gene encoding levansucrase (an enzyme whose activity in the presence of sucrose leads to the accumulation of toxic compounds in the bacterial cell, causing a lethal effect), and the other carries the Flp recombinase gene under the control of the *tetR* regulator (induced with anhydrotetracycline). Both plasmids also contain an FRT recombination site. Induction of Flp recombinase expression causes excision of the target gene cluster (together with accompanying plasmid sequences) from the chromid, which results in an unmarked mutation (Fig. 2) [3].

1. Design oligonucleotide primers and amplify DNA regions of about 0.5–1 kb each from both sides of the targeted gene cluster from a chromid DNA template by PCR (primer pairs A1/A2 and B1/B2 in Fig. 2).
2. Purify the amplified DNA fragments and digest with appropriate restriction enzymes to facilitate their cloning.
3. Linearize plasmids pJQFRT and pKFRT/FLP using the same restriction enzymes used previously to cleave the PCR products.

4. Ligate linear pJQFRT and pKFRT/FLP with the PCR products representing the regions upstream and downstream of the target gene cluster, respectively, using T4 DNA ligase.
5. Prepare chemically competent *Escherichia coli* DH5α cells using a standard procedure (e.g., [5]) and transform with the ligation mixtures.
6. Use colony PCR to identify clones containing recombinant constructs (suicide gene-replacement plasmids) (*see* **Note 14**).
7. Introduce the obtained derivative of plasmid pJQFRT into the recipient strain via triparental mating [37], as previously described in Subheading 3.3, using gentamicin as a selection marker for transconjugants (*see* **Note 15**).
8. Confirm the nature of the integration of the pJQFRT derivative by PCR using primer FRT-SP6R and a forward primer complementary to the upstream region of the plasmid integration site (primer A1, Fig. 2).
9. Introduce the pKFRT/FLP derivative into the obtained pJQFRT derivative-containing strain via triparental mating [37] using gentamicin and kanamycin as the selection markers for transconjugants (*see* **Note 15**).
10. Confirm the nature of the integration of the plasmid by PCR using primer FRT-leftF and a reverse primer complementary to the downstream region of the plasmid integration site (primer B2, Fig. 2).
11. Inoculate LB medium supplemented with gentamicin and kanamycin with the obtained strain and incubate at the appropriate temperature overnight.
12. Dilute the culture (1:100) in LB medium (without antibiotics) and grow to mid-logarithmic phase.
13. Add anhydrotetracycline to a final concentration of 400 ng/ml.
14. After 6 h of induction with anhydrotetracycline spread 100 μl of the culture onto a plate of LB agar medium supplemented with 50 mg/ml (i.e., 5 %) sucrose and incubate overnight.
15. Test the resulting sucrose-resistant colonies for their susceptibility to gentamicin and kanamycin by replica plating. The ability to obtain clones susceptible to gentamicin and kanamycin indicates that the analyzed DNA region can be deleted from the genome of the predicted chromid. When analyzing secondary chromids, a pool of strains containing deletion mutants (obtained in a rich medium) should be tested for their ability to grow in minimal media (see Subheading 3.4). Growth inhibition of certain clones points to the loss of conditionally essential genes (*see* **Note 16**).

In the case of primary chromids, the deletion of DNA segments containing housekeeping genes is lethal for the host strain, and therefore it will not be possible to obtain such mutants (*see* **Note 17**). To confirm the presence of essential genes within a given DNA fragment, further complementation analysis should be performed. This requires cloning of the DNA segment (or particular genes) to be deleted, in a broad host range vector (e.g., derived from pBBR1) and the introduction of this plasmid into the strain described in Subheading 3.5 (point 10). In the presence of the complementation plasmid, it should be possible to detect anhydrotetracycline-induced deletion. Further trimming of the cloned fragment should permit identification of the essential genes.

4 Notes

1. When the identification of plasmid/chromid open reading frames is based on amino acid sequence homology, we suggest using a less stringent cut-off for *e*-values (e.g., *e*-value < 1×10^{-5}), since extrachromosomal replicons are much more variable than chromosomes, and some significant hits may be lost.
2. A multiple sequence alignment has to be prepared for phylogenetic analysis (e.g., using MUSCLE). Normally, such alignments are manually corrected, but in the case of large-scale phylogenetic analyses, detailed inspection of the alignment quality is impractical. In such cases, programs like GBLOCKS, which mask the nonmatching parts of alignments, can be applied.
3. These parameters cannot be treated as ultimate determinants for the classification of a particular replicon as either a plasmid or a chromid.
4. Chromid identification using DEG can be equivocal and requires careful manual verification.
5. Use a high-fidelity polymerase to reduce amplification errors.
6. For *Alphaproteobacteria* a good choice is either pABW1 [38] or a suicide vector carrying the R6K replication origin [requires the *trans*-encoded *pir* gene product (Pi protein) to function].
7. When using pABW1 it is possible to sequence the insert using M13 universal oligonucleotide primers [M13 (-21)—5′-TGTA AAACGACGGCCAGT-3′ and M13 reverse—5′-CAGGAAA CAGCTATGACC-3′].
8. Sometimes a lack of transconjugants may be due to the presence of toxin-antitoxin (TA) or restriction-modification (R-M) modules within the analyzed plasmid/chromid. Both modules are stabilization systems, responsible for post-segregational elimination of replicon-less cells from the bacterial population. Therefore, the inability to obtain clones deprived of a replicon (containing TA or R-M) may erroneously suggest its indispensability, which is a

typical feature of primary chromids. To eliminate this problem, TA and RM modules of the tested replicon should be included within the shuttle plasmid used for the incompatibility analysis.

9. When analyzing replicons of bacteria other than *Alphaproteobacteria* (e.g., of the family *Enterobacteriaceae*; *Gammaproteobacteria*) biparental conjugal mating is recommended. This omits the use of the helper plasmid pRK2013 (functional in members of the *Enterobacteriaceae*). A strain such as *E. coli* S17-1, which contains a chromosomally encoded transfer system of plasmid RK2, may be used as a helper strain for biparental mating. This approach also enables the use of suicide vectors carrying the R6K replication origin (e.g., pDS132) for the construction of shuttle plasmids (*E. coli* S17-1 encodes the Pi protein of R6K, required for initiation of replication of these plasmids).
10. The mating procedure should be optimized for each particular strain, e.g., for some bacteria, mating occurs in a liquid environment, so the procedure requires the use of liquid media.
11. Sometimes the introduced plasmid may form a cointegrate with the analyzed replicon as a result of homologous recombination or a transposition event.
12. It is important to monitor the growth rate of the cultures in rich medium. Sometimes strains deprived of a secondary chromid may grow significantly more slowly.
13. The decrease in the growth rate of strains deprived of secondary chromids in minimal media occurs irrespective of the carbon source used in the experiment.
14. Plasmids pJQFRT and pKFRT/FLP carry the p15A and ColE1 replication systems, respectively, which are highly specific for *Enterobacteriaceae*. Therefore, they act as suicide vectors in other Gram-negative bacteria, including *Alphaproteobacteria*.
15. The introduced plasmid cannot replicate in the recipient strain; therefore gentamicin enables selection of clones in which the plasmid has integrated into the analyzed chromid via homologous recombination. The expected frequency of such events is very low. Alternatively, biparental conjugation using a strain such as *E. coli* S17-1 can be employed.
16. The genes responsible for the conditional indispensability of secondary chromids may be identified by mutation complementation tests. This requires cloning of the predicted conditionally essential genes and their introduction into the mutant strain. If the wild-type phenotype (the ability to grow on minimal media) is restored, then the particular gene is deemed essential for this trait.
17. The loss of the DNA fragment can be additionally confirmed by PCR using appropriate primers or by DNA–DNA hybridization analysis.

Acknowledgement

This work was supported by the National Science Centre, Poland (grant 2013/09/B/NZ1/00133).

References

1. Harrison PW, Lower RP, Kim NK et al (2010) Introducing the bacterial 'chromid': not a chromosome, not a plasmid. Trends Microbiol 18:141–148
2. Dziewit L, Czarnecki J, Wibberg D et al (2014) Architecture and functions of a multipartite genome of the methylotrophic bacterium *Paracoccus aminophilus* JCM 7686, containing primary and secondary chromids. BMC Genomics 15:124
3. Ishikawa M, Hori K (2013) A new simple method for introducing an unmarked mutation into a large gene of non-competent Gram-negative bacteria by FLP/FRT recombination. BMC Microbiol 13:86
4. Sambrook J, Russell DW (2001) Molecular cloning: a laboratory manual. Cold Spring Harbor Laboratory, New York, NY
5. Kushner SR (1978) An improved method for transformation of *E. coli* with ColE1 derived plasmids. In: Boyer HB, Nicosia S (eds) Genetic engineering. Elsevier/North-Holland, Amsterdam, pp 17–23
6. Eckhardt T (1978) A rapid method for the identification of plasmid desoxyribonucleic acid in bacteria. Plasmid 1:584–588
7. Wheatcroft R, McRae GD, Miller RW (1990) Changes in the *Rhizobium meliloti* genome and the ability to detect supercoiled plasmids during bacteroid development. Mol Plant-Microbe Interact 3:9–17
8. Tuovinen OH, Kelly DP (1973) Studies on the growth of *Thiobacillus ferrooxidans*. I. Use of membrane filters and ferrous iron agar to determine viable numbers, and comparison with 14 CO 2 -fixation and iron oxidation as measures of growth. Arch Mikrobiol 88:285–298
9. Wood AP, Kelly DP (1977) Heterotrophic growth of *Thiobacillus* A2 on sugars and organic acids. Arch Microbiol 113:257–264
10. Petersen J, Frank O, Goker M et al (2013) Extrachromosomal, extraordinary and essential–the plasmids of the Roseobacter clade. Appl Microbiol Biotechnol 97:2805–2815
11. Carver T, Berriman M, Tivey A et al (2008) Artemis and ACT: viewing, annotating and comparing sequences stored in a relational database. Bioinformatics 24:2672–2676
12. Aziz RK, Bartels D, Best AA et al (2008) The RAST server: rapid annotations using subsystems technology. BMC Genomics 9:75
13. Meyer F, Goesmann A, McHardy AC et al (2003) GenDB – an open source genome annotation system for prokaryote genomes. Nucleic Acids Res 31:2187–2195
14. Altschul SF, Madden TL, Schaffer AA et al (1997) Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. Nucleic Acids Res 25:3389–3402
15. Claudel-Renard C, Chevalet C, Faraut T et al (2003) Enzyme-specific profiles for genome annotation: PRIAM. Nucleic Acids Res 31: 6633–6639
16. Mahadevan P, Seto D (2010) Rapid pair-wise synteny analysis of large bacterial genomes using web-based GeneOrder4.0. BMC Res. Notes 3, 41
17. Sullivan MJ, Petty NK, Beatson SA (2011) Easyfig: a genome comparison visualizer. Bioinformatics 27:1009–1010
18. Darling AC, Mau B, Blattner FR et al (2004) Mauve: multiple alignment of conserved genomic sequence with rearrangements. Genome Res 14:1394–1403
19. Treangen TJ, Messeguer X (2006) M-GCAT: interactively and efficiently constructing large-scale multiple genome comparison frameworks in closely related species. BMC Bioinformatics 7:433
20. Chen H, Boutros PC (2011) VennDiagram: a package for the generation of highly-customizable Venn and Euler diagrams in R. BMC Bioinformatics 12:35
21. Blom J, Albaum SP, Doppmeier D et al (2009) EDGAR: a software framework for the comparative analysis of prokaryotic genomes. BMC Bioinformatics 10:154
22. Petersen J, Brinkmann H, Berger M et al (2011) Origin and evolution of a novel DnaA-like plasmid replication type in *Rhodobacterales*. Mol Biol Evol 28:1229–1240
23. Garcillan-Barcia MP, Francia MV, de la Cruz F (2009) The diversity of conjugative relaxases and its application in plasmid classification. FEMS Microbiol Rev 33:657–687

24. Tamura K, Stecher G, Peterson D et al (2013) MEGA6: molecular evolutionary genetics analysis version 6.0. Mol Biol Evol 30:2725–2729
25. Edgar RC (2004) MUSCLE: multiple sequence alignment with high accuracy and high throughput. Nucleic Acids Res 32:1792–1797
26. Talavera G, Castresana J (2007) Improvement of phylogenies after removing divergent and ambiguously aligned blocks from protein sequence alignments. Syst Biol 56:564–577
27. Felsenstein J (1989) PHYLIP – phylogeny inference package (version 3.2). Cladistics 5:164–166
28. Fondi M, Bacci G, Brilli M et al (2010) Exploring the evolutionary dynamics of plasmids: the *Acinetobacter* pan-plasmidome. BMC Evol Biol 10:59
29. Tamminen M, Virta M, Fani R et al (2012) Large-scale analysis of plasmid relationships through gene-sharing networks. Mol Biol Evol 29:1225–1240
30. Krzywinski M, Schein J, Birol I et al (2009) Circos: an information aesthetic for comparative genomics. Genome Res 19:1639–1645
31. Darzentas N (2010) Circoletto: visualizing sequence similarity with Circos. Bioinformatics 26:2620–2621
32. Sharp PM, Li WH (1986) Codon usage in regulatory genes in *Escherichia coli* does not reflect selection for 'rare' codons. Nucleic Acids Res 14:7737–7749
33. Puigbo P, Bravo IG, Garcia-Vallve S (2008) CAIcal: a combined set of tools to assess codon usage adaptation. Biol Direct 3:38
34. Li L, Stoeckert CJ Jr, Roos DS (2003) OrthoMCL: identification of ortholog groups for eukaryotic genomes. Genome Res 13: 2178–2189
35. Luo H, Lin Y, Gao F et al (2013) DEG 10, an update of the database of essential genes that includes both protein-coding genes and noncoding genomic elements. Nucleic Acids Res 42:D574–D580
36. Kanehisa M, Goto S, Hattori M et al (2006) From genomics to chemical genomics: new developments in KEGG. Nucleic Acids Res 34:D354–D357
37. Bartosik D, Szymanik M, Wysocka E (2001) Identification of the partitioning site within the repABC-type replicon of the composite *Paracoccus versutus* plasmid pTAV1. J Bacteriol 183:6234–6243
38. Bartosik D, Bialkowska A, Baj J et al (1997) Construction of mobilizable cloning vectors derived from pBGS18 and their application for analysis of replicator region of a pTAV202 mini-derivative of *Paracoccus versutus* pTAV1 plasmid. Acta Microbiol Pol 46:387–392
39. Hanahan D (1983) Studies on transformation of *Escherichia coli* with plasmids. J Mol Biol 166:557–580
40. Antoine R, Locht C (1992) Isolation and molecular characterization of a novel broad-host-range plasmid from *Bordetella bronchiseptica* with sequence similarities to plasmids from gram-positive organisms. Mol Microbiol 6:1785–1799
41. Ditta G, Stanfield S, Corbin D et al (1980) Broad host range DNA cloning system for gram-negative bacteria: construction of a gene bank of *Rhizobium meliloti*. Proc Natl Acad Sci U S A 77:7347–7351

Chapter 3

Choice of Next-Generation Sequencing Pipelines

F. Del Chierico, M. Ancora, M. Marcacci, C. Cammà, L. Putignani, and Salvatore Conti

Abstract

The next-generation sequencing (NGS) technologies are revolutionary tools which have made possible achieving remarkable advances in genetics since the beginning of the twenty-first century. Thanks to the possibility to produce large amount of sequence data, these tools are going to completely substitute other high-throughput technologies. Moreover, the large applications of NGS protocols are increasing the genetic decoding of biological systems through studies of genome anatomy and gene mapping, coupled to the transcriptome pictures. The application of NGS pipelines such as (1) de-novo genomic sequencing by mate-paired and whole-genome shotgun strategies; (2) specific gene sequencing on large bacterial communities; and (3) RNA-seq methods including whole transcriptome sequencing and Serial Analysis of Gene Expression (Sage-analysis) are fundamental in the genome-wide fields like metagenomics. Recently, the availability of these advanced protocols has allowed to overcome the usual sequencing technical issues related to the mapping specificity over standard shotgun library sequencing, the detection of large structural genomes variations and bridging sequencing gaps, as well as more precise gene annotation. In this chapter we will discuss how to manage a successful NGS pipeline from the planning of sequencing projects through the choice of the platforms up to the data analysis management.

Key words NGS, Metagenomics, Whole-genome sequencing, 16S rRNA gene, Gene mapping, RNA-seq, Library preparation, Template preparation, NGS platforms

1 Introduction

All NGS experimental approaches forecast a quite similar experimental protocol composed of sample collection and nucleic acid extraction, followed by typical next NGS steps, shared by several NGS platforms: (1) library and template preparation; (2) sequencing protocols completed by genome alignment and read assembly during the data analysis (Fig. 1). Experimental designs are mainly modulated by sample collection and ad hoc library preparation. The metagenomics is the "deep" study of the genetic material of several microorganisms (metagenome) directly recovered from complex and environmental samples [1–5]. Metagenomic analysis is characterized by different challenges than those typical of

Alessio Mengoni et al. (eds.), *Bacterial Pangenomics: Methods and Protocols*, Methods in Molecular Biology, vol. 1231, DOI 10.1007/978-1-4939-1720-4_3,

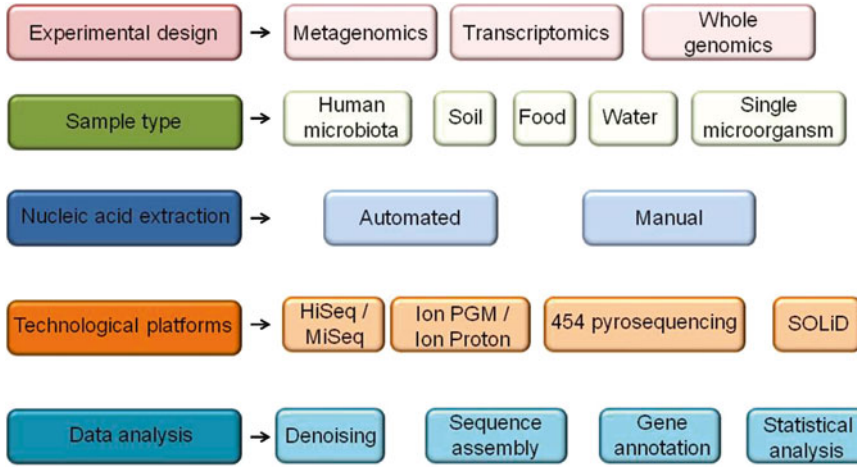

Fig. 1 First experimental design for NGS applications. Depending on the purpose of the experiment (e.g., metagenomics, transcriptomics, or whole genomics), NGS pipelines can differ in the different steps. The chosen pipeline can better fit with specific sample collection protocol or with different technological platforms up to the final data analysis

singular species genome studies, as microbiome samples contain thousands of species [4]. The main issue arising from conventional bacterial genome sequencing, often microbial isolation, is the loss of biodiversity due to cultivation-based methods, while Whole-Genome NGS technologies (WG-NGS) are able to get largely unbiased information of all genes from all the members of bacterial communities by employing different DNA and RNA sequencing methods, joined to the availability of fast and reliable bioinformatic tools [6, 7]. For instance, to get the best genome assembly, especially from metagenome materials, the most suitable NGS pipeline is the "mate-paired sequencing," which provides a particular library preparation protocol [9]. Fast whole-genome sequencing of clinically relevant organisms is possible but the final assembly inevitably contains gaps in the sequence [7]. Shotgun fragment library data can be augmented with mate-paired library data to produce a high quality assembled sequence from large contigs and only few scaffolds [6–8, 12]. The basic sequencing pipeline for low complex samples or unique species is composed of (1) sample isolation; (2) DNA extraction; and (3) shotgun fragment library sequencing, a simple protocol that may be sufficient to assemble bacterial genomes depending on read length [5, 7]. Many applications find utility in the use of this simple pipeline such as rapid whole-genome sequencing for detection and characterization of microorganisms directly from clinical samples (e.g., studies on evolution of antibiotic resistance in vivo) [13]. Other approaches such as the targeted sequencing of the 16S rRNA gene is one of the genetic tools to produce diversity profiling in natural samples by using a shotgun

NGS approach. This targeted gene application finds a wide range of applications, including the characterization of bacterial populations belonging to human gut-, airflow-, skin-microbiota [1–4, 10], and to other biological systems like bioremediation environment of contaminated water [11]. NGS strategies may differ depending on NGS platforms of choice [5] (Fig. 2a), because each platform presents different features and advantages, consistently with the final targets. All these platforms propose different library preparation protocols and sequencing outputs (e.g., number and length of the reads), characteristics that can be suitable for different sequencing strategies and purposes. In this chapter we show the main available NGS pipelines, depending on the sequencing need, from the single bacterial genome assembly to the decoding of complex metagenomic materials and transcriptomes.

2 Evaluation of NGS Platforms

Each NGS platform presents suitable and different sequencing features and metrics depending on the chosen sequencing strategy. For instance, the 454 Flex Lifescience Roche provides the longest read performance, reaching up to 1,000 bp for single read, an attribute rightly appropriate with de novo sequencing by mate-paired and shotgun strategies for complex samples like metagenomes [4–6]. Other NGS platforms, like Illumina Genome Analyzer and Hiseq series, make available a throughput with the highest number of reads with a range from 250 million up to 2 billion, allowing a huge multiplex sequencing of several complex samples [6, 15]. In the last 2 years the Ion Torrent by Life Technologies Company has launched the NGS planet with the "PGM" and "Proton" sequencers, with a chemistry based on the pH detection due to nucleotide incorporation during the DNA strand extension (Fig. 3). These systems are breaking out in the NGS scenery thanks to their sequencing properties, like scalability and speed, well appropriate in the world of bacterial genomics application [1–3, 7–14]. The sequencing workflow for each NGS platform consists of three principal steps: (1) library preparation; (2) template preparation; and (3) sequencing reaction (Fig. 2a). The NGS platforms, based on single molecule sequencing technology, not discussed in this chapter, exclude the template preparation step. For the step (1), library preparation, several procedures are available and though all platforms share the same library applications for DNA or RNA sequencing, these can provide different advantages. For instance, regarding the sequencing of complex genomes, the usual NGS library used is the mate-paired one and the available platforms propose different protocols discussed in the next paragraph. All NGS library protocols produce a complex library composed of double strand DNA fragments with different adapters linked to their ends [16, 17].

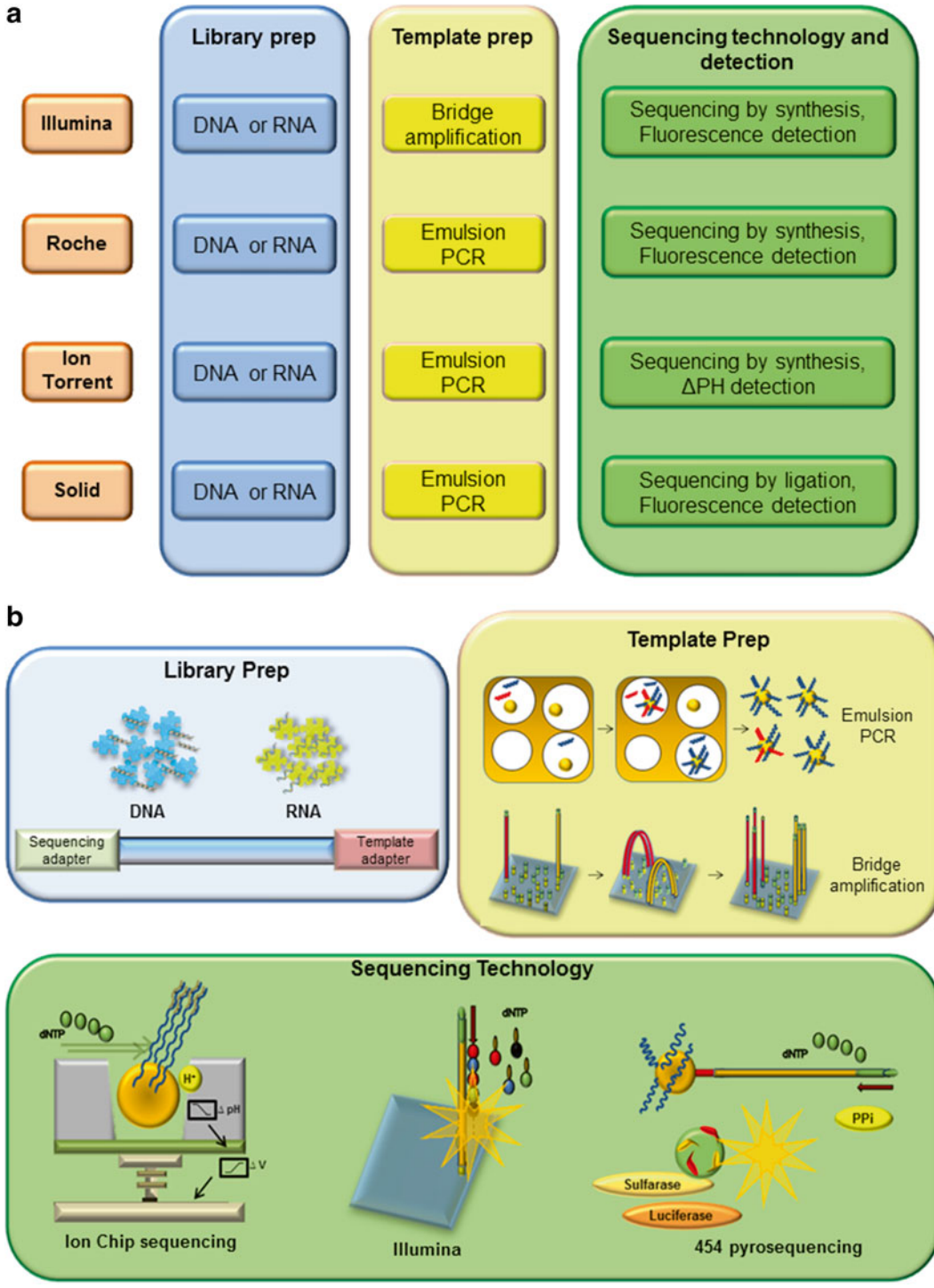

Fig. 2 Principal NGS platforms from library preparation to sequencing technology. (**a**) All NGS platforms make use of ePCR in order to carry on the clonal amplification, except for the Illumina Solexa technology performing this step by the bridge amplification method. Illumina, Roche, and Ion Torrent platforms produce sequences from template by DNA chain synthesis, while Solid platform by a ligation technology. The usual base recognition, as a result of dNTPs incorporation or ligation, is achieved by fluorescent detection. Ion Torrent technology does not make use of any optical system owing to the detection of pH decreasing during dNTPs incorporation.

These adapters will be functional for the next (2) and (3) NGS steps (Fig. 2b). The template preparation step provides the clonal amplification of each fragment in order to get a subsequent strong signal during the sequencing reaction. As shown in Fig. 2, Solid, Ion torrent, and Roche systems share the same template strategies that make use of the emulsion PCR (ePCR) in order to get clonal (i.e., monoclonal and polyclonal) sphere particles. Regarding ePCR, the Solexa technology, by Illumina systems, makes use of the "Bridge Amplification" on a solid surface as a flow cell, in order to get clonal amplification. This flow cell with linked amplified library fragments is directly useful for the sequencing step carried out by the chemistry of the fluorescent reversible terminator. The last step of sequencing reaction provides the elongation of DNA chain, obtained by synthesis for Illumina, Roche, and Ion torrent and by ligation for Solid system. The NGS platforms require an optical system for the detection of incorporated nucleotide except for the Ion Torrent system that directly detects the ΔpH during the DNA extension over the ion semiconductor chip (Figs. 2b and 3).

All these NGS platforms provide several instruments with a large throughput range from the bench top machines (Ion Torrent® PGM, Roche® 454 Junior, Illumina® Miseq), suitable for targets with low complexity, to the highest throughput instruments like Solid 5500 and Illumina Hiseq series, for complex metagenomes that require a high number of reads and different methodological and data analysis approaches such as mate-paired sequencing and positional contig data assembly.

3 Whole-Genome Sequencing by "Mate-Paired" and "Shotgun" Sequencing

The bacterial sequencing can match with a very hard challenge given that metagenomes can be composed of hundreds of bacterial species. Getting a very high number of short reads by "shotgun" sequencing may not be enough extensive to solve the right genomic composition for each bacterium [6]. To simplify this issue, the "mate-paired" sequencing proposes a specific sequencing pipelines aiming to increase the mapping specificity over standard fragment

Fig. 2 (continued) (**b**) Images show each step of the NGS workflows. Every library preparation protocol provides final double strand DNA fragments with linked adaptors at both ends. In the template step, ePCR millions of spheres are covered by each library fragment; in the bridge amplification, each fragment is amplified on a solid flow cell with the aim of getting "clusters" useful for the next sequencing detection. The three different sequencing technologies are shown on the *bottom*. Illumina and 454 Roche platforms detect dNTPs incorporation by fluorescence releasing: the first one by the reversible terminator chemistry and the latter by the pyrosequencing cascade reaction. The Ion Torrent technology detects directly the hydrogen ion release, turning it in a voltage signal by a semiconductor technology

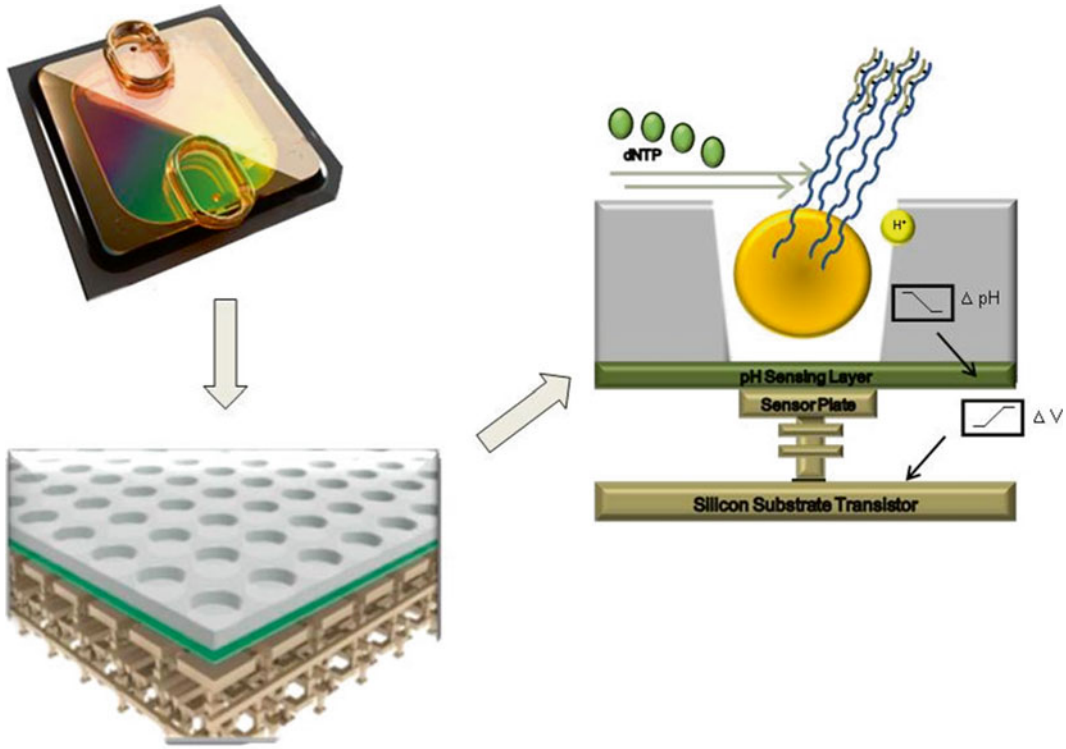

Fig. 3 Ion Torrent semiconductor technology. On the *right top side* is the PI chip of the Ion Torrent Proton instrument followed by a chip cross section. The cross section shows the Ion semiconductor structure composed of a microfluidic top layer able to lodge the ISPs from ePCR covered by amplified fragments of library. Under this level there is a *green* layer sensitive to the ΔpH that transmits the signal to the underlying *brown* transistor able to convert it in a ΔVoltage signal. The image on the *right* shows the dNTPs flow in a chip well with an ISP releasing hydrogen ion from each DNA fragments

library, to detect large structural variations in the genome, and to bridge sequencing gaps [18]. Figure 4 represents the mate-paired library preparation from Solid and Ion Torrent strategies [9, 20]. Such a library consists of pairs of DNA fragments that are "mates" arising from the two ends of the same genomic DNA fragment [9, 18, 19]. The two mate-paired adaptors, linked to both DNA fragment ends (e.g., 3–10 kb of size), form an internal adaptor, connecting the DNA mate-paired together, after intermolecular DNA circularization, in a very dilute DNA solution. This specific protocol involves a circularization method different from that used in other NGS platform [9, 20]. In fact, the resulting DNA circle has one nick in each strand, due to the missing of the 5′ phosphate in oligonucleotides internal adaptor sequence. Nick translation, using *Escherichia coli* DNA polymerase I, "pushes" the nick into the genomic DNA region in the 5′–3′ direction. The length of nick-translated DNA can be controlled by adjusting temperature and time of the reaction. In this way the method can provide fragments

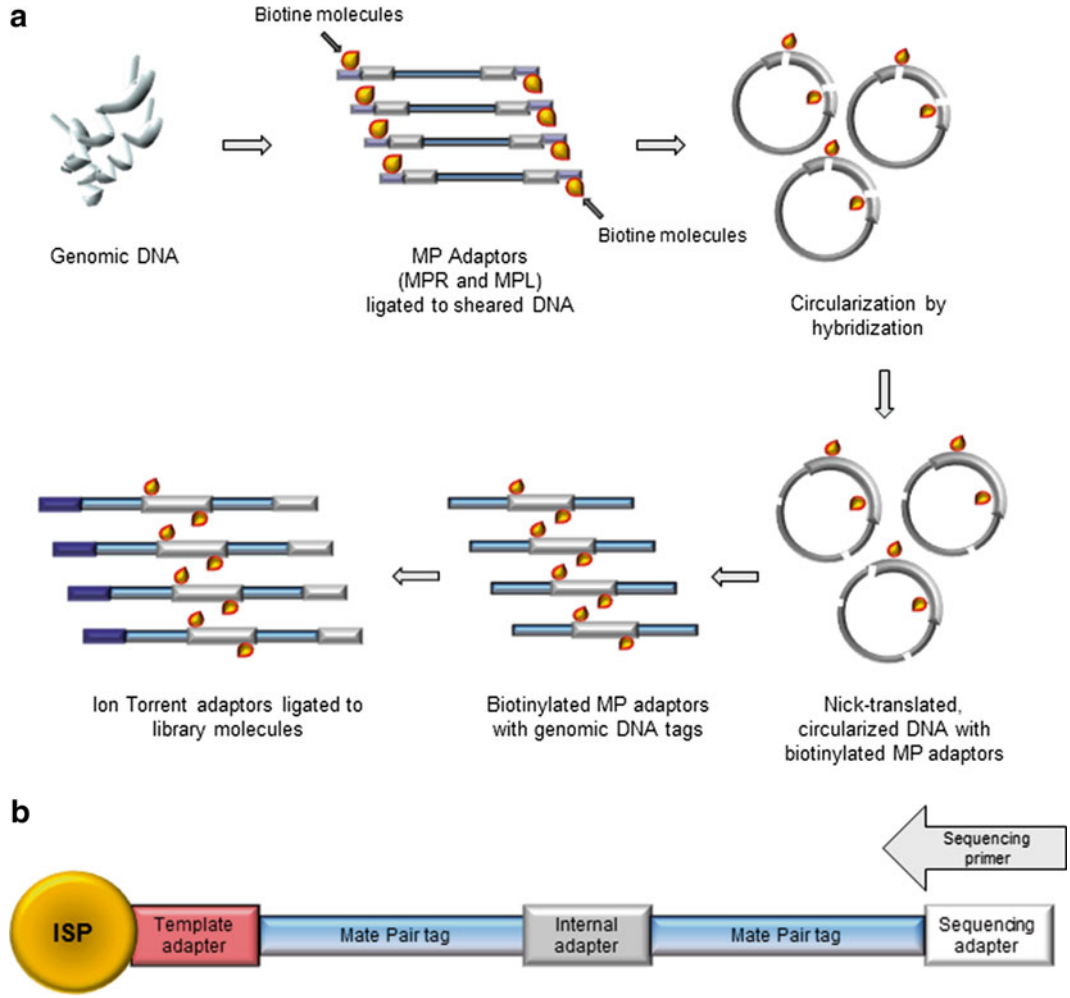

Fig. 4 Ion torrent mate-paired library preparation protocol and sequencing. (**a**) Critical mate-paired library preparation workflow. After DNA shearing, size-selected genomic DNA fragments are ligated to intramolecular circularization (MPR and MPL) adaptors, then self-circularized by hybridization in a very dilute solution. The resulting DNA circle has one nick in each strand "pushed" by *E. coli* DNA polymerase I in 5′–3′ direction in a controlled manner. T7 Exonuclease and S1 Nuclease digestion cut the DNA at the position opposite to the nick and releases the DNA mate pair. Ion adaptors are then ligated to the ends of the mate-paired library. (**b**) The final library products ligated to an ISP. The final product consists of several fragments composed of an internal adapter with known sequence surrounded by the two DNA ends from a specific fragment named Mate pair Tag. From the sequencing adaptor the system produces reads with positional info for both ends of a predefined size selected DNA molecule

with different size tag, ranging from 30 up to 100 bp. Afterwards, T7 Exonuclease and S1 Nuclease digestion cuts the DNA at the position opposite to the nick and releases the DNA mate-paired. Then, the useful adapters for the subsequent template and sequencing steps are ligated to the ends of the mate-paired library (Fig. 4a).

The final products of this pipeline consist of several fragments composed of an internal adapter, with known sequence, surrounded by the two DNA ends from a specific fragment named Mate pair

Tag. After the sequencing run, each read contains two tags that permit to determine the contig order and orientation, during genome assembly (Fig. 4b). Thanks to the known gap size, this approach helps to fill gaps among several contigs and improves the de novo assemblies for identifying novel structural and functional genomic arrangements in newly sequenced strains [9, 20]. This strategy can be especially useful when it is not possible to have access at reference strain sequences or when it can be coupled with other sequencing or mate-paired data with different gap size. The possibility to get longer Mate Pair Tags will improve genome assembly avoiding issues, usually related to repeated sequences and regions of low complexity, that typically hamper accurate assembly [6, 18]. As already mentioned above, the shotgun pipeline is the largest sequencing strategy used for low complexity targets, either single genomes or metagenomes. This pipeline is fast and accurate and its efficiency is depending on the number and the length of the reads as well as on sequencing strategies (e.g., single or paired end reads). Also this sequencing pipeline can be useful for de novo and re-sequencing projects [7, 8]. In the study of Ancora et al. [21], this pipeline has been used for sequencing genomic DNA from *Brucella ceti* ST26 strain, isolated from Italian striped dolphins (*Stenella coeruleoalba*). The genome of *Brucella* is composed of two circular chromosomes without any plasmids. In this pipeline, as first step the genomic DNA is purified and quantified by Qubit® dsDNA HS Assay Kit (Invitrogen™, Life Technologies). A quantity equal to 1 μg of bacterial DNA is employed for enzymatic fragmentation and Ion torrent adapters ligation, using the Ion Plus fragment library kit (Ion Torrent™, Life Technologies). Enzymatic shearing step is set at 37 °C for 5 min in a thermocycler. After the adapters ligation a size selection step is performed in order to get fragment libraries with 200 bp size, using the E-Gel® SizeSelect™ 2 % agarose gel (Invitrogen™, Life Technologies) and a 50 bp ladder (Invitrogen™, Life Technologies) on E-Gel® SizeSelect™ system (Invitrogen™, Life Technologies) (Fig. 5). After the size selection, a further step of library amplification is performed by PCR according to the manufacturer's instructions (Ion Torrent™, Life Technologies), completed with a final purification step by XP Ampure beads (Beckman™). Purified libraries are then quantified and validated for quality by running an aliquot on High Sensitivity Bioanalyzer Chip by using Agilent Bioanalyzer 2100 instrument (Agilent™) (Fig. 6). For template preparation step, libraries are diluted down to a concentration of 26 pM and exploited in the ePCR reaction carried on by the Ion Torrent OneTouch™ system (Ion Torrent™, Life Technologies). The use of OneTouch™ instrument allows fragments produced from libraries to be ligated to Ion Sphere Particles (ISPs) and, after this step, ISPs are enriched by using the Ion OneTouch™ ES instrument (Ion Torrent™, Life

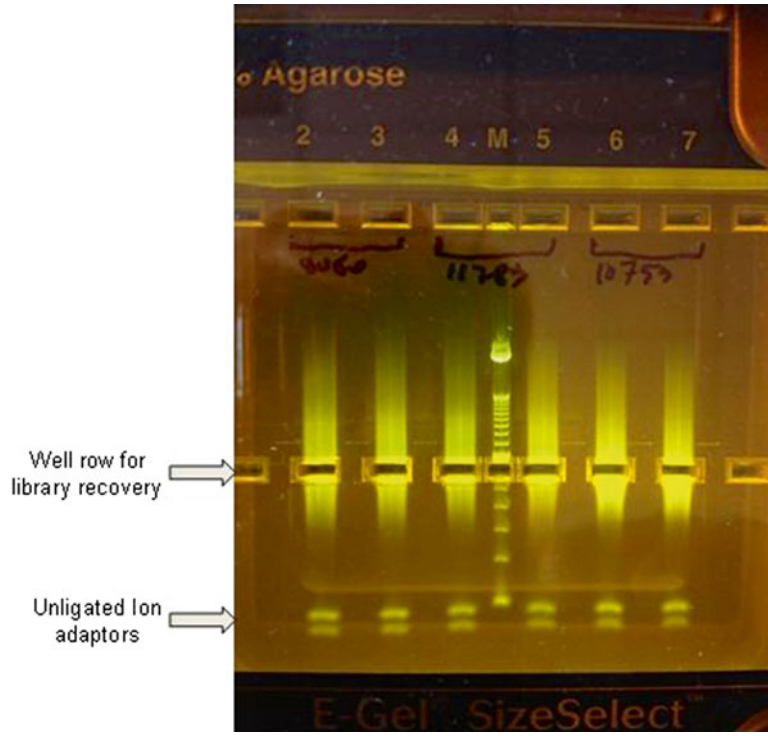

Fig. 5 Size selection step. This step was performed by using E-Gel® SizeSelect™ system (Invitrogen™, Life Technologies). Following this protocol, it is possible to recover fragments with selected sizes; in this case an insert sized 200 bp is recovered by directly aspirating samples from the well row in the middle of a 2 % agarose gel depending on a reference line for the 50 bp ladder (Invitrogen™, Life Technologies) in the upper well M. On the bottom of the gel unligated adaptors are visible

Technologies) in order to get only the covered ISPs to be loaded on the semiconductor sequencing chip [22]. All these template steps are performed using the Ion OneTouch 200 template kit version 2 DL (Ion Torrent™, Life Technologies). The enriched ISPs are loaded onto the Ion 314 Chip v2 (Ion Torrent™, Life Technologies), containing up to 1.2 million wells, and sequenced in the Ion Torrent PGM™ platform with the Ion PGM™ Sequencing 200 Kit v2 using 500 dNTPs flows and the Torrent Server Suite™ 3.4.2 version (Ion Torrent™, Life Technologies) (Fig. 7). All low-quality bases are trimmed from the sequence reads, and the remaining reads are de novo assembled by using Velvet software version 1.1.0 [23]. Genomes are finished by *in-house* developed python packages and genome annotation is performed by Prokka (Prokaryotic Genome Annotation System—http://vicbioinformatics.com/) followed by manual inspection.

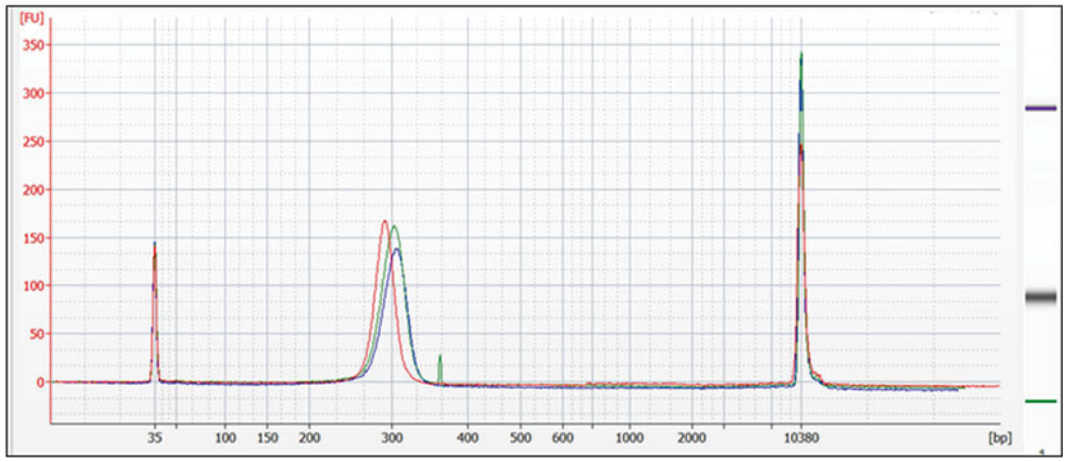

Fig. 6 Library quality check on Bioanalyzer. The electropherogram shows a merging of three *Listeria* libraries with an insert size of 200 bp. All three libraries show a quite perfect alignment with an average size of 330 bp due to ligated adaptors of 60 bp each one

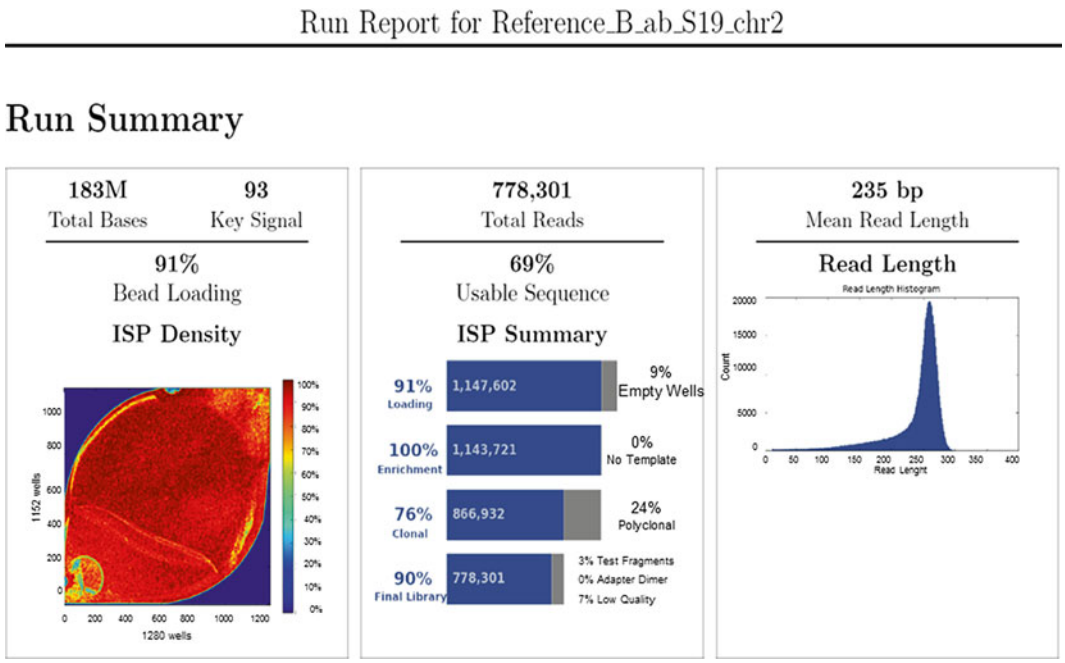

Fig. 7 Ion torrent semiconductor sequencing run report. More than 7.7 thousand filtered usable reads were obtained by this Ion Torrent run using a 314 chip v1. In this schematic visualization the loading percentage reached up to 91 % of chip capacity getting a final throughput of 183 Megabases with an average coverage depth of more than 56× (data not shown). On the *left side* the final filtered length reads is shown as a homogenous distribution around 235 bp

4 Bacterial Transcriptome Analysis

The whole-genome (WG)-NGS technologies have considerably increased the number of microbial genomes deposited in databanks [24]. However, genomic studies are not sufficient to elucidate post-genomic processes like bacterial adaptation to diverse environments, response to stress, virulence, host association, and "quorum sensing" [25]. The transcriptomics approach is useful to elucidate all process above and have helped to assess a deeper genome assembly by newly identified transcriptionally active regions of the genome [26].

The transcriptomics is the quantitative and qualitative study of the total RNAs present in a cell in a determined development stage or during a physiological condition [27]. The application of NGS of RNA in the gene expression profiling, named RNA-seq, has allowed to overcome techniques like Northern blot and Quantitative real-time PCR that permit the single analysis of one or few genes, and the subtractive hybridization, serial analysis of gene expression (SAGE) and microarray, that analyze a large number of genes, to come to a "ultra large scale" technology [28]. Early transcriptomic projects utilized SAGE scaled up versions, the Digital Gene Expression (DGE), with 18 bp reads, then substituted with 25 bp of mRNA-seq, that enabled unique mapping of randomly fragmented cDNA reads. Illumina sequencing technology has increased read length and overall number of reads generated per run starting from short single reads (25–40 bp) up to long paired-end strand-specific reads [29]. These longer paired-end reads permit better identification (ID) and mapping of spliced reads, improving the transcriptome assembly in absence of a reference genome. Today, the latest instruments can generate more than one billion reads of >150 bp in a single run [30]. The RNA-seq studies usually provide wide information on transcription start sites (TSSs) and the location of the 5′ and 3′ UTRs of genes, and discover new ORFs and previously unknown small noncoding RNAs. Furthermore, the regulation of gene expression involves multistep transcript modification and processing. A study on *Helicobacter pylori* introduced a novel differential RNA-seq approach to characterize the transcriptome, involving selective total RNA pretreatment with an exonuclease that degrades processed RNAs (containing a 5′ monophosphate), but not the primary mRNAs (with a 5′ triphosphate) [31].

However, the routine application of RNA-seq transcriptomic to the host response to pathogen promises to provide exciting new insights into the infection process. The steps to obtain the transcripts are (1) total RNA isolation; (2) library preparation provided by fragmentation, cDNA synthesis, and final amplification; (3) NGS sequencing (Fig. 8). The first step in an RNA-seq experiment is to isolate the total RNA, which should be done as rapidly as possible. However, this may not be practical under certain conditions,

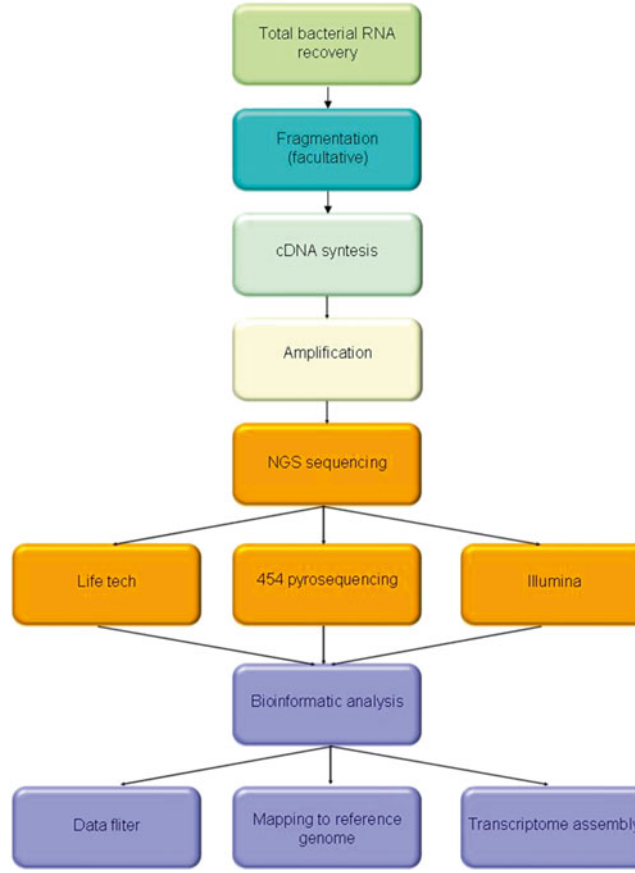

Fig. 8 Operative workflow for transcriptomic approaches. The steps to obtain the transcripts are (1) total RNA isolation; (2) library preparation provided by fragmentation, cDNA synthesis, and final amplification; (3) NGS sequencing; (4) bioinformatic analysis

so cells must be fixed to maintain transcriptome integrity during these steps, but fixation may cause partial fragmentation of the RNA [32]. It is also important to remove the genomic DNA to reduce background noise; however, many cDNA library preparation protocols ligate sequence-specific linkers directly to the RNA molecule depleting genomic DNA indirectly without the DNase treatment [30].

Moreover, in transcriptomes the rRNA represents the 80 % of total RNA, whereas mRNA constitutes only a 5 %, and many transcriptomic studies have tried to increase the information content by depleting rRNA. A lot of protocols and commercial kits for rRNA depletion are based on sequence-specific oligonucleotides bound to magnetic beads, or on reverse transcription using a pool of primers free of rRNA annealing sites [33]. Furthermore, these kits frequently have different efficiencies and may add biases [34].

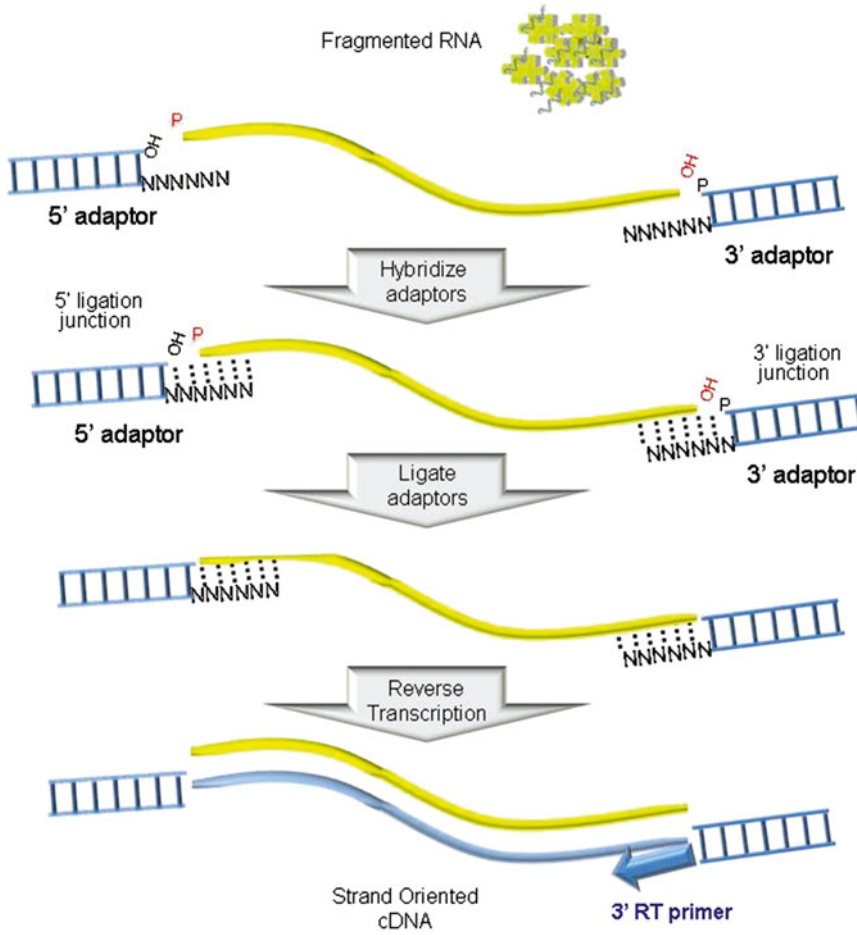

Fig. 9 Ion Total RNA-Seq approach. The protocol for RNA library preparation provides the adaptors ligation in a simultaneous and directional manner. A separate RT primer is used to generate cDNA. After RT, a purification step removes unwanted molecules. This strategy of directional ligation maintains strand orientation during sequencing

The main step to ensure a productive result passes through the library preparation step. All RNA library protocols provide a cDNA preparation that depends on the NGS platform used (Fig. 9). To obtain a uniform fragments size the RNA should be fragmented by mechanical, chemical, or enzymatical methods [27]. Moreover, the cDNA library protocol must preserve strand information, to allow the ID of antisense transcription (Fig. 9). A particular protocol provides the use of a deoxy-UTP (dUTP) second-strand-marking protocol, in which the actinomycin D is added to the reverse transcription reaction to inhibit DNA-dependent DNA synthesis, reducing genomic contamination, and the dUTP is incorporated into the second cDNA strand allowing selective destruction of this strand [29]. In bacteria, strand specificity has

also been achieved by omitting second-strand synthesis or by adding bar-code sequences for multiplexing, integrated either at RNA level by direct ligation or at cDNA level as part of the PCR primer sequence [31].

Furthermore, the technical reproducibility for RNA-seq has been claimed to be high but should be checked for each data set, especially when the coverage is low [35]. To provide a reference method that allows inter-experimental comparisons, biological samples should always be spiked-in with RNAs. The sequence of any spiked-in RNA must be confirmed bioinformatically to be absent from all the genomes under investigation [36]. Despite the advantages, sequencing data involve gigabytes of information, and consequently the analysis of these data requires bioinformatic steps and computational processes using powerful servers.

5 16S rRNA Gene Sequencing for Bacterial Identification

Metagenomics refers to culture-independent studies of microbial communities to explore microbial consortia that inhabit specific niches in plants or in animal hosts, such as mucosal surfaces and human skin [37]. Initial studies on bacterial phylogeny and taxonomy were based on the Sanger sequencing of the most common housekeeping marker that is 16S rRNA gene [38]. The choice of this locus resides in a number of reasons: (1) the presence in almost all bacteria, often existing as multicopy gene; (2) the presence of conserved nucleotide sequence region interspersed by nine high variable regions (ranging from 50 to 100 bases in length) [39]; (3) the 16S rRNA gene length (1,500 bp) [40] (Fig. 10). However, the genus- and species-level microbial ID could be optimized, with a reasonable amount of confidence, on less than half of the coding

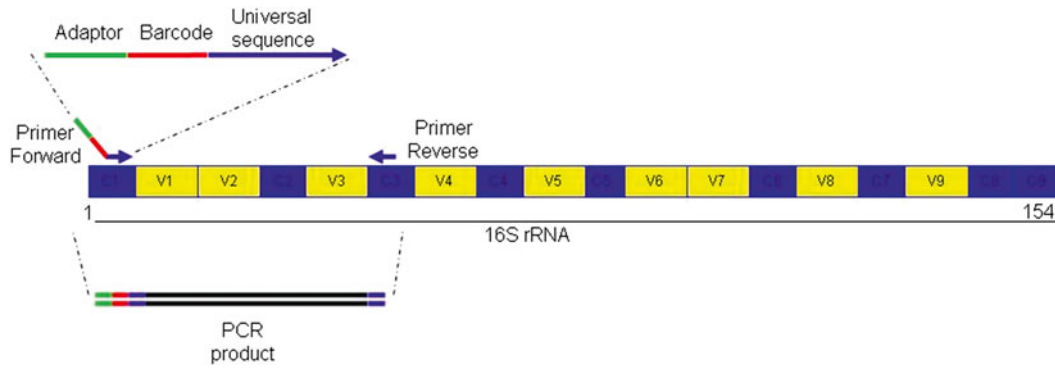

Fig. 10 Conserved and hypervariable regions in the 16S rRNA gene. The intercepted conserved regions C1–C9 are represented in *blue*, while the hypervariable regions are in *yellow*. An example of barcoded primer for amplification and pyrosequencing is reported

sequence, approximately 500 bp, including several hypervariable regions, by NGS technology [41]. The recent introduction of pyrosequencing and its combination with DNA barcoding of multiplexing samples have greatly increased the yield of bacterial community analysis. These improvements have enabled large-scale studies involving hundreds of different individuals or time points [42]. In a typical pyrosequencing experiment, a short segment covering one or two variable regions of the 16S rRNA is amplified with 16S specific primers fused with library adaptor sequences and sequenced. The length of the segment is around 300 or 500 bases depending on the use of Illumina paired-end or Roche 454 machines, respectively [43]. By amplifying selected regions within 16S rRNA genes, bacteria and Archaea can be identified by the use universal primers probing conserved regions flanking variable sequences that facilitate genus and species identification. The hypervariable regions showed variable efficacy with respect to different species, and the V2–V3 regions were most effective for universal genus ID [39]. The identity and frequency of bacteria in a sample are determined by assigning reads to known 16S rRNA database sequences via sequence homology [44], or by clustering reads [45]. Genus and species are typically distinguished at levels of 95 and 97 % pairwise sequence identities, respectively, and strains may be distinguished at the level of 99 % pairwise sequence identity [46]. For ID different taxonomic classifications are used, and different species may be identified depending on the taxonomic scheme. Multiple online databases provide access to large ribosomal RNA sequence databases. The most prominent databases include the Ribosomal Database Project II (RDP II) (http://rdp.cme.msu.edu/) [47], Greengenes (greengenes.lbl.gov) [48], and ARB-Silva [49]. RDP II is based on Bergey's taxonomy which contains a relatively small number of phyla. Greengenes includes multiple taxonomic schemes using different classification systems. The ARB-Silva database also offers an option of microbial taxonomies, though it is more limited in its plasticity than Greengenes [48].

Online ribosomal RNA databases include a multiplicity of software tools for sequence classification and multiple sequence alignments in order to facilitate microbial identification. All this databases contain sequence query tools, sequence alignment programs, and sequence editors. Greengenes provides different query and sequence alignment tools for sequence-based microbial ID [48]. Greengenes uses the NAST aligner tool and generates outputs that are compatible with ARB software tools [48]. Also different supervised sequence classifier tools are available for matching test with query sequences. Compared to BLAST, supervised classifiers like RDP Seqmatch demonstrate greater accuracy in finding most similar rDNA sequences [44]. Furthermore, despite microbial 16S rRNA sequencing is the gold standard for microbial communities characterization, this technique could be improved using 454 sequencing of whole microbial genomes [50].

References

1. Luo G, Wang W, Angelidaki I (2013) Anaerobic digestion for simultaneous sewage sludge treatment and CO biomethanation: process performance and microbial ecology. Environ Sci Technol 47:10685–10693
2. Salipante SJ, Sengupta DJ, Hoogestraat DR et al (2013) Molecular diagnosis of Actinomadura madurae infection by 16S rRNA deep sequencing. J Clin Microbiol 51:4262–4265
3. Salipante SJ, Sengupta DJ, Rosenthal C et al (2013) Rapid 16S rRNA next-generation sequencing clinical of polymicrobial samples for diagnosis of complex bacterial infections. PLoS One. doi:10.1371/journal.pone.0065226
4. Thomas T, Gilbert J, Meyer F (2012) Metagenomics—a guide from sampling to data analysis. Microb Inform Exp. doi:10.1186/2042-5783-2-3
5. Luo C, Tsementzi D, Kyrpides N et al (2012) Direct comparisons of Illumina vs. Roche 454 sequencing technologies on the same microbial community DNA sample. PLoS One. doi:10.1371/journal.pone.0030087
6. Schatz MC, Delcher AL, Salzberg SL et al (2010) Assembly of large genomes using second-generation sequencing. Genome Res 20:1165–1173
7. Powers JG, Weigman VJ, Shu J et al (2013) Efficient and accurate whole genome assembly and methylome profiling of E. coli. BMC Genomics 14:675
8. Durfee T, Nelson R, Baldwin S et al (2008) The complete genome sequence of Escherichia coli DH10B: insights into the biology of a laboratory workhorse. J Bacteriol 190:2597–2606
9. Jucá Ramos RT, Ribeiro Carneiro A, De Castro Soares S et al (2013) High efficiency application of a mate-paired library from next-generation sequencing to postlight sequencing: *Corynebacterium pseudotuberculosis* as a case study for microbial *de novo* genome assembly. J Microbiol Methods 95:441–447
10. Milani C, Hevia A, Foroni E et al (2013) Assessing the fecal microbiota: an optimized ion torrent 16S rRNA gene-based analysis protocol. PLoS One. doi:10.1371/journal.pone.0068739
11. White AG, Watts GS, Lu Z et al (2014) Environmental arsenic exposure and microbiota in induced sputum. Int J Environ Res Public Health 21:2299–2313
12. Hasman H, Saputra D, Sicheritz-Ponten T et al (2014) Rapid whole-genome sequencing for detection and characterization of microorganisms directly from clinical samples. J Clin Microbiol 52:139–146
13. Van Hal SJ, Steen JA, Espedido BA et al (2014) In vivo evolution of antimicrobial resistance in a series of Staphylococcus aureus patient isolates: the entire picture or a cautionary tale? J Antimicrob Chemother 69:363–367
14. Tyakht AV, Kostryukova ES, Popenko AS et al (2013) Human gut microbiota community structures in urban and rural populations in Russia. Nat Commun. doi:10.1038/ncomms3469
15. Zhang T, Zhang XX, Ye L (2011) Plasmid metagenome reveals high levels of antibiotic resistance genes and mobile genetic elements in activated sludge. PLoS One. doi:10.1371/journal.pone.0026041
16. Lai Z, Zou Y, Kane NC et al (2012) Preparation of normalized cDNA libraries for 454 Titanium transcriptome sequencing. Methods Mol Biol 888:119–133
17. Wan M, Faruq J, Rosenberg JN et al (2013) Achieving high throughput sequencing of a cDNA library utilizing an alternative protocol for the bench top next-generation sequencing system. J Microbiol Methods 92: 122–126
18. Chaisson MJ, Pevzner PA (2008) Short read fragment assembly of bacterial genomes. Genome Res 18:324–330
19. Rodrigue S, Materna AC, Timberlake SC et al (2010) Unlocking short read sequencing for metagenomics. PLoS One. doi:10.1371/journal.pone.0011840
20. Umemura M, Koyama Y, Takeda I (2013) Fine *de novo* sequencing of a fungal genome using only SOLiD short read data: verification on *Aspergillus oryzae* RIB40. PLoS One. doi:10.1371/journal.pone.0063673
21. Ancora M, Marcacci M, Orsini M et al (2014) Complete genome sequence of a Brucella ceti ST26 strain isolated from a striped Dolphin (Stenella coeruleoalba) on the coast of Italy. Genome Announc. doi:10.1128/genomeA.00068-14
22. Merriman B, Ion Torrent R&D Team, Rothberg JM (2012) Progress in ion torrent semiconductor chip based sequencing. Electrophoresis 33:397–417
23. Zerbino DR, Birney E (2008) Velvet: algorithms for *de novo* short read assembly using de Bruijn graphs. Genome Res 18:821–829
24. Chain PSG, Grafham DV, Fulton RS et al (2009) Genome project standards in a new era of sequencing. Science 326:236–237
25. Toledo-Arana A, Repoila F, Cossart P (2007) Small noncoding RNAs controlling pathogenesis. Curr Opin Microbiol 10:182–188

26. Pierlé SA, Dark MJ, Dahmen D et al (2012) Comparative genomics and transcriptomics of trait-gene association. BMC Genomics 13:669
27. Wang Z, Gerstein M, Snyder M (2009) RNA-Seq: a revolutionary tool for transcriptomics. Nat Rev Genet 10:57–63
28. Pinto AC, Melo-Barbosa HP, Miyoshi A et al (2011) Application of RNA-seq to reveal the transcript profile in bacteria. Genet Mol Res 10:1707–1718
29. Parkhomchuk D, Borodina T, Amstislavskiy V et al (2009) Transcriptome analysis by strand-specific sequencing of complementary DNA. Nucleic Acids Res 37:e123
30. Westermann AJ, Gorski SA, Vogel J (2012) Dual RNA-seq of pathogen and host. Nat Rev Microbiol 10:618–630
31. Sharma CM, Hoffmann S, Darfeuille F et al (2010) The primary transcriptome of the major human pathogen Helicobacter pylori. Nature 464:250–255
32. Cox ML, Eddy SM, Stewart ZS et al (2008) Investigating fixative-induced changes in RNA quality and utility by microarray analysis. Exp Mol Pathol 84:156–172
33. Armour CD, Castle JC, Chen R et al (2009) Digital transcriptome profiling using selective hexamer priming for cDNA synthesis. Nat Methods 6:647–649
34. Giannoukos G, Ciulla DM, Huang K et al (2012) Efficient and robust RNA-seq process for cultured bacteria and complex community transcriptomes. Genome Biol 13:R23
35. McIntyre LM, Lopiano KK, Morse AM et al (2011) RNA-seq: technical variability and sampling. BMC Genomics 12:293
36. Jiang L, Schlesinger F, Davis CA et al (2011) Synthetic spike-in standards for RNA-seq experiments. Genome Res 21:1543–1551
37. Del Chierico F, Gnani D, Vernocchi P et al (2014) Meta-omic platforms to assist in the understanding of NAFLD gut microbiota alterations: tools and applications. Int J Mol Sci 15:684–711
38. Janda JM, Abbott SL (2007) 16S rRNA gene sequencing for bacterial identification in the diagnostic laboratory: pluses, perils, and pitfalls. J Clin Microbiol 45:2761–2764
39. Chakravorty S, Helb D, Burday M et al (2007) A detailed analysis of 16S ribosomal RNA gene segments for the diagnosis of pathogenic bacteria. J Microbiol Methods 69:330–339
40. Patel JB (2001) 16S rRNA gene sequencing for bacterial pathogen identification in the clinical laboratory. Mol Diagn 6:313–321
41. Margulies M, Egholm M, Altman WE et al (2005) Genome sequencing in microfabricated high-density picolitre reactors. Nature 437:376–380
42. Human Microbiome Project Consortium (2012) Structure, function and diversity of the healthy human microbiome. Nature 486:207–214
43. Amir A, Zeisel A, Zuk O et al (2013) High-resolution microbial community reconstruction by integrating short reads from multiple 16S rRNA regions. Nucleic Acids Res 41:e205. doi:10.1093/nar/gkt1070
44. Wang Q, Garrity GM, Tiedje JM et al (2007) Naïve Bayesian classifier for rapid assignment of rRNA sequences into the new bacterial taxonomy. Appl Environ Microbiol 73:5261–5267
45. Kuczynski J, Lauber CL, Walters WA et al (2012) Experimental and analytical tools for studying the human microbiome. Nat Rev Genet 13:47–58
46. Peterson DA, Frank DN, Pace NR et al (2008) Metagenomic approaches for defining the pathogenesis of inflammatory bowel diseases. Cell Host Microbe 3:417–427
47. Cole JR, Chai B, Farris RJ et al (2007) The ribosomal database project (RDP-II): introducing myRDP space and quality controlled public data. Nucleic Acids Res 35:D169–D172
48. DeSantis TZ, Hugenholtz P, Larsen N et al (2006) Greengenes, a chimerachecked 16S rRNA gene database and workbench compatible with ARB. Appl Environ Microbiol 72: 5069–5072
49. Pruesse E, Quast C, Knittel K et al (2007) SILVA: a comprehensive online resource for quality checked and aligned ribosomal RNA sequence data compatible with ARB. Nucleic Acids Res 35:7188–7196
50. Petrosino JF, Highlander S, Luna RA et al (2009) Metagenomic pyrosequencing and microbial identification. Clin Chem 55:856–866

Chapter 4

The Pyrosequencing Protocol for Bacterial Genomes

Ermanno Rizzi

Abstract

The pyrosequencing methodology was applied in 2005 by 454 Lifesciences to the emerging field of next generation sequencing (NGS), revolutionizing the way of DNA sequencing. In the last years the same strategy grew up and was technologically updated, reaching a high throughput in terms of amount of generated sequences (reads) per run and in terms of length of sequence up to values of 800–1,000 bases. These features of pyrosequencing perfectly fit to bacterial genome sequencing for the de novo assemblies and resequencing as well. The approaches of shotgun and paired ends sequencing allow the bacterial genome finishing providing a high-quality data in few days with unprecedented results.

Key words NGS, Pyrosequencing, Reads, Bacterial genomes, Paired ends

1 Introduction

The next generation sequencing approach, invented by 454 Lifesciences [1], is now available on the platforms FLX-Titanium and Junior Genome Sequencer (Roche/454) and could be applied in different biology fields. The sequencing chemistry of the Roche/454 methodology is the pyrosequencing based on the single nucleotide addition resulting in the emission of a spotlight every time the correct nucleotide is incorporated. Briefly, as shown in Fig. 1, the enzymes involved in the pyrosequencing reaction cascade are the DNA polymerase, the ATP sulfurylase, the luciferase, and the apyrase [2]. The four deoxynucleotide triphosphates (dNTPs) are separately added on the growing chain by the DNA polymerase, and the inorganic pyrophosphate (PPi) is released after the addition of the correct dNTP. The released PPi is then converted to ATP by the enzyme ATP sulfurylase in the presence of adenosine 5′ phosphosulfate (APS). The amount of produced ATP is proportional to the amount of light emitted by the conversion of luciferin to oxyluciferin, a reaction catalyzed by the enzyme luciferase. A key enzyme of the pyrosequencing procedure is the apyrase that degrades the unincorporated free dNTPs, so that each

Alessio Mengoni et al. (eds.), *Bacterial Pangenomics: Methods and Protocols*, Methods in Molecular Biology, vol. 1231, DOI 10.1007/978-1-4939-1720-4_4,

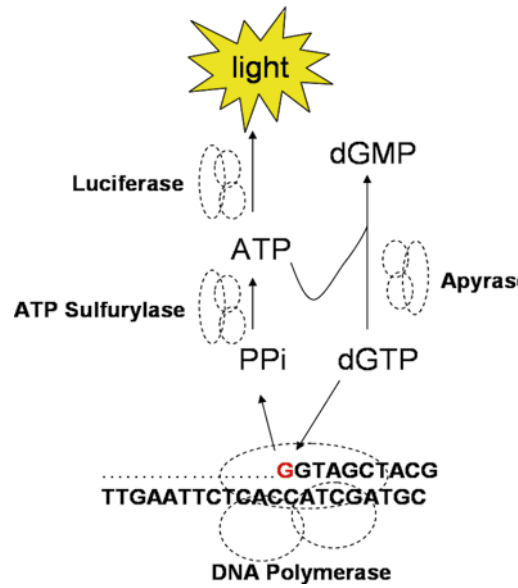

Fig. 1 The pyrosequencing reaction cascade

dNTP addition is specifically related to a certain amount of light. In the Roche/454 platforms, a Charge-Coupled Device (CCD) camera, placed in front of the pyrosequencing reaction site, detects and measures the emitted light as a result of dNTP incorporation.

The Roche/454 platforms are currently applied to sequence different kind of targets such as genomic DNA, target genome regions (PCR amplicons or target enriched genome portions), degraded and ancient DNA [3], and target for transcriptomic studies (mRNA, cDNA, and microRNA) [4] and chromatin immunoprecipitation (ChIP) sequencing studies [5]. The sequencing of bacterial genomes exploits the high-throughput and the read size provided by the Roche/454 pyrosequencing. These latter features are very important for the bacterial genome sequencing, easing the genome scaffolding for the complete genome finishing. Many bacterial genomes were completed in the last few years, using the pyrosequencing approach, unraveling the sequence of pathogens [6], antibiotic producers [7], environmental stains [8], and bioremediation-related bacteria [9].

Roche/454 platforms generate long reads, in particular the FLX-Titanium version Plus, generate 800–1,000 bases long reads that speed-up the genome assembly; in addition, the protocols for sample preparation are optimized for sequencing GC rich genomes.

For all sequencing applications, the workflow as shown in Fig. 2 is the same and consists of three steps: (1) library preparation, (2) emulsion PCR (emPCR), and (3) pyrosequencing. These protocols will be shown in the next sessions of this chapter.

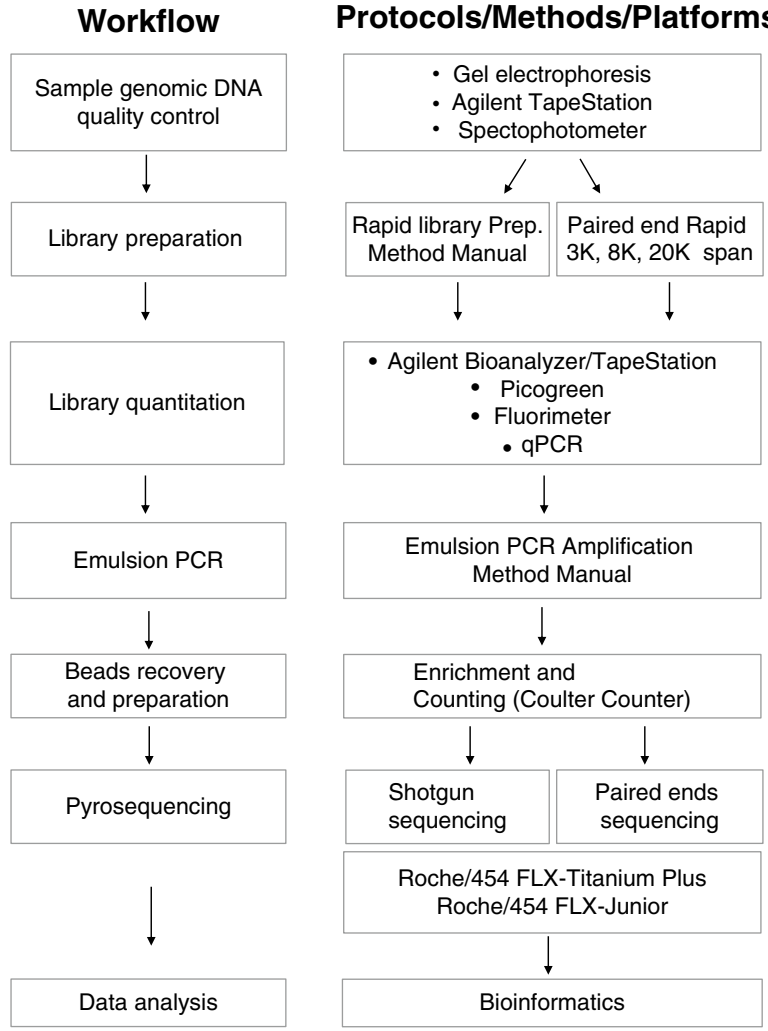

Fig. 2 Workflow for sample preparation and related protocols

The first step is the conversion of the genomic DNA (gDNA) into an amplifiable and sequenceable library. In the NGS field, the library is the mixture of DNA fragments ligated with specific oligonucleotides adapters bringing specific sequences that prime the subsequent amplification and sequencing. Briefly, as schematically shown in Fig. 3, after the DNA fragmentation made with nitrogen nebulization, the DNA fragments are ligated to "Y-shape" oligo-adapters that are subsequently recognized by the amplification machinery and the following pyrosequencing. In addition, the "Y-shape" oligo-adapters are end-labeled by a fluorophore (FAM) that allows the library quantification by the use of a spectrofluorimeter.

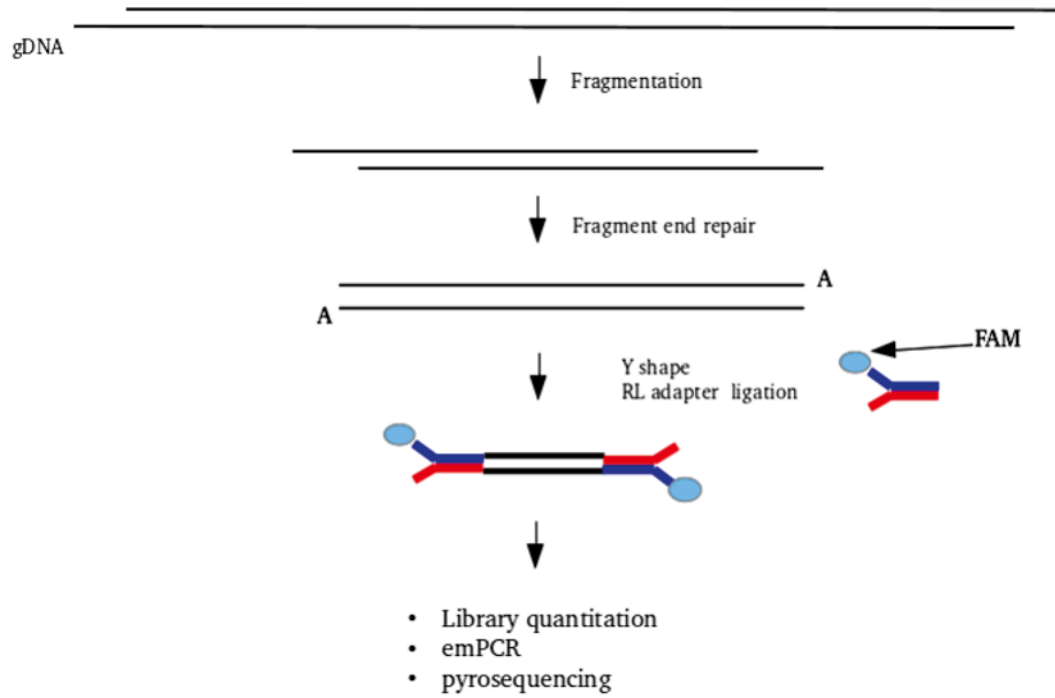

Fig. 3 Schematic procedure of the Rapid library preparation

The library quantification is a crucial step to obtain valuable results; in fact other than a fluorimetric approach, often the sample library is quantitated by real-time PCR. Because of the random ligation of adapters at the fragments' DNA ends, each fragment is sequenced in both forward and reverse directions.

During library preparation, it is possible the ligation of the Multiplex Identifiers (MIDs), short nucleotide sequences added to the standard adapter that allow the multiplexing, that is the procedure of sequencing together different samples. About 12 different MIDs are available (the number could be incremented up to 100) allowing the parallel sequencing of 12 different samples in the same region. At the end of the sequencing run, the identification of MID-related samples is ensured by the de-multiplexing bioinformatic procedure.

After library preparation and quantitation, ligated DNA is amplified. This procedure is needed because the sequencing platform is not able to identify and measure the sequencing signal (luminescence in the case of pyrosequencing) generated by a single molecule, but from a very high amount (millions) of DNA molecules. The library amplification is performed by an emulsion PCR (emPCR); in this phase microdroplets of water act as microreactors and are mixed with oil creating an emulsion. Inside the microdroplet of water, with DNA polymerase, buffers and all reagents for a

PCR, the DNA library is attached onto the surface of a sepharose bead were is immobilized an oligonucleotides with a sequence that is complementary to the sequence of one adapter at the end of DNA fragments. Using a very high diluted library solution, the emPCR is conducted with a ratio between the number of molecules (or copies) and beads that must be closer to one so that one molecule of library binds one bead inside one water droplet. This procedure is performed to obtain the emulsio-clonal amplification of a single molecule. After emPCR, the surface of each bead is covered by millions copies of a DNA fragment, so all DNA fragments of the library are amplified in a single emPCR. At the end of the emPCR, all beads are recovered by breaking the emulsion using alcoholic solutions and buffers, resulting in the retrieval of all beads in suspension.

After the amplification and emulsion breaking, the beads generated are classified into three categories: null beads, DNA beads, and the so-called mixed beads. The null beads are those that during emPCR did not bind any DNA fragments; conversely the mixed beads are those with more than one type of DNA fragment attached and amplified onto the beads surface, while the DNA beads are considered as "good" beads bringing the clonal amplification of one molecule of library DNA. The null beads are removed from the mixture of all beads, by the procedure called enrichment, which physically separates the beads bringing DNA from null beads. After this procedure the beads are counted by the use of a Coulter Counter and the number of enriched beads is used to calculate the enrichment yield. The enriched beads are a mixture of good and mixed beads, and because these latter must be excluded by the sequencing, if the enrichment yield is higher than a certain threshold, the sample must be discarded. Only samples beads that generate an enrichment yield lower than the threshold could be considered for the subsequent pyrosequencing (see later).

The good beads are then loaded onto the PicoTiterPlate (PTP), the specific support that hosts the DNA beads other than specific enzyme beads and packing beads. The PTP is a fiberglass support able to host from 500,000 (Junior) to 2 million beads (FLX-Titanium). During the PTP loading, the beads fit inside a well that has a diameter size that is able to host just one bead; accordingly, each well contains one DNA beads, so that the sequencing signal from a specific well is related to a bead that is related to a single sequence.

The sequencing step itself is performed inside the NGS platform, FLX-Titanium or Junior where the loaded PTP is inserted in.

During the sequencing run, flowgrams are generated, which are histograms in which for each nucleotide correspond a light intensity. Because of the presence of four specific bases at the beginning of the DNA library (part of the adapters added to DNA fragments by ligation), the system can normalize the measured

light and can refer a certain amount of light to a certain number of added nucleotides. The normalization procedure allows the identification of homopolymeric stretch. Unfortunately, pyrosequencing has an important, but very well-known sequencing error that occurs in homopolymeric regions. These errors are intrinsic to the pyrosequencing chemistry and take place in long homopolymeric regions, where the proportionality between light intensity and number of added nucleotide is lost. This error results in fake insertions or deletion in region with a homopolymeric length of about five or six nucleotides.

As final data, the sequencing run generates a Standard Flowgram Format file (".sff") that includes both quality and sequence information. The files could be used for assembly of genomes or mapping if a reference sequence is available in databases.

The pyrosequencing strategy applied by Roche/454 platforms provides a very low sequencing error rate that permits the genome finishing with a moderate depth of sequencing (also referred as "X": 10X, 20X, etc.), that is the number of reads mapping on a certain single position of a reference sequence. To calculate the amount of bases that must be generated by a sequencing project that is strictly related to the number of sequencing run, the expected depth of sequencing and the genome size must be known. For a 10 million bases bacterial genome and the 20X expected depth, one must generate 200 million bases that in terms of Roche/454 FLX-Titanium means about half PTP.

Roche/454 provides two different sequencing approaches: the shotgun and the paired-ends sequencing. The shotgun approach allows the sequencing of each DNA fragment converted into library and is usually performed to create a scaffold sequence in bacterial genome sequencing projects. The assembly of all reads generated from shotgun sequencing generates contigs and their number and size depend on the amount of generated reads, the genome size, and the features of bacterial genome such as GC% and number of repeats. The long reads generated by the Plus version of FLX-Titanium, up to 1,000 bases, well fit with the scaffolding of a bacterial genome. The genome finishing could be difficult in case of de novo sequencing or when the bacterial genome is rich in repeated traits. In these cases, the paired-ends approach could be very useful to accomplish the final goal.

The paired-ends protocol allows the sequencing of both ends of the same DNA molecule of known size, so that the two ends are sequenced as paired reads. This procedure eases the complete genome sequencing, and in fact the information from paired-ends can help to order the scaffolds obtained from a shotgun sequencing. The size of the DNA molecule is a crucial value to identify the distance between the two reads sequenced in pair and their relative orientation allows understanding the orientation of an entire contig or scaffold.

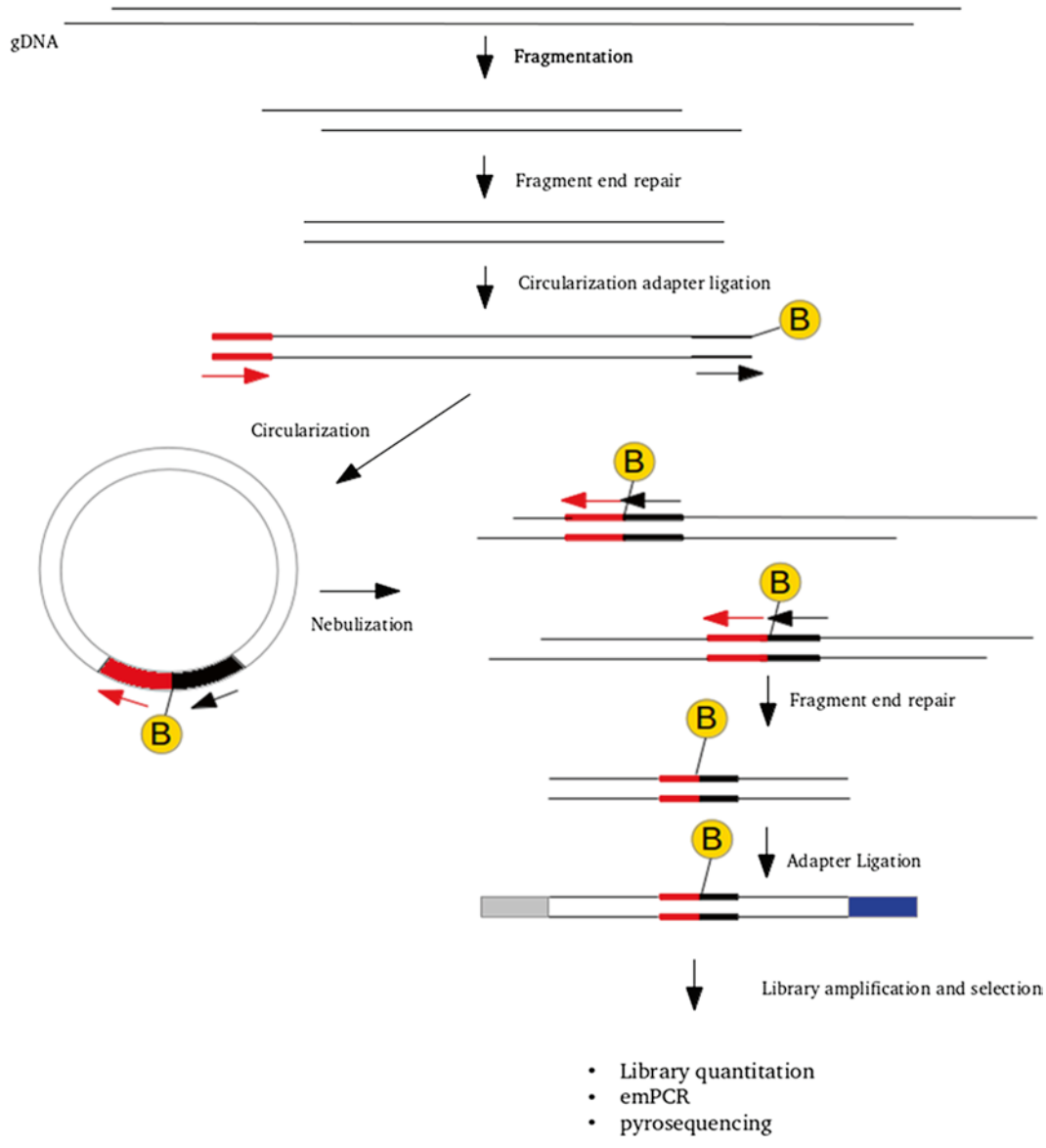

Fig. 4 Schematic procedure of the paired-ends preparation

The DNA molecule size is defined by the starting fragmentation that is obtained by a precise shearing strategy of whole gDNA. The procedures proposed by Roche/454 encompass three different multi-span paired end sizes: 3, 8, and 20 kb. The closer is the size to the specified value, the more the final assembly is precise. These fragment sizes could be obtained applying a hydrodynamic or ultrasonic acoustic energy, with the Hydroshear or Covaris instruments, respectively. Once the fragmentation in performed, the DNA is converted to sequenceable library following a specific protocol (Fig. 4) that will be described later.

The combination of sequences obtained from a shotgun and a paired-ends sequencing run could provide enough data to close an entire genome. In addition, the combination of more paired-ends sequencing, with different DNA fragment sizes, such as 3 kb plus 8 or 20 kb paired ends libraries, could reach the final goal even without any shotgun sequencing.

2 Materials and Areas

The 454/Roche procedures could be performed using the 454/Roche proprietary kits or similar kits or reagents provided by other companies when available (*see* **Note 1**). To the author of this chapter are unknown other companies, other than 454/Roche, that provide kits or reagents for the following procedures: Rapid Library Preparation for Paired End sequencing (3, 8, and 20 kb Span), Emulsion PCR Amplification, Pyrosequencing. The following protocols are intended for sequencing the sample on the 454/Roche FLX-Titanium Plus version platform.

For the libraries preparation protocols, the sample DNA should have the following features: double-stranded, OD260/280 ≥ 1.8, concentration ≥10 ng/μl, fragment size >2 kb.

3 Methods

In this session the protocols will be shown, as reported by Roche/454 Manuals, for the following steps:

- Rapid Library Preparation Method for shotgun sequencing.
- Paired End Rapid Library Preparation Manual for 3, 8, and 20 kb Span paired end sequencing.
- Emulsion PCR and beads enrichment.
- Pyrosequencing.

3.1 Rapid Library Preparation Method for Shotgun Sequencing

1. Start with 1 μg of sample DNA in a 1.7 ml microcentrifuge tube, dissolved in TE or water (*see* **Notes 2** and **3**).
2. Add TE Buffer to a final volume of 100 μl.

3.2 DNA Fragmentation by Nebulization

3. Using sterile gloves, affix a Nebulizer condenser tube around the Aspiration tube. To ensure proper function, make sure to push the condenser tube all the way down around the base of the aspiration tube, being careful not to rotate the aspiration tube, and press the vented cap into the Nebulizer top (Fig. 5). This procedure must be performed in a dedicated hood (*see* **Note 4**) to avoid sample and laboratory contamination.

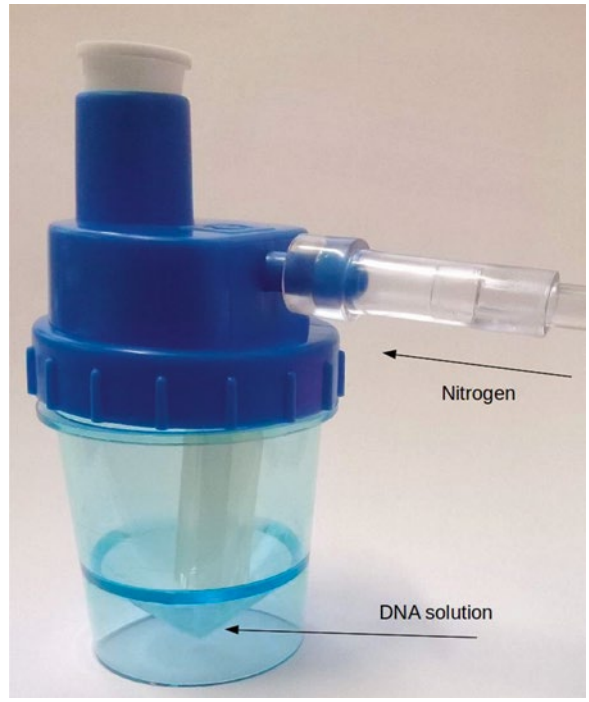

Fig. 5 Assembled nebulizer for fragmentation by nitrogen nebulization

4. Set the assembled Nebulizer top with the aspiration tube pointing upwards, making sure that the inside parts do not contact any contaminated surfaces (counter top, hands).
5. Pipet the 100 µl DNA sample in the Nebulizer cup.
6. Add 500 µl of Nebulization Buffer, pipet up and down to mix.
7. Screw the Nebulizer top to the cup, transfer the cup to the external vented hood, and connect the tubing to the nitrogen tank (Fig. 5).
8. Apply 15 psi (1 bar) of nitrogen for 1 min.
9. Disconnect the tubing and remove the cup from the hood.
10. Remove the Nebulizer top from the cup.
11. Add 2.5 ml of PBI Buffer.
12. Pipet up and down to mix.
13. Purify the nebulized DNA sample on a column from the Qiagen MinElute PCR Purification kit, as follows, with all centrifugation steps carried out at 13,000 rpm in a tabletop centrifuge.
14. Load 750 µl of the nebulized DNA at a time into a single column.
15. Centrifuge for 15 s and discard the flow-through.
16. Repeat **steps a** and **b** three more times, using the same column.

17. After the last centrifugation in **step b**, centrifuge for 1 min. Discard all the flow-through.
18. Add 750 μl of PE Buffer, centrifuge for 1 min, and discard the flow-through.
19. Centrifuge 1 min, rotate the column 180°, centrifuge 1 min.
20. Elute in new tube with 16 μl of TE Buffer. Wait 1 min before centrifuging for 1 min.
21. Transfer the sample to a 200 μl PCR tube.

The nitrogen nebulization here described could be replaced by other DNA fragmentation procedures, *see* **Notes 5** and **6**.

3.3 Fragment End Repair

1. In a 1.7 ml microcentrifuge tube, prepare the End Repair mix, as follows:
   - 2.5 μl RL 10× PNK Buffer.
   - 2.5 μl RL ATP.
   - 1 μl RL dNTP.
   - 1 μl RL T4 Polymerase.
   - 1 μl RL PNK.
   - 1 μl RL Taq Polymerase.
   - 9 μl Total volume.
2. Pipet up and down to mix, and add the 9 μl of End Repair mix to the DNA sample.
3. Vortex for 5 s, then spin for 2 s in a minicentrifuge.
4. Run the End Repair program on a thermocycler, with the heated lid on:
   - 25 °C for 20 min.
   - 72 °C for 20 min.
   - 4 °C on hold.
5. While the program is running, you can prepare the Agencourt AMPure beads as described in Subheading 3.4. You will use them in Subheading 3.6, *see* **Notes 7** and **8**.

3.4 Agencourt AMPure Bead Preparation

1. Vortex the AMPure bead bottle for 20 s, or until the beads are completely resuspended.
2. Aliquot 125 μl of AMPure beads in a 1.7 ml microcentrifuge tube.
3. Place the tube on the Magnetic Particle Concentrator (MPC).
4. When the beads have completely pelleted on the side of the tube, leaving the tube in the MPC, remove and discard all supernatant, without disturbing the beads.

5. Add 163 µl of TE Buffer to the beads, remove the tube from the MPC, and vortex for 5 s.
6. Add 500 µl of Sizing Solution to the beads, vortex for 5 s, quick spin for 2 s in a minicentrifuge. Keep the tube on ice, until you reach section 5.
7. Prepare a new 5 ml of 70 % ethanol, by adding 3.5 ml of 100 % ethanol to 1.5 ml Molecular Biology Grade Water, and vortex. You will use this 70 % ethanol solution in Subheading 3.6.

3.5 Adaptor Ligation

1. Once the End Repair program has completed, add 1 µl of RL Adaptor or RL MID Adaptor to the reaction tube.
2. Add 1 µl of RL Ligase to the reaction tube.
3. Vortex for 5 s, then centrifuge for 2 s in a minicentrifuge.
4. Incubate at 25 °C for 10 min on a thermocycler.

3.6 Small Fragment Removal

1. Add the sample to the AMPure beads already prepared. Vortex for 5 s and spin for 2 s in a minicentrifuge.
2. Incubate at room temperature for 5 min.
3. Place the tube on the MPC.
4. When the beads have fully pelleted on the wall of the tube, carefully remove and discard the supernatant, while maintaining the tube in the MPC.
5. Add 190 µl of TE Buffer. Vortex for 5 s.
6. Add 500 µl of Sizing Solution. Vortex for 5 s.
7. Incubate at room temperature for 5 min.
8. Place the tube on the MPC.
9. When the beads have fully pelleted on the wall of the tube, carefully remove and discard the supernatant, while maintaining the tube in the MPC.
10. Repeat steps 5.5–5.9, once.
11. Keeping the tube on the MPC, wash the beads twice, as follows:
    - Add 1 ml of 70 % ethanol avoiding disturbing the pellet.
    - Wait 30 s. Completely remove and discard the ethanol.
    - Keeping the tube on the MPC, uncap the tube and air-dry the pellet at room temperature for 2 min.
    - Remove the tube from the MPC.
12. Add 53 µl of TE Buffer. Vortex for 5 s and spin for 2 s in a minicentrifuge.
13. Place the tube on the MPC, wait for the beads to pellet on the wall of the tube and transfer 50 µl of the SUPERNATANT, containing the library, to a new, labeled 1.7 ml microcentrifuge tube.

Make sure not to carry over any beads in this process as they will cause incorrect readings during library quantitation.

3.7 Library Quantitation

Use either a single cuvette or a 96-well plate fluorometer to quantitate the DNA library. We recommend the TBS 380 Fluorometer (Turner Biosystems) for single use cuvette.

4 Paired End Rapid Library Preparation Manual for 3, 8, and 20 kb Span Paired End Sequencing

This procedure shows the preparation of a paired end library starting from 3, 8, or 20 kb DNA fragments. There are few and small differences between the procedures for the three span size options: the starting material, the shearing values and the volumes for Fragment end repairs (Subheading 4.2), and circularization adapter ligation (Subheading 4.3) reactions. These differences are listed in Table 1.

Start with the amount of sample DNA as shown in Table 1 and using a 1.7 ml microcentrifuge tube and adding TE Buffer to a final volume of 200 µl, *see* **Notes 9** and **10**.

4.1 DNA Fragmentation

1. Shear the DNA with the Hydroshear (Digilab Inc, MA-USA); *see* Table 1 for settings and conditions. Purify the sheared DNA using AMPure XP and at the end resuspend the beads with 52 µl of Tris–HCl. Elute the sheared DNA from AMPure XP beads.

4.2 Fragment End Repair

1. In a microcentrifuge tube, add the following reagents:
   - 24.5 µl Molecular Biology Grade Water.
   - 10.0 µl 10× PNK Buffer (free of precipitate. If any, warm buffer at +37 °C and vortex.).
   - 0.5 µl Bovine Serum Albumin (20 mg/ml).
   - 1.0 µl ATP, lithium salt, pH 7 (100 mM).
   - 4.0 µl PCR Nucleotide Mix (10 mM each).

Table 1
Reaction and Hydroshearing conditions for 3, 8, and 20 kb paired ends library preparation

Span size (kb)	Starting genomic DNA (µg)	Number of cycles (Hydroshearing)	Speed settings (Hydroshearing)	Volumes for Subheadings 4.2 and 4.3
3	5	20	12	1×
8	15	20	15	2×
20	30	20	16	2×

- 50.0 µl sheared DNA.
- The volumes of this reaction must be doubled for the 8 and 20 kb procedure.

2. Vortex, and then spin for 2 s in a minicentrifuge. Place the tube on ice and add the following enzymes:
   - 5 µl T4 DNA Polymerase (1 U/µl).
   - 5 µl Polynucleotide Kinase (PNK, 10 U/µl).
   - 100 µl Total volume.
3. Vortex, spin for 2 s in a minicentrifuge, and incubate the polishing reaction at +25 °C for 20 min.
4. Immediately after, purify the polished fragments with a QIAquick column, following the manufacturer's instructions.
5. Elute with 35 µl of Buffer EB at room temperature.

4.3 Circularization Adaptor Ligation

1. To the tube containing the purified, polished DNA, add the following reagents, in the order indicated:
   - ~35 µl Sheared and polished DNA (already in the tube).
   - 50 µl Rapid Ligase Buffer, 2× Conc.
   - 10 µl Circularization Adaptors (20 µM).
   - 95 µl Total volume.
2. Vortex and then spin for 2 s in a minicentrifuge.
3. Add 5 µl of Rapid Ligase.
4. Vortex, spin for 2 s in a minicentrifuge, and incubate the ligation reaction at +25 °C for 15 min.
5. Purify using a QIAquick column following the manufacturer's instructions.
6. Elute with 100 µl of Buffer EB at room temperature.
   - The volumes of this reaction must be doubled for 8 and 20 kb procedure.

4.4 Library Cleanup

1. Add 50 µl of AMPure XP beads to the 100 µl eluate from the previous step.
2. Vortex, spin for 2 s in a minicentrifuge, and incubate at room temperature for 5 min.
3. Using the MPC, pellet the beads against the wall of the tube, and remove the supernatant.
4. Wash the beads three times with 500 µl of 70 % ethanol, maintaining the tube in the MPC.
5. Remove all supernatant and spin for 2 s in a minicentrifuge. Remove any residual ethanol.
6. Air-dry the bead pellet for 2 min.

7. Remove the tube from the MPC, add 42 µl of Tris–HCl, and vortex to resuspend the beads.
8. Using the MPC, pellet the beads against the wall of the tube.
9. Transfer 40 µl of supernatant containing the DNA to a new microcentrifuge tube.

4.5 Fill-In Reaction

1. In a microcentrifuge tube, add the following reagents:
   - 40 µl Circularization-adapted DNA.
   - 5 µl 10× ThermoPol Buffer.
   - 2 µl PCR Nucleotide Mix (10 mM each).
   - 3 µl Bst DNA polymerase, large fragment (8 U/µl).
   - 50 µl Total volume.
2. Vortex and incubate at 50 °C for 15 min.
3. Purify using Qiaquick.
4. Elute with 52 µl of EB Buffer at RT.
5. Quantitate the eluted DNA using Pico Green dsDNA.

4.6 DNA Circularization-Adapted

1. Prepare a stock of 1 M DTT by dissolving 1.54 g of 1–4 dithiothreitol (DTT) in water in a final volume of 10 ml, followed by filtration through a 0.45 µm filter. Store in single use aliquot at –20 °C.
2. Prepare a 100 ng aliquot of the filled-in DNA in a total volume of 80 µl volume.
3. In a 0.2 µl PCR tube, add the following reagents:
   - 10 µl 10× Cre buffer.
   - 80 µl filled-in DNA (100 ng).
   - 10 µl Cre recombinase (1 U/µl) total volume: 100 µl.
4. Vortex and spin for 2 s in a minicentrifuge.
5. Incubate in a thermocycler: 37 °C for 30 min; 70 °C for 10 min, hold at 4 °C.
6. Prepare a fresh 100 mM DTT from the 1 M DTT stock.
7. Add 1.1 µl DTT (100 mM).
8. Vortex and spin for 2 s in a minicentrifuge.
9. Add the following reagents to the sample:
   - 1.1 µl ATP, lithium salt, pH 7 (100 mM).
   - 5.0 µl Plasmid-safe ATP-Dependent DNase (10 U/µl).
   - 3.0 µl Exonuclease I (20 U/µl).
10. Vortex and spin for 2 s in a minicentrifuge.
11. Incubate at 37 °C for 30 min.
12. Add 1 µl of Carrier DNA to the sample and mix gently.

13. Purify circularized DNA using a QiaQuick column.
14. Elute with 100 µl Tris–HCl at RT.

4.7 Circularized DNA Fragmentation

1. *Nebulizer assembly*. Follow the procedure as reported in "Rapid Library Preparation" protocol.
2. Pipet the sample in the nebulizer buffer.
3. Screw assembled nebulized cap onto cup.
4. Transfer the assembled nebulized to the externally vented nebulization hood.
5. Connect the loose end of nebulizer tubing to the nitrogen tank.

4.8 DNA Nebulization and Collection/ Purification of the Fragmented DNA

1. Direct 45 psi (3.1 bar) of nitrogen through the nebulizer for 2 min.
2. After nebulization, turn off the nitrogen gas flow.
3. Disconnect the tubing from the nebulizer and the nitrogen tank.
4. Tap the nebulizer on a table top to collect as much as possible to the bottom of the cup.
5. Carefully unscrew the nebulizer and measure the volume of nebulized material. Total recovery should be greater than 300 µl.
6. Add 2.4 ml of Qiagen's Buffer PBI directly into the nebulizer and swirl to collect all material droplets and mix the sample.
7. Purify using one MinElute column, loading four times the sample in the same column. Final elution is with 16 µl of EB buffer.

4.9 Fragment End Repair

1. Using the Rapid Library preparation kit, prepare the end repair mix in a 1.7 ml tube, following steps of fragment end repair of Rapid Library Preparation Method.

4.10 Immobilization Bead Preparation

1. Prepare a stock solution of 2× Library Binding Buffer by mixing 5.9 ml of water, 4.0 ml of 5 M NaCl, and 0.1 ml of 100× TE.
2. Transfer 50 µl of Dynal M-270 streptavidin beads to a new microcentrifuge tube.
3. Using MPC, pellet the beads and remove the buffer.
4. Wash the beads twice with 100 µl 2× Library Binding Buffer. Remove the supernatant at the end.
5. Resuspend the beads in 50 µl of 2× Library Binding Buffer.

4.11 Adaptor Ligation

1. Add 1 µl of RL Adaptor to the reaction tube.
2. Add 1 µl of RL ligase to the reaction tube.
3. Vortex for 5 s, then centrifuge for 2 s in a minicentrifuge.
4. Incubate at 25 °C for 10 min on the thermocycler.

4.12 Library Immobilization

1. Add 23 µl of Tris–HCl to the adapted DNA sample to a final volume of 50 µl. Transfer to a new microcentrifuge tube.
2. Add the 50 µl of washed Dynal M-270 streptavidin beads to the 50 µl of adapted DNA.
3. Vortex and place on a tube rotator at RT for 15 min.
4. Spin in a microcentrifuge.
5. Using the MPC wash the immobilized library four times with 500 µl of TE buffer. At the end remove all TE.
6. Resuspend the beads in 20 µl Tris–HCl.

4.13 Library Amplification

1. In a 200 µl tube add:
   - 30 µl water.
   - 5 µl 10× PCR Reaction Buffer with MgCl2.
   - 2 µl dNTP mix (10 mM each).
   - 2 µl amplification primers (100 µM).
   - 10 µl adapted paired end library beads suspension.
   - 1 µl faststart enzyme (5 U/µl) total volume: 50 µl.

4.14 Mix Well and Run the Following Program in a Thermocycler

- 94 °C for 11 min.
- 94 °C for 60 s.
- 60 °C for 60 s.
- 72 °C for 60 s.
- 72 °C for 10 min.
- Hold at 4 °C.

4.15 Sizing Mix Preparation

1. Aliquot 250 µl of AMPure XP beads in a 1.7 ml tube.
2. Place the tube on the MPC, when the beads have completely collected on the side of the tube, discard the supernatant.
3. Add 500 µl of Sizing Solution to the beads, vortex, and spin.
4. Keep the tube on ice and prepare an aliquot on 70 % ethanol.

4.16 Final Library Selection

1. Transfer amplified PCR product to a new 1.7 ml tube.
2. Add 125 µl of the sizing mix previously prepared to the sample.
3. Vortex and spin.

4. Incubate for 5 min at 235 °C.
5. Place sample in MPC and pellet the beads.
6. Once the beads are completely pelleted, transfer the supernatant to the tube containing the remaining 375 µl of the sizing mix previously prepared.
7. Vortex and spin in a minicentrifuge.
8. Incubate for 5 min at 25 °C.
9. Place the sample in MPC to pellet the beads.
10. Once the beads are pelleted, carefully remove and discard the supernatant.
11. Add 100 µl of TE buffer. Vortex for 5 s then spin.
12. Add 500 µl of sizing solution. Vortex for 5 s then spin.
13. Incubate at RT for 5 min.
14. Place the tube on the MPC.
15. Once the beads are pelleted, carefully remove and discard the supernatant.
16. Repeat Subheadings 4.12–4.14, once.
17. Keeping the tube on the MPC, wash the beads twice with 1 ml of 70 % ethanol.
18. Air-dry the pellet at RT.
19. Remove the tube from the MPC.
20. Add 23 µl of TE, vortex, and spin in a minicentrifuge.
21. Place the tube on the MPC, wait for the beads to pellet on the wall of the tube, and transfer 21 µl of the supernatant to a new 1.7 ml tube.

4.17 Library Quantity Assessment

There are several options for the library quantitation, while for the qualitative analysis the use of the miniaturized capillary electrophoresis by Agilent Bioanalyzer or Agilent TapeStation and High Sensitivity DNA chip is suggested (*see* **Note 11**). This measure provides the size distribution of DNA library fragments that must be between 500 and 600 bp with a lower size cut-off: <10 % below 300 bp. The Agilent miniaturized capillary electrophoresis provides both the quality and quantity assessment showing the fragment library migration as electropherogram, where the fluorescence is proportional to the amount of DNA and the migration time (seconds) is proportional to the fragments size, as shown in Fig. 6.

A quantitative assessment could be performed using a picogreen assay or a quantitative Real-Time PCR (qPCR). For this latter method there are many commercially available kits (Kapa Biosystems, NEBNext) all based on the use of SYBR® Green and specific amplification primers, complementary to the adaptor ligated to the DNA fragment ends, *see* **Notes 12** and **13**.

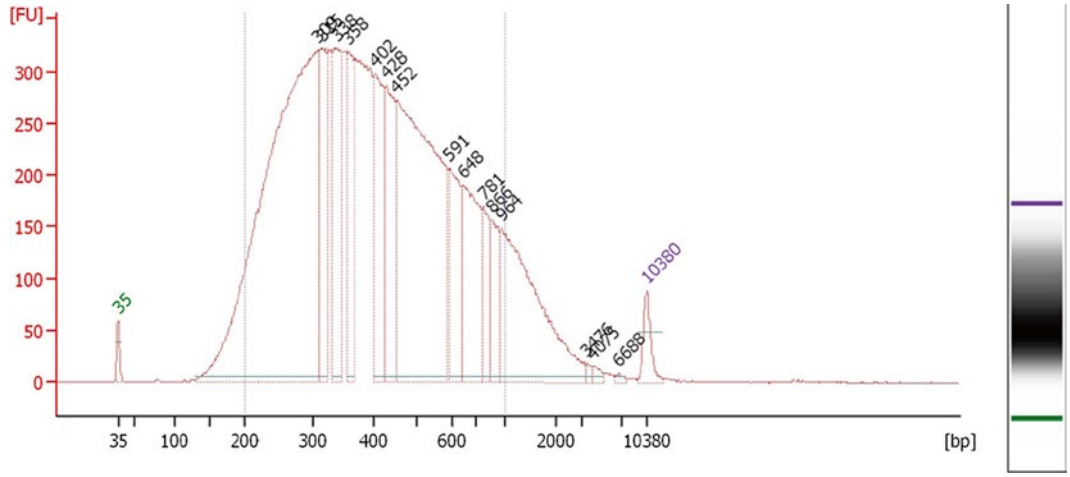

Fig. 6 Agilent Bioanalyzer electropherogram

Once the library is quantitated, the concentration expressed in ng/μl must be converted into molecules/μl, using the following equation:

$$\text{Molecules} / \mu\text{l} = \frac{\text{sample concentration}\left(\text{ng} / \mu\text{l}\right) \times N}{\left(660 \times 10^{9}\right) \times \left(\text{average fragments length}\left(\text{bp}\right)\right)}$$

where N is the Avogadro's number $= 6.022 \times 10^{23}$ molecules/mol and 660×10^{9} is the average molecular weight of a double stranded nucleotide in g/mol.

Once the library is prepared, by the use of paired-ends or rapid procedures and quantitated using fluorimetric measure, Agilent Bioanalyzer, Picogreen assay, or quantitative real-time PCR (qPCR). The library solution must be diluted and aliquoted in working stocks with a concentration of 1×10^{7} molecules/μl and a volume of about 25 μl. These aliquots can be stored at −20 °C for up to 2 months.

5 Emulsion PCR and Beads Enrichment

Conversely to the library preparation methods, the only protocols and kits available for the emulsion PCR and for pyrosequencing are those provided by Roche/454. In the following session the procedures will be briefly described, while for the detailed protocols all information are available on the Roche/454 website. Few changes to the protocol are listed in **Note 14**.

5.1 Emulsion PCR (emPCR)

The procedure for the emulsion PCR (emPCR) provides the amplification of library molecules on a support of microbeads in a water-in-oil emulsion. Libraries generated with the paired ends (3, 8, or 20 kb) or the rapid procedures will be amplified using the same protocol.

The emPCR procedure is subdivided into three parts:

- Mix preparation and emulsification.
- Amplification.
- Emulsion breaking.

During the mix preparation, the library is added to the capture beads using a precise value of copies (molecules of library) per beads (cpb), then mixed with primers, DNA polymerase, Ppiase, buffer, and water. The amplification mix is then added to Castor oil to create the emulsion by shaking with a TissueLyser with setted frequency (15 or 25 Hz) and time (2 or 5 min). Once the emulsification is performed, each water droplet containing one capture beads and one library molecule is aliquoted in a 96 wells plate. Each emPCR reaction is performed with 2.4×10^6 beads for the small volume (SV) version or 10×10^6 beads for the large (LV) volume version. The SV procedure is usually used to reach the best cpb value following a kind of titration, while the LV is used when the cpb value is already defined and obviously allows the generation of an higher number of DNA beads.

The amplification step is performed placing the 96 wells plate in a thermocycler, following this thermal profile:

- 1×: 4 min at 94 °C
- 50×: 30 s at 94 °C
- 10 min at 60 °C
- on hold at 10 °C

for libraries that will be sequenced on the FLX-Titanium Plus version. This reaction takes about 8–9 h to complete.

The emulsion breaking is performed at the end of the amplification step and allows the recovery of the DNA beads from the water-in-oil emulsion. The emulsion breaking is performed using isopropanol and the recovery and washing of the beads is made by the use of buffer and specific filters mounted on the top of a syringe (SV) or using a specific tool connected to vacuum (LV).

5.2 Enrichment

The emPCR provide three categories of beads: DNA beads, null beads, and mixed beads, so once the beads are recovered from the emulsion, an enrichment step must be performed to recover the DNA beads. After amplification, some beads could be empty of DNA so the enrichment step eliminates this category. The enrichment step is performed using a specific primer (enrichment primer)

that is complementary to one library adaptor. The library attached on the beads surface is first made single strand using a melting solution (0.0125 N NaOH) and then incubated with the enrichment primer that brings a biotin at the end allowing the separation using streptavidinated magnetic beads. After the enrichment step the beads are counted using the Coulter Counted (*see* **Note 15**) and an enrichment yield is calculated as in the following equation:

$$\text{Enrichment yield} = \frac{\text{Enriched beads}}{\text{total beads}} \times 100$$

If the enrichment yield is higher than a pre-set threshold (20 %), the reaction generated many mixed beads, those with more than one library molecule amplified onto the bead surface. This kind of beads must be excluded because they are not readable by the pyrosequencing reaction. A high value of enrichment yield is related to the amount of library molecule added to the capture beads during the amplification mix preparation. High values of cpb or wrong library quantitation result in high enrichment yield. If the enrichment yield of an amplification reaction is higher than 20 %, the reaction must be repeated using a lower value of cpb.

Once the correct value of cpb is defined, the DNA beads are annealed with the sequencing primer and are ready for the pyrosequencing reaction (*see* **Note 16**).

6 Pyrosequencing

The only NGS platforms based on the pyrosequencing methodology available in commerce are the Roche/454 FLX-Titanium or Junior. For both platforms, the prepared DNA beads, enriched, annealed, and counted, are loaded in the PicoTiterPlate (PTP) that is the physical support where the pyrosequencing takes place. The pyrosequencing step is subdivided into four phases:

- Sample beads preparation.
- PTP loading.
- Sequencing cassette preparation.
- Pyrosequencing run.

In the first step, the DNA sample beads are mixed with the enzyme beads, control beads, and packing beads to obtain a mixture containing almost all reagent needed for the sequencing; other enzymes will flow during the pyrosequencing run. The control beads added to the mix are needed to check the quality of the sequencing at the end of the run and are beads covered by DNA fragment with known sequence and length. The enzyme beads are magnetic beads covered by attached enzymes such as

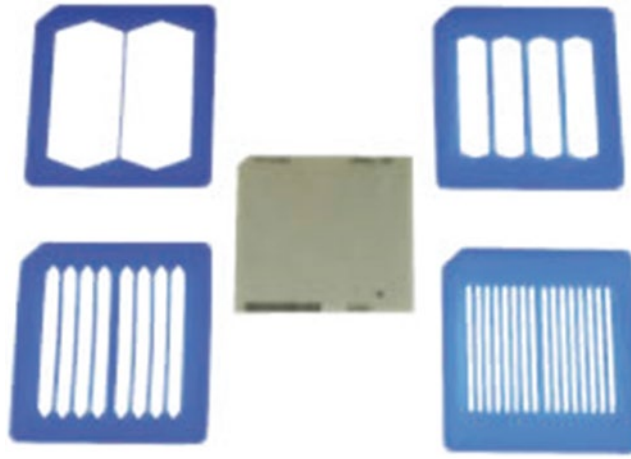

Fig. 7 PTP (in *centre*) and blue gasket used for physical multi-region separation of PTP

ATP-Sulfurylase and Phosphatase. The DNA polymerase is added as solution to the mixture. The packing beads are tiny beads that are needed to create a compact layer inside the PTP wells.

The PTP is loaded with four layers, each containing a combination of control and DNA sample beads, packing beads, and enzyme beads. The loading is performed using a specific plastic tool named Bead Deposition Device (BDD) that hosts the PTP and the blue gasket used to subdivide the PTP in 2 or more regions (up to 16), as shown in Fig. 7.

The multi-regions gaskets are used to obtain a physical separation of the PTP that could be loaded with different samples in performing a physical multiplexing. Obviously, the more regions are used, the less reads are obtained at the end of sequencing run; in fact the amount of DNA sample beads to be loaded in each region decreases with the number of regions. The Bead Deposition Device is used for all loading steps and is also used as a support for the PTP during the centrifugation needed to compact all layers (Fig. 8). The centrifugation steps are 5 or 10 min long and are performed at very high speed (1,630 rpm).

During the PTP loading and centrifugation steps, the sequencing cassette is prepared with all reagents needed for the pyrosequencing. The sequencing reagents in the cassette, such as nucleotides solution, washing solution, apyrase solutions, and buffers, are inserted in the right part of the sequencer, where fluidics, made of pumps, valves, and tubing, allow the flowing of reagents on the PTP. Figure 9 shows the FLX-Titanium Roche/454 sequencer; its right part contains the reagent cassette, while on the left of the instrument there is the signal detection and measure apparatus (*see* **Note 17**). The PTP is inserted in the

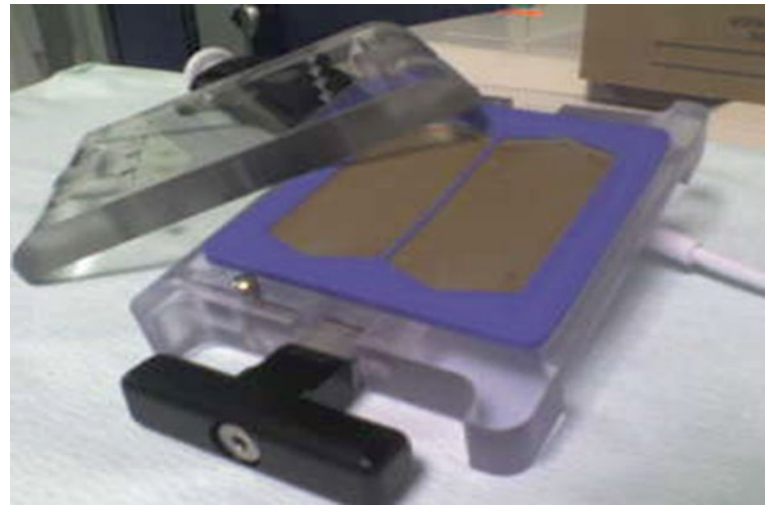

Fig. 8 The Beads Deposition Device (BDD) used to load the sample beads and the PTP divided into two regions

Fig. 9 The FLX-Titanium Roche/454 sequencer. The fluidic part at the *right side* and the signal detection device in the *left side*

cartridge placed in the left part of the instrument, where the signal is detected by a CCD camera in front of the PTP cartridge as shown in Figs. 9 and 10.

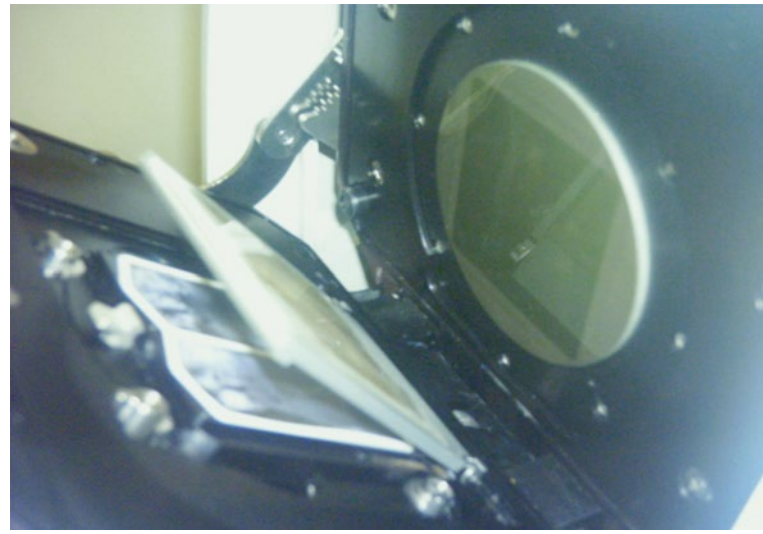

Fig. 10 The signal detection device in the *left part* of the sequencer. The PTP loaded inside the cartridge, located in *front* of the CCD camera

Once the cassette is inserted and all centrifugations steps are performed, the PTP and the sequencer are ready for the run. At the run launching step, some options are available, such as the number of sequencing cycles that result in the read length, the PTP separation (from 2 to 16 regions as briefly reported in **Note 18**), and just for the Roche/454 FLX-Titanium Plus version it is possible to launch an "acyclic" flowing of sequencing reagents. In the standard flowing (named pattern A) the nucleotides pass through the PTP by following a defined pattern that is: T, C, A, G. In addition is also available the acyclic flowing named pattern B that randomly lets the reagents flow onto the PTP surface. This latter pattern allows the increase of sequencing reads in terms of length and quality. During the run, the pyrosequencing reaction cascade (Fig. 1) takes place for each nucleotide flowing on the PTP and each light signal is measured and recorded.

7 Notes

1. The library preparation kits available in commerce other than those provided by Roche/454 are:
   - Kapa Biosystems (Kapa Biosystems Pty, South Africa Cape Town), kit named DNA NGS Library Preparation Kit.
   - Lucigen, (Middletown, WI—USA), kit named NxSeq™ DNA Sample Prep Kits for 454.
   - NEBNext, (New England Biolabs, MA—USA), kit named NEBNext Quick DNA Library Prep Master Mix Set.

2. The water used for dilutions or solutions must be ultra pure and DNAse and RNAse free.
3. All consumables such as pipette tips and plastic tubes must be DNAse and RNAse free and autoclaved.
4. To avoid contamination, the laboratory where the emulsion breaking and beads enrichment procedures are performed (high copy number laboratory) must be physically isolated from the library preparation area (low copy number laboratory).
5. The genomic DNA fragmentation could be performed with other techniques, other than nitrogen nebulization, or with the Hydroshear. The Covaris AFA instrument (Covaris, MA—USA) could be used to perform a DNA fragmentation resulting in variable ranges of sizes.
6. Few enzymatic procedures are available for the fragmentation of double strand DNA for NGS applications. The enzyme fragmentase could be used for the DNA fragmentation (NEBNext® dsDNA Fragmentase® by New England Biolabs, MA—USA).
7. The purification steps require the use of Agencourt AMPure XP (Beckman Coulter, Inc., CA—USA); in some cases this kit could be replaced by the MinElute PCR purification kit by Qiagen (Qiagen, Hilden, Germany) or similar silica columns purification kits.
8. A magnetic rack is needed for sample purification with Ampure Beads; some available in commerce are the DynaMag™ from Invitrogen-Life.
9. Once the paired ends protocol is approached, it could be very useful to test independently the enzymes before applying directly on the sample. Some enzymes used in paired end library preparation step are very sensitive.
10. The library immobilization step during the paired end procedure (step 2 uses a biotinylated circularization adapter (step 2.6)), so that the final paired-end library can be recovered by the use of streptavidinated magnetic beads, the Dynal M-270 (Invitrogen-Life).
11. The quantitative and qualitative analysis of libraries can be performed using the Agilent Bioanalyser 2100 or TapeStation 2200 and the related DNA analyses kits (Agilent Technologies, CA—USA).
12. Once a library is obtained, the better way to quantitate it is the qPCR. The PCR could be performed using a standard master mix for real time (DNA polymerase, $MgCl_2$, buffer etc.) and specific primer complementary to the Roche/454 adapters added during library preparation. In addition it is mandatory the use of DNA fragments with a known size and concentration bringing the same sequences of Roche/454 adapters

at the ends. One known commercial qPCR kit for Roche/454 libraries is the Kapa Biosystems kit, named Library Quantification Kit—454 Titanium (Lib-L)/Universal.

13. The qPCR method has also the advantage to provide a qualitative information. At the end of qPCR reaction, the amount of molecule can be easily calculated performing a standard curve with reference control DNA. Once the reaction is completed, in addition to the quantitative analysis it is also possible to analyze the size of the amplified DNA fragment. A little amount of qPCR mix could be run on an Agilent Bioanalyser to check the real size of library fragments.
14. The emPCR steps (from PCR reaction setting to the enrichment) must be performed as reported by the Roche/454 manual; few changes or advices can be applied at this stage:
    (a) The emulsion breaking must be performed as soon as possible after the end of the reaction. Due to the long duration of the emPCR, it is usually performed overnight starting the reaction as late as possible, or pretty soon in the morning so that the breaking could be performed the same day.
    (b) After the reaction that takes about 5 h and half the beads are inside the water-in-oil emulsion that could be visually observed to understand if any problems occurred during the reaction. Due to the possibility that air entry inside the well plate during the reaction, through the adhesive cover, the reaction could have a very low yield. If air entered the plate during emPCR, the emulsion is no more homogeneous but two distinct layers can be observed: the water layer down in the well and the oil as upper layer. If this two-layer emulsion is observed, the emPCR could be problematic and is recommended the repetition of the emPCR.
    (c) The melting solution used during the enrichment step must be freshly prepared every time, because if the NaOH tube (Falcon) is left open, the CO_2 in the air could react resulting in a modification value of pH that means a decreased melting power capacity.
    (d) During the enrichment step, once the null beads are discarded and the DNA sample beads are washed and transferred to a new tube, these latter could be counted using the Coulter Counter before performing the annealing step. The count of the enriched beads and the calculation of the enrichment yield at this stage avoid continuing the protocol on beads that couldn't be sequenced because of their high enrichment yield.

15. The beads count is usually performed using the Coulter Counter machine (Beckman Coulter), but it could be possible to count the beads using a Fisher chamber.
16. The enriched beads could be stored at 4 °C for 2 weeks before performing the annealing with the sequencing primer. It is recommended to perform the annealing on the DNA sample beads just before the sequencing run.
17. The sequencer is made of two parts, the fluidic one and the image generation and reading. In the fluidic part the bottles containing the nucleotides solution, buffers, and enzymes are connected to the other part with pumps, valves, and tubes that make flow one nucleotide by one and between a flow and the other are flows of degrading of un-reacted nucleotide (by apyrase) and washing by buffers. The pyrosequencing reaction cascade takes place on the PTP where the nucleotides are flown on. After each nucleotide flow the growing chain is elongated by one nucleotide and this adding reaction results in the generation of a spotlight that is recorded and measured by a CCD camera. The intensity of emitted light is proportional to the number of identical nucleotides added in a single flow. The whole sequencing run consists in flowing the nucleotides for hundreds times (cycles), depending on the required reads length. The higher read length obtainable by the FLX-Titanium platform is about 1,000 bases and is reached by 400 cycles.
18. The FLX-Titanium PTP could be physically partitioned in many regions, from 2 to 16, to sequence different samples in the same run. Obviously, each region has different sequencing yields in terms of number of reads; the higher is the separation (16) the lower is the amount of reads, which for the 16 regions separation is 25,000–40,000 reads per region, in opposition with the 2 regions separation that allows obtaining about 500,000 reads per region.

Acknowledgements

The author acknowledges the two "FIRB-Futuro in Ricerca" grants from the Italian Minister of Education, Universities and Research (MIUR): RBFR08U07M and RBFR126B8I.

References

1. Margulies M, Egholm M, Altman WE, Attiya S, Bader JS, Bemben L et al (2005) Genome sequencing in microfabricated high-density picolitre reactors. Nature 437:376–380
2. Ronaghi M (2001) Pyrosequencing sheds light on DNA sequencing. Genome Res 11: 3–11
3. Rizzi E, Lari M, Gigli E, De Bellis G, Caramelli D (2012) Ancient DNA studies: new perspectives on old samples. Genet Sel Evol 44(1):21–29
4. Morozova O, Hirst M, Marra M (2009) Applications of new sequencing technologies for transcriptome analysis. Annu Rev Genomics Hum Genet 10:135–151

5. Park PJ (2009) ChIP-seq: advantages and challenges of a maturing technology. Nat Rev Genet 10:669–680
6. Holt KE, Parkhill J, Mazzoni CJ, Roumagnac P, Weill F-X, Goodhead I, Rance R, Baker S, Maskell DJ, Wain J, Dolecek C, Achtman M, Dougan G (2008) High-throughput sequencing provides insights into genome variation and evolution in Salmonella Typhi. Nat Genet 40: 987–993
7. Peano C, Bicciato S, Corti G, Ferrari F, Rizzi E, Bonnal RJ, Bordoni R, Albertini A, Bernardi LR, Donadio S, De Bellis G (2007) Complete gene expression profiling of Saccharopolyspora erythraea using GeneChip DNA microarrays. Microb Cell Fact 6:37–53
8. Busam D, Feldblyum T, Ferriera S, Friedman R, Halpern A, Khouri H et al (2006) A Sanger/pyrosequencing hybrid approach for the generation of high-quality draft assemblies of marine microbial genomes. Proc Natl Acad Sci U S A 103:11240–11245
9. Fondi M, Rizzi E, Emiliani G, Orlandini V, Berna L, Papaleo MC, Perrin E, Maida I, Corti G, De Bellis G, Baldi F, Dijkshoorn L, Vaneechoutte M, Fani R (2013) The genome sequence of the hydrocarbon-degrading Acinetobacter venetianus VE-C3. Res Microbiol 164:439–449

Chapter 5

Bacterial Metabarcoding by 16S rRNA Gene Ion Torrent Amplicon Sequencing

Elio Fantini, Giulio Gianese, Giovanni Giuliano, and Alessia Fiore

Abstract

Ion Torrent is a next generation sequencing technology based on the detection of hydrogen ions produced during DNA chain elongation; this technology allows analyzing and characterizing genomes, genes, and species. Here, we describe an Ion Torrent procedure applied to the metagenomic analysis of 16S rRNA gene amplicons to study the bacterial diversity in food and environmental samples.

Key words Metagenomics, 16S rDNA, Ion PGM

1 Introduction

Next generation sequencing (NGS) technologies have opened new frontiers in microbial community analysis by providing a large amount of information to identify the microbial phylotypes present in different samples. It is known that more than 99 % of the environmental microorganisms are unculturable with the common laboratory methods, so their identification relies heavily on the sequencing of DNA from environmental samples. The advent of NGS gave an enormous impulse to these studies; in fact, NGS has revolutionized our understanding of the microbial communities in soil [1–4], sea [5], and our bodies [6, 7]; this revolution in sequencing technology, combined with the development of advanced bioinformatics tools, has revived metagenomic studies based on the 16S rRNA gene [8]. Because it is an excellent phylogenetic marker, analysis of 16S rRNA provides an accurate view of which microbial taxa are present in a given environmental sample [9].

In this context, we applied the Ion Torrent technology [10], a light-independent sequencing method based on the detection of

Alessio Mengoni et al. (eds.), *Bacterial Pangenomics: Methods and Protocols*, Methods in Molecular Biology, vol. 1231, DOI 10.1007/978-1-4939-1720-4_5,

Table 1
Primer pairs suitable for amplification of 16S

16S region	Domain	Name	Sequence	Amplicon length (bp)	Reference
V3	Eubacteria	338F	ACTCCTACGGGAGGCAGC	181	Mao [9]
V3	Eubacteria	519R	GTATTACCGCGGCKGCTG		
V1-V2	Universal	27F	AGAGTTTGATYMTGGCTCAG	311	Guss [11]
V1-V2	Universal	338R	GCTGCCTCCCGTAGGAGT		
V4-V5	Universal	515F	GTGCCAGCMGCCGCGGTAA	392	Turner [12]
V4-V5	Universal	907R	CCGTCAATTCMTTTRAGTTT		
V7-8-9	Universal	1100F	CAACGAGCGCAACCCT	392	Baker [13]
V7-8-9	Universal	1492R	GGTTACCTTGTTAYGACTT		

hydrogen ions generated by nucleotide addition to the nascent DNA chain, to the analysis of the microbial composition of food samples (packaged salad) and environmental samples (anaerobic digester slurry) by sequencing of 16S rDNA. The application of Ion Torrent sequencing to these samples allowed a much greater depth of sampling of the microbial diversity than either sequencing of cloned libraries or culturing of bacteria.

Most of the primers actually used in metagenomic analysis fail to amplify all bacterial and archaeal phyla in uncultured samples; a search of bibliography and of 16S rRNA gene sequences identified candidate primers corresponding to the following criteria: high coverage rate [9], size fragment not exceeding 400 bp (maximum present read length of the Ion Torrent chemistry), and absence of single mismatches in the four nucleotides close to the 3′ end of the primer [9]. Several pairs of primers corresponded to these parameters and targeting conservative regions of the 16S rDNAs gene were chosen (Table 1) [9, 11–13] and used to generate 16S rRNA gene amplicons informative for taxonomic assignment.

On the basis of the above criteria and a series of experimental tests, we chose the 338F and 519R primers, amplifying a 180-bp fragment of the hypervariable V3 region of the 16S rRNA gene; in this chapter, we describe the detailed protocols for creating amplicon libraries useful for Ion Torrent sequencing aimed at unveiling the microbial diversity in the samples. Up to 150,000 reads were produced from a single run, using a 314 chip; the reads were analyzed using appropriate bioinformatic tools, leading to the identification of more than 70 different bacterial genera/sample.

2 Materials

General purpose:

- Thermal cycler.
- 1.5-mL Eppendorf LoBind Tube.
- 0.2 mL PCR tube.
- P2, P20, P200, P1000 μL pipette set and filtered tips.
- Microcentrifuge.

Library preparation:

- Agarose for gel electrophoresis and gel electrophoresis apparatus.
- Phusion High-Fidelity DNA Polymerase (NEB) (optional).
- Primers: 338F (ACTCCTACGGGAGGCAGC), 519R (GTATTACCGCGGCKGCTG).
- Ion Plus Fragment Library Kit (Life Technologies) (*see* **Note 1**).
- Ion Xpress Barcode Adapters 1–16 Kit (optional for barcoded libraries, Life Technologies).
- Agencourt AMPure XP Reagent (Beckman Coulter).
- Magnet support DynaMag-2 magnet (Life Technologies).
- Freshly prepared 70 % ethanol.
- Library Quantification Kit For Ion Torrent platform (KAPA) or Ion Library Quantitation Kit (Life Technologies).
- 2100 Bioanalyzer and Agilent High Sensitivity DNA chip (Agilent Technologies).

Template-Positive Ion One Touch Ion sphere Particles preparation, enrichment, and quantification:

- Ion OneTouch Kit 200 template kit V2 (Life Technologies).
- Ion Sphere Quality control kit (Life Technologies).
- Ion One Touch 2 system (OT2 and ES instruments) (Life Technologies).
- Qubit 2.0 Fluorometer (Life Technologies).

Sequencing

- Ion PGM 200 Sequencing Kit.
- Ion PGM system.

Prepare all solutions using ultrapure water (prepared by purifying deionized water to attain a sensitivity of 18 MΩ cm at 25 °C) and analytical grade reagents.

2.1 Library Solutions

1. 70 % ethanol: 70 % solution in water.

2.2 Enrichment Solutions

1. Melt-Off solution: mix 865 μL of nuclease-free water, 125 μL of NaOH 1 M, and 10 μL of Tween 10 % (*see* **Note 2**).

3 Methods

3.1 Library Preparation

Unless otherwise indicated, all the steps are carried out at room temperature.

1. Amplicon production: prepare a PCR reaction according to your high-fidelity DNA polymerase guidelines (*see* **Note 3**). Amplification conditions:

   Hold: 98 °C 30 s,

   35 cycles: 98 °C for 10 s, 58 °C for 20 s, 72 °C for 10 s.

2. Amplicon verification: run an aliquot (2.5–5 μL) of the PCR reaction in a 2 % agarose gel to check the size and to verify the absence of nonspecific amplicons.

3. PCR product purification (*see* **Note 4**): transfer the PCR reaction (50 μL) to a 1.5 mL tube, add 90 μL of well-resuspended Agencourt AMPure XP Reagent (vortexed at max speed for 1 min), mix completely by pipetting, and incubate the mixture at room temperature for 5 min. Pulse-spin the tube and place it on a magnet support (DynaMag-2 magnet); when the solution is clear (about 2 min), remove and discard the supernatant without disturbing the bead pellet. Perform two 30-s washes with 300 μL of 70 % ethanol (*see* **Note 5**). The pellet must be covered by the ethanol. During the washes, turn the tube a little clockwise and counterclockwise. Carefully remove and discard all the supernatant. Air-dry the beads for 3–5 min, then remove the tube from the magnet and add 15 μL of nuclease-free water directly on the pellet. Mix completely by pipetting. Place again the tube on the magnet and, after the solution clears (about 1 min), transfer the supernatant containing the purified amplicons to a new tube without disturbing the pellet (*see* **Note 6**).

4. End repair: quantify the purified PCR product and elute 20–50 ng in a total volume of 79 μL of nuclease-free water in a 1.5 mL tube. Add 20 μL of 5× End Repair Buffer and 1 μL of End Repair Enzyme, mix by pipetting and incubate for 20 min at room temperature. Add 180 μL of Agencourt AMPure XP Reagent to the reaction and perform two washes of 30 s, as described in **step 3**, with 500 μL of 70 % ethanol and with a final elution in 25 μL of Low TE instead of 15 μL of nuclease-free water. Transfer the final supernatant to a 0.2 mL tube.

5. Adapter ligation (*see* **Note 7**): add to the 25 μL of end-repaired DNA the following reagents:

   For nonbarcoded libraries

   - 10× Ligase Buffer: 10 μL
   - Adapters: 2 μL.
   - dNTP Mix: 2 μL.
   - Nuclease-free Water: 51 μL.
   - DNA Ligase: 2 μL.
   - Nick Repair Polymerase: 8 μL.

   For barcoded libraries

   - 10× Ligase Buffer: 10 μL.
   - Adapters: 2 μL.
   - Ion P1 Adapter: 2 μL.
   - Ion Xpress Barcode X: 2 μL (X: chosen barcode).
   - dNTP Mix: 2 μL.
   - Nuclease-free Water: 49 μL.
   - DNA Ligase: 2 μL.
   - Nick Repair Polymerase: 8 μL.

   Then incubate the sample in a thermal cycler at 25 °C for 15 min and 72 °C for 5 min.

   Transfer the sample to a 1.5 mL tube and add 140 μL (for 100 base-read library use 180 μL) of Agencourt AMPure XP reagent and perform two washes of 30 s, as described in **step 3**, with 500 μL of 70 % ethanol and with a final elution in 25 μL of Low TE instead of 15 μL of nuclease-free water. Transfer the final supernatant to a 0.2 mL tube.

6. Library amplification (*see* **Note 8**): add to the sample from previous step 100 μL of Platinum PCR SuperMix High Fidelity and 5 μL of Library Amplification Primer Mix. Split the reaction in two 0.2 mL tubes (about 65 μL each) and run the following PCR program:

   Hold: 95 °C for 5 min.

   Five to seven cycles*: 95 °C for 15 s, 58 °C for 15 s, 70 °C for 1 min.

   *Five cycles for 50 ng of initial purified PCR product, seven cycles for 20 ng.

   Combine the two samples in a 1.5 mL tube and add 195 μL (for 100 base-read library use 180 μL) of Agencourt AMPure XP reagent and perform two washes of 30 s, as described in **step 3**, with 500 μL of 70 % ethanol and with a final elution in 20 μL of Low TE instead of 15 μL of nuclease-free water. Transfer the final supernatant to a 0.2 mL tube.

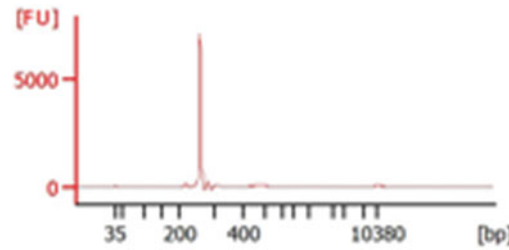

Fig. 1 Library qualification with Agilent High Sensitivity DNA chip. A single peak of about 250 bp (180 bp of amplicon + 70 bp of adapters) can be observed

STORAGE: split the amplified library in ten 2-μL aliquots in 0.2 mL tubes and store them at −20 °C in order to reduce the number of freeze-thaw cycles (*see* **Note 9**). You can store immediately eight aliquots and keep one on ice for **step 7** and one in the fridge for the emulsion PCR (ePCR), if you plan to run it within 24 h.

7. Library qualification and quantification: make a dilution 1:10 with 1 μL of amplified library and analyze two replicas on the Agilent Bioanalyzer with an Agilent High Sensitivity DNA chip. Possible contaminants that can be observed in the electropherogram are nonspecific amplicons, primer-dimers (close to the smaller marker), or concatemers. They reduce strongly the efficiency of the ePCR and of the following sequencing. Working with a single amplicon (the pooling of different amplicons in equimolar amounts before **step 4** is an option), a single narrow peak (Fig. 1) represents the best results in order to proceed with quantification and ePCR.

   Quantify the library with a qPCR kit (e.g., Ion Library Quantitation Kit (TaqMan) or KAPA Library Quantification Kit for Ion Torrent (Sybr Green)) and calculate the dilution factor according to the kit instructions, in order to obtain a diluted library with a molarity of 26 pM (*see* **Notes 10** and **11**).

3.2 "Template-Positive Ion One Touch Ion Sphere Particles" Preparation

1. Prepare the Ion OneTouch DL instrument and install all the instrument disposables according to the instrument instruction manual (*see* **Note 12**).
2. After installation of disposable injector, prepare the amplification solution by mixing 280 μL of nuclease-free water, 500 μL of Ion OneTouch 2× reagent mix, 100 μL of Ion OneTouch enzyme mix, and 20 μL of diluted library in a 1.5 mL eppendorf.
3. Vortex the Ion sphere Particles very well (at least 1 min), add 100 μL to the amplification solution, and pipette up and down.
4. Fill the Ion OneTouch reaction filter assembly with the amplification solution (1,000 μL) by pipetting the solution slowly into the sample port.

5. Add 1,000 μL of reaction oil by pipetting the solution slowly into the sample port.
6. Change the tip and add other 500 μL of reaction oil by pipetting the solution slowly into the sample port.
7. Gently keep the Ion OneTouch reaction filter assembly and rotate the assembly until it is completely inverted and the three ports are face down.
8. Insert the three ports of the Ion OneTouch reaction filter assembly into the three holes of the instrument.
9. Select the appropriate run program and press run.
10. After the run, immediately remove all the liquid in the recovery tubes but keeping almost 50 μL without disturbing the pellet (*see* **Note 13**).
11. Resuspend the pellet (template-positive ion sphere particles) by pipetting up and down.
12. In a new tube, merge the two samples and add Ion OneTouch Wash solution to 1 mL.
13. Centrifuge for 2.5 min at 15,500 × *g*.
14. At the end of the run, immediately remove all the liquid in the recovery tubes but keeping almost 100 μL without disturbing the pellet.
15. Resuspend the pellet (template-positive ion sphere particles) by vortexing 5 s and transfer 2 μL in a new 0.2 mL PCR tube to perform Qubit quality control of the unenriched Ion Sphere Particles.

3.3 "Template-Positive Ion One Touch Ion Sphere Particles" Enrichment

1. Prepare the Ion OneTouch ES instrument according to the instrument instruction manual.
2. Resuspend the MyOne Streptavidin C1 beads by vortexing for 30 s and add 13 μL to a new 1.5 mL tube.
3. Add 130 μL of MyOne Streptavidin beads wash solution to the tube containing the MyOne Streptavidin C1 beads, pipette up and down, and transfer in a DynaMag magnet for 2 min.
4. Remove and discard the liquid avoiding touching the pellet (Streptavidin beads).
5. Add 130 μL of MyOne Streptavidin beads wash solution to the pellet, vortex for 30 s, and centrifuge for few seconds.
6. Transfer the 130 μL of the resuspended MyOne Streptavidin beads into well 2 of the eight-well strip.
7. Prepare and transfer each solution in the appropriate well of the eight-well strip:
   - Well 1: template positive sample.
   - Well 2: MyOne Streptavidin beads.

- Well 3, 4, 5: Ion OneTouch wash solution.
- Well 6, 8: empty.
- Well 7: 300 μL of melt-off solution.

8. Perform the run following instrument instructions (the run takes about 30 min).
9. At the end of the run, ensure that the volume in the 0.2 mL tube is approximately 200 μL (*see* **Note 14**).
10. Centrifuge the 0.2 mL tube containing the enriched sphere particles at 15,500 × *g* for 1.5 min.
11. Remove all but 10 μL of supernatant by slowly pipetting and add 200 μL of IonTouch wash solution (*see* **Note 15**).
12. Pipet up and down to resuspend the pellet and centrifuge at 15,500 × *g* for 1.5 min.
13. If a brown pellet is not visible, proceed directly with **step 14**. Otherwise, if you can see a brown pellet, this means that MyOne Streptavidin beads are present in the pellet; therefore further cleaning with the Dynamag-2 magnet is necessary; in the latter case, insert the tube containing the enrichment solution in a Dynamag-2 magnet for 4 min and recover the supernatant containing the enriched sphere particles by pipetting gently; then you can proceed with the **step 14**.
14. Remove all but 10 μL of supernatant and add 90 μL of IonTouch wash solution (*see* **Note 15**), resuspend by gently pipetting in a total volume of 100 μL. Enriched ISPs can be stored at 2–8 °C for up to 1 week.
15. Transfer 10 μL in a new 0.2 mL PCR tube to perform Qubit quality control.
16. Perform ion sphere particle quality control through a Qubit 2.0 Fluorimeter.

3.4 Quality Control of Ion OneTouch Ion Sphere Particles

1. Add 19 μL of Annealing Buffer and 1 μL of Ion Probes to the two tubes containing the enriched and un-enriched Ion sphere Particles and perform the following protocol in a thermal cycler: 95 °C for 2 min and 37 °C for 2 min (*see* **Note 16**).
2. At the end of the run, add 200 μL of Quality Control Wash Buffer to the two tubes and centrifuge at 15,500 × *g* for 1.5 min.
3. Remove the supernatant without disturbing the pellet (*see* **Note 15**) leaving approximately 10 μL.
4. Repeat **steps 2** and **3** two times and after the final wash add 190 μL of Quality Control Wash Buffer to have 200 μL of finale volume.
5. Set the Qubit 2.0 Fluorimeter according to the instrument instruction manual and read the standards (Alexa Fluor 488

Calibration Standard and Alexa Fluor 647 Calibration Standard); record the readings.

6. Read the two samples (enriched and unenriched ion sphere particles) and negative control (200 μL of Quality Control Wash Buffer) in both wavelengths (488 and 647); record the readings.
7. Download the Qubit 2.0 Easy calculator Spreadsheet file (*see* **Note 17**) and enter all the readings each one in the appropriate cell.
8. Enter also the conversion factor derived from the lot of the kit used (*see* **Note 17**).
9. The percent of template ISPs is automatically calculated in the sheet; this percent must comprise between 10 and 30 % for unenriched and more than 50 % for enriched (*see* **Note 18**).

3.5 314 Ion Chip Loading and Sequencing

1. Clean and initialize the Ion PGM system according to the instrument instruction manual (*see* **Notes 19–21**).
2. When the initialization is complete, mix completely by pipetting the enriched ISPs and transfer half of the volume (about 45 μL, *see* **Note 22**) to a new 0.2 mL tube. Add 5 μL of vortexed Control Ion Sphere Particles and 100 μL of Annealing Buffer. Mix completely by pipetting and centrifuge the tube for 2 min at 15,500×*g*. Carefully remove all the supernatant except 3 μL without disturbing the pellet and add 3 μL of Sequencing Primer, for a total of 6 μL. Mix completely by pipetting in order to dissolve the pellet. Incubate in a thermal cycler at 95 °C for 2 min and then at 37 °C for 2 min. Transfer the tube to room temperature and proceed with the chip check and wash as described in the instrument instruction manual.
3. After the chip wash, add 1 μL of Ion PGM 200 Sequencing Polymerase (for a total of 7 μL) and mix completely by pipetting, setting the micropipette to 4 μL in order to avoid bubble formation. Incubate for 5 min at room temperature and load the chip according to the latest developed procedures.
4. Proceed with the sequencing run, using the AmpliSeq application and setting the instrument to 500 flows (125 cycles) for the 200-base reads.

3.6 Quality Check and Filtering of Raw Sequencing Reads

1. Select reads matching the degenerated PCR primers (forward and reverse) and trim the primer sequences by using Cutadapt [14] with a minimum primer overlap of ten residues and an error rate of 0.2.
2. Trim the selected reads at the 3′ end with the Mott algorithm, then remove reads shorter than 80 nucleotides (the minimum distance of the V3 variable region) and with an average quality score (Sanger) <20. For this purpose use the Perl script

trim-fastq.pl included in PoPoolation [15], a collection of tools to facilitate population genetic studies of next generation sequencing data from pooled individuals.

3. Convert the sequence file from fastq to fasta format, the latter required for **step 4**.
4. Screen the high quality reads for artificial chimeric formations by using the UCHIME algorithm [16] and, as a reference of 16S rRNA gene sequences, the "Gold" database (http://drive5.com/uchime/gold.fa).
5. After the removal of chimeric formations, convert the sequence file from fasta to fastq format, the latter usable in **step 6** (but also fasta format is supported), replenishing the quality scores.
6. De-noise the read sequences by applying a modified version of run-length encoding [17] implemented in Acacia [18]. Maintain the default configuration parameters, with the exception of: AVG_QUALITY_CUTOFF (=20), FASTA (=FALSE), FASTQ (=TRUE), REPRESENTATIVE_SEQUENCE (=Max), SIGNIFICANCE_LEVEL (=−4).

3.7 Taxonomy Assignment and Taxonomic Analysis

1. Calculate the Operational Taxonomic Units (OTUs) on the basis of a clustering analysis. Cluster the high quality reads (de-noised by Acacia) at 97 % identity threshold using the complete linkage clustering method employed by ESPRIT [19].
2. Calculate the species richness estimator ACE and rarefaction curves by ESPRIT, excluding the reads from singleton OTUs to avoid the problem of species overestimation [20].
3. Assign the taxonomic annotation with GAST [21] process, using, as a reference, the VAMPS rRNA gene database [22]. Keep the default parameters with the exception of the identity threshold used to select the closest reference(s) of each sequence during the comparison to the reference database: set it at 90 %.
4. Calculate Shannon and Simpson diversity indices [23] and the Pielou's evenness index [24], using the GAST assignments at the taxonomic rank of *genus*. Computation can be performed with the Perl function *genus_assignment*, whose argument is the name of the file generated by the GAST process.

4 Notes

1. Amplicon libraries can be prepared with a dedicated kit (Ion AmpliSeq Library Kit) but also with the genomic DNA kit (Ion Plus Fragment Library Kit). Since the latter contains reagents for at least ten library preparations, and since the cost for single preparation is quite similar, a non-high-throughput

lab that works both with genome and amplicon sequencing could take advantage from the second option.

2. Melt-Off solution and NaOH must be prepared fresh (each enrichment experiment).
3. High-fidelity DNA polymerase is highly recommended in order to avoid errors that could affect the reliability of the sequencing results. Platinum PCR SuperMix High Fidelity is an option and is furnished with the Ion Plus Fragment Library Kit, since it is required for the library amplification. We used Phusion High-Fidelity DNA Polymerase.
4. For amplicons greater than 100 bp, an efficient removal system of dNTPs, primers, primer dimers, salts, and other contaminants is the Agencourt AMPure XP Reagent. Aliquot the Agencourt AMPure XP Reagent in small aliquots, in order to warm only the required amount of Reagent. Allow the Reagent to reach room temperature (about 30 min) prior to use. All the discarded supernatants from the purification steps can be tested on the magnet to check for beads presence.
5. Always use freshly prepared 70 % ethanol since higher concentrations are inefficient in washing DNA molecules smaller than 100 bp (dNTPs, primers, primer-dimers), while lower concentrations could wash also your sample. Fresh ethanol can be prepared during the warming of the Agencourt AMPure XP Reagent.
6. After each purification step it is possible to stop the procedure and store the samples at –20 °C. However, since the procedure is quite rapid, we suggest avoiding freeze-thaw cycles and completing the library preparation without stops.
7. To perform multiple libraries sequencing in a single run, barcoded adapters are available with the Ion Xpress Barcode kit.
8. Library amplification is optional and depends on the concentration of the unamplified library. To verify this, a qPCR quantification is suggested by the kit guidelines. If the library does not need an amplification, you can proceed directly to the dilution factor calculation with the data obtained. However we strongly recommend to avoid the quantification and to always amplify the library. In our experience the quantification of unamplified libraries always indicated that no amplification was required (e.g., dilution factor of 50, that means you can perform about 50 ePCR runs) and always the ePCR was inefficient and the sequencing poor. The amplification of the libraries always solves the problem. Moreover, a qPCR is time and money consuming. If the quantification pre-amplification indicates that an amplification is required, after this amplification a second qPCR will be necessary to calculate the dilution factor.

9. Ion Torrent libraries tend to degrade in few months, especially if they suffer repeated freeze-thaw cycles. If you need to run an ePCR with a library older than 2 months, it is recommended to run it on an Agilent High Sensitivity DNA chip to verify the library integrity.
10. Although a quantification via qPCR is more accurate, it is possible to quantify amplified libraries directly with the Agilent High Sensitivity DNA chip, but at least three replicas are required. Since the molarity of the diluted library required for an ePCR is 26 pM, the dilution factor is easily computable using the molarity value of the peak representing the library (corrected for the dilution factor used for the chip loading).
11. The primers used in the amplification are designed on the adapters, as the ones of the available qPCR kits for library quantification. The purification step after the amplification is fundamental in order to remove primer-dimers that can otherwise produce a false signal during the qPCR.
12. Before proceeding with the installation of the new disposables, we suggest performing this procedure: insert an old disposable injector into the upper port of the Ion OneTouch Injector Hub, keeping a paper towel under the lower port of the hub. This operation allows removing liquid residues from the previous run.
13. At the end of the Ion Touch run, if more than few minutes have passed, centrifuge one more time to avoid sphere particles resuspension.
14. If the volume of the enriched ISPs is less than 200 μL, the ES system is probably not calibrated and you have lost your sample. You can proceed with the calibration of the instrument according to the instruction manual. In order to avoid this, routinely perform the residual volume test according to the instrument instruction manual.
15. During the removal of the supernatant, make sure not to disturb the pellet; remove the supernatant by pipetting very slowly.
16. The Alexa Fluor 488 and 647 are very photosensitive, so be careful and perform all steps away from strong light sources.
17. Download the Qubit 2.0 Easy calculator Spreadsheet file and Conversion factor from: http://ioncommunity.lifetechnologies.com/community/products/pgm/user_guides_and_bulletins.
18. The optimal amount of unenriched library corresponds to a range of 10–30 %, but this range is not so stringent, in fact we recommended to observe especially the amount of enriched library; if this amount is more than 50 %, even slightly, proceed to sequencing; if the amount of enriched library is less than

50 %, you can proceed to sequencing but the results cannot be satisfactory (reads number, low quality reads, polyclonal spheres, etc.).

19. Be sure that the room temperature is lower than 25 °C and that the Ion PGM has enough free space on its back to let the fan work. Sequencing efficiency is strongly affected by temperature and overheating of the instrument leads to sequencing errors (easily detectable by a low 50AQ17/Num ratio of the test fragments).
20. For the initialization you can use an old undamaged chip properly stored in its bag.
21. With a 200-base reads sequencing, two runs can be performed with one initialization of the instrument. You can run two different libraries or perform a replica of the same library, but this requires the use of a 314 chip (*see* **Note 22**).
22. Unlike the 316 and 318 chip types, the 314 chip type requires only half of the enriched ISPs. You can load a second chip as a replica of the sequencing. However, if you do not plan to perform a replica, you can use the enriched ISPs for a single run and force the chip loading density using a higher amount of enriched ISPs. Chip loading is still not an exact science and optimal chip loading is not always obtained. Perform this attempt only if you observe frequently a poor loading of the 314 chip type.

Acknowledgments

Work supported by the Italian Ministry of Agriculture (Biomassval project), the Italian Ministry of Economic Development (Foodflavr project), and the Italian Ministry of Research ("Integrated Knowledge for the Sustainability and Innovation of Italian Agri-Food" project).

References

1. Venter JC, Remington K, Heidelberg JF, Halpern AL, Rusch D, Eisen JA, Wu D, Paulsen I, Nelson KE, Nelson W et al (2004) Environmental genome shotgun sequencing of the Sargasso Sea. Science 304:66–74
2. Tyson GW, Chapman J, Hugenholtz P, Allen EE, Ram RJ, Richardson PM, Solovyev VV, Rubin EM, Rokhsar DS, Banfield JF (2004) Community structure and metabolism through reconstruction of microbial genomes from the environment. Nature 428:37–43
3. Roesch LFW, Fulthorpe RR, Riva A, Casella G, Hadwin AKM, Kent AD, Daroub SH, Camargo FAO, Farmerie WG, Triplett EW (2007) Pyrosequencing enumerates and contrasts soil microbial diversity. ISME J 1: 283–290
4. Lauber CL, Hamady M, Knight R, Fierer N (2009) Pyrosequencing-based assessment of soil pH as a predictor of soil bacterial community structure at the continental scale. Appl Environ Microbiol 75:5111–5120
5. Sogin ML, Morrison HG, Huber JA, Mark Welch D, Huse SM, Neal PR, Arrieta JM, Herndl GJ (2006) Microbial diversity in the deep sea and the underexplored "rare

biosphere". Proc Natl Acad Sci U S A 103: 12115–12120

6. The human Microbiome Consortium (2012) Structure, function and diversity of the healthy human microbiome. Nature 486: 207–214
7. Grice EA, Kong HH, Conlan S, Deming CB, Davis J, Young AC, Bouffard GG, Blakesley RW, Murray PR, Green ED et al (2009) Topographical and temporal diversity of the human skin microbiome. Science 324: 1190–1192
8. Tringe SG, Hugenholtz P (2008) A renaissance for the pioneering 16S rRNA gene. Curr Opin Microbiol 11:442–446
9. Mao D-P, Zhou Q, Chen C-Y, Quan Z-X (2012) Coverage evaluation of universal bacterial primers using the metagenomic datasets. BMC Microbiol 12:66
10. Junemann S, Prior K, Szczepanowski R, Harks I, Ehmke B, Goesmann A, Stoye J, Harmsen D (2012) Bacterial community shift in treated periodontitis patients revealed by ion torrent 16S rRNA gene amplicon sequencing. PLoS One 7:e41606
11. Guss AM, Roeselers G, Newton ILG, Young CR, Klepac-Ceraj V, Lory S, Cavanaugh CM (2011) Phylogenetic and metabolic diversity of bacteria associated with cystic fibrosis. ISME J 5:20–29
12. Turner S, Pryer KM, Miao VP, Palmer JD (1999) Investigating deep phylogenetic relationships among cyanobacteria and plastids by small subunit rRNA sequence analysis. J Eukaryot Microbiol 46:327–338
13. Baker GC, Smith JJ, Cowan DA (2003) Review and re-analysis of domain-specific 16S primers. J Microbiol Methods 55: 541–555
14. Martin M (2011) Cutadapt removes adapter sequences from high-throughput sequencing reads. EMBnet J 17(1):10–12
15. Kofler R, Orozco-terWengel P, De Maio N, Pandey RV, Nolte V, Futschik A, Kosiol C, Schlotterer C (2011) PoPoolation: a toolbox for population genetic analysis of next generation sequencing data from pooled individuals. PLoS One 6:e15925
16. Edgar RC, Haas BJ, Clemente JC, Quince C, Knight R (2011) UCHIME improves sensitivity and speed of chimera detection. Bioinformatics 27:2194–2200
17. Reeder J, Knight R (2010) Rapidly denoising pyrosequencing amplicon reads by exploiting rank-abundance distributions. Nat Methods 7: 668–669
18. Bragg L, Stone G, Imelfort M, Hugenholtz P, Tyson GW (2012) Fast, accurate error-correction of amplicon pyrosequences using Acacia. Nat Methods 9:425–426
19. Sun Y, Cai Y, Liu L, Yu F, Farrell ML, McKendree W, Farmerie W (2009) ESPRIT: estimating species richness using large collections of 16S rRNA pyrosequences. Nucleic Acids Res 37:e76
20. Kunin V, Engelbrektson A, Ochman H, Hugenholtz P (2010) Wrinkles in the rare biosphere: pyrosequencing errors can lead to artificial inflation of diversity estimates. Environ Microbiol 12:118–123
21. Huse SM, Dethlefsen L, Huber JA, Mark Welch D, Relman DA, Sogin ML (2008) Exploring microbial diversity and taxonomy using SSU rRNA hypervariable tag sequencing. PLoS Genet 4:e1000255
22. Huse SM, Welch DM, Morrison HG, Sogin ML (2010) Ironing out the wrinkles in the rare biosphere through improved OTU clustering. Environ Microbiol 12:1889–1898
23. Magurran AE (2003) Measuring biological diversity. Blackwell, Oxford
24. Pielou E (1966) The measurement of diversity in different types of biological collections. J Theor Biol 13:131–144

Chapter 6

The Illumina-Solexa Sequencing Protocol for Bacterial Genomes

Zhenfei Hu, Lei Cheng, and Hai Wang

Abstract

Based on reversible dye-terminators technology, the Illumina-solexa sequencing platform enables rapid sequencing-by-synthesis (SBS) of large DNA stretches spanning entire genomes, with the latest instruments capable of producing hundreds of gigabases of data in a single sequencing run. Illumina's NGS instruments powerfully combine the flexibility of single reads with short- and long-insert paired-end reads, and enable a wide range of DNA sequencing applications. Here, we describe the paired-end library preparation with an average insert size of 470 bp, 2 kbp, and 6 kbp, together with the DNA cluster generation and sequencing procedure of *E. coli* O104:H4 genome on Illumina Hiseq 2000 platform.

Key words Illumina, Solexa, Hiseq, Reversible dye-terminators, Sequencing by synthesis, Single reads, Paired-end reads, Library preparation, Cluster generation

1 Introduction

Next-generation sequencing technology is an exciting tool that may help us to decipher the genome architecture and the evolution of bacteria. One of the most powerful and user-friendly NGS platform is Hiseq by Illumina Inc. In this approach, genomic DNA is fragmented, end-repaired, and ligated to unique sequencing adaptors to form Sequencing-by-synthesis DNA libraries. After verification and quantization, DNA library is loaded to the sequencing flowcell and form DNA cluster through bridge PCR. Flowcells with clusters can be loaded to the Sequencer and analyzed automatically.

Here, we describe the sequencing of the *E. coli* O104:H4 strain genome on the Illumina HiSeq platform, in accordance with the manufacturer's instructions [1]. An initial single-end run was used to correct errors in the previously reported Ion Torrent sequence, principally in homopolymeric tracts. We later performed paired-end and mate-pair sequencing on this platform, exploiting libraries with insert sizes of 470 bp, 2 kb, and 6 kb, and generated

Alessio Mengoni et al. (eds.), *Bacterial Pangenomics: Methods and Protocols*, Methods in Molecular Biology, vol. 1231, DOI 10.1007/978-1-4939-1720-4_6,

enough data (1 Gb, 576 Mb, and 576 Mb from each library, respectively) to create a high-quality draft genome sequence within 2 weeks after receipt of the DNA samples.

2 Materials

Prepare all solutions using ultrapure water (prepared by purifying deionized water to attain a sensitivity of 18 MΩ cm at 25 °C) and analytical-grade reagents. Prepare and store all reagents at room temperature (unless indicated otherwise). Diligently follow all waste disposal regulations when disposing waste materials. Do not add sodium azide to the reagents.

2.1 Equipment Required for Library Preparation

Benchtop microcentrifuge.

Benchtop centrifuge with swing-out rotor.

Dark Reader transilluminator or UV transilluminator.

Disposable scalpels.

Electrophoresis unit.

Gel trays and tank.

Thermal cycler or heat block.

Covaris S2 sonicator.

2.2 Reagents and Consumables

Genomic DNA sample prep kit containing:

(a) T4 DNA ligase buffer with 10 mM ATP, part # 1000534.

(b) Klenow DNA polymerase, part # 1000515.

(c) Klenow buffer, part # 1000535.

(d) 2× DNA ligase buffer, part # 1000523.

(e) Phusion DNA polymerase (Finnzymes Oy), part # 1000524.

(f) 10 mM dNTPs mix, part # 1001932.

(g) T4 PNK, part # 1000519.

(h) 1 mM dATP, part # 1000520.

(i) Adapter oligo mix, part # 1000521.

(j) PCR primer 1.1, part # 1000537.

(k) T4 DNA polymerase, part # 1000514.

(l) Empty.

(m) Klenow fragment (3′ to 5′ exo minus), part # 1000536.

(n) DNA ligase, part # 1000522.

(o) PCR primer 2.1, part # 1000538.

TruSeq PE Cluster Kit v3-cBot-HS, Illumina catalog # PE-401-3001.
TruSeq SBS Kit v3-HS, Illumina catalog # FC-401-3001.
TE buffer: 10 mM Tris, 1 mM EDTA, pH 8.0 (*see* **Note 1**).
QIAquick PCR purification kit (QIAGEN, #28104).
MinElute PCR purification kit (QIAGEN, part # 28004).
Certified low-range Ultra Agarose (BIO-RAD, part # 161-3106).
50× TAE buffer: 2 M Tris, 1 M Acetic, 100 mM EDTA, pH 8.5 (*see* **Note 1**).
Ethidium bromide.
Loading buffer (50 mM Tris pH 8.0, 40 mM EDTA, 40 % (w/v) sucrose).
Low molecular weight DNA ladder (NEB, part # N3233L).

3 Methods

Carry out all procedures at room temperature unless otherwise specified.

Genomic DNA was extracted and purified using a conventional SDS lysis and phenol–chloroform method. The DNA sample to be processed should be highly pure, having an OD260/280 ratio of between 1.8 and 2, and should be as intact as possible.

3.1 Fragment the Genomic DNA

1. 3 μg of DNA was dissolved in TE buffer to a total volume of 100 μl and fragmented by sonication (Covaris S2, Massachusetts, USA) to a size distribution of 50–300 bp (*see* **Note 2**).
2. Follow the instructions in the QIAquick PCR Purification Kit to purify the sample solution and concentrate it on one QIAquick column, eluting in 30 μl of EB.

3.2 Perform the End-Repair

1. Prepare the following reaction mix:
   (a) DNA sample (30 μl)
   (b) Water (45 μl)
   (c) T4 DNA ligase buffer with 10 mM ATP (10 μl)
   (d) dNTPs mix (4 μl)
   (e) T4 DNA polymerase (5 μl)
   (f) Klenow DNA polymerase (1 μl)
   (g) T4 PNK (5 μl)
   The total volume should be 100 μl.
2. Incubate in the thermal cycler for 30 min at 20 °C (*see* **Notes 3** and **4**).
3. Follow the instructions in the QIAquick PCR Purification Kit to purify on one QIAquick column, eluting in 32 μl of EB.

3.3 Add "A" Bases to the 3′ End of the DNA Fragments

1. Prepare the following reaction mix:
   (a) DNA sample (32 μl)
   (b) Klenow buffer (5 μl)
   (c) dATP (10 μl)
   (d) Klenow exo (3′ to 5′ exo minus) (3 μl)
   The total volume should be 50 μl.
2. Incubate for 30 min at 37 °C (*see* **Note 4**).
3. Follow the instructions in the MinElute PCR Purification Kit to purify on one QIAquick MinElute column, eluting in 10 μl of EB.

3.4 Ligate Adapters to DNA Fragments

1. Prepare the following reaction mix:
   (a) DNA sample (10 μl)
   (b) DNA ligase buffer (25 μl)
   (c) Adapter oligo mix (10 μl)
   (d) DNA ligase (5 μl)
   The total volume should be 50 μl.
2. Incubate in a thermal cycler for 15 min at 20 °C.
3. Follow the instructions in the QIAquick PCR Purification Kit to purify on one QIAquick column, eluting in 30 μl of EB.

3.5 Purify Ligation Products

1. Prepare a 50 ml, 2 % agarose gel with distilled water and TAE. Final concentration of TAE should be 1× at 50 ml (*see* **Note 5**).
2. Add ethidium bromide (EtBr) after the TAE-agarose has cooled. Final concentration of EtBr should be 400 ng/ml (i.e., add 20 μg EtBr to 50 ml of 1× TAE) (*see* **Note 6**).
3. Add 3 μl of loading buffer to 8 μl of the ladder. Add 10 μl of loading buffer to 30 μl of the DNA from the purified ligation reaction.
4. Load all of the ladder solution to one lane of the gel. Load the entire sample in another lane of the gel, leaving a gap of at least one empty lane between ladder and sample (*see* **Note 7**).
5. Run the gel at 120 V for 60 min. View the gel on a Dark Reader transilluminator or a UV transilluminator (*see* **Note 8**).
6. Excise a region of gel with a clean scalpel. The gel slice should contain the material in the 180–200 bp range (*see* **Note 9**).
7. Using a Gel Extraction Kit, do one of the following:
   (a) If the gel slice is less than 400 mg, use one column from a QIAquick Gel Extraction Kit and elute in 30 μl EB.
   (b) If the gel slice is more than 400 mg, use two MinElute columns, elute each one in 15 μl EB, and pool.

3.6 Enrich the Adapter-Modified DNA Fragments by PCR

1. Prepare the following PCR reaction mix:
   (a) DNA (1 μl)
   (b) Phusion DNA polymerase (Finnzymes Oy) (25 μl)
   (c) PCR primer 1.1 (1 μl)
   (d) PCR primer 2.1 (1 μl)
   (e) Water (22 μl)
   The total volume should be 50 μl.
2. Amplify using the following PCR protocol:
   (a) 30 s at 98 °C
   (b) 10 cycles of:
   - 10 s at 98 °C
   - 30 s at 65 °C
   - 30 s at 72 °C

   (c) 5 min at 72 °C
   (d) Hold at 4 °C
3. Follow the instructions in the QIAquick PCR Purification Kit to purify on one QIAquick column, eluting in 30 μl of EB (*see* **Note 10**)

3.7 Verify the Library

1. Determine the concentration of the library by measuring its absorbance at 260 nm. The yield from the protocol should be between 500 and 1,000 ng of DNA.
2. Measure the 260/280 ratio. It should be approximately 1.8.
3. Load 10 % of the volume of the library on a gel and check that the size range is as expected. It should be similar in size to the size-range excised during the gel purification step.

3.8 Perform the Cluster Generation Using the cBot System

Use the TruSeq PE Cluster Kit v3-cBot-HS to perform the cluster generation exactly following the Reagent Preparation Guide for TruSeq® PE Cluster Kit v3 and TruSeq® Dual Indexing Sequencing Primer Box (Part# 15023336 Rev. E) and cBot User Guide(Illumina Part# 15006165 Rev.K).

3.9 Perform the Sequencing Run Using the Hiseq 2000 System

Use the TruSeq SBS Kit v3-HS to perform the sequencing run exactly following the TruSeq™ SBS Kit v3(200 Cycles) Reagent Preparation Guide(Illumina Part#15023333 Rev.C) and HiSeq® 2000 System User Guide(Illumina Part # 15011190 Rev. T).

3.10 Creation of a Hybrid Assembly Using Ion Torrent PGM Data and Illumina Single-End Data

Ion Torrent and Illumina read data were quality filtered before assembly including removal of adapter contamination. The Ion Torrent PGM assembly from seven chips of Ion Torrent 314 data were assembled with Newbler 2.0.00.22. Illumina single-end data (taken from paired-end in-progress pair-end run) were assembled

using SOAPdenovo 1.06. (with k-mer of 51 and parameters "-d 1,-R") [2]. Assemblies were combined using AMOS minimus2 1.59 with parameters REFCOUNT = 0, OVERLAP = 50, MINID = 94, MAXTRIM = 10^2. The resulting assembly consisted of 451 contigs greater than 200 bp with an N50 of 53,266 bp. The largest contig was 204,342 bp.

3.11 Creation of a Draft Genome Scaffold Assembly Using Illumina Paired-End and Mate-Pair Reads [3]

A draft de novo assembly was produced using SOAPdenovo version 1.05. Contigs were first assembled using the 470 bp paired-end library initially using a k-mer value of 45 for de Bruijn graph construction. These were subsequently scaffolded in a hierarchical fashion using 2 kb followed by 6 kb mate-pair libraries by way of the rank parameter in the SOAPdenovo configuration file. Other parameters supplied to SOAPdenovo included -F to attempt to fill gaps in scaffolds. Where possible, in order to fill remaining scaffold gaps, local information available from the abundant mate-pair data was utilized by the GapCloser utility which was run over the assembly output with a k-mer size of 23. Both scaffolds and un-scaffolded contigs were used in further analysis, with the exception of contigs smaller than 200 bp, which were excluded.

De novo assembly produced 24 scaffolds plus 75 un-scaffolded contigs. The largest scaffold was 757,969 bp, the smallest was 552 bp. Scaffold N50 was 403,980 bp. After gap filling the scaffolds contained 143 distinct stretches of gaps (represented as ambiguous 'N' bases) comprising 94,491 bp of sequence.

4 Notes

1. Concentrated HCl (12 N) can be used at first to narrow the gap from the starting pH to the required pH. From then on it would be better to use a series of HCl (e.g., 6 N and 1 N) with lower ionic strengths to avoid a sudden drop in pH below the required pH.
2. Fill the water tank of the Covaris S2 sonicator to required water level using the ultra pure water. Do not open the lid of water tank during fragmentation.
3. When not doing thermocycling incubation, do not close the heat lid of the thermocycler.
4. After the incubation step of end-repair and A-tailing, the product should be purified immediately.
5. Chemical Hazard: TAE buffer.
6. Bio Hazard: Ethidium bromide (EtBr).
7. Purifying multiple samples on a single gel is not recommended due to the risk of cross-contamination between libraries.
8. Whenever working with darkreader, please wear appropriate eye protector.

9. In this step, it is recommended dispose the scalpel after each cut.
10. To avoid amplicon contamination, the PCR product should not be purified in the PCR-setup room.

Acknowledgements

This work was supported by *E. coli* O104:H4 Genome Analysis Crowd-Sourcing Consortium.

References

1. Rohde H, Qin J, Cui Y (2011) Open-source genomic analysis of Shiga-toxin-producing *E. coli* O104:H4. N Engl J Med 365:718–724
2. Frank C, Werber D, Cramer JP et al (2011) Epidemic profile of Shiga-toxin-producing *Escherichia coli* O104:H4 outbreak in Germany—preliminary report. N Engl J Med 365:1771–1780
3. Robert Koch Institute (2011) Information update on EHEC/HUS outbreak. http://www.rki.de/nn_217400/EN/Home/PM082011.html

Chapter 7

High-Throughput Phenomics

Carlo Viti, Francesca Decorosi, Emmanuela Marchi, Marco Galardini, and Luciana Giovannetti

Abstract

Standard protocols are available in order to apply Phenotype MicroArray (PM) technology to characterize different groups of microorganisms. Nevertheless, there is the need to pay attention to several crucial steps in order to obtain high-quality and reproducible data from PM, such as the choice of the Dye mix, the type and concentration of the carbon source in metabolic experiments, the use of a buffered medium. A systematic research of auxotrophies in strains to be tested should be carefully evaluated before starting with PM experiments. Detailed protocols to obtain defined and reproducible phenotypic profiles for bacteria and yeasts are shown. Moreover, the innovative software opm R packages and DuctApe suite for the analysis of kinetic data produced by PM and panphenome description are reported.

Key words Phenotype MicroArray, Phenomics, Microbial metabolism, Chemical sensitivity

1 Introduction

In recent years, *omic* approaches, such as microarrays and next generation sequencing (NGS) techniques for genome and transcriptome analysis, and proteomics technologies, have evolved rapidly. A boosting number of studies have been performed for profiling RNA expression, identifying novel transcripts and microRNA, describing patterns of transcripts/proteins in association with specific physiological states of cells [1, 2]. Classical approaches exploited to deepen phenotypic characterization of microorganisms typically lack speed, simplicity, and sensitivity, and didn't encounter great progresses. Actually only an integrated approach connecting genomics, transcriptomics, and proteomics with a deep phenotypic characterization can provide a cell-wide perspective leading to a more comprehensive and systematic investigation of cell physiology [3, 4].

A whole phenotypic characterization of microorganisms is the last major area of interest becoming amenable to efficient overall analysis. Classical approaches allow to investigate the phenotype

Alessio Mengoni et al. (eds.), *Bacterial Pangenomics: Methods and Protocols*, Methods in Molecular Biology, vol. 1231, DOI 10.1007/978-1-4939-1720-4_7,

characters one at a time, taking quite long time for experiment. Furthermore the definition of phenotypes often refers to vague qualitative descriptors. Therefore there is the need of an efficient method with appropriate sensitivity, specificity, and wideness for a satisfactory microorganisms phenotypic global analysis.

In 2001 Biolog Inc. released a high-throughput technology, which allows to test thousands of phenotypes at the same time. This technology, called Phenotype MicroArray (PM), arises in the wake of genomics, transcriptomics, and proteomics and can be classified as "phenomics."

PM technology permits to investigate the panphenome of microorganisms analyzing in a single experiment the ability to use nearly 200 C-sources, 400 N-sources, 100 P- and S-sources, 100 nutrient supplements, and the response to 240 toxic compounds (each one at 4 increasing concentrations), and to a range of pH values and osmolyte concentrations. PM uses tetrazolium dyes as colorimetric reporters of cellular metabolic activity.

The integrated application of PM technology with molecular approaches widen the fields of investigation and make possible a more comprehensive exploration of microorganisms.

2 Materials

2.1 Equipment and Software

Turbidimeter: Biolog turbidimeter (Biolog, Cat #3531), which is preset to 590 nm, contains a well that accepts dedicates glass tube (20 mm diameter, 150 mm length) to prepare standardized bacterial suspensions.

Turbidity standards: Turbidity standard glass tubes (65 % Turbidity Standard—Biolog, Cat #3440; 85 % Turbidity Standard—Biolog cat#3441) are used for the calibration of the Biolog turbidimeter.

OmniLog microplates reader: The OmniLog is an incubator and an automated microplate reader, which contains up to 50 PM microplates. Plates reading is performed by a CCD camera housed inside the rear of the instrument that captures the image of each plate every 15 min throughout the user-defined incubation period.

Software: Three modules form the OmniLog Phenotype MicroArray software: (1) Data Collection module, the basic OmniLog operating program, which for each plate generates a data file containing the kinetic information recorded from the 96 wells; (2) File Management Kinetic module, which helps to manage data files, assembles data lists, draws and superimposes kinetic curves; (3) Parametric module, which provides tools for data analysis, including comparison functions, calculation of specific kinetic curve parameters (height, area, slope, lag time, etc.) as well as export functions.

The opm R package [5] and the DuctApe suite [6] which can also calculate the specific kinetic curve parameters, handle metadata about the experiment and prepare a series of summary plots that are used to obtain panphenome of microorganisms. These Software analyze the PM ".csv" data files exported by the Parametric module of OmniLog Phenotype MicroArray.

2.2 Media

All media and solutions must be sterilized appropriately.

2.2.1 Media for Bacteria

BUG agar (Biolog, Cat #70101) and BUG+B agar (Biolog Cat #71102) are suggested as agarized media for the cultivation of bacteria (*see* **Note 2**).

M9 medium, required for the preliminary classification of the bacterial strains, contains: 12.8 g/l $Na_2HPO_4{\cdot}7H_2O$, 3 g/l KH_2PO_4, 0.5 g/l NaCl, 1 g/l NH_4Cl, 0.24 g/l $MgSO_4$, 2 g/l carbon source (i.e. D-glucose, pyruvate, succinate).

IF-0 1.2× (Biolog, Cat #72268), IF 10a 1.2× (Biolog, Cat #72256), IF 10b 1.2× (Biolog, Cat #72266) are used for the inoculation of PM panels.

2.2.2 Media for Yeasts

Agarized medium BUY (Biolog, Cat #70005) is used for yeast growth (*see* **Note 27**).

IF Y-0 1.2× (Biolog, Cat #72232) is used to inoculate metabolic panels.

SCG medium, required for the inoculation of chemical sensitivity panels is prepared as follow: 8.04 g/l Yeast Nitrogen Base, 2.4 g/l Drop Out Mix Complete, 21.5 g/l D-glucose.

2.3 Dye Mixes

Biolog Redox Dye mix A (Biolog, Cat #74221), Biolog Redox Dye mix D (Biolog, Cat # 74224), Biolog Redox Dye mix E (Biolog, Cat #74225), Biolog Redox Dye mix F (Biolog, Cat #74226), Biolog Redox Dye mix G (Biolog, Cat #74227), Biolog Redox Dye mix H (Biolog, Cat #74228).

2.4 PM Panels

Biolog currently manufactures 25 PM microplates relevant to microorganisms (bacteria and fungi) performing nearly 2,000 phenotypic assays: eight metabolic panels for measuring the utilization of various carbon, nitrogen, phosphorus, and sulfur sources; two panels to measure the osmotic/ionic and pH responses; ten and five panels to test the sensitivity to toxic chemical compounds in bacteria and fungi, respectively (Table 1).

2.5 Disposables

Long cotton-tipped swabs, sterile reservoir for multichannel pipettor, filter tips, sterile glass test tubes (20 mm diameter, 150 length), sterile plastic vials (50 ml, 120 ml), gas permeable membranes Breathe-easy (Sigma Aldrich Cat. No. Z380059).

Table 1
PM panels

	Panel	Biolog Cat#	Assay	Organisms
Metabolism	PM1	12111	Carbon utilization	Bacteria and yeasts
	PM2	12112	Carbon utilization	Bacteria and yeasts
	PM3	12121	Nitrogen utilization	Bacteria and yeasts
	PM4	12131	Phosphorus and sulfur utilization	Bacteria and yeasts
	PM5	12141	Growth promoters	Bacteria and yeasts
	PM6	12181	Nitrogen (di- and tri-peptides) utilization	Bacteria and yeasts
	PM7	12182	Nitrogen (di- and tri-peptides) utilization	Bacteria and yeasts
	PM8	12183	Nitrogen (di- and tri-peptides) utilization	Bacteria and yeasts
Chemical sensitivity	PM9	12161	Osmotic/Ionic response	Bacteria and yeasts
	PM10	12162	pH sensitivity	Bacteria and yeasts
	PM11	12211	Drug/chemical sensitivities	Bacteria
	PM12	12212	Drug/chemical sensitivities	Bacteria
	PM13	12213	Drug/chemical sensitivities	Bacteria
	PM14	12214	Drug/chemical sensitivities	Bacteria
	PM15	12215	Drug/chemical sensitivities	Bacteria
	PM16	12216	Drug/chemical sensitivities	Bacteria
	PM17	12217	Drug/chemical sensitivities	Bacteria
	PM18	12218	Drug/chemical sensitivities	Bacteria
	PM19	12219	Drug/chemical sensitivities	Bacteria
	PM20	12220	Drug/chemical sensitivities	Bacteria
	PM21	12221	Drug/chemical sensitivities	Yeasts
	PM22	12222	Drug/chemical sensitivities	Yeasts
	PM23	12223	Drug/chemical sensitivities	Yeasts
	PM24	12224	Drug/chemical sensitivities	Yeasts
	PM25	12225	Drug/chemical sensitivities	Yeasts

3 Methods

PM application involves the steps showed in Fig. 1. The first important warning when performing a PM experiment concerns the great influence of medium, temperature, and other growth parameters on the results of the assays. All growth parameters can affect an organism's phenotype, so it is fundamental to standardize all the steps starting from the pre-inoculum, in which even slight differences might be responsible for variable phenotypic responses highlighted by PM analysis. Moreover, it is fundamental the identification of the specific nutritional requirements of the strains under analysis.

3.1 Phenotype MicroArray for Bacteria

3.1.1 Metabolic Panels (PM1–8)

Inoculation in metabolic panels requires a minimal chemically defined medium whose composition depends on the nutritional requirements of each strain. Therefore, preliminary experiments must be conducted in order to identify the nutritional requirements of the strains.

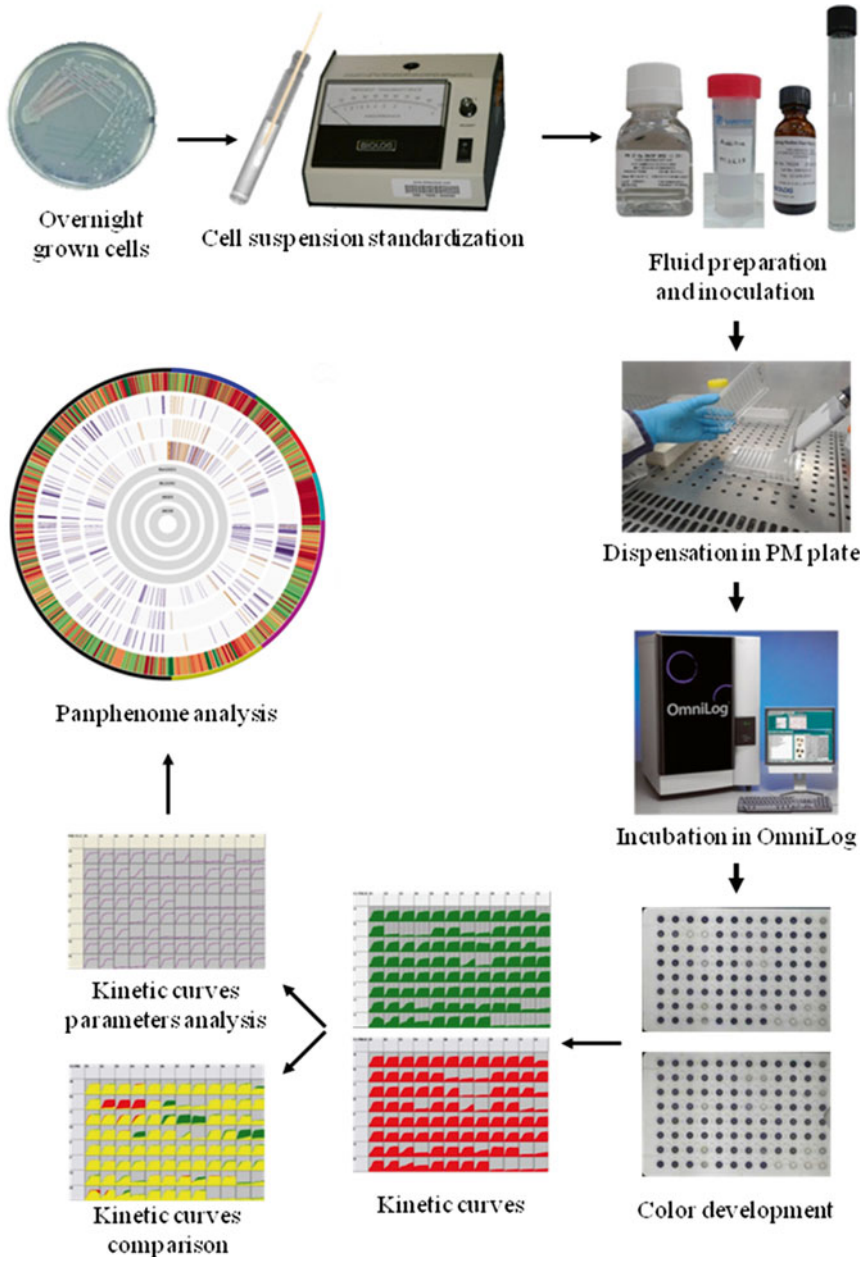

Fig. 1 Workflow of PM experiments

3.1.1.1 Preliminary Test for the Characterization of Strains

Three different types of strain can be identified on the bases of their requirements of nutrients: (1) strains with minimum requirement of nutrients (SMRN), (2) strains with complex but known requirement of nutrients (SCKRN), (3) strains with complex and unknown requirement of nutrients (SCURN).

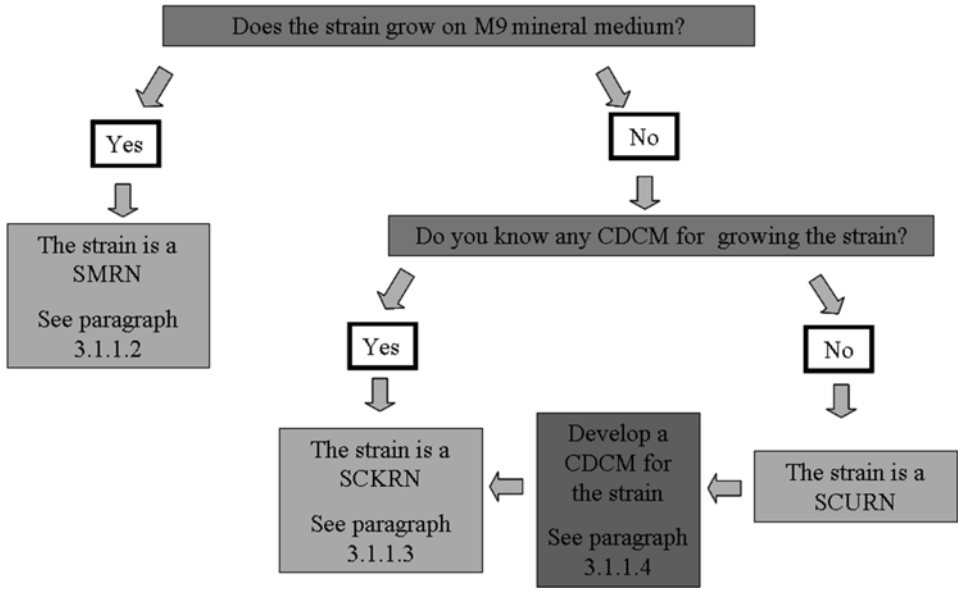

Fig. 2 Scheme for the classification of bacterial strains for PM metabolic analysis. *CDCM* chemically defined complex medium (*see* **Note 1**), *SMRN* strain with minimal requirement of nutrients, *SCKRN* strain with complex and known requirement of nutrients, *SCURN* strain with complex and unknown requirement of nutrients

In order to classify a strain as a SMRN, SCKRN, or SCURN proceed as follow:

1. Streak the bacterial strain on the agarized medium usually used for its cultivation, and incubate overnight at the optimal growth temperature.
2. Transfer a colony from the agarized medium into sterile tubes containing M9 medium added with different carbon sources (i.e.: M9 plus D-glucose, M9 plus succinate, M9 plus pyruvate, etc.). Incubate at the optimal temperature of growth for 24 h or more. Check whether the bacterium is grown at least in one of the M9 media tested.
3. Classify the strain as SMRN or SCKRN or SCURN referring to Fig. 2.

3.1.1.2 PM Characterization of Strains with Minimum Requirement of Nutrients (SMRN) on Metabolic Panels

Preparation of Standardized Cell Suspension

1. Streak, using a three-sector streaking method, a small portion of the frozen glycerol cell stock on BUG agar or BUG + B agar (*see* **Note 2**).
2. Incubate the plate at optimal growth temperature until colonies are clearly visible.
3. Check the purity of the culture.
4. Subculture on the same medium a second time (BUG agar or BUG + B agar) (*see* **Note 2**).

Table 2
SMNR—fluids for metabolic PM panels

Panels	Fluid
PM1, 2	IF-0 (1×)
PM3–8	IF-0 (1×) added with a carbon source (*see* **Notes 6, 7** and **8**)

5. Transfer cells, using a sterile cotton swab, into a sterile glass tube containing 15 ml of IF-0 1× (*see* **Note 3**). Mix with the swab avoiding turbulence (*see* **Note 4**) until you obtain a homogeneous suspension.
6. Check turbidity and adjust to reach a suitable transmittance (12× in respect to the density in the PM inoculation fluid) (*see* **Note 5**).

Fluid Preparation

1. Prepare 22 ml of IF-0 1× (*see* **Note 3**) for PM1 and PM2 (11 ml each plate) (Table 2).
2. Prepare 66 ml of IF-0 1× amended with a carbon source (*see* **Notes 6, 7** and **8**) for PM3–8 (11 ml each plate) (Table 2).

Plate Inoculation and Incubation in the OmniLog

PM1, 2

1. Add 2 ml of cell suspension and 0.24 ml of Dye mix (*see* **Note 9**) 100× to 22 ml of inoculation fluid for PM1, 2.
2. Using a multichannel pipette, dispense 100 μl of inoculum into each well.
3. Seal the PM plates with gas permeable membrane if necessary (*see* **Note 10**).
4. Lodge the inoculated PM plates closed with their own cap into the vessels of the OmniLog (*see* **Note 11**).

PM3–8

1. Add 8 ml of cell suspension and 0.96 ml of Dye mix (*see* **Note 9**) 100× to 88 ml of inoculation fluid for PM3–8.
2. Using a multichannel pipette, dispense 100 μl of inoculum into each well.
3. Seal the PM plates with gas permeable membrane if necessary (*see* **Note 10**).
4. Lodge the inoculated PM plates closed with their own cap into the vessels of the OmniLog (*see* **Note 11**).

Table 3
SCKRN—fluids for metabolic PM panels

Panels	Fluid
PM1, 2	CDCM (1×) depleted of C- source
PM3, 6–8	CDCM (1×)depleted of N- source (*see* **Note 8**)
PM4 (rows A–E)	CDCM (1×) depleted of P-source (*see* **Note 8**)
PM4 (rows F–H)	CDCM (1×) depleted of S-source (*see* **Note 8**)
PM5	CDCM (1×)

3.1.1.3 PM Characterization of Strains with Complex and Known Requirement of Nutrients (SCKRN) on Metabolic Panels

The inoculation of metabolic PM panels with SCKRN requires a CDCM (*see* **Note 1**) sustaining the growth of the bacterium.

Preparation of Standardized Cell Suspension

1. Streak, using a three-sector streaking method, a small portion of the frozen glycerol cell stock on BUG agar or BUG + B agar (*see* **Note 2**).
2. Incubate the plate at optimal growth temperature until colonies are clearly visible.
3. Check the purity of the culture.
4. Subculture a second time on the same medium (BUG agar or BUG + B agar) (*see* **Note 2**).
5. Transfer cells, using a sterile cotton swab into a sterile glass tube containing 15 ml of CDCM depleted of C-, N-, P-, S-sources. Mix with the swab avoiding turbulence (*see* **Note 4**) until you obtain a homogeneous suspension.
6. Check turbidity and adjust to reach a suitable transmittance (12× in respect to the density in the PM inoculation fluid) (*see* **Note 5**).

Fluids Preparation

Prepare the fluids for metabolic panels according to Table 3. For each panel 11 ml of fluid must be prepared.

PM Inoculation and Incubation in the OmniLog

PM1, 2

1. Add 2 ml of cell suspension and 0.24 ml of Dye mix (*see* **Note 9**) 100× to 22 ml of inoculation fluid for PM1, 2.
2. Using a multichannel pipette, dispense 100 μl of inoculum into each well.
3. Seal the PM plates with gas permeable membrane if necessary (*see* **Note 10**).
4. Lodge the inoculated PM plates closed with their own cap into the vessels of the OmniLog (*see* **Note 11**).

PM3, 6–8

1. Add 8 ml of cell suspension and 0.96 ml of Dye mix (*see* **Note 9**) 100× to 88 ml of inoculation fluid for PM3, 6–8.
2. Using a multichannel pipette, dispense 100 μl of inoculum into each well.
3. Seal the PM plates with gas permeable membrane if necessary (*see* **Note 10**).
4. Lodge the inoculated PM plates closed with their own cap into the vessels of the OmniLog (*see* **Note 11**).

PM4

1. Add 1 ml of cell suspension and 0.12 ml of Dye mix (*see* **Note 9**) 100× to 11 ml of inoculation fluid for PM4 rows A–E.
2. Add 1 ml of cell suspension and 0.12 ml of Dye mix (*see* **Note 9**) 100× to 11 ml of inoculation fluid for PM4 rows F–H.
3. Using a multichannel pipette, dispense 100 μl of inoculum into each well.
4. Seal the PM plates with gas permeable membrane if necessary (*see* **Note 10**).
5. Lodge the inoculated PM plates closed with their own cap into the vessels of the OmniLog (*see* **Note 11**).

PM5

1. Add 1 ml of cell suspension and 0.12 ml of Dye mix (*see* **Note 9**) 100× to 11 ml of inoculation fluid for PM5.
2. Using a multichannel pipette, dispense 100 μl of inoculum into each well.
3. Seal the PM plate with gas permeable membrane if necessary (*see* **Note 10**).
4. Lodge the inoculated PM plates closed with their own cap into the vessels of the OmniLog (*see* **Note 11**).

3.1.1.4 PM Characterization of Strains with Complex Unknown Requirement of Nutrients (SCURN) on Metabolic Panels

In order to develop a suitable CDCM sustaining the growth of a SCURN (*see* **Note 12**) proceed as follow.

1. Prepare different media according to Table 4 (*see* **Note 13**).
2. Streak the bacterial strain on BUG agar or BUG+B agar (*see* **Note 2**), and incubate overnight at the optimal growth temperature.
3. Harvest the bacterial cells from agarized medium and inoculate the different CDCM media. Incubate at the optimal temperature of growth for 24 h or more.
4. Check the growth of the bacterial culture in order to identify CDCM sustaining the growth of the strain.

Table 4
Composition of the medium for the growth of a SCURN

Components	Concentration (*see* **Note 14**)
IF-0 (Biolog)	1×
Tricarballylic acid pH 7 (*see* **Note 15**)	20 mM
Carbon source (*see* **Note 7**)	0.2 % (v/w)
Nitrogen source (*see* **Note 16**)	1 mM
Phosphorus source (*see* **Note 17**)	0.5 mM
Sulfur source (*see* **Note 18**)	0.5 mM
$MgCl_2$ (*see* **Note 19**)	240 mM
$CaCl_2$ (*see* **Note 20**)	120 mM
Ferric citrate (*see* **Note 21**)	200 μM
Aminoacids, purines and pyrimidines bases, etc. (*see* **Note 22**)	50 μM
Vitamines (*see* **Note 22**)	0.5 μM
Tween 80/tween 40 (*see* **Note 23**)	0.005 %

5. Perform metabolic PM analysis as described in (Subheading 3.1.1.3 PM Characterization of Strains with Complex and and Known Requirement of Nutrients (SCKRN) on Metabolic Panels). Known Requirement of Nutrients (SCKRN) on Metabolic Panels.

3.1.2 Chemical Sensitivity Panels (PM9–20)

Chemical sensitivity panels require a complex medium (CM) (*see* **Note 24**) sustaining the growth of the strain. Thus, to perform chemical sensitivity analysis it is not needed to know the minimal nutritional requirement of the strains.

3.1.2.1 Preparation of Cell Suspension

1. Streak, using a three-sector streaking method, a small portion of the frozen glycerol cells stock on BUG agar or BUG + B agar (*see* **Note 1**).
2. Incubate the plate at optimal growth temperature until colonies are clearly visible.
3. Check the purity of the culture.
4. Subculture a second time on the same medium (BUG agar or BUG + B agar) (*see* **Note 1**).
5. Transfer cells, using a sterile cotton swab, into a glass sterile tube containing 15 ml of IF-0 1×. Mix with the swab avoiding turbulence (*see* **Note 4**) until you obtain a homogeneous suspension.

Table 5
Fluids for chemical sensitivity PM panels

	PM panels	Fluid
Not-fermenting bacteria	PM9–20	CM (1.1×)
Fermenting bacteria	PM10 PM9, PM11–20	CM (1.1×) CM (1.1×) added with 30 mM Na-phosphate buffer pH 6–8 (*see* **Note 26**)

6. Check turbidity and adjust to reach a suitable transmittance (12× in respect to the density in the PM inoculation fluid) (*see* **Note 5**).

3.1.2.2 Fluid Preparation

The preparation of the CM for chemical sensitivity analysis of bacteria differs on the basis of the bacterial metabolism. If the bacterium has a fermentative metabolism and the CM contains a carbon source which is fermented, Na-phosphate buffer pH 6–8 must be added to the CM to prevent its acidification (*see* **Note 25**) (Table 5).

Fluid Preparation for Not Fermentative Bacteria

1. Prepare a CM for the growth of the strain at a concentration 1.2× in respect to the working concentration.
2. Prepare the fluid for PM9–20 (CM 1.1×) by adding 120 ml CM 1.2× with 12 ml sterile water.

Fluid Preparation for Fermentative Bacteria

1. Prepare a CM for the growth of the strain at a concentration 1.2× in respect to the working concentration.
2. Prepare the fluid for PM10 (CM 1.1×) by adding 10 ml CM 1.2× with 1 ml sterile water.
3. Prepare the fluid for PM9, 11–20 (CM 1.1× plus 30 mM Na-phosphate buffer pH 6–8) (*see* **Note 26**) by adding 110 ml CM 1.2× with 11 ml 360 mM Na-phosphate buffer pH 6–8 (*see* **Note 26**).

3.1.2.3 PM Inoculation and Incubation in the OmniLog

PM Inoculation for Not Fermentative Bacteria

1. Add 12 ml of cell suspension and 1.44 ml of Dye mix (*see* **Note 9**) 100× to 132 ml of inoculation fluid for PM9–20.
2. Using a multichannel pipette, dispense 100 μl of inoculum into each well.
3. Seal the PM plates with gas permeable membrane if required (*see* **Note 10**).
4. Lodge the inoculated PM plates closed with their own cap into the vessels of the OmniLog (*see* **Note 11**).

PM Inoculation for Fermentative Bacteria

1. Add 1 ml of cell suspension and 0.12 ml of Dye mix (*see* **Note 9**) 100× to 11 ml of inoculation fluid for PM10.
2. Add 11 ml of cell suspension and 1.32 of Dye mix (*see* **Note 9**) 100× to 121 ml of inoculation fluid for PM9, 11–20.
3. Using a multichannel pipette, dispense 100 μl of inoculum into each well.
4. Seal the PM plates with gas permeable membrane if required (*see* **Note 10**).
5. Lodge the inoculated PM plates closed with their own cap into the vessels of the OmniLog (*see* **Note 11**).

3.2 Phenotype MicroArray for Yeasts

To carry out a successful metabolic analysis of yeasts it is required to know nutritional requirement of the strain (prototroph or auxotroph). For auxotrophic yeasts all growth factors needed for growth must be known or preliminarily identified.

Protocols for metabolic analysis of prototrophic and auxotrophic strains differ only in the preparation of cell suspension, the former are suspended in water and the latter in a nutrient supplement solution (NS) containing all the growth factor needed for their growth. The phases of fluid preparation and PM inoculation are identical for auxotrophic and prototrophic strains.

3.2.1 Metabolic Panels (PM1–8)

3.2.1.1 Preparation of Cell Suspension

Prototrophic Strains

1. Streak, using a three-sector streaking method, a small portion of the frozen glycerol stock on BUY agar (*see* **Note 27**).
2. Incubate the plate at the optimal growth temperature of the strain until colonies are clearly visible.
3. Check the purity of the culture.
4. Subculture a second time on the same medium (BUY agar) (*see* **Note 27**).
5. Using a sterile cotton swab transfer an aliquot of colonies into a sterile tube containing 15 ml of sterile water. Mix with the swab avoiding turbulence (*see* **Note 4**) until you obtain a homogeneous suspension. Check turbidity and adjust to reach a transmittance of 62 % (*see* **Note 28**).

Auxotrophic Strains

1. Streak, using a three-sector streaking method, a small portion of the frozen glycerol stock on BUY agar (*see* **Note 27**).
2. Incubate the plate at the optimal growth temperature of the strain until colonies are clearly visible.
3. Check the purity of the culture.
4. Subculture a second time on the same medium (BUY agar) (*see* **Note 27**).
5. Prepare a NS solution containing all the compounds satisfying the auxotrophies of the strain. Add each compound at concentration 48× in respect to the minimal concentration needed

Table 6
Most common nutritional requirements for auxotrophic yeast strains. For each compound the final concentration required in the culture and the concentration in the nutrient supplement solution (NS) stock solution (48×) are reported

Components	48× NS stock solution	Concentration in the inoculums (*see* Note 29)
Adenine HCl	2.4 mM	0.05 mM
L-histidine HCl monohydrate	0.48 mM	0.01 mM
L-leucine	4.8 mM	0.10 mM
L-lysine HCl	2.4 mM	0.05 mM
L-methionine (*see* **Note 30**)	1.2 mM	0.025 mM
L-tryptophan	1.2 mM	0.025 mM
Uracil	1.44 mM	0.03 mM

Table 7
Yeast—fluids for metabolic PM panels

Panels	Fluid
PM1, 2	IF-Y0 (1.1×) added with N-, P-, S-, sources
PM3, 6–8	IF-Y0 (1.1×) added with C-, P-, S sources
PM4	IF-Y0 (1.1×) added with C-, N- sources
PM5, 9 (*see* **Note 31**)	IF-Y0 (1.1×) added with C-, N-, P-, S- sources

for the strain growth. In Table 6 the concentrations of the most common nutritional requirements of yeast are reported as a suggestion. About 20 ml of NS solution is required for each strain. Sterilize by filtration and store at 4 °C.

6. Using a sterile cotton swab transfer an aliquot of colonies into a sterile tube containing 15 ml of NS. Mix with the swab avoiding turbulence (*see* **Note 4**) until you obtain a homogeneous suspension. Check turbidity and adjust to reach a transmittance of 62 % (*see* **Note 28**).

3.2.1.2 Fluid Preparation

IF-Y0, a solution without organic carbon, nitrogen, sulfur, and phosphorus sources, is suggested by Biolog as basic fluid for the inoculation of metabolic PM panels. IF-Y0 must be added with the appropriate organic sources (prepared as additive solutions) depending on the PM panel you are going to inoculate (Table 7). The procedure to prepare the fluids mentioned in the table is described in the following paragraphs.

Table 8
Composition of 12× nitrogen/phosphorus/sulfur (NPS) additive. The concentrations of the components in the inoculum are also reported

Components (*see* Note 32)	12× NPS additive	Concentration in the inoculums (*see* Note 33)
L-glutamic acid monosodium	60 mM	5 mM
Potassium phosphate monobasic anhydrous (pH 6.0)	60 mM	5 mM
Sodium sulfate	24 mM	2 mM

Table 9
Composition of 12× carbon/phosphorus/sulfur (CPS) additive. The concentrations of the components in the inoculum are also reported

Components (*see* Note 32)	12× CPS additive	Concentration in the inoculums (*see* Note 33)
D-glucose	1,200 mM	100 mM (*see* **Note 34**)
Potassium phosphate monobasic anhydrous (pH 6.0)	60 mM	5 mM
Sodium sulfate	24 mM	2 mM

Table 10
Composition of 12× carbon/nitrogen (CN) additive. The concentrations of the components in the inoculum are also reported

Components (*see* Note 32)	12× CN additive	Concentration in the inoculums (*see* Note 33)
D-glucose	1,200 mM	100 mM (*see* **Note 34**)
L-glutamic acid monosodium	60 mM	5 mM

Preparation of Additive Solutions

Nitrogen/phosphorus/sulfur (NPS) additive

1. Prepare a 12× NPS additive (50 ml) as described in Table 8.
2. Sterilize 12× NPS solution by filtration and store at 4 °C.

Phosphorus/sulfur (CPS) additive

1. Prepare a 12× CPS additive (50 ml) as described in Table 9.
2. Sterilize 12× CPS solution by filtration and store at 4 °C.

Carbon/nitrogen (CN) additive

1. Prepare a 12× CN additive (50 ml) as reported in Table 10.
2. Sterilize 12× CN additive by filtration and store at 4 °C.

Table 11
Composition of 12× CNPS additive. The concentrations of the components in the inoculum are also reported

Components (*see* Note 32)	12× CN additive	Concentration in the inoculum (*see* Note 33)
D-glucose	1,200 mM	100 mM (*see* **Note 34**)
L-glutamic acid monosodium	60 mM	5 mM
Potassium phosphate monobasic anhydrous (pH 6.0)	60 mM	5 mM
Sodium sulfate	24 mM	2 mM

Carbon/nitrogen/phosphorus/sulfur (CNPS) additive

1. Prepare a 12× CNPS additive (50 ml) as reported in Table 11.
2. Sterilize 12× CNPS additive by filtration and store at 4 °C.

Preparation of the Inoculation Fluids

1. Prepare the fluid for PM1, 2 (Table 7) by adding 2 ml of NPS additive to 20 ml IF-Y0 1.2×.
2. Prepare the fluid for PM3, 6, 7, 8 (Table 7) by adding 4 ml of CPS additive to 40 ml IF-Y0 1.2×.
3. Prepare the fluid for PM4 (Table 7) by adding 1 ml of CN additive to 10 ml IF-Y0 1.2×.
4. Prepare the fluid for PM5, 9 (Table 7) by adding 2 ml of CNPS additive to 20 ml IF-Y0 1.2×.

3.2.1.3 PM Inoculation and Incubation in the OmniLog

PM1, 2

1. Add 0.50 ml of 62 % T cell suspension and 0.32 ml Dye mix D (75×) to 20 ml of inoculation fluid for PM1, 2.
2. Add 1.18 sterile water to a final volume of 24.0 ml
3. Dispense the inoculation fluid (100 µl per well) in PM1, 2.
4. Seal the PM plates with gas permeable membrane if required (*see* **Note 10**).
5. Lodge the inoculated PM plates closed with their own cap into the vessels of the OmniLog (*see* **Note 11**).

PM3, 6, 7, 8

1. Add 1 ml of 62 % T cell suspension and 0.64 ml Dye mix D (75×) to 40 ml of inoculation fluid for PM3, 6, 7, 8.
2. Add 2.36 ml of sterile water to a final volume of 48.0 ml.
3. Dispense the inoculation fluid (100 µl per well) in PM3, 6, 7, 8.

4. Seal the PM plates with gas permeable membrane if required (*see* **Note 10**).
5. Lodge the inoculated PM plates closed with their own cap into the vessels of the OmniLog (*see* **Note 11**).

PM4

1. Add 0.25 ml of 62 % T cell suspension and 0.16 ml Dye mix D (75×) to 10 ml of inoculation fluid for PM4.
2. Add 0.59 ml of sterile water to a final volume of 12 ml.
3. Dispense the inoculation fluid (100 μl per well) in PM4.
4. Seal the PM plates with gas permeable membrane if required (*see* **Note 10**).
5. Lodge the inoculated PM plates closed with their own cap into the vessels of the OmniLog (*see* **Note 11**).

PM5, 9 (*see* **Note 31**)

1. Add 0.50 ml of 62 % T cell suspension and 0.32 ml Dye mix D (75×) to 20 ml of inoculation fluid for PM5, 9.
2. Add 1.18 sterile water to a final volume of 24.0 ml.
3. Dispense the inoculation fluid (100 μl per well) in PM5, 9.
4. Seal the PM plates with gas permeable membrane if required OmniLog (*see* **Note 10**).
5. Lodge the inoculated PM plates closed with their own cap into the vessels of the OmniLog (*see* **Note 11**).

3.2.2 Chemical Sensitivity Panels (PM21–25)

3.2.2.1 Preparation of Cell Suspension

1. Streak, using a three-sector streaking method, a small portion of the frozen glycerol cell stock on BUY agar (*see* **Note 27**).
2. Incubate the plate at the optimal growth temperature until colonies are clearly visible.
3. Check the purity of the culture.
4. Subculture a second time on the same medium (BUY agar) (*see* **Note 27**).
5. Transfer cells using a sterile cotton swab into a sterile glass tube containing 15 ml of sterile water. Mix with the swab avoiding turbulence (*see* **Note 4**) until you obtain a homogeneous suspension.
6. Check turbidity and adjust to reach a suitable transmittance (62 %) (*see* **Note 28**).

3.2.2.2 Fluid Preparation

Inoculation in chemical sensitivity panels requires a complex medium (SCG). Thus the identification of the auxotrophies is not required.

Prepare 50 ml of SCG medium 1.2×.

3.2.2.3 PM Inoculation and Incubation in the OmniLog

1. Add 0.6 ml 100× of Dye mix E and 1.25 ml of cell suspension 62 % T to 50 ml of SCG 1.2×.
2. Add 8.15 ml sterile water to a final volume of 60 ml.
3. Dispense the inoculation fluid in PM21–25 plates 100 µl per well.
4. Seal the PM plates with gas permeable membrane if required (*see* **Note 10**).
5. Lodge the inoculated PM plates closed with their own cap into the vessels of the OmniLog (*see* **Note 11**).

3.3 Phenotype MicroArray Data Analysis

The analysis of PM data is generally performed using the File Management Kinetic and Parametric modules of the PM software which have been extensively described by Shea et al. [7]. In this paragraph alternative and advanced software to analyze the PM data are described: the opm R package [5] and the DuctApe suite [6].

3.3.1 Kinetic Data Export with the OmniLog Phenotype MicroArray Software

The raw kinetic data to be analyzed can be obtained by the File Management Kinetic module of the PM software. Once you have defined the data list use the button "Go: export all hours" in the "EXPORT" window. You can chose to export the data of a single panel or of a set of panels.

3.3.2 PM Data Analysis with the opm R Package

The easiest way to perform the PM data analysis through the opm R package is to use the RStudio graphical interface (http://www.rstudio.com/ide/). In this section the PM example data provided in the opm R package will be used (vaas_4). Further information on each command options can be found by adding `??` in front of each command.

3.3.2.1 Installation of the opm R Package

1. Open RStudio (or the R shell) and type (*see* **Note 35**):
   ```
   install.packages('opm')
   ```
2. Install more example data (optional).
   ```
   install.packages('opmdata')
   ```
3. Load the opm package and the sample data.
   ```
   library('opm')
   ```

3.3.2.2 Import PM Data Files

1. Load a PM *.csv*, *.yml*, or *.json* format file (*see* **Note 36**).
   vaas_4<- read_opm('vaas_4.yml')

3.3.2.3 Calculate PM Curves Parameters and Discretised Values

1. Calculate the curve parameters for each PM curve, using the spline method (*see* **Note 37**) and performing 100 bootstrap replicates to obtain the 95 % confidence intervals.
   ```
   op<- set_spline_options(type="smooth.spline")
   do_aggr(vaas_4,  boot=10,  method="spline",
   options=op)
   ```

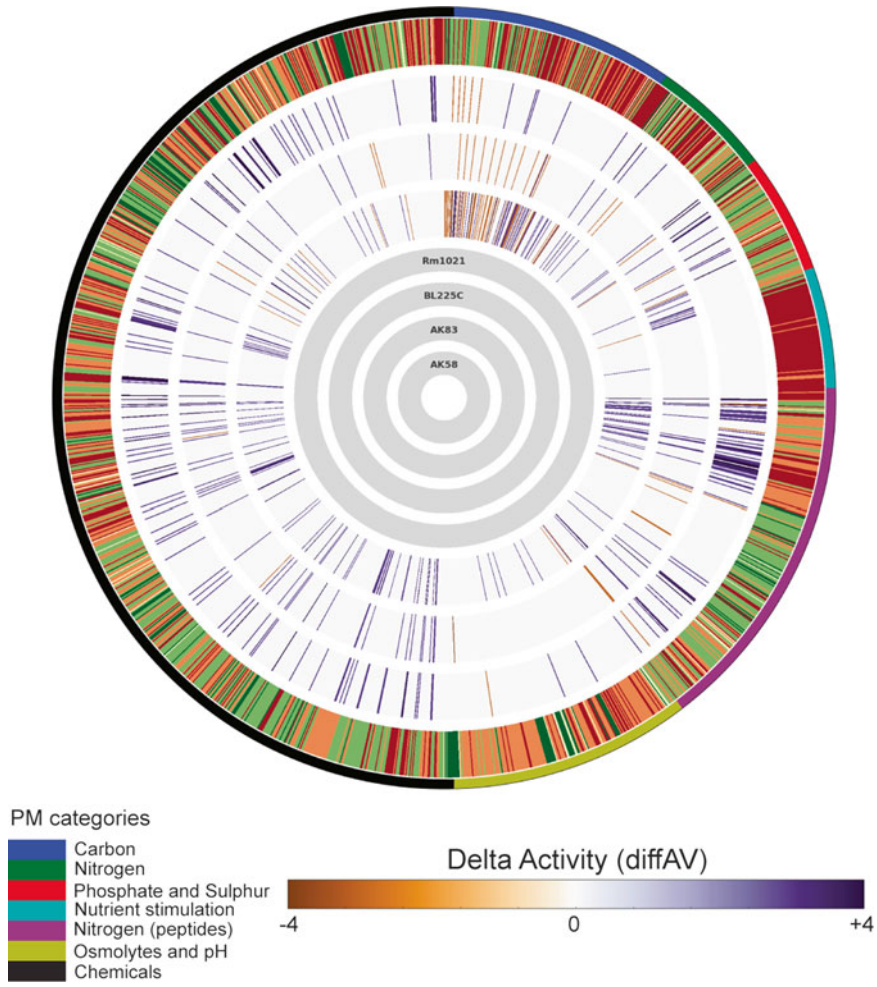

Fig. 3 Example output from the DuctApe suite

2. Compute the discretised values for each PM curve, using the k-means clustering with two clusters (active curves vs. inactive curves).

```
vaas_4<- do_disc(vaas_4, cutoff=FALSE)
```

3.3.2.4 Generate PM Curve Plots

1. Generate plate-wise plots.

```
xy_plot(vaas_4)
```

2. A subset of the curves of a plate can be generated, as well as single curves plots.

```
xy_plot(vaas_4[,, 1:12])
xy_plot(vaas_4[,, "A02"])
```

3. Generate level plots, alternative to traditional curve plots (a subset of the curves can also be used) (Fig. 3).

```
level_plot(vaas_4)
level_plot(vaas_4[,, 1:12])
```

4. Generate plate-wise heatmaps (a subset of the curves can also be used).

```
heat_map(vaas_4,  as.labels="Strain",  as.
groups="Species")
heat_map(vaas_4[,,1:12], as.labels="Strain",
as.groups="Species")
```

5. Plot the 95 % confidence intervals for a curves subset.

```
ci_plot(vaas_4[,,c("A02",  'B01')],  as.
labels=list("Species", "Strain"))
```

6. Generate a radial plot for a curves subset.

```
radial_plot(vaas_4[,,1:12],  as.
labels="Strain")
```

3.3.2.5 Export PM Data

1. Export PM .yml files (readable by the DuctApe suite), including PM curves parameters and discretised values.

```
write(to_yaml(vaas_4), 'vaas_4.yml')
```

3.3.3 PM Data Analysis with the DuctApe Suite

The DuctApe suite contains a series of command line programs that can be used in any UNIX-like shell, such as bash, zsh, or cygwin. The suite contains three modules: dape, used for the project setup, dgenome, used to analyze genomic data and dphenome, which is used to analyze PM kinetic data. In this section the PM example data provided in the ductape_data repository will be used (https://github.com/combogenomics/ductape_data, folder smeliloti); we also assume that we are using a bash terminal on Ubuntu. Further information on each command options can be found by adding `-h` to each command.

3.3.3.1 Installation of the DuctApe Suite

1. Open a terminal and type (*see* **Notes 35** and **38**):

```
sudo pip install DuctApe
```

2. Download the example data (*see* **Note 35**) and move to the working directory.

```
wget  https://github.com/combogenomics/duc-
tape_data/archive/master.tar.gz
tar -xvf master.tar.gz
 cd ductape_data-master/smeliloti
```

3.3.3.2 DuctApe Project Setup

1. Create a DuctApe project.

```
dape init
```

2. Use dape to add the organism names.

```
dape add Rm1021 -c red
dape add BL225C -c green
dape add AK83 -c blue
dape add AK58 -c orange
```

3.3.3.3 Import PM Data Files

1. Automatically load all the PM .csv files (*see* **Note 39**).

```
dphenome add-dir phenome
```

2. Control signal subtraction (optional).

```
dphenome zero
```

3.3.3.4 Calculate PM Curves Parameters and Discretised Values

1. Calculate the curve parameters for each PM curve.

```
dphenome start -f -g
```

2. Perform an elbow test to determine the optimal number of discretised categories for the PM curves.

```
dphenome start -e -g
eog elbow.png
```

Choose the number of clusters that cause the highest reduction in the sum of squared errors for most of the analyzed curve parameters (five in this case).

3. Compute the discretised values for each PM curve, using five clusters.

```
dphenome start -n 5 -f -g
```

3.3.3.5 Generate PM Curve Plots

1. Generate plate-wise plots, single curve plots, and plate-wise heatmaps.

```
dphenome plot
```

2. Generate a whole experiment ring plot with discretised values.

```
dphenome rings
```

3. Generate a ring plot with the distretised values comparison against the reference strain, showing only those PM curves with a difference in the discretised value ≥ 2 (Fig. 4).

```
dphenome rings -o Rm1021 -d 2
```

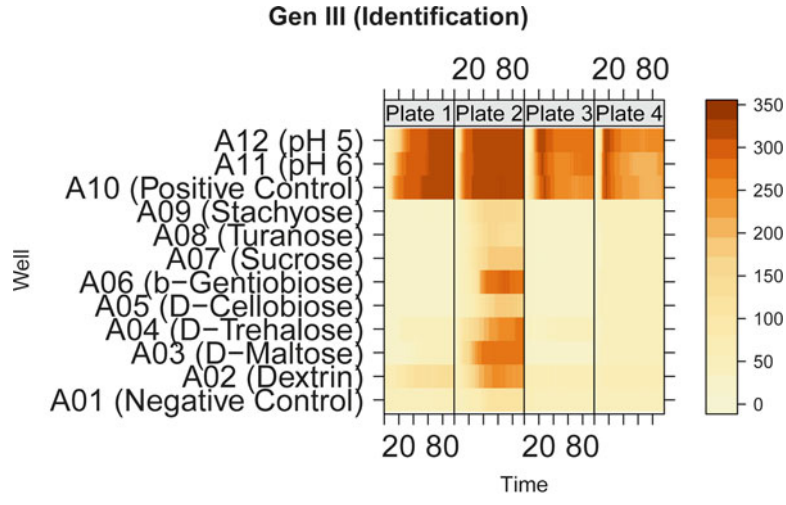

Fig. 4 Example output from the opm package

4. Generate various plots and tables about the computed discretised values, using a value threshold of 3 and a difference threshold between each organism of 2.

```
dphenome stats -a 3 -d 2
```

3.3.3.6 Export PM Data

1. Export PM yml files (readable by the opm package) and PM curves parameters and discretised values.

```
dphenome export
```

4 Notes

1. A "chemically defined complex medium" (CDCM) is a medium, more complex than M9, for which the composition is exactly known. Examples of CDCM are mineral media added with some amino acids or vitamins to satisfy the auxotrophies of a strain, or with some organic sources of nitrogen, sulfur, or phosphorus. Complex media, such as Luria Bertani or Tryptic Soy Broth, whose composition is not chemically defined, are not CDCM.
2. You can use any other suitable medium. , If the strain does not grow on BUG agar or BUG + B agar (generally used for fastidious bacteria) The same medium must be used for strains whose phenotype must be compared.
3. Prepare IF-0 1× diluting IF-0 1.2× (Biolog) with sterile water.
4. Do not vortex the cell suspension.
5. The choice of the starting cell density in the inoculation fluids can be made by the operator, and should be adjusted in order to obtain an adequate kinetic curve in a limited time (from 24 to 96 h), and minimize the response of negative controls in metabolic panels PM3–8 (false positive can occur when the cell concentration is too high). As a suggestion, cell suspensions at 82 % and 42 % transmittance (corresponding respectively at about 0.07 and 0.38 absorbance) which will be diluted 12 times in the inoculation fluids are suitable respectively for several fast growing and slow growing bacteria. It is crucial to use the same starting cell density for the strains whose phenotype must be compared.
6. To obtain IF-0 1× amended with carbon sources add an appropriate volume of a sterile stock solution of a carbon source (100× or less) to IF-0 1.2×, then add sterile water to achieve IF-0 1×.
7. A carbon source certainly sustaining the strain growth must be used. Biolog protocol indicates to use 20 mM succinate or 20 mM pyruvate for Gram-negative bacteria, 5 mM glucose plus 2.5 mM pyruvate for Gram-positive bacteria. Nevertheless, the carbon source can be chosen by the operator. The same

carbon source must be used for the stains whose phenotypes must be compared.

8. If A01 well in PM3, 6–8, and wells A01 and F01 in PM04 (negative controls) show color development, the concentration of the carbon source can be reduced to half or one quarter, in order to reduce the background.
9. Biolog purchases different types of Dye mixes for bacteria (A, F, G, H). Dye mix A is generally used for Gram-negative bacteria, Dye mix F for fast growing Gram-positive bacteria, Dye mixes G and H for slow growing Gram-positive bacteria. Before starting the PM experiment, the best Dye mix must be selected. Grow the strain in the medium selected for the inoculation of PM panels added with Dye mix A (1×), or Dye mix F (1×), or Dye mix G (1×), or Dye mix H (1×), and select the Dye mix that gives the best color development. Furthermore, to check that the selected Dye mix is not abiotically (negative control) reduced in the medium add the Dye mix (1×) to the not inoculated medium (cell free), and verify that it does not turn colored during the time of culture growth (24–96 h).
10. The sealing of the plates with gas permeable membrane limits the fluid drying and therefore is required primarily for long time incubation and/or incubation at high temperature (above 37 °C). Close the plates sticking adhesive tape in order to be sure that the cap fits tightly to the base of the plate. The OmniLog should be damage during the phase of image capturing if the plates are not perfectly closed.
11. Set the OmniLog at the growth temperature of the strains about a half an hour before starting with the incubation.
12. SCURNs can have a wide range of nutrient requirements (from simple to highly complex requirements). Some SCURNs likely require only an organic source of N- and/or S- and/or P- to growth or compounds satisfying their auxotrophies (i.e. amino acids, vitamins, fatty acids). Several attempts should be done before finding the suitable CDCM sustaining its growth, if the strain has complex nutritional requirements. Nevertheless, some strains have so complex nutrient requirements that any attempt to formulate a CDCM could be unsuccessful.
13. Table does not describe a fluid with a defined composition, it only gives a generic scheme on the basis of which you should prepare different inoculation fluids varying C-, N-, S-, P-sources, the content of amino acids, purines and pyrimidines bases, vitamins, and of any other growth factor that should be required for the growth of the strain.
14. The reported concentrations are indicative and they should be adjusted by the operator.

15. Biolog protocol indicates tricarballylic acid to buffer the medium. It is usually required if the strain has a fermentative metabolism and produces acids during growth. The Dye mixes are pH sensitive, thus they are inactivated, and they do not turn colored, if the pH of the medium becomes low the Dye mix. Tricarballylic acid can be used for all Gram-positive bacteria. Nevertheless, it should be checked that the strain does not use tricarballylic acid as a carbon source. In this case, another suitable buffer should be used. pH of the buffer solution must be adjusted on the basis of the need of the strain. For the majority of Gram-positive bacteria, pH 7 should be used, however lactobacilli could prefer pH 5.5. Tricarballylic acid can be prepared as an 800 mM stock solution as follow: add 14.088 g to 55 ml of water, add NaOH to reach pH 5.5 or pH 7, then add water to 100 ml.
16. A suitable N-source, NH_4^+ or organic compounds (i.e. L-glutamate or L-glutamine) should be used.
17. A suitable P-source, PO_4^{3-}, $P_2O_7^{4-}$, or organic compounds (i.e. uridine-5′-monophosphate) should be used.
18. A suitable S-source SO_4^{2-} or organic compounds (i.e. methionine, cystine, or thiosulfate) should be used. Use 1 mM thiosulfate for *Yersinia*, *Proteus,* and *Obesumbacterium*.
19. Biolog suggests using $MgCl_2$ for the growth of Gram-positive bacteria.
20. Biolog suggests using $CaCl_2$ for the growth of Gram-positive bacteria.
21. Ferric citrate must be added to satisfy the requirement of iron.
22. The defined growth factors (vitamins, amino acids, purines and pyrimidines bases, etc.) can be replaced by yeast extract and/or hydrolyzed protein (i.e. peptone) at low concentration, if the auxotrophies of the strain are not known. Typically 0.005–0.01 % total organic matter is used as a growth promoter. Higher concentrations should be used, if this concentration is not enough to sustain the growth of the strain. However, in order to be sure that the amount of organic matter added is used as a growth factor and not as C-, N-, S-, P- sources.
23. Add tween 80 or tween 40 as anticlumping agents in fluids for Gram-positive bacteria.
24. Complex media typically contain materials of biological origin (blood, milk, yeast extract, beef extract, etc.,), therefore the exact chemical composition is obviously undetermined. These media provide the full range of growth factors that may be required by an organism. Thus they can be use to cultivate bacteria whose nutritional requirements are complex or/and unknown. Biolog suggests using IF-0a and IF-0b for

Gram-negative and Gram-positive bacteria, respectively. However, it is possible to use the complex medium typically used to culture the strain, if these media do not sustain the growth of the strains. Use the same medium for the strains whose phenotype must be compared.

25. Growing fermenting bacteria produce organic acids, as a consequence the pH of the medium decreases during the bacterial growth. Dyes are pH sensitive, and when the pH becomes acid they are inactivated. Thus, Na-phosphate buffer is added to prevent the acidification of the medium. The final concentration of Na-phosphate buffer in the media is 30 mM. As an example, a suitable chemical sensitivity analysis of *S. thermophilus* should be obtained using a complex medium added with 0.3 % lactose or sucrose and 30 mM Na-phosphate [8].
26. The pH of the buffer must be equal or close to the optimal pH for the growth of the strain.
27. BUY agar can be replaced with YPD agar (yeast extract 10 g/l, peptone 20 g/l, D-glucose 20 g/l, agar 16 g/l).
28. 62 % T coincides with ~0.2 in absorbance.
29. The concentrations are indicative, they should be the lowest concentrations sustaining the growth of the strain and should be adjusted by the operator.
30. 0.12 mM pyridoxine may be used in place of L-methionine for *met15Δ0* strains. Methionine interferes with sulfur source testing in PM4 whereas pyridoxine does not.
31. PM9 is generally considered a chemical sensitivity panel because it tests the sensitivity of the microorganism to osmolytes. Nevertheless, for yeast analysis, PM9 is usually inoculated with a chemically defined medium commonly used for metabolic panels.
32. The L-glutamic acid and/or potassium phosphate and/or sodium sulfate and/or D-glucose can be used as N-, P-, S-, C-sources, respectively. Nevertheless, the suggested compounds can be replaced with others, if the operator deems them more suitable for the growth of the strain. The same N-, P-, S-, C-sources must be used for strains whose phenotypes must be compared.
33. The reported concentrations of N-, P-, S-, C- organic sources are indicative and can be adjusted by the operator.
34. Lower D-glucose concentration up to five times (20 mM), if the negative controls (A01 wells in PM3, 5–8, and wells A01 and F01 in PM4) show color development.
35. The computer needs to be connected to the internet.

36. The vaas_4 dataset is already present in the R workspace once the opm R package that has been loaded.
37. Other method options for curve parameters extraction are available.
38. Installation instructions for the software dependencies are listed in the README file or the online documentation.
39. Single PM .csv files can be loaded with the dphenome add command.

References

1. Methé BA, Lasa I (2013) Microbiology in the 'omics era: from the study of single cells to communities and beyond. Curr Opin Microbiol 16: 602–604
2. Nelson KE (2013) Microbiomes. Microb Ecol 65:916–919
3. Arakawa K, Tomita M (2013) Merging multiple omics datasets in silico: statistical analyses and data interpretation. Methods Mol Biol 985:459–470
4. Chaston J, Douglas AE (2012) Making the most of "omics" for symbiosis research. Biol Bull 223: 21–29
5. Vaas LA, Sikorski J, Hofner B, Fiebig A, Buddruhs N, Klenk HP, Göker M (2013) opm: an R package for analysing OmniLog® Phenotype MicroArray data. Bioinformatics 29:1823–1824
6. Galardini M, Mengoni A, Biondi EG, Semeraro R, Florio A, Bazzicalupo M, Benedetti A, Mocali S (2014) DuctApe, a suite for the analysis and correlation of genomic and OmniLog™ Phenotype MicroArray data. Genomics 103(1):1–10
7. Shea A, Wolcott M, Daefler S, Rozak DA (2012) Biolog phenotype microarrays. In: Microbial systems biology: methods and protocols. Methods in Molecular Biology vol 881, pp 331–373
8. Decorosi F, Santopolo L, Mora D, Viti C, Giovannetti L (2011) The improvement of a Phenotype MicroArray protocol for the chemical sensitivity analysis of *Streptococcus thermophilus*. J Microbiol Methods 86:258–261

Chapter 8

Comparative Analysis of Gene Expression: Uncovering Expression Conservation and Divergence Between *Salmonella enterica* Serovar Typhimurium Strains LT2 and 14028S

Paolo Sonego, Pieter Meysman, Marco Moretto, Roberto Viola, Kris Laukens, Duccio Cavalieri, and Kristof Engelen

Abstract

Different strains of the same organism can share a large amount of their genetic material, the so called core pangenome. Nevertheless, these species can display different lifestyles and it is still not well known to what extent the core pangenome plays a role in the divergence of lifestyles between the two organisms. Here, we present a procedure for uncovering the conservation and divergence of gene expression by using large expression compendia. We will use data from two *Salmonella enterica* serovar Typhimurium strains as an example here, strain LT2 and strain 14028S, to assess if there are orthologous gene pairs with different expression domains related in both strains.

Key words Gene expression, Expression divergence, Expression conservation, Salmonella

1 Introduction

Organisms that share a large part of their genome and present a high similarity at the sequence level can nevertheless show significant divergence of expression regulation for this shared core pangenome. In practice, this kind of behavior is nigh impossible to explain by analyzing the sequence alone. Instead it is more convenient to rely on gene expression measurements. To study such phenomena, ideally we would like to compare the expression profiles of different strains measured for the same set of biological conditions. Such data, however, is rarely available and generally gene public expression data of different strains is measured for different conditions and cannot be directly compared. In order to overcome this issue and inspect the role of the core pangenome in this divergence, the representation of the compendia can be changed from a "gene × conditions" matrix to a "gene × gene" correlation matrix of

Alessio Mengoni et al. (eds.), *Bacterial Pangenomics: Methods and Protocols*, Methods in Molecular Biology, vol. 1231, DOI 10.1007/978-1-4939-1720-4_8,

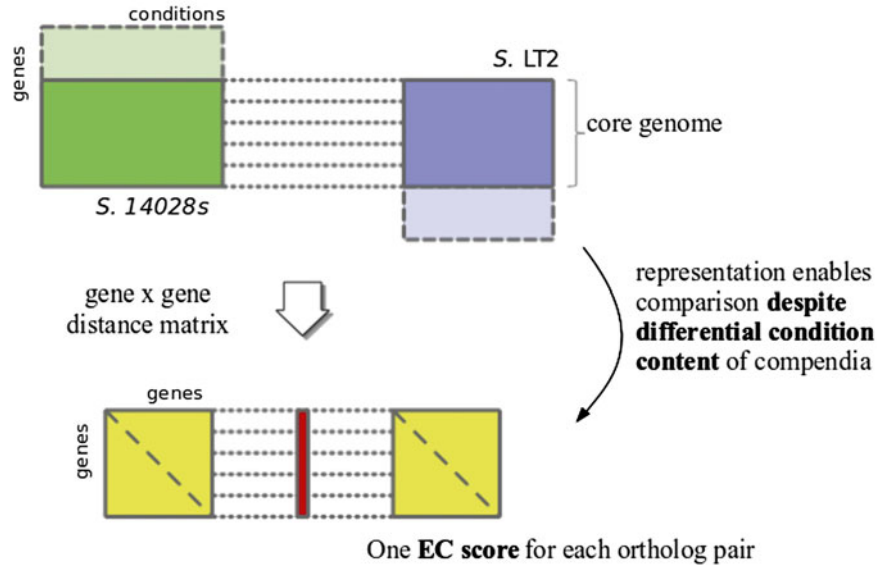

Fig. 1 Schema of the representation used by the ICC methodology for comparing compendia with differential condition content

one-to-one orthologous pairs independent from the specific contrast in each of the original compendia (represented in Fig. 1). Such a representation opens the door to cross-strain comparisons, e.g., by constructing networks from the correlation matrices and evaluating their congruency between different strains. The procedure we describe here though relies on the full matrices and uses the "Iterative Comparison of Co-expression" methodology [1] for comparing them. As a case study, we will compare the expression profiles of two strains of the same species: *Salmonella enterica* serovar Typhimurium LT2 and *S. enterica* serovar Typhimurium 14028S.

2 Materials

For the purpose of this chapter, we worked with the genomes of *S. enterica* serovar Typhimurium LT2 (*NC_003197.1*) and *S. enterica* serovar Typhimurium 14028S (*NC_016856.1*).

2.1 Gene Expression Data

The gene expression data sets for the analysis were built using the backend technology behind COLOMBOS 2.0 [2]. These data consisted of 208 measured conditions for *S. enterica* serovar Typhimurium LT2 and 645 conditions for *S. enterica* serovar Typhimurium 14028S (*see* **Note 1**).

2.2 Software and Code Used in the Analysis

Orthologous genes were identified using the OrthoMCL v.1.4 algorithm with the default settings on the protein sequences for both strains [3]. In this manner, we found 4,183 genes with only a single homolog in either strain (one-to-one mapping). All the computations were performed in MATLAB (requiring Statistics toolbox) [4] and together with all the data and results of the analysis are available as a compressed zip file at http://colombos.fmach.it/~ke/supple/Sonego_etal_BacterialPangenomics.zip.

3 Methods

The analysis begins from a collection of MATLAB objects containing both the gene expression data and the corresponding ortholog information for the two Salmonella strains under examination. The procedure, based on the Iterative Comparison of Co-expression (ICC) methodology described in [1], estimates the level of "expression conservation" (EC) of a single orthologous gene pair from the compendia of different strains (or organisms): for each gene pair the EC score is calculated by estimating the retention of the similarity in expression domains to all other genes in the core pangenome.

Before starting, download both the data and the MATLAB scripts for the analysis from http://colombos.fmach.it/~ke/supple/Sonego_etal_BacterialPangenomics.zip and unzip them in a single directory. Alternatively, you can put the included functions (all files with extension .m except analysis_procedure.m) elsewhere and add them to your MATLAB path. The file analysis_procedure.m contains all the commands used in the analysis. You can copy and paste them into MATLAB line by line to retrace each step of the analysis.

3.1 EC Score Calculation

The procedure for getting the EC scores is implemented in the analysis_procedure.m script, in the section titled "Expression Conservation (EC) score calculation":

(A) *Setup*: The expression matrices need to be made row by row comparable with the same amount of genes (already the case for the two gene expression compendia). Genes with too many NaNs (typically more than half of the expression matrix) need to be removed from both compendia.

(B) *Gene correlation matrices*: The correlation matrix of all genes vs. all genes for the same compendium needs to be calculated. It is important to account for NaNs when you are calculating correlations. This procedure is implemented in "geneUncCorr.m".

(C) *EC scores*: The correlation of the correlation matrices is calculated: the EC score of gene i is the correlation of row i from correlation matrix A with row i from correlation matrix B. Because the EC score is based on the other genes in the

compendium, we need to correct for any genes that might be very different. So we redo the previous step but with a weighted correlation, where the weight is equal to the EC score from the previous step: so genes which have not changed have an EC and therefore a weight of 1, those that are doing something completely different have a weight of 0, and so on. This procedure is implemented in "ortholScoreEC.m".

1. A correlation matrix of 4002×4002 was constructed for both strains LT2 and 14028S by calculating the uncentered Pearson correlation coefficient between the expression profiles of each pair of ortholog genes: each element of the matrix is the correlation value of the gene on the row versus the gene on the column across every measured condition. (Note that the correlation matrices are naturally ordered so that the equivalent rows correspond to the correlation profiles of a pair of orthologs.)
2. As the two matrices share the same dimensionality, we can compare the equivalent rows by calculating their Pearson correlation.
3. For correcting the bias due to the orthologous genes whose expression has diverged, we iteratively recalculate the correlation giving higher weights to genes of which expression has been conserved between the two species. This process is iterated at least ten times until an optimum is reached.

The values assigned to each orthologous gene pair vary between −1 and 1, where 1 signifies perfect conservation of expression with respect to the correlation with all the other genes, 0 means no conservation, and −1 signifies a complete reverse of expression regulation (i.e., genes correlated with the expression of the orthologous gene in a single compendium are anticorrelated with the ortholog in the other compendium).

3.2 Background EC Distribution Calculation

In this section we estimate "background" distributions for the EC scores that represent either perfect conservation or complete divergence of gene expression. Using these background distributions we can estimate the number of genes we expect to have diverged expression, under the assumption that the observed EC distribution is a mixture between genes that have conserved their expression domains and those that have diverged. The distribution of the EC scores of the orthologous gene pairs is depicted in Fig. 2 together with the distribution of the estimated score given no conservation of expression and the estimated score given perfect conservation of expression. We can see that the EC score distribution shows two peaks, one at 0.3 and a smaller one at 0.6; this bimodal nature of the EC scores' distribution we assume is due to the overall expression divergence and conservation levels. The EC

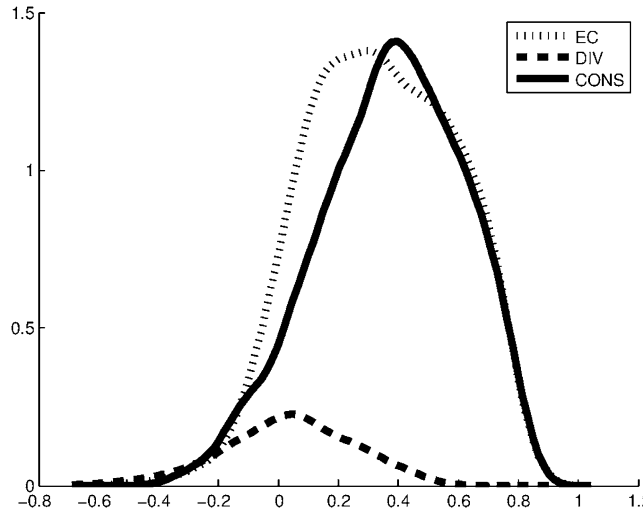

Fig. 2 Distribution of the EC scores between orthologous genes of *Salmonella enterica* serovar Typhimurium LT and 14028s depicted by its kernel smoothed density estimates (*blue line*). The background distributions for the EC scores that represent either perfect conservation (*red line*) or complete divergence (*green line*) (Color figure online)

score of gene pairs with complete expression divergence values varies between −0.7 and 0.6. The correlation score of a gene pair with perfectly conserved expression regulation can vary between −0.6 and 1: this extremely low score for an expression distribution which should be perfectly conserved can be attributed to the condition dependency of EC score: it will not be trivial to decide whether a gene is truly diverged or merely *seems* diverged because it was measured in different conditions for both strains. This step shows that the bimodal nature of the EC scores' distribution is due to the overall expression divergence and conservation levels.

As each background represents either diverged or conserved expression domains the most likely combination of the two background distributions into the found EC distribution between the two compendia can be used as a measure for the fraction of diverged genes.

1. The divergence background distribution calculation is implemented in the function "backgrndEC.m" (*see* **Note 5**). The background distribution for the diverged gene expression domains is estimated by permuting the expression values of a single gene in one of the compendia, recreate the entire correlation matrix, and recalculate the EC score. The process is iterated for every gene pair, and the score for the permutated gene is kept.
2. The conserved background distribution calculation is implemented in the function "backgrndEC.m". The background distribution for the conserved gene expression domains is esti-

mated by splitting the expression compendium into two equal halves (multiple times), with each half containing a different set of expression experiments: the two splitted compendia are then compared against each other. This procedure allows us to simulate perfect conservation by comparing a species to itself, but accounting for the presence of different experimental conditions in both expression compendia.

3.3 Selecting Candidate Genes from the Entire Data Set

In this section we will be using the background distributions calculated in the previous section to select candidate genes with conserved and diverged expression regulation between the two strains.

3.3.1 Conserved Expression Regulation

Since the estimated background distribution for non-conserved genes never gets a score higher than 0.6, it is reasonable to assume that gene pairs with a higher EC score are very likely to have conserved expression domains. This reasoning leads to a set of 682 genes with conserved expression at a cutoff of 0.6.

3.3.2 Diverged Expression Regulation

Defining a set of diverged genes using the conserved background is a much harder task as this distribution overlaps with the entire EC distribution (*see* Fig. 2). Given our conserved EC score distribution which varies from −0.6 to 1 we could take a cutoff of −0.6 but there are no genes in the EC distribution with a value smaller than this arbitrary cutoff. Setting a higher cutoff cannot guarantee that we only get diverged genes: some of them might actually be conserved genes. To overcome this issue we can estimate how many genes are expected to be conserved for each cutoff. There are two steps to this procedure:

1. As we can see from Fig. 2 the EC scores comparing the two Salmonella strains shows a bimodal distribution which could be seen as a mixture between the divergent and conserved background distributions previously calculated. The function "mixtureEC.m" estimates the fraction of divergent and conserved genes in the core pangenome of the two *Salmonella* strains by trying to maximize the overlapping between the EC score distribution and a mixture of the two background distributions. The fraction of genes with conserved expression domains for the given background distributions and the EC scores then corresponds to 88 %. This analysis estimated that approximately 12 % of the genes have divergent expression domains between these two *Salmonella* strains.
2. Given the two background distributions and the 12 % of divergent genes we estimated before, we take a reasonable cutoff of −0.1 (*see* **Note 6**) for which we can expect a false discovery rate of 0.53, that is, 53 % of gene pairs with an EC score lower than can be expected to be diverged. The FDR for this cutoff is still extremely high, and better results can be obtained by separating the genes in different classes first, as is done in the next section.

3.4 Selecting Candidate Genes by Relying on "Functional Expression Classes"

A more comprehensive analysis of the entire core pangenome can be performed by defining "functional expression classes" (FEC) for each organism and doing the selection of diverged genes on each class separately. The FECs are created independently from the EC scores, but if some of these can be characterized as either more diverged or conserved than the overall EC score distribution, this will improve the selection (*see* **Note 7**).

The procedure is implemented in the script analysis_procedure.m in the section entitled "Selecting candidate diverged genes relying on functional expression classes". The FECs are created with the function "makeFEC.m", and are defined based on a k-means clustering of the correlation matrices (k-means algorithm of Matlab R2013a) (*see* **Note 7**). Genes are not grouped together based on the similarity of expression profiles under specific conditions but based on a similar expression correlation toward the other genes in the compendium. Motivated by our previous study [5], we decided to select four classes for both *S. enterica* serovar Typhimurium LT2 and 14028S (*see* **Note 9**).

3.4.1 Cutoff Selection

Following the same procedure we did before for the background EC score calculation in Subheading 3.2, **step 2**, we now compare the EC score distribution for every single FEC in each *Salmonella* strain with the previously calculated conserved and diverged background distributions. As before, we estimate the percentage of genes that have divergent expression domains and retrieve the cutoffs that allow the selection of the diverged and conserved genes; only now, we do it separately for each FEC. As expected, some FEC shows better results than others. This can be explained taking a look at their distributions (Fig. 3): if one FEC can be characterized as either more diverged or conserved than the overall EC score distribution it will improve the selection.

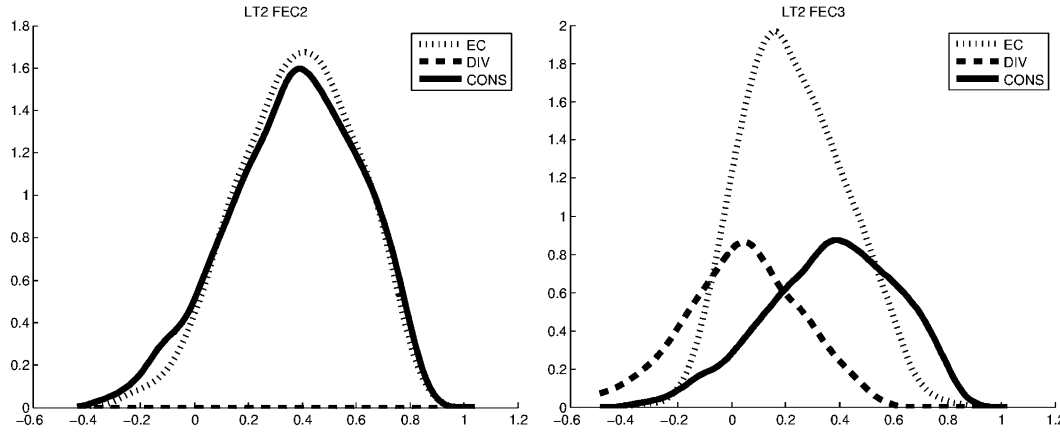

Fig. 3 Distributions of EC score, diverged and conserved distributions for two functional expression classes (FEC) of Salmonella LT2

3.4.2 GO Enrichment

GO enrichment calculation was achieved by applying a one-sided hypergeometric distribution to each biological process ontology present in the various gene set collected in our analysis. The enrichment calculation is implemented in the perl script go_enrichment. pl.

1. In order to run the script on your system you need a perl installation, which is commonly a default in many Linux/Unix distributions (Mac OS X included) and four files: you can download two files for the Gene Ontology Annotation from http://www.geneontology.org/ontology/obo_format_1_2/ (gene_ontology_ext.obo) and ftp://ftp.ebi.ac.uk/pub/databases/GO/goa/proteomes/ (35140.S_typhimurium_14028s.goa and 69.S_typhimurium_ATCC_700720.goa) and two files, one containing the full set of gene id (LocusTag here) for the species under investigation, and the other presenting the gene set ids which will be enriched. All the files and the perl script should be run from the some directory.
2. With any proper editor (vim, emacs, etc.) edit the perl file and substitute the right side of variable `my$inputfile='my_gene_set_ids.txt';` with the name of the file containing the ids of the gene set you want to be enriched and the right side of the variable `my$bgfile='s_14028s_all_LocusTag.txt';` with the name of the file containing all the LocusTag of the strain under investigation.
3. Run the perl script from the shell:
   `perl go_enrich.pl>go_enrich_my_gene_set. txt`

   In the file generated from the analysis you will find the GO TERM description of interest for you gene set plus the corresponding *p*-value.

4 Notes

1. The COLOMBOS 2.0 [2] database (http://www.colombos.net) features comprehensive expression compendia that combine microarray and RNA-seq data for seven bacterial model organisms. It is supported by a fully interactive web portal with extensive search, visualization, and analysis options; incorporates information from main curated microbial databases; and includes a formal sample annotation and ontology, and a web API for programmatic access to the database. The compendia for *S. enterica* serovar Typhimurium LT2 and 14028S were retrieved from a work-in-progress version of COLOMBOS, which will be made publicly available to the scientific community with the next release. The *Salmonella* compendia used in this chapter represent an improvement over the current COLOMBOS release: they properly separate the experiments for each strain (and hence map the measurements

to the proper genome sequence) where the current release aggregates all the experiments in a single compendium regardless of the strain origin.

2. The data and the scripts for the analysis should be decompressed in the same directory! Alternatively, you can put the included functions (all files with extension .m except analysis_procedure.m) elsewhere and add them to your MATLAB path.
3. Although the EC score was specifically developed to avoid bias in expression compendia with different biological conditions, these conditions do affect the observed correlation between genes of the same species [1, 5].
4. The background distribution analysis shows that 12 % of the genes in the core pangenome have divergent expression domains, but at the same time it shows that the EC score is very susceptible to changing conditions as is made clear comparing the same strain to itself.
5. The calculation of the background distribution for the case of diverged gene expression domains took ~8 h on a 2.9GHz Core i7. The calculation of the conserved background distribution is faster, but still took around ~2 h.
6. From a figure such as Fig. 4 (generated by "plotFDR.m") we can determine a cutoff that best fits the particular study and goals:
   (a) We would select a bigger cutoff if we prefer to work with a larger gene set, with as many diverged genes as possible, while not being concerned with a large number of false positives.

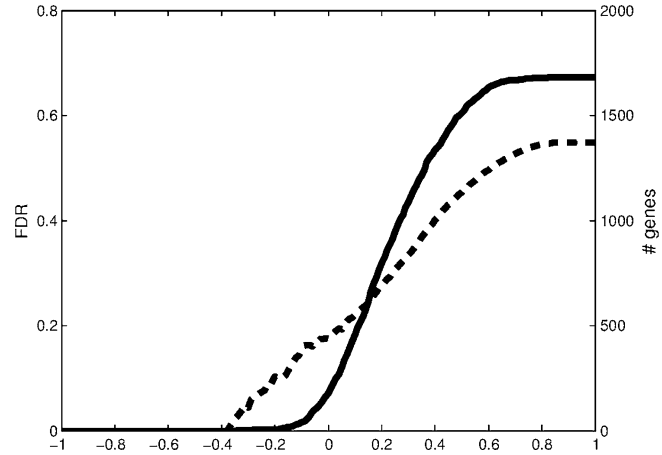

Fig. 4 Scatterplot representing both the variation of the number of genes selected according to different cutoffs and the trend of the false discovery rate. It allows the selection of gene sets based on the percentage of gene pairs that can be expected to be conserved. *The full line represents the number of genes and the dashed line the FDR*

(b) On the other hand, we would select a bigger cutoff if we decide to be more strict and be sure to get nearly only diverged genes (i.e., restrict the number of false positives) at the expense of getting a smaller number of genes and potentially missing many other diverged genes.

7. In our analysis we used a k-means clustering approach to identify FEC of interest. The user can decide to use or implement any other partitioning method that is independent from the EC score.
8. Our previous experience [5] showed that with an approach that separates the genes in classes independently from the EC score calculated with the ICC methodology [1] we can improve the quality of the gene set selection, when the classes have the propensity to be either more diverged or conserved than the overall EC score distribution. The results of [5] showed that for *Escherichia coli* and *S. enterica* serovar Typhimurium, two different but closely related species, the diverged and conserved contribution for *E. coli* could be explained in three main functional classes and for the *S. enterica* with five clusters. In our case study with data from two strains of the same species we decided to select four FEC. From Fig. 5, which shows the EC scores of the genes in each expression class, we can infer that almost every functional express class can be characterized as either being more conserved or more diverged than average. The distributions of the functional classes in both species follow a common pattern as either a diverged or a conserved background distribution. The conserved classes have a distribution with a peak at higher EC scores and a tail to the left, whereas the diverged ones follow a more normal-like distribution centered around a lower EC score. It leaps to the eye that only FEC1 and FEC2 (for 14028s) and FEC3 (for LT2) would generate meaningful results because their distributions show a clear diverged behavior.

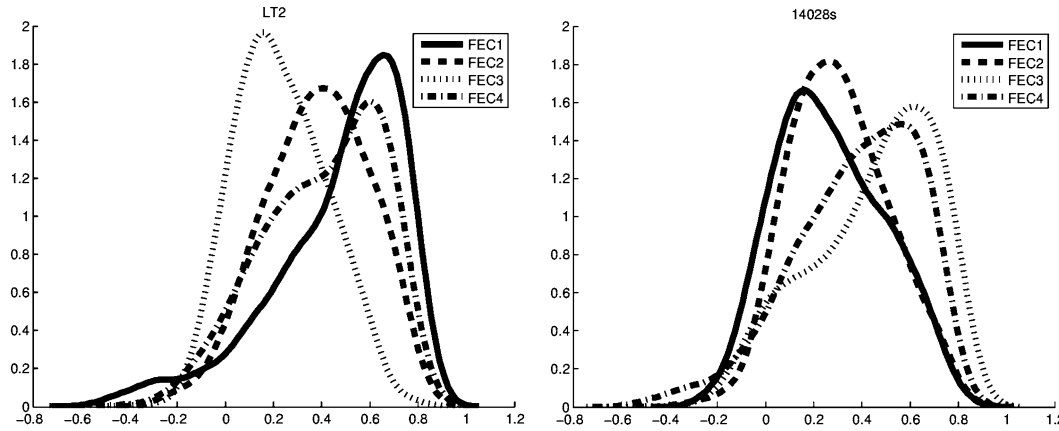

Fig. 5 EC score distributions for the individual FECs for both strains

References

1. Tirosh I, Barkai N (2007) Comparative analysis indicates regulatory neofunctionalization of yeast duplicates. Genome Biol 8:R50
2. Meysman P, Sonego P, Bianco L, Fu Q, Ledezma-Tejeida D, Gama-Castro S, Liebens V, Michiels J, Laukens K, Marchal K, Collado-Vides J, Engelen K (2014) COLOMBOS v2.0: an ever expanding collection of bacterial expression compendia. Nucleic Acids Res 42:D649–653
3. Li L, Stoeckert CJ, Roos DS (2003) OrthoMCL: identification of ortholog groups for eukaryotic genomes. Genome Res 13:2178–2189
4. MATLAB and Statistics Toolbox Release (2012) The MathWorks, Inc., Natick, MA, USA
5. Meysman P, Sanchez-Rodríguez A, Fu Q, Marchal K, Engelen K (2013) Expression divergence between *Escherichia coli* and *Salmonella enterica* serovar Typhimurium reflects their lifestyles. Mol Biol Evol 30:1302–1314

Chapter 9

Raw Sequence Data and Quality Control

Giovanni Bacci

Abstract

Next-generation sequencing technologies are extensively used in many fields of biology. One of the problems, related to the utilization of this kind of data, is the analysis of raw sequence quality and removal (trimming) of low-quality segments while retaining sufficient information for subsequent analyses. Here, we present a series of methods useful for converting and for refinishing one or more sequence files. One of the methods proposed, based on dynamic trimming, as implemented in the software StreamingTrim allows a fast and accurate trimming of sequence files, with low memory requirement.

Key words Next-generation sequencing, DNA sequence, Trimming, FASTQ, FASTA, QUAL, Base-calling

1 Introduction

DNA sequencing is the process of determining the order of the nucleotides that composed a DNA molecule. Knowledge of DNA sequences is becoming indispensable for a great number of biological fields such as diagnostic, biotechnology, forensic biology, systems biology, and evolutionary biology [1]. The increasing speed of sequencing reached with modern DNA sequencing technology has been crucial in the sequencing of longer and longer complete DNA sequences. In recent years this process has led to the sequencing of entire genomes of numerous types and species of life such as human genome [2], plant genomes, and complete genomes of several microbial species.

When we speak about DNA sequences, normally we refer to "already processed" sequences present in a dedicated database such as NCBI or EMBL. However, we have to know that the first type of sequence produced by "next-generation sequencing" machine is the so-called flowgram or chromatogram. These sequence types are represented by a series of peaks along time where each peak is the signal intensity and the time is the order of

Alessio Mengoni et al. (eds.), *Bacterial Pangenomics: Methods and Protocols*, Methods in Molecular Biology, vol. 1231, DOI 10.1007/978-1-4939-1720-4_9,

the bases within the DNA sequence. As a consequence, if we want to transform a chromatogram or a flowgram into a simple DNA sequence (in other words a series of bases) there are several steps that we have to perform.

First of all, we have to use a "base calling algorithm" in order to assign a nucleotide to each peak present in the raw file. The most common "base calling algorithm" is Phred [3]; in fact the quality of each nucleotide inside a DNA sequence is commonly expressed as "Phred quality score". Phred's algorithm uses a probabilistic based quality score estimated using the per-base error probabilities. The quality score, Q, assigned to a base is proportional to its error probability, P, and is calculated using this formula:

$$Q = -10 \log_{10} P$$

Accordingly, a Phred quality score of 30 corresponds to an error probability of 0.1 %. There are also other base caller algorithms as TraceTuner (http://sourceforge.net/projects/tracetuner/) or LifeTrace [4] but, for the purpose of this chapter, their differences are very small and we have no specific recommendations from the ones here described.

After the base calling step, two different files are generated: one file containing the sequence data (the nucleotide sequence, normally in FASTA format) and the other file containing a series of quality scores separated by a white space. This file format is called QUAL file and is one of the standard file formats used by bioinformaticians [5]. However, this is not the only file format used for storing nucleotide data and quality data. In fact, a different file format able to store a numeric quality score associated with each nucleotide in a sequence is commonly used and is becoming the de facto standard for storing the output of high-throughput sequencing instruments. This format is called the FASTQ format; no doubt because of its simplicity, the FASTQ format has become widely used as a simple interchange file format. Unfortunately the FASTQ format suffers from the absence of a clear definition bringing to light some incompatibilities between its different encodings.

Normally, a FASTQ file uses four different lines to store a DNA sequence with its quality. The first line contains the id of the sequence and is preceded by a "@" character followed by the sequence identifier. The second line contains the DNA sequence itself as a repetition of four characters, one per each nucleotide ("A" for adenine, "C" for cytosine, "T" for thymine, and "G" for guanine). The third line starts with a "+" character that may be followed by a repetition of the sequence id (the same contained in the first line) or not. Finally, the fourth line contains the quality values, and must contain the same number of symbols

as letters in the sequence. Here is an example of a FASTQ sequence as reported in [5]:

```
@SRR014849.1 EIXKN4201CFU84 length=93
GGGGGGGGGGGGGGGGCTTTTTTTGTTTGGAACCGAAAGGGTTTTGAATTTCAAACCCTTTTCGGTTTCCAACCTTCCA
AAGCAATGCCAATA
+SRR014849.1 EIXKN4201CFU84 length=93
3+&$#""""""""""7F@71,'";C?,B;?6B;:EA1EA1EA5'9B:?:#9EA0D@2EA5':>5?:%A;A8A;?9B;D
@/=<?7=9<2A8==

@ [title and optional description]
sequence line(s)
+ [optional repeat of title line]
quality line(s)
```

The quality line is encoded using a simple ASCII character for each base of the sequence. In fact, each ASCII character can be represented as an integer ranging from 0 to 128 [6]. However, the quality score of a nucleotide can range from 0 to 40 Phred (41 for the most recent Illumina machines). In order to scale ASCII values according to Phred quality scores we have to subtract an offset from each ASCII value and use only a portion of the ASCII scale. Unfortunately, there are more than one different encoding for FASTQ quality format but the conversion between these different standards is very straightforward. Thus, all we have to do is to subtract to the ASCII value an offset specified by the FASTQ encoding. The offsets most commonly used are 33 and 64 as reported in the Wikipedia FASTQ file format page [7]:

```
SSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSS.............................
.......................XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX..
..............................IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII..
.................................JJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJ..
..LLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLL.............................
!"#$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ(\)^_`abcdefghij
|                         |    |        |                              |
33                        59   64       73                             104

S - Sanger - Phred + 33,  raw reads typically (0, 40)

X - Solexa - Phred + 64, raw reads typically (-5, 40)

I - Illumina 1.3 - Phred + 64,  raw reads typically (0, 40)

J - Illumina 1.5 - Phred + 64,  raw reads typically (3, 40)

L - Illumina 1.8 - Phred + 33,  raw reads typically (0, 41)
```

Despite all this different encoding formats and the absence of a clear definition, the FASTQ file format has a great advantage in respect to the FASTA+QUAL file format: it uses only 1 character for the encoding of a nucleotide quality instead of 2 or 3 character (1 or 2 for the quality and 1 for the withe space) used by the

QUAL file. In fact, if we consider that a simple character uses 1 byte to store its value, a FASTQ sequence of 1,000 nucleotides will use about 2,000 bytes of space while a FASTA + QUAL sequence of the same length will use from 3,000 to 4,000 bytes. In addition, if we consider that DNA sequencing cost is decreasing year by year at the same speed that DNA sequencing data is increasing in size, using a "more compressed" file format to store DNA sequences and their quality values is certainly a better choice.

When all the steps described above have been completed, it is time for the central steps of this chapter: the quality control step. One of the most important problems related to the production and utilization of DNA sequence reads is the analysis of base quality and removal (trimming) of low-quality segments while retaining sufficient information for subsequent analyses [8]. Several trimming algorithms and software programs have been developed to cope with the cleanup of DNA sequence reads, e.g., SolexaQA DynamicTrim [9], FASTX-ToolKit (http://hannonlab.cshl.edu/fastx_toolkit), ConDeTri [10], and NGS QC Toolkit [11]. However, all these software were developed in order to be used by expert bioinformaticians; in fact they have not been equipped with a graphical user interface and the setting of their parameters has to be hand made by the user.

To overcome this limitation imposed by the existing trimming software programs, we have developed StreamingTrim [12] using standard Java language and BioJava libraries [13] (included in the package). This software uses a very flexible "dynamic window" algorithm to remove low-quality segments of DNA sequences, beginning from the end of each read in a sequence file. This approach is very useful because it allows users to set a more stringent quality cutoff, which increases the read quality and reduces the risk of losing too much information. In addition, due to its graphical user interface, StreamingTrim can be simply installed and launched, allowing the software to be used even by inexperienced bioinformaticians, easily permitting "wet lab" molecular ecologists to analyze their data.

In Fig. 1 we report a comparison of StreamingTrim and other four commonly used trimming software (SolexaQA DynamicTrim, ConDeTri, NGS QC Toolkit, and Mothur [14]). In order to compare the number of removed bases and the quality increment in two sample datasets using a single metric, we introduced a trimming performance estimator, called *Z*-score. This estimator is proportional to the ratio between the increase in quality and the decrease in the number of bases for each dataset. The *Z*-score was calculated as follows:

$$Z_{\text{score}} = \log_{10}\left(\frac{Q_{\text{diff}}}{|L_{\text{diff}}|}\right)$$

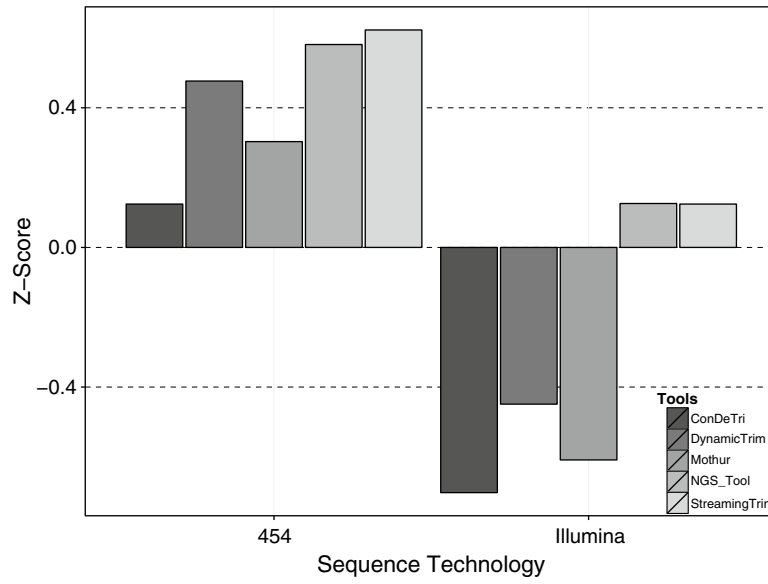

Fig. 1 *Z*-score of different trimming software programs. Bar charts of the *Z*-score after executing the trimming on two datasets (Illumina and 454) are shown. *Negative values* of the *Z*-score indicate that the percentage of bases lost during the trimming process is higher than the percentage of increase in quality. *Positive values* of the *Z*-score indicate that the quality increase is higher than the percentage of bases lost

where:

$$Q_{\text{diff}} = \frac{\left(Q_f - Q_i\right)}{\left(Q_{max} - Q_i\right)} \quad \text{and} \quad L_{\text{diff}} = \frac{\left(L_f - L_i\right)}{\left(L_{min} - L_i\right)}$$

with:

Q_i = initial average quality ; L_i = initial number of bases

Q_f = final average quality ; L_f = final number of bases

L_{min} = minimum final number of bases
(if users do not specify the minimum length parameter, this value is set to 0)

Q_{max} = maximum final quality
(for Phred score this parameter is set to 40)

The results obtained with all tested trimming tools considered on the 454 and Illumina datasets showed that StreamingTrim had the highest *Z*-score values (Fig. 1), indicating the presence of a good compromise between base conservation and increase in read quality.

1.1 Note to This Chapter

As you may have noticed in this manual we use some type-setting conventions. We use:

```
this format
```

in order to refer to command line input or output, but also to refer to external text (for example a DNA sequence contained in a sequence file); when we want to indicate a program menu or function we use <this format>. If you see something like <File → Open File> it means that we refer to the Open File item in the File menu.

2 Materials

All software used in this chapter can be downloaded for free. StreamingTrim is distributed under the BSD-2-Clause license; if you want to learn more about this kind of license visit the page http://opensource.org/licenses/BSD-2-Clause. Since StreamingTrim keeps in memory only one sequence at a time, it can be used even with a standard desktop PC or a laptop. However we recommend having at least 1 or 2 Gigabytes free for each 500 Megabytes of raw data. In this chapter we assume that you have your sequences in FASTQ file format; however, if it is not your case, here we report a two-step procedure in order to convert your chromatogram files into FASTQ file. If you have your sequences already in FASTQ file format you can ignore the two subheadings described below.

2.1 Obtaining Sequence Data from Chromatograms

In order to generate a sequence file you have to perform at least one base calling step as described in Subheading 1.

1. Download and install Phred from http://www.phrap.org/phredphrapconsed.html.
2. Run Phred on your raw sequence file. Here is an example using the standard Phred analysis:

```
phred -id chromat_dir -sa seqs_fasta -qa seqs_fasta.qual
```

Running this line will convert all chromatogram files present in the chromate_dir directory into two files: a FASTA file called `seqs_fasta` and a QUAL file called `seqs_fasta.qual`.

2.2 Converting the FASTA + QUAL Files into One FASTQ File

There are many tools able to encode a FASTQ file starting from a FASTA file and a QUAL file; here we report only one script developed by the Bio-Linux community [15] (http://nebc.nerc.ac.uk/) in order to be as simple as possible.

1. Download and install Phyton from http://www.python.org/download/.
2. Download and install Biophyton from http://biopython.org/wiki/Download.

3. Download the script called `fasta_to_fastq.py` from the Bio-Linux community: http://nebc.nerc.ac.uk/tools/code-corner/scripts/sequence-formatting-and-other-text-manipulation.
4. Run the script as described below:

```
fasta_to_fastq.py input.fna
```

The script does not care if you use a different FASTA extension but there must be a file named `input.qual` containing the phred quality scores; otherwise the FASTQ file will not be generated.

2.3 Downloading StreamingTrim

StreamingTrim is a software built using Java 1.7, so you have to ensure that you have at least Java 1.7.0 version installed on your system. In order to do this you have to open your command windows (cmd.exe in Windows systems and terminal in OS systems) and type this:

```
java -version
```

If you receive an error message it means that you do not have Java installed on your system. Otherwise, if you receive a message like this one:

```
java version "1.7.0_09"
OpenJDK Runtime Environment (IcedTea7 2.3.4)
OpenJDK 64-Bit Server VM (build 23.2-b09, mixed mode)
```

If the number between brackets is smaller than 1.7.0 it means that you have Java installed on your system but you have an old version of the software. In both cases you have to install an up-to-date *Java Runtime Environment*; you can download it from the oracle website: http://www.java.com/en/download/ (if you have an old version of Java it is recommended that you uninstall it before installing the new version). Otherwise, if your Java version is up to date you can proceed to download the software from the GitHub repository at https://github.com/GiBacci/StreamingTrim and save it in a folder of your choice.

2.4 Running StreamingTrim for the First Time

Once you have downloaded the software you can launch it by double clicking one of the two launchers present in the software's folder. If you have a Microsoft Windows-based system you have to use the `windowsLauncher.bat` file, while if you have a Linux-based system or a Mac OS-based system, you can launch it with the `unixLauncher.sh` file (remember to allow executing file as an application). If everything has gone well you would be able to see the main window of StreamingTrim software. Now you are able to analyze your FASTQ files and trim them using this trimmer.

2.5 StreamingTrim Workflows

StreamingTrim algorithm workflows and example steps are reported in Fig. 2. Given a DNA sequence of length N, the algorithm starts

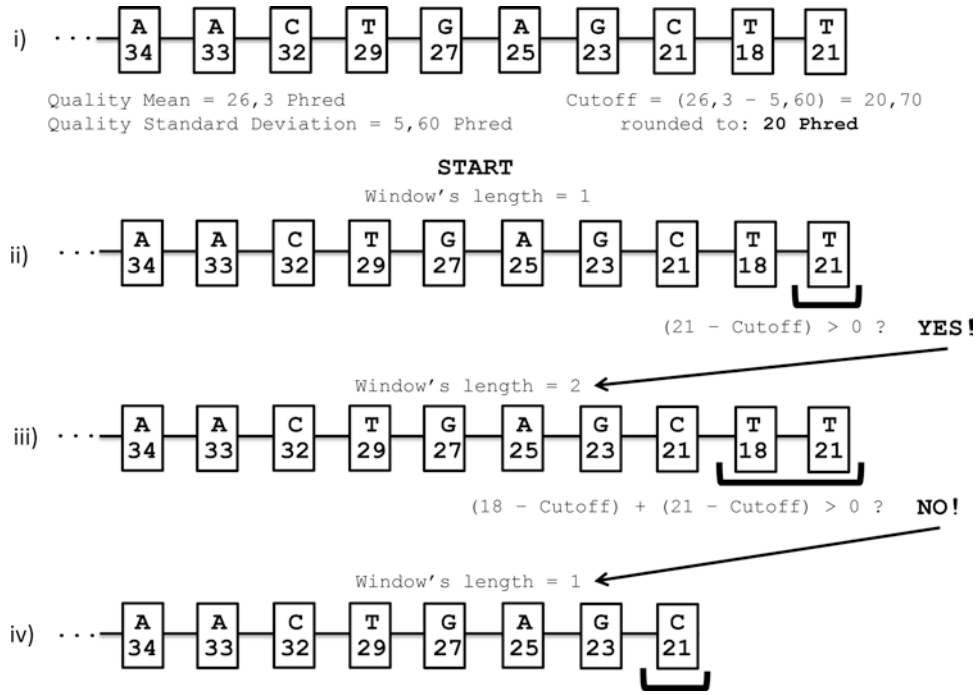

Fig. 2 Workflow of the StreamingTrim algorithm. First (1), a sample sequence is selected from a sequence file with a mean quality of 26,30 Phred and a quality standard deviation (SD) of 5,60. Then (1), a quality cutoff is calculated by subtracting one SD from the quality mean. Next (2), the last base of the sequence is analyzed by subtracting the previously obtained cutoff from its quality value. If this result is bigger than 0, the base is maintained and (3) the analysis window is increased by one. Now, the quality of each base is analyzed as in step (2) and the results are summed up. In the displayed example, the result is less than 0 and, consequently (4), the two bases are removed from the sequence and the size of the analysis window is set again to 1. All these steps are repeated until the sequence has been entirely analyzed

from the last nucleotide (the n^{th} nucleotide), using a window length (W) of 1 and checks if:

$$\left(\text{Quality}_{n^{th}} - \text{cutoff}\right) \geq 0$$

If this is true, the algorithm will proceed by enlarging the window length by 1 (in this case putting $W = 2$); otherwise the n^{th} nucleotide is removed. N is then decreased by the number of removed nucleotides (in this case 1) and W is set to 1. This process is repeated until the algorithm reaches the first nucleotide of the DNA sequence ($N = 1$), or if the trimmed sequence length goes below a minimum value previously chosen by the user (default 1). A formal description of the algorithm is shown here:

$$N = \text{sequence length};\ W = \text{window length};\ M = \left(N - W\right)$$

$$T = \sum_{M < k \leq N} \left(\text{Nucl}_k - \text{cutoff}\right)$$

$$\text{If } T \geq 0 \rightarrow (W+1); \text{If } T < 0 \rightarrow N = (N-W) \,\&\, W = 1$$

Continue with the test T until $(N - W) \leq 0$ or $N <$ minimum length.

The above reported algorithm has been developed in order to be as conservative as possible. In fact, a DNA segment is deleted only if all its nucleotides are considered to be of low quality. If there are only a few low-quality bases in a sequence, the segment is maintained in order to prevent loss of information.

3 Operating Procedure

Here we describe the crucial steps to perform in order to check the quality of a sequence file.

3.1 Analyzing the Reads

In order to prepare the trimmer for the quality refinement, it is better to perform at least one quality control step.

3.1.1 Open a FASTQ File

To open a sequences file in the program the user can click on <File → Open File> in the main window of the program or type the "Ctrl + o" shortcut on his or her keyboard. After that, the file open windows will appear on the screen and the user can select the file to open. Unfortunately, the FASTQ file format does not have a well-defined set of extensions; .fastq, .fq, and .txt are the most used. If the user has a FASTQ file with another extension he or she must select the "All file" option in the extension menu in the <File Open> windows and then select the right file to open; otherwise he or she will not be able to see and select his or her file. After delecting the file and pressing the <Open File> button the <Input File> section in the main windows will fill with the path to the selected file.

3.1.2 Analyzing the File

After a sequence file is successfully opened the user can analyze it in order to see the quality and length distribution of the DNA sequences present in the file. If the user has not opened a file yet, when he or she presses the <Analyze> button, an <Open File> window will appear and he or she can select the interested file from here.

In order to analyze the file the user has to press the <Analyze> button in the <Controls> section of the software main window. When the user presses the button, the <Progress Bar> will begin to move and the file will be analyzed. After that, the <Reads Properties> window will display all the statistics related to the file. If the user wants a more accurate description of quality and length distribution, he or she can press the <Plot> button in the <Controls> section of the main window; <Plot Window> opens and the software begins to deeply analyze all the sequences in the file. When the program has finished analyzing data a plot will appear in the <Raw data> section of the <Plot Window>.

Two different kinds of plot can be displayed in the <Plot Window>:

1. <Deviation Plot> is a representation of the DNA base quality distribution along each sequence. In the *x*-axis the length of the sequences is reported. If there are sequences with different lengths, then the length of this axis is the length of the longest sequence. In the *y*-axis the quality values from 0 to 40 are reported. The mean quality is represented as a bold line while the range between maximum quality value and minimum quality value is represented as a blue surface. In this way the user can see the distribution of every base quality, and not only the mean or the standard deviation.
2. <Box Plot>: This is a standard box plot representation of the quality distribution for each sequence in the sequence file. If you have reads longer than 200 nucleotides, this type of visualization can be very difficult to read; otherwise if you have short reads (about 100–150 nt) this plot can be very useful since also the median and the first and third quartile (as a normal boxplot) are reported.

There is also another kind of plot that can be displayed in the <Plot Window>, the so-called length plot. This plot gives the user a bar chart representation of the read length distribution. Here, only one type of plot is possible, where in the *x*-axis the sequence length values (they can change by changing the input file) are reported and in the *y*-axis the number of reads in the file that has the corresponding length value is shown.

The user can zoom anywhere in the plot, by simple clicking and dragging with the mouse the part of the plot that he or she wants to zoom. In the bottom of the plot there is the number of reads that are found in the plotted file.

The user can now save the chosen plots by simply right clicking them and choosing the "Save as" option in the pop-up menu.

3.2 Parameter Settings

In the <Advanced Option> window (accessed through <Window → Show advanced option>) the user can specify some trimming parameters in order to adjust the trimming process to his or her will. Here, all the advanced options are described in order to understand the complete StreamingTrim functionality.

3.2.1 Cutoff

This parameter represents the quality cutoff to be used by the software during the trimming process. Typically, the quality range of a FASTQ sequence file goes from 0 to 40, representing hypothetical error probabilities of 100 % and 0.01 %, respectively. If this parameter is not selected, the trimmer chooses a cutoff automatically based on the mean quality and the standard deviation of the reads in the given file (e.g., if we have a file with a mean quality of 31.46 and a standard deviation of 6.54, the quality cutoff is set to 31.46–6.54 = 24.92 and approximated up to 25).

The user can change this parameter in order to perform a more or less stringent quality refinement by using higher or lower cutoff values, respectively.

3.2.2 Offset

This parameter indicates the number of bases to eliminate at the beginning of every reads. Setting a value higher than 0 is useful when the presence of adapters or some unwanted region at the beginning of each sequence is known. Otherwise it is recommended to leave this parameter unchecked.

3.2.3 Minimum Length

With this parameter the user can specify a length cutoff (in bases). Sequences that, after the trimming process, have a length lower than this parameter are not saved in the output file. This parameter is very useful in amplicon-based analysis, where reads that result too short after trimming are useless for the following analyses (e.g., taxonomic identification).

3.2.4 General Considerations

It is recommended to choose this set of parameter based on the previously done analysis of the sequence quality. In fact, for example, choosing a cutoff parameter too small in a very-poor-quality sequence file could lead to inconclusive results. On the other hand, choosing a too high value of cutoff for a very-poor-quality FASTQ file could generate a file with too few sequences. If the user is not sure about the setting of these parameters, the better choice is to let everything unchecked.

3.3 Trimming

The principal function of StreamingTrim is to cut low-quality bases from each sequence in a DNA sequence file. First of all, in order to start the trimming process, the user has to open a valid input file as described in Subheading 3.1.1. Then, the user can proceed to start the analysis clicking on the <Trim> button in the main window of the StreamingTrim interface. When the <Trim> button is pressed a <Save File> window appears and the user can choose the destination and the name of the file containing the trimmed reads. After that the <Progress Bar> begins to move and the trimming process starts using the default trimming parameters or the user-defined parameters (if previously specified, *see* Subheading 3.2).

When the trimming process reaches the end an output file will be saved as previously specified by the user. The output file will be in the same format as the input file and will use the same FASTQ offset (*see* Subheading 1).

3.3.1 The <Trim to FASTA> Function

StreamingTrim can convert a trimmed file into FASTA format while the trimming process goes on. If the checkbox <Trim to FASTA> in the main window is selected, when the user starts the trimming process the software simultaneously converts the output file to FASTA format. When the checkbox is selected from the user, a <Save FASTA file> window opens and the user can choose the directory and the file name he or she prefers.

This function is very useful if there is a need to trim more than one file with the same parameters, without analyzing them each time. In this way the trimming and conversion processes are speeded up.

3.3.2 Controlling Results

Results obtained after the trimming process can be analyzed as described in Subheading 3.1. In the plot window the user can compare the two graphic representations of the sequence file before and after the trimming process. This can be useful in order to check if the result obtained with the set of parameters chosen is satisfactory or not.

If the average quality of the trimmed reads is still too low, the user can repeat the trimming process specifying a more stringent cutoff value. It is recommended to trim the original file again in order to be as much reproducible as possible. If the user attempts to trim an already trimmed file he or she will not be able to repeat the same analysis unless he or she does not perform again the two trimming processes with exactly the same parameters. On the other hand, if the user chooses to trim the original file he or she will be able to reach the same results with only one step.

3.4 Converting Raw Sequencing Data

When the quality refinement step has reached a satisfactory conclusion, it is recommended to convert the raw sequence file (in this case in FASTQ format) into a more suitable sequence format. The most used file format for DNA sequences is the FASTA file format. StreamingTrim can convert FASTQ file into FASTA after the end of the trimming process or even in the same time (as seen in Subheading 3.3.1). If the user wants to convert the refined FASTQ file all he or she has to do is to click the <FASTA> button in the main window of the program. Doing this will cause the <Progress Bar> to start moving and a FASTA file will be created.

References

1. Pettersson E, Lundeberg J, Ahmadian A (2009) Generations of sequencing technologies. Genomics 93:105–111
2. Sawicki MP, Samara G, Hurwitz M, Passaro E (1993) Human genome project. Am J Surg 165:258–264
3. Ewing B, Hillier L, Wendl MC, Green P (1998) Base-calling of automated sequencer traces using Phred. I. Accuracy assessment. Genome Res 8:175–185
4. Walther D, Bartha G, Morris M (2001) Base calling with lifetrace. Genome Res 11: 875–888
5. Cock PJ, Fields CJ, Goto N, Heuer ML, Rice PM (2010) The Sanger FASTQ file format for sequences with quality scores, and the Solexa/Illumina FASTQ variants. Nucleic Acids Res 38:1767–1771
6. Wikipedia (2014) ASCII. Wikipedia, the free encyclopedia
7. Wikipedia (2014) FASTQ format. Wikipedia, the free encyclopedia
8. Kunin V, Copeland A, Lapidus A, Mavromatis K, Hugenholtz P (2008) A bioinformatician's guide to metagenomics. Microbiol Mol Biol Rev 72:557–578
9. Cox MP, Peterson DA, Biggs PJ (2010) SolexaQA: at-a-glance quality assessment of Illumina second-generation sequencing data. BMC Bioinformatics 11:485

10. Smeds L, Künstner A (2011) ConDeTri-a content dependent read trimmer for Illumina data. PLoS One 6:e26314
11. Patel RK, Jain M (2012) NGS QC Toolkit: a toolkit for quality control of next generation sequencing data. PLoS One 7:e30619
12. Bacci G, Bazzicalupo M, Benedetti A, Mengoni A (2014) StreamingTrim 1.0: a Java software for dynamic trimming of 16S rRNA sequence data from metagenetic studies. Mol Ecol Resour 14:426–434
13. Holland RC, Down TA, Pocock M, Prlić A, Huen D, James K, Foisy S, Dräger A, Yates A, Heuer M (2008) BioJava: an open-source framework for bioinformatics. Bioinformatics 24:2096–2097
14. Schloss PD, Westcott SL, Ryabin T, Hall JR, Hartmann M, Hollister EB, Lesniewski RA, Oakley BB, Parks DH, Robinson CJ (2009) Introducing mothur: open-source, platform-independent, community-supported software for describing and comparing microbial communities. Appl Environ Microbiol 75: 7537–7541
15. Field D, Tiwari B, Booth T, Houten S, Swan D, Bertrand N, Thurston M (2006) Open software for biologists: from famine to feast. Nat Biotechnol 24:801–804

Chapter 10

Methods for Assembling Reads and Producing Contigs

Valerio Orlandini, Marco Fondi, and Renato Fani

Abstract

Determining the genome sequence of an organism is often the first step towards its molecular characterization. Once a difficult and expensive task, nowadays it is an almost routine practice in many molecular biology labs. In this chapter we discuss in depth the various methods to assemble the short sequences (called reads) obtained from a massive sequencing system, using different software and strategies, and how to perform some fundamental quality controls on the data obtained.

Key words Genome assembly, Assemblers, Reads, Bioinformatics, Genomics

1 Introduction

The ease at which genomes are currently sequenced has conferred to genomics a platform-like role in the overall path to a system-level comprehension of microorganisms and ecosystems. Indeed, microbial genome sequencing has become affordable for most of the laboratories worldwide. Both time and cost for single run have dropped dramatically in recent years, allowing fast (e.g., 2 h with Ion Proton, Life Technologies) and cheap (e.g., less than 100€/genome on a multiplexed Illumina plate) sequencing of a medium-sized microbial genome. Nevertheless genomics data post-processing still represents a major bottleneck in the overall cascade of analyses that can be performed to gain a complete picture of microbial genomes.

Sequencing reads are the typical result of a sequencing run. For bacterial genomes, depending on the technique used, the number of these reads may vary from a few hundred thousand (e.g., with 454 Roche sequencing technology) up to several millions (e.g., adopting IlluminaHiSeq). Also, the length of reads may greatly vary among the different sequencing platforms ranging from few hundred bases ("short" reads, Illumina and Ion Torrent) up to more than 1,000 bases ("long" reads, PacBio technology). With the exception of those cases in which sequencing reads are

Alessio Mengoni et al. (eds.), *Bacterial Pangenomics: Methods and Protocols*, Methods in Molecular Biology, vol. 1231, DOI 10.1007/978-1-4939-1720-4_10,

enough to guide the exploration of particular biological features [e.g., single-nucleotide polymorphism (SNP) detection], sequence assembly is the first challenge encountered in a typical computational genomics pipeline. It basically involves the merging and the ordering of shorter sequence fragments (reads) with the aim to get as close as possible to the original larger sequence (genome).

Assemblers rely on the basic assumption that two reads (two strings of letters produced by the sequencing machine) sharing a common sequence of bases may have originated from the same place in the genome. Using such basic concept, although with different flavors (see below), assemblers can join the contiguous sequences together into a consensus sequence, eventually producing a collection of unique sequences, also called contigs. Early genome assemblers used a "greedy" algorithm, according to which all reads are compared with each other, and the ones that overlap most are merged first. At the end of this procedure, the reads with the longest overlap are concatenated to form a contig. For some assemblers, this resulted in the construction of an overlap graph, in which each read is represented by a node and (weighted) edges account for the overlaps among the reads [1].

However, currently widely exploited sequencing technology (e.g., IlluminaHiSeq, IonTorrent) is characterized by a huge number of sequenced bases for each run (i.e., high coverage) and an overall shorter length of produced reads.

Although, in principle, the overlap graph-based strategy for genome assembly may still be applied to shorter sequencing, it becomes computationally infeasible and less accurate as the number of reads increases. To address these issues, a new generation of genome assemblers has been developed. These, rather than using an overlap graph, use a de Bruijn graph algorithm [2]. In this approach, the reads are decomposed into *k-mers* that in turn become the nodes of a de Bruijn graph. A directed edge between nodes indicates that the *k-mers* on those nodes occur consecutively in one or more reads. Contigs can then be derived from this network by walking all the paths formed by unambiguous stretches of "connected" sequences.

The output of a de novo assembly is typically a draft genome, consisting of a set of contigs (i.e., contiguous sequence fragments) that may be ordered and oriented into scaffold sequences, with gaps between them, representing regions of uncertainty [3]. In this chapter we present the basic usage of some of the most widely adopted assembly tools, including both "short" and "long" read assembler. Also, possible parameters for understanding and estimating the reliability of output data are described.

2 Materials

In this section we describe what is needed to assemble a genome starting from the raw reads obtained from the sequencer, and how to perform some basic analyses on it.

2.1 Raw Read File(s)

Modern massive genome sequencers use different technologies to sequence DNA molecules, but basically most of them return one or more text files containing the reads in FASTQ format. The FASTQ is similar to the FASTA file, but for every base there is a quality (hence the Q) information. It is represented by an ASCII character, and, in practice, indicates how much the information about the corresponding base is reliable. In the previous chapter you learnt how to refine the raw reads by trimming them, so that the low-quality regions are wiped out and do not interfere in the assembly process.

Some sequencers can return the reads in a different format, like the Roche 454 that gives `.sff` files. These files can be easily converted into FASTQ or, with some software, used as they are.

Depending on the sequencing method, you could have single-end or paired-end reads. In the latter case, they can be joined in one single file or split into two files with similar names. In any case, if you have more than one set of reads (resulting from multiple sequencing runs), you can use them all together (if you have enough RAM and CPU power) or assemble each set separately and then try to assemble the resulting contigs with a dedicated software (like Phrap, about which we will discuss later on).

2.2 Hardware and Software

One of the aims of the modern assembly software is to be able to run on computers with limited hardware resources, making it possible to use common desktop computers to perform tasks once accomplished on large mainframe systems. The limiting factors you have to take into account are the CPU power (using a multicore processor is rather important) and the RAM amount, crucial to load the reads to be assembled into memory. For bacterial genomes, 3–4GB is usually fine; for larger genomes you will need much more memory. Anyway, today additional RAM banks are very cheap, so remember to use a computer with a motherboard supporting at least 16GB of RAM: you will be able to add it in a second time.

As for the operating system, most bioinformatics tools, including assemblers, are developed and optimized for the GNU/Linux operating system or for any UNIX and UNIX-like OS (thus including the various BSD variants, the commercial UNIX distributions, and, to a certain extent, Mac OSX). Most of the assemblers can run on Windows, too. Anyway, the use of GNU/Linux is strongly encouraged as it is the development platform of most of the stuffs you are going to use. Any distribution is fine, as long as it is regularly updated and does not ship with outdated software.

3 Methods

Short and long reads saw the development of different assembly software. Here we present the most common and efficient assemblers in these two categories. Keep in mind that the distinction between short and long reads is not marked by an exact value, and with the latest short read sequencers giving longer and longer reads, the separation line becomes more and more blurred. It is the same for the assemblers: most of the times an assembler designed for short or long reads simply means that it has been tested and performs better for one kind of reads, without being unusable for the other one.

3.1 Short Read Assembly

3.1.1 ABySS

ABySS [4] is a rather widespread assembler developed by the Canada's Michael Smith Genome Sciences Centre and released with GPL open-source license for noncommercial uses. It is designed for short reads and its points of strength are ease to use, portability, and a general good quality of the assemblies.

For single-end reads, the command to perform the assembly is

```
ABYSS -k [int] reads.fastq -o my_assembly.fna
```

For paired-end reads, a script called `abyss-pe` is used. You need at least a couple of paired-end reads; the assembly is performed with the command

```
abyss-pe in='my_reads_1.fastq my_reads_2.fastq' k=[int] out=my_assembly
```

Several options can be passed to `ABYSS` and `abyss-pe`, but for the most used cases, the ones shown in these examples are the only ones you have to be aware of. The first and the only compulsory one (besides the read file names) is the k-value (`-k` or `k` option, for `ABYSS` and `abyss-pe`, respectively). The k-value is related to dimension of the k-mers, which are the subsequences in which the assembler splits the reads to make the comparisons leading to a possible overlap. To find the k-value that brings to the best possible assembly, you should try several k-values, comprised in the interval between half the mean read length of your set and the mean read length itself. The standard version of ABySS (i.e., the one you usually find precompiled in your distro repository) supports k-values up to 64. This value is rather low for most read sets, since, as written before, it is advisable to keep this value to at least half of the mean read length. To overcome this issue, you can build ABySS yourself, specifying the maximum k-value that the assembler can support. A maximum value of 96 will be enough for most of the short read sets. Another useful option is `-out=`, used to declare a prefix to all the files generated by the current ABySS run. Since it is advisable to perform more than one assembly, each at a different k-value, you will likely want to append the k-value of each assembly you perform. If you want to automate this process, you can use a

utility like Assembl-o-matic, which will perform different assemblies (at different k-values) and choose the best one for you (available at http://www.dbefcb.unifi.it/CMpro-v-p-8.html).

3.1.2 Velvet

Velvet [5] is another popular option among genome assemblers. Developed by the European Bioinformatics Institute, it is suitable for both short and long read assembly, be they single or paired ends.

The assembly with Velvet consists in two distinct steps, using the programs `velveth` and `velvetg`, respectively. The first one constructs a data set from the reads, used in the second step to construct a de Bruijn graph which finds the overlaps and builds the contigs.

You have to create a directory containing the reads you want to assemble: Velvet will work inside it and generate all its outputs there. In a bash terminal you will have to run the following command:

```
mkdir my_directory
```

Then, run `velveth` with

```
velveth my_directory [hash_length] [file options] my_assembly
```

Hash length is substantially the same as the k-value. It must be an odd integer not higher than 149. The file options to be specified concern the format of the reads: for example, if you have short paired-end reads in FASTQ format, the command will look like

```
velveth my_directory [hash length] -shortPaired -fastq my_assembly
```

Once this first step is finished, finalize your assembly with

```
velvetg my_directory
```

With a series of well-documented options you can fine-tune your assembly. The output will consist of three files: the FASTA file with the sequences, a tab-separated text file with the statistics of the assembly (useful to make adjustments in the parameters used), and a file with the last de Bruijn graph used for the read assembly.

3.1.3 Ray

Ray [6] is another powerful genome assembler, optimized to be used on multicore systems. Running Ray requires the usual set of information: read files, k-value, output file prefix. For paired-end reads:

```
Ray -k [int] -p my_reads_1.fastq my_reads_2.fastq -o my_assembly
```

If you have a multicore system and OpenMPI (Open Source High Performance Computing, available at http://www.open-mpi.org/) installed, you can use multiple cores by appending

```
mpiexec -n [int]
```

before the command. The `-n` option indicates how many cores you want to use.

3.1.4 Edena

Edena [7, 8] is probably a less known assembler compared to the previous ones, but it represents a very good alternative. Its main points of strength lie in fast execution, identification (where the resulting contigs allow this) of circular molecules such as plasmids and complete chromosomes, stability, and a two-step process that permits multiple assemblies with different parameters without having to start all the assembly from the beginning each time.

As mentioned before, assembling with Edena is a two-step process (like Velvet): in the first one Edena runs in overlap mode, and in the second one in assembly mode. Once the overlap file from step one is obtained, several assemblies with different parameters can be performed quickly from this intermediate output.

If you have one or more single-end read file(s), overlap mode is launched with

```
edena -singleEnd reads.fastq -minOverlap [int] -prefix my_overlaps
```

`-minOverlap`, like the k-value, should be at least half of the mean read length. Other options are available: the most notable one is `-truncate [int]`, which truncates all the reads from 3′ end at the wanted length. For paired-end reads, the command is the same, but with `-paired` instead of `-singleEnd`, followed by the pair(s) of read files.

Once this step, which is the most time consuming, has been carried out, you get an overlap file with the `.ovl` extension. This will be the input of the second, much faster, step:

```
edena -e my_overlaps.ovl -prefix my_assembly
```

Also in this case, there are several options worthy of attention and experimentation. The ones you will use most often are (1) `-m` followed by the minimum overlap size to consider, (2) `-c` followed by the minimum contig size (default is 1.5 times the read length), and (3) `-trim`, followed by a value of the coverage cutoff for contig ends (default is 4, with value 1 the contig ends are not trimmed at all).

3.2 Long Read Assembly

Here we present two assemblers that, for different reasons, are suitable for long read assembly. Newbler is a proprietary software designed to assemble the output of the Roche 454 sequencer that for its nature generates long reads. Phrap is a general-purpose assembler that, being able to find overlaps and merge even already assembled sequences of any length, is a good choice for long reads. Minimo is another good assembler, released with an open-source license and part of a larger set of utilities called Amos Tools [9].

3.2.1 Newbler

Newbler is a proprietary software developed by Roche Life Sciences and specifically aimed at the reads obtained from their 454 pyrosequencer. These reads are stored in a binary format whose extension is `.sff`. Utilities exist that can convert a `.sff` file into FASTQ format (for example http://bioinf.comav.upv.es/seq_crumbs/ and http://github.com/indraniel/sff2fastq).

Newbler has a graphical interface (`gsAssembler`) and a command line one (`runAssembly`). To run the latter on a read set, just run

```
runAssembly my_reads.sff
```

The graphical version allows exploring the multiple assembly options with ease. FASTA files, with or without quality information, are accepted by Newbler too.

3.2.2 Phrap

We can call Phrap a general-purpose assembler; in fact it can efficiently handle any kind of FASTA file, be it a collection of short reads or already assembled contigs. Phrap is free for noncommercial use, but it has to be requested with an e-mail to its authors. Building it from source is very easy, and should work on a vast spectrum of operating systems.

The basic command line for Phrap is as easy as

```
phrap my_reads.fa
```

Nevertheless, the program offers a vast range of options to adapt the assembler to specific needs. A very handy possibility offered is to generate an `.ace` file, which contains information about the merged sequences, including the mismatches and the gaps generated. A software like Tablet (http://ics.hutton.ac.uk/tablet/) can be used to open such files and visually shows the results of your assembly. To generate an `.ace` file, just append `-ace` to the command line. Another option worth some tweaking is `-minmatch`, which sets the minimum length of the word size (i.e., a subsequence that the software algorithm uses to make comparisons) for a match to be considered. Default value is 14; increasing it makes an assembly more accurate, but some valid matches can be lost (especially when short sequences are used and a long read can be considered a short sequence, since Phrap handles contigs as well). By decreasing the value more matches can be found, but it may be easier to have false positives.

When checking the results, remember to consider not only the contig file, but also the "singlet" one, which features the sequences not assembled into larger units.

3.2.3 Minimo

Minimo is part of the suite AMOS [9], including a large series of bioinformatics utilities. Like Phrap, it can handle reads of all sizes and contigs, too. Since it is fully open-source licensed and freely available to download, it is a viable alternative to Phrap for those who want to incorporate it inside a pipeline free to distribute. The basic command line is as simple as

```
Minimo my_reads.fa
```

Like Phrap, Minimo can export a `.ace` file. Other useful options allow setting the minimum overlap length and the minimum identity between reads. A command line using all these possibilities may look like this:

```
Minimo my_reads.fa -D ACE_EXP=1 -D
MIN_LEN=[int] -D MIN_IDENT=[int]
```

3.3 Understanding and Analyzing Output Data

3.3.1 First Evaluation About Assembly Quality

Every assembler outputs different files, depending on the way it assembles the input reads and on the developers' choices. Anyway, you obviously always find a FASTA file containing the assembled contigs. This is your starting point if you want to evaluate how good and reliable the assembly is. If you know the expected size of the genome you are working on, the first thing to do is to compare it with the total amount of bases assembled. If the two values are too different, and provided that the reads have an average good quality and are not contaminated, the main reasons could be the following:

1. The parameters that you specified (or that you left to the default values) to the assembler led to a bad assembly. Solution: Retry to assemble the reads changing the parameters.
2. The assembler, for different reasons, is not able to efficiently work on your reads. Solution: Try to use another assembler and see if the results significantly differ.
3. The experimental esteem of the genome size is wrong. Solution: If you do not have any other similar strain to make a comparison, repeat the experiments (pulsed field gel electrophoresis) that led to the assumed wrong data.

Other basic procedures for assessing the quality of the sequencing run that should be mentioned here include the following:

1. A comparison of the average GC content %. One should compare the GC content of the newly assembled genome with that of closely related organisms and be sure that the two values fall in the same range.
2. 16SrDNA sequence similarity check: Once assembled, the draft genome can be searched for the presence of 16S rDNA coding sequence and, once retrieved, this can be compared to the one of closely related microbes. Again, if the sample has been correctly sequenced and no contamination occurred, the sequence similarity between the two should fall within the expected range.

The next sessions deal with checking how many contigs you have obtained and their overall length. If you have tried to assemble the reads multiple times with different parameters (and this should be done), usually the best assembly is the one with the least number of contigs with the longest size (provided that its total size is in line with the expected genome size). To clean your assembly, it is a good practice to filter out the contigs that are shorter than a certain length, usually 500 bp.

Following these simple guidelines it is often already sufficient to discriminate between the good assembly and the one that should be repeated. However, there are other technical statistics that should be considered if the assembly of a genome is not just an accessory part of a work, but its main aim.

3.3.2 N50 and NG50 Statistics

Related to a certain extent to what has been discussed so far, there is the N50 statistics to be considered. N50 is a statistical measure of average length of a set of sequences, and is defined as the length for which the series of all contigs of a given length or longer contains at least half of the total length of the contigs, and for which the series of all contigs of that length or shorter contains at least half of the total of the lengths of the contigs. This gives an idea of how the assembly is structured. A high N50 value, for example, means that half of the assembly size is contained in a series of rather long contigs, and the other half in a series of short contigs that significantly outnumber those in the half of the longest ones. With a lower N50, the distribution of contigs from the shortest to the longest one is more even.

Various tools exist to calculate the N50; a simple search on the Web will take you to lots of public domain scripts to obtain the N50 of the sequences contained in a multiFASTA file.

A similar measure, nowadays preferred to the N50, is the NG50, which uses the estimated genome size instead of the total size of the obtained assembly. In this way, the NG50 gives a more accurate esteem of the assembly quality, since it also takes into consideration the coverage of the assembly across the genome.

3.3.3 Improving the Assembly

For most purposes, the assembly you have obtained at this point is informative enough to be used even to release a genome announcement paper. These include, for example, the analysis of the overall gene repertoire of the organism under study and/or the search of specific coding capabilities. There are situations, however, when you need a complete genome (i.e., the full chromosome sequence, not split into multiple contigs), or at least an assembly made up of few contigs, so that, for example, it is easier to find complete gene clusters in one single un-gapped sequence. In these cases there are different strategies you can follow:

1. If the contigs are just a handful or you have to fill only some gaps which you know where they are positioned inside the genome, the most accurate and easy way to accomplish this aim is to perform a PCR using as primers the edges of the two contigs you want to link.
2. You can sequence the genome again with a different technology, then perform the assembly of the new set of reads, and finally merge the two assemblies with a software like Phrap. Due to the differences between two technologies, some parts of the genome could be sequenced only with one method.
3. You can assemble the same set of reads with different assemblers, or with different parameters within the same assembler, and then merge the contigs like in the above point. This is far less effective, because you start with the same set of reads, but it could lead just to some improvement and it is expensive in terms of time.

4. A large panel of tools exist (here a comprehensive list: http://omictools.com/scaffolding/) [10], enabling the scaffolding of draft genomes and/or the closure of the gaps that are commonly present.

If a reference genome (i.e., the complete genome of a phylogenetically near organism) is already available, it is far easier to improve the assembly, or at least give an order to the contigs. To map the contigs on a reference genome several utilities exist. For example, CONTIGuator [11] is an open-source software which aligns the contigs to a given reference genome, providing the user with useful figures and PCR primers, too.

4 Notes

1. How to obtain the software mentioned in this chapter.

Software	Website	Notes
ABySS	http://www.bcgsc.ca/platform/bioinfo/software/abyss	
Assembl-o-matic	http://www.dbefcb.unifi.it/CMpro-v-p-8.html	
CONTIGuator	http://contiguator.sourceforge.net/	
Edena	http://www.genomic.ch/edena.php	
Minimo	http://amos.sourceforge.net/	Part of the AMOS tools suite
Newbler	http://www.my454.com/	Commercial software sold with the Roche 454 sequencer
Phrap	http://www.phrap.org/	Software is free, but has to be requested by mail
Ray	http://denovoassembler.sourceforge.net/	
Velvet	https://www.ebi.ac.uk/~zerbino/velvet/	

References

1. Schatz MC, Delcher AL, Salzberg SL (2010) *Assembly of large genomes using second-generation sequencing*. Genome Res 20:1165–1173
2. Pevzner PA, Tang H, Waterman MS (2001) *An Eulerian path approach to DNA fragment assembly*. Proc Natl Acad Sci U S A 98:9748–9753
3. Earl D, Bradnam K, St John J, Darling A, Lin D, Fass J, Yu HO, Buffalo V, Zerbino DR, Diekhans M et al (2011) *Assemblathon 1: a competitive assessment of de novo short read assembly methods*. Genome Res 21:2224–2241
4. Simpson JT, Wong K, Jackman SD, Schein JE, Jones SJ, Birol I (2009) *ABySS: a parallel assembler for short read sequence data*. Genome Res 19:1117–1123
5. Zerbino DR, Birney E (2008) *Velvet: algorithms for de novo short read assembly using de Bruijn graphs*. Genome Res 18:821–829
6. Boisvert S, Laviolette F, Corbeil J (2010) *Ray: simultaneous assembly of reads from a mix of high-throughput sequencing technologies*. J Comput Biol 17:1519–1533

7. Hernandez D, Francois P, Farinelli L, Osteras M, Schrenzel J (2008) *De novo bacterial genome sequencing: millions of very short reads assembled on a desktop computer*. Genome Res 18:802–809
8. Hernandez D, Tewhey R, Veyrieras JB, Farinelli L, Osteras M, Francois P, Schrenzel J (2013) *De novo finished 2.8 Mbp Staphylococcus aureus genome assembly from 100 bp short and long range paired-end reads*. Bioinformatics 30:40–49
9. Treangen TJ, Sommer DD, Angly FE, Koren S, Pop M (2011) *Next generation sequence assembly with AMOS*. Curr Protoc Bioinformatics. Chapter 11, Unit 11 18
10. Fondi M, Orlandini V, Corti G, Severgnini M, Galardini M, Pietrelli A, Fuligni F, Iacono M, Rizzi E, De Bellis G et al (2014) *Enly: Improving Draft Genomes through Reads Recycling*. J Genomics 2:89–93
11. Galardini M, Biondi EG, Bazzicalupo M, Mengoni A (2011) *Contiguator: a bacterial genomes finishing tool for structural insights on draft genomes*. Source Code Biol Med 6:11

Chapter 11

Mapping Contigs Using CONTIGuator

Marco Galardini, Alessio Mengoni, and Marco Bazzicalupo

Abstract

Obtaining bacterial genomic sequences has become a routine task in today's biology. The emergence of the comparative genomics approach has led to an increasing number of bacterial species having more than one strain sequenced, thus facilitating the annotation process. On the other hand, many genomic sequences are now left in the "draft" status, as a series of contigs, mainly for the labor-intensive finishing task. As a result, many genomic analyses are incomplete (e.g., in their annotation) or impossible to be performed (e.g., structural genomics analysis). Many approaches have been recently developed to facilitate the finishing process or at least to produce higher quality scaffolds; taking advantage of the comparative genomics paradigm, closely related genomes are used to align the contigs and determine their relative orientation, thus facilitating the finishing process, but also producing higher quality scaffolds.

In this chapter we present the use of the CONTIGuator algorithm, which aligns the contigs from a draft genome to a closely related closed genome and resolves their relative orientation based on this alignment, producing a scaffold and a series of PCR primer pairs for the finishing process. The CONTIGuator algorithm is also capable of handling multipartite genomes (i.e., genomes having chromosomes and other plasmids), univocally mapping contigs to the most similar replicon. The program also produces a series of contig maps that allow to perform structural genomics analysis on the draft genome. The functionalities of the web interface, as well as the command line version, are presented.

Key words Bacterial genomics, Genome finishing, Contig mapping, Next-generation sequencing, Scaffolding, Structural genomics

1 Introduction

The field of microbial genetics has seen a strong acceleration in recent years, thanks to the introduction of the so-called next-generation sequencing technologies (NGS). Obtaining the complete sequence of a bacterial isolate has now become a routine task that requires a relative minimum effort, when compared to the previous sequencing technologies. As a result, almost 10,000 bacterial genomes are now available in the GenBank database (9,744, as of January 2014).

Alessio Mengoni et al. (eds.), *Bacterial Pangenomics: Methods and Protocols*, Methods in Molecular Biology, vol. 1231, DOI 10.1007/978-1-4939-1720-4_11,

Another interesting outcome of the advent of the NGS technologies is the establishment of the comparative genomics and pangenome paradigms. For many bacterial species, there are now many genomic sequences available belonging to different strains or isolates, which allowed the development of new analysis focused on highlighting the common and divergent genetic components inside the species pangenome [10, 15]. Such analysis may have a critical impact in applied microbiology, such as clinical comparative studies that may be able to highlight the genetic determinants of pathogenic strains. In fact, 21 % of the 3,376 bacterial species available in the GenBank genome database have more than one strain sequenced, with an average number of 9.91 genomes per species. As the number of available genomes increases, it is expectable that the number of species for which the comparative genomics approach can be applied will increase.

On the other hand, the bacterial genomics revolution has one important drawback: an increase in the number of genomes left in the draft form, such as a series of contigs or scaffolds. Several factors contribute to the impossibility to obtain a closed sequence directly after the sequencing step: the average length of the sequence reads can lead to unsolvable ambiguities in the assembly step, the presence of repetitive elements in virtually all bacterial genomes which also lead to ambiguities, and the limits of the current assembly algorithms [5]. As a result of these limitations, a series of unlinked contigs are produced by the assembly step; to obtain a complete genomic sequence the relative orientation of the obtained contigs has to be resolved and experimentally confirmed, in the so-called finishing phase. Such phase usually involves the generation of PCR primer pairs that span the edges of two contigs, in order to understand if they are linked, and the exact conjunction sequence. When no prior estimate of the relative position of the contigs is available, the number of PCR reactions needed to close the genome (and the consequent cost and labor that are needed as well) is usually too high. By looking at the GenBank bacterial genome database, this resistance in performing the finishing phase is evident: only about 29 % of the 9,744 bacterial genomes is in the final closed form.

The lack of the complete genome sequence may reduce the information that is obtainable from a genomic sequence. The annotation of the genome may be incomplete, since some genetic features (i.e., genes, RNAs) may be split between the edges of two contigs, thus leading to an underestimation in the number of genes or in an error in the estimation of orthology relationships between two genomes [1]; also the information on the number and composition of bacterial operons may be incomplete or erroneous. Other genomic analysis may even become impossible when dealing with a draft genome, such as structural comparisons between two

genomes; the presence of translocations, inversions, and insertions/deletions may not be determined, especially considering that the repetitive elements—which are usually related to such structural variants—are often missed or underestimated in a draft assembly. Therefore, to ensure a complete and comprehensive genomic analysis, a complete genomic sequence is needed, or at least a reliable scaffold, for which the relative position of each contig is established with sufficient confidence. Given the cost of the finishing process, a reliable scaffolding may be the best sustainable option for many genomics projects.

Luckily, comparative genomics gives solution to the draft genome problem. In fact, for many bacterial species there is at least one complete genome sequence available (about 44 % of the 3,376 available species); this number is even higher when considering that some species are particularly close to each other. Given that strains of the same species usually have a high similarity at the nucleotide level, a valuable approach to resolve the relative orientation of the contigs is to align them to the most similar complete genome available. Once the contigs are mapped, a scaffold can be produced, as well as a series of PCR primer pairs for the finishing process, whose number would be significantly lower than having no mapping information. A number of algorithms have been developed in latest years to facilitate such mapping, each one using different methods to calculate the alignment between the contigs and the reference closed genome, as well as different strategies to resolve ambiguous mappings and produce the final scaffold. Projector2 uses BLAT or BLAST [16]; OSlay uses BLAST or MUMmer [12]; and ABACAS [2], scaffold_builder [14], SIS [6], and fillScaffolds [11] use MUMmer [9]. In this chapter we present the CONTIGuator [7] algorithm and its use through the web interface, as well as the command line tool.

CONTIGuator, similarly to the other mapping algorithms, uses BLAST [3] to align the draft genome contigs to the reference genome. The outputs of the program comprise one scaffold for each reference replicon (chromosome, chromids, or plasmids), as well as a series of maps that highlight the structural differences between the two genomes, both statically (as PDF files) and interactively (viewable using ACT, Artemis Comparison Tool [4]). The program can also output a series of PCR primer pairs for genome finishing, using the ABACAS interface to primer3 [13].

One of the main differences between CONTIGuator and the other mapping algorithm is the ability to unequivocally map the contigs to reference genomes having more than one replicon. In fact in many cases bacterial species can harbor extrachromosomial molecules with a size comparable to the chromosome (i.e., megaplasmids and chromids [8]); the ability to take into account such problem ensures that also this species can be automatically and unequivocally scaffolded. The ability to easily gain insights into the

structural features of the draft genome, either by the graphical maps or the output files, is also a powerful feature that could reduce the need to obtain a complete genome sequence.

1.1 Note on This Chapter

In this chapter we use some type-setting conventions. We use

```
this format
```

in order to refer to command line input or output, but also to refer to external text.

2 Materials

The materials needed to perform the mapping process with CONTIGuator are divided into two groups: those needed by the command line tool and those in common with the web interface version.

2.1 Common Materials

2.1.1 Genomic Sequences

Both the draft and the reference genome should be provided as nucleotide sequences in separate FASTA format. The FASTA format looks as follows:

```
>sequence_ID description
AGTACAGTAGACAGATATCCAGAT
>sequence2_ID description
AAATGGACCACAGTTAGCACAGAT
TTTACAGGACCAGATAC
```

Both the draft and the reference genome FASTA file can contain more than one nucleotide sequence: in general, it is expected that the draft genome contains a higher number of short nucleotide sequences than the closed reference genome. For performance reason, the CONTIGuator web server accepts a maximum file size of 50 MB, which is far above the average size for a bacterial genome FASTA file, which is usually around 5 MB.

2.1.2 Reference Genome PTT Files (Optional)

Additional analysis can be performed on the unaligned portions of the reference genome if one PTT file for each reference nucleotide molecule is provided. Example PTT files (also called protein tables) can be found in the NCBI FTP bacterial genomes folder (i.e., ftp://ftp.ncbi.nlm.nih.gov/genomes/Bacteria/Escherichia_coli_K_12_substr__MG1655_uid57779/NC_000913.ptt) and its format is as follows:

```
Escherichia coli str. K-12 substr. MG1655, complete
   genome. - 1..4641652
4141 proteins
Location    Strand      Length      PID   Gene
   Synonym  Code  COG   Product
190..255    +     21    16127995    thrL  b0001 -
   -  thr operon leader peptide
```

```
337..2799   +     820   16127996    thrA  b0002 -
   COG0527E fused aspartokinase I and homoserine
   dehydrogenase I
2801..3733  +     310   16127997    thrB  b0003 -
   COG0083E homoserine kinase
```

One PTT file for each reference nucleotide sequence should be provided, whose name must follow the notation sequence_ID.ptt, where sequence_ID is the same sequence ID found in the reference genome FASTA file.

2.2 Command Line Tool Materials

The CONTIGuator command line tool can be used in any UNIX-like shell, such as bash, zsh, or cygwin. This chapter assumes that the bash shell is being used in a Linux operating system such as Ubuntu 12.04.

2.2.1 Environment

2.2.2 Software Dependencies

The CONTIGuator software has two kinds of dependencies: mandatory or optional, and are all listed in Table 1. All the dependencies can be installed following the instructions provided in the project website (http://contiguator.sourceforge.net/); however if a package manager is present in the operating system (such as the apt-get command in Ubuntu), most of the dependencies can be installed directly from the package manager.

Table 1
CONTIGuator command line tool dependencies

Name	Source	Description	Version	Optional
CONTIGuator	http://contiguator.sourceforge.net/	Main script	≥2.6	No
Python	http://www.python.org/	Used to run the main script	≥2.6	No
BioPython	http://biopython.org	Used by the main script	≥1.59	No
BLAST+	http://blast.ncbi.nlm.nih.gov	Used to map the contigs	Any	No
Perl	http://www.perl.org/	Used to obtain PCR primers	Any	Yes
ABACAS	http://abacas.sourceforge.net/	Used to obtain PCR primers	≥1.3.1	Yes
MUMmer	http://mummer.sourceforge.net/	Used to obtain PCR primers	≥3	Yes
Primer3	http://primer3.sourceforge.net/	Used to obtain PCR primers	≥2	Yes
ACT	http://www.sanger.ac.uk/resources/software/act/	Used to inspect contig maps	Any	Yes

3 Methods

This section is divided into three parts: first, the web interface workflow is discussed, and then the command line workflow and options are discussed. The final part discusses the content and analysis of the outputs of the program, which are common to both workflows.

3.1 Web Interface Workflow

The CONTIGuator web interface (Fig. 1) is reachable through the project website (http://contiguator.sourceforge.net/) or directly (http://bazzigroup.dbe.unifi.it/contiguator/).

3.1.1 Upload Files and Set Parameters

The web server main page allows the user to provide all the input files (the draft and reference genomes in FASTA format, and optionally the reference PTT files) and to set all the analysis parameters. The two FASTA files are mandatory to start the analysis, as well as an e-mail address, while all the other parameters can be left with their default value. If javascript is enabled in the browser, the input form will check for the correctness of the provided values. For performance reasons, the maximum size of each input file is limited to 50 MB; analysis using files bigger than this threshold is advised to follow the command line workflow.

- Provide the draft genome FASTA file using the file upload form under the "Contig file" label.

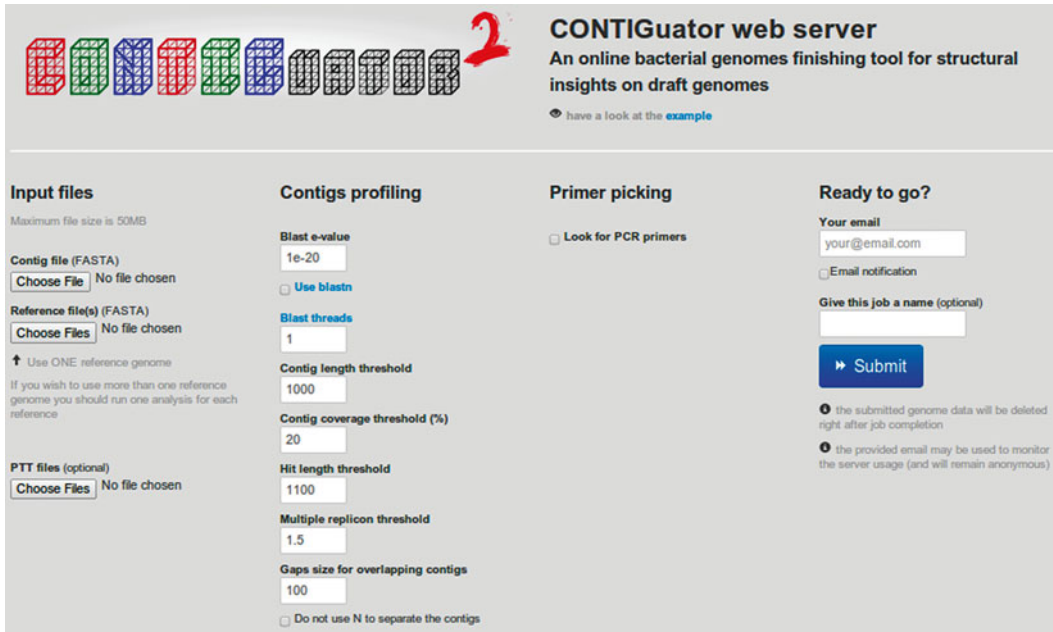

Fig. 1 The CONTIGuator web server main page

- Provide the reference genome FASTA file (or files) using the file upload form under the "Reference file(s)" label.
- Optionally provide one PTT file for each reference genome nucleotide sequence.
- Optionally change the analysis parameters.
- Provide an e-mail address and optionally provide a name for the analysis.

3.1.2 Submit and Wait for the Results

Depending on the size of the input genomes and on the analysis parameters, CONTIGuator may take up to several minutes to complete the analysis, especially if the program has to provide the PCR primers.

- Once all the analysis parameters are set, click on the "Submit" button.
- Wait for the results page to appear or bookmark the waiting page and try to open the results page later.

3.1.3 Inspect the Results Page

An example excerpt of a CONTIGuator result web page is showed in Fig. 2. The result page contains a link to download the whole result files as a tar.gz archive, detailed information about the

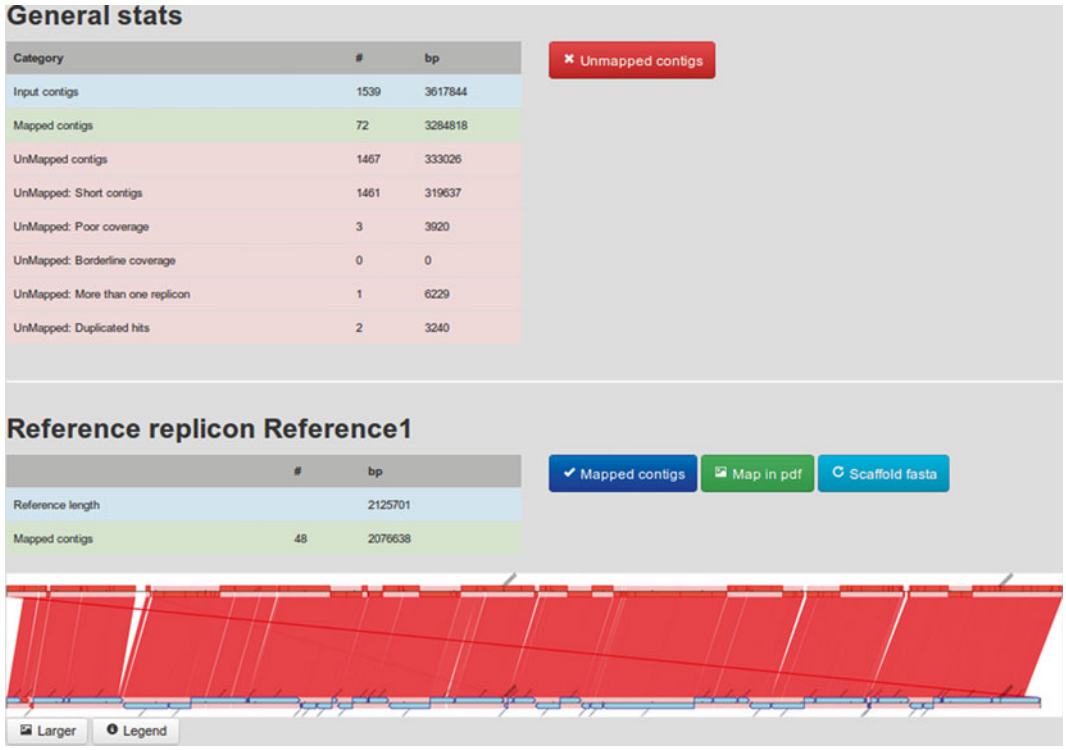

Fig. 2 The CONTIGuator web server results page

analysis parameters, a summary table about the mapped and unmapped contigs (Fig. 2, upper part), and a summary table and detailed alignment for each generated scaffold (Fig. 2, lower part).

- Look at the "General stats" table to inspect the number of draft contigs mapped to the reference genome ("Mapped contigs"), as well as the number of draft contigs that were not mapped ("UnMapped contigs"), further divided by category, which indicates the reason why they have not been mapped to the reference genome (i.e., "UnMapped; poor coverage").
- Click on the "Unmapped contigs" button to show the unmapped contig ID and size.
- For each reference genome nucleotide sequence inspect the summary table to see how many contigs participate in the scaffold.
- Click on the "Mapped contigs" button to show the scaffold contig ID and size.
- Click on the "Scaffold fasta" button to obtain the scaffold nucleotide sequence in FASTA format.
- Inspect the alignment between the reference genome and the draft scaffold to gain structural insights on the draft contigs.
- Larger and more detailed alignments are available by clicking on the "Larger" or "Map in pdf" buttons.
- A detailed legend explaining the alignment graphical features is available by clicking on the "Legend" button.

3.1.4 Compute the PCR Primer Pairs

Optionally CONTIGuator can produce the PCR primer pairs, which can be used for genome finishing. By default only those PCR primer pairs whose PCR product is predicted to span two adjacent contigs in a scaffold are considered.

- From the main web page check the "Look for PCR primers" box.
- Change the PCR primer parameters if needed.
- On the result page, for each scaffold click on the "PCR primers" button to download a summary table on the primers for genome finishing.

Please note that the computation of PCR primers may take a significant amount of time, which depends on the genome size and number of contigs mapped.

3.1.5 Download the Results

A gzipped tar archive containing all the results files is available for download in the CONTIGuator results page. Its content is identical to the command line outputs and it will be discussed in Subheading 3.3.

- Click on the "Download the results" button (on top of the results page).
- Choose a proper location to save the CONTIGuator_results.tar.gz file.

3.2 Command Line Workflow

In this section we assume that a bash shell is used in a Ubuntu Linux operating system, although the same results should be obtained using different shells/operating systems. We also assume that the input files that will be used are the ones provided in the "examples" directory of the CONTIGuator installation archive; in particular the draft genome FASTA file will be called "contigs.fna," while the reference genome FASTA file will be called "references.fna."

3.2.1 Default Run

- Open a shell terminal and move to the directory where the CONTIGuator.py script has been downloaded.
- Type the following and press "enter" to start a CONTIGuator run with default parameters:

```
python CONTIGuator.py -c contigs.fna -r references.fna
```

If all the dependencies have correctly been installed and no errors have been encountered the following messages should appear together with other messages on screen:

```
Input contigs: 1539, 3617844 bp
Mapped contigs: 72, 3284818 bp
UnMapped contigs: 1467, 333026 bp
UnMapped categories:
   Short contigs: 1461, 319637 bp
   Contigs with poor coverage: 3, 3920 bp
   Contigs with nearly good coverage: 0, 0 bp
   Contigs mapped to more than one replicon: 1, 6229 bp
   Contigs discarded due to duplicated hits: 2, 3240 bp
```

which indicates a successful CONTIGuator run; roughly 90 % of the input nucleotide sequences have been mapped to the reference genome, leaving most of the smaller contigs (below 1,000 bp) as unmapped.

3.2.2 Parameter Tuning

The CONTIGuator command line script has many options that allow the user to fine-tune the analysis.

- Type the following in a terminal to get the complete list of available options:

```
python CONTIGuator.py -h
```

Some options are used to determine the number and type of output files, while some others may change the number of contigs mapped to each reference genome nucleotide molecule. In particular:

- Increase the value of the –e option (Blast E-value) to allow the inclusion of poorer quality alignments.
- Use the –b option to use the blastn algorithm instead of megablast, which may be more sensitive when distance between the draft and the complete genome is substantial.
- Decrease the value of the –L option (minimum contig length) to consider also the smaller contigs in the mapping process.
- Decrease the value of the –C option (minimum contig coverage percentage) to also map those contigs with a smaller aligned portion.
- Decrease the value of the –B option (minimum hit size) to consider also the smaller alignments in the mapping process.
- When dealing with reference genomes having more than one nucleotide sequence, lower the value of the –R option (multiple replicon ratio) to reduce the number of unmapped contigs due to unambiguous contig mappings.

To change the output number and type, the user can use the following options:

- Use the –M option to obtain more output files.
- Use the –f option to set a prefix for the output directories, which is useful when running more than one analysis on the same dataset. Using this option, each run will not override existing directories.
- Change the value of the –n option to change the number of N bases that separate each separate contig of the scaffolds.
- Use the –N option to concatenate the contigs in the scaffolds; by default 100 N bases are inserted between each contig in the scaffolds.

3.2.3 Compute the PCR Primer Pairs

Optionally CONTIGuator can produce the PCR primer pairs, which can be used for genome finishing. The ABACAS interface to primer3 is used: by default only those PCR primer pairs whose PCR product is predicted to span two adjacent contigs in a scaffold are considered.

- Add the –P option to let CONTIGuator compute the PCR primer pairs.
- When asked by the program, provide the PCR primer parameters.
- Add the –A option to use the default parameters in PCR primer creation.

Please note that the computation of PCR primers may take a significant amount of time, which depends on the genome size and number of contigs mapped.

3.2.4 Inspect the Alignments

There are two kinds of graphical outputs for the alignments generated by CONTIGuator: interactive maps analyzed using the Artemis Comparison Tool (ACT) or static maps which can be opened with any PDF viewer.

- Open the Artemis Comparison Tool (ACT).
- Press on File → Open.
- Load the Reference.embl file (which can be found in each one of the output folders) in the "Sequence file 1" field.
- Load the Pseudocrunch.embl file (which can be found in each one of the output folders) in the "Comparison file 1" field.
- Load the PseudoContig.embl file (which can be found in each one of the output folders) in the "Sequence file 2" field.
- Press "Apply," and then ignore the warnings to open the interactive map.

An example of the alignment map interactive representation is shown in Fig. 3. The same color codes of the static maps (which are also shown in the web interface version) are used, and can be inspected by visiting this web page (http://bazzigroup.dbe.unifi.it/contiguator/legend.html).

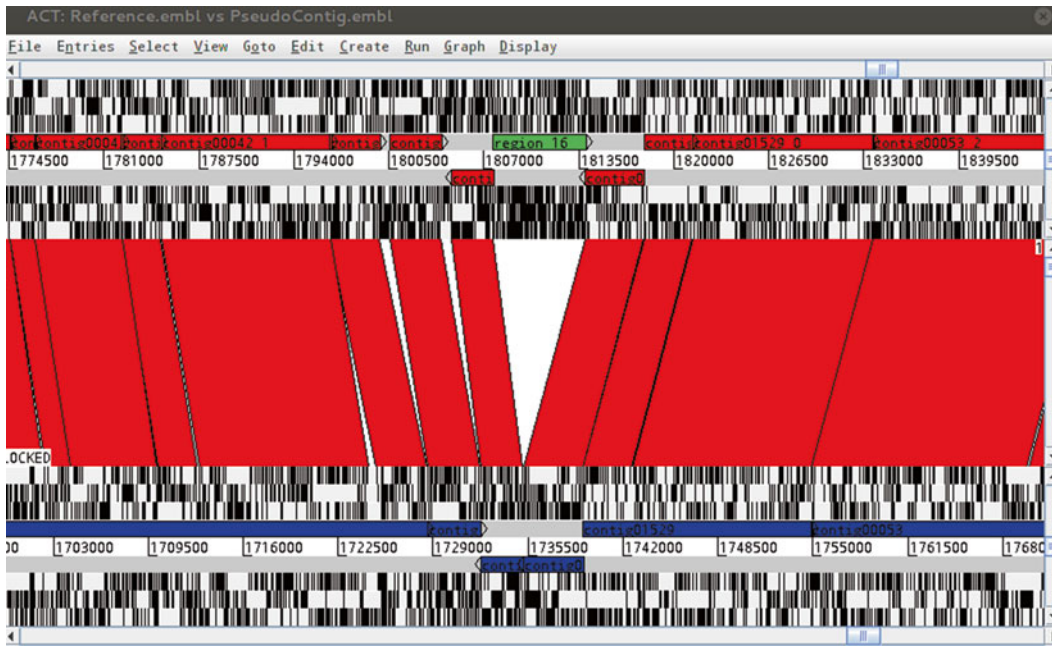

Fig. 3 The ACT interactive alignment map

If ACT is already present in the computer when running the analysis, CONTIGuator will try to produce a shell script that can be used to automatically open the interactive maps; when the attempt succeeds, a similar output should be seen in the terminal:

```
16:46:33 Will try to prepare the ACT launchers…
16:46:33 Searching the ACT executable in your system
16:46:35 ACT binary:/opt/artemis/act
16:46:35 Writing the ACT launcher scripts
16:46:35 Reference1 launcher: Map_Reference1/Reference1.sh
16:46:35 Reference2 launcher: Map_Reference2/Reference2.sh
```

To open the first interactive map simply type the following command in a terminal:

- `sh Map_Reference1/Reference1.sh`
- Add the –l option to the CONTIGuator command to automatically open the interactive maps at each run.

3.3 Output Files

Both the web interface and the command line workflow produce the same output files, which are discussed in this section, divided by folder.

3.3.1 Map Folders

According to the number of nucleotide sequences present in the reference genome, there will be the same number of directories whose name starts by "Map_," followed by the ID of the reference sequences. The following files will be found inside:

- Reference.embl: Reference genome nucleotide sequence in EMBL format, which contains the position of the aligned regions.
- PseudoContig.fsa: Scaffold nucleotide sequence in FASTA format.
- PseudoContig.crunch: ACT comparison file between the reference genome and the scaffold.
- PseudoContig.embl: Scaffold nucleotide sequence in EMBL format, which contains the position of the mapped contigs and their alignments with the reference molecule.
- MappedContigs.txt: Names (and lengths) of the contigs mapped to the reference molecule.
- A shell script to open the ACT map.
- A PDF containing a publication-quality alignment map.

If option -M was selected in the command line workflow the following files will also be present (in the web interface workflow they will always be produced):

- AlignDetails.tab: Tab-delimited file containing details about the alignment position in the reference molecule and on the contigs.
- AlignedContigsHits.fsa: Mapped hit nucleotide sequences in FASTA format (on contigs).
- AlignedReferenceHits.fsa: Mapped hit nucleotide sequences in FASTA format (on the reference molecule).
- UnAlignedContigsHits.fsa: Unmapped region nucleotide sequences in FASTA format (on contigs).
- UnAlignedReferenceHits.fsa: Unmapped region nucleotide sequences in FASTA format (on reference).

If the primer picking option was selected (-P) the folder will contain another file:

- PCRPrimers.tsv: Table containing details about the PCR primers generated by the program.

3.3.2 Unmapped Folders

The "UnMappedContigs" folder contains information on those contigs that were not mapped to the reference genome and therefore are not present in any scaffold.

- Excluded.fsa: All the unmapped contig nucleotide sequences in FASTA format.
- Multi.fsa: Contig nucleotide sequences mapped to more than one reference molecule in FASTA format.
- Short.fsa: Nucleotide sequence of those contigs below the length threshold in FASTA format.
- NoCoverage.fsa: Nucleotide sequence of those contigs below the coverage threshold in FASTA format.
- CoverageBorderLine.fsa: Nucleotide sequence of those contigs near the coverage threshold in FASTA format.
- Discarded.fsa: Nucleotide sequence of those contigs discarded due to duplicated hits in FASTA format.
- UnMappedContigsHits.tab: Contains the list of the excluded contigs with the number of blastn hits, if the PTT files have been provided.
- UnMappedReferenceRegions.tab: Contains the reference genome unmapped regions with at least one blastn hit, if the PTT files have been provided.
- UnMappedContigs.txt: Names (and lengths) of the contigs not mapped to any reference genome molecule.

References

1. Angiuoli S, Hotopp JD, Salzberg S, Tettelin H (2011) Improving pan-genome annotation using whole genome multiple alignment. BMC Bioinformatics 12:272
2. Assefa S, Keane TM, Otto TD, Newbold C, Berriman M (2009) ABACAS: algorithm-based automatic contiguation of assembled sequences. Bioinformatics 25:1968–1969
3. Camacho C, Coulouris G, Avagyan V et al (2009) BLAST+: architecture and applications. BMC Bioinformatics 10:421
4. Carver TJ, Rutherford KM, Berriman M et al (2005) ACT: the Artemis comparison tool. Bioinformatics 21:3422–3423
5. Compeau PE, Pevzner PA, Tesler G (2011) How to apply de Bruijn graphs to genome assembly. Nat Biotechnol 29:987–991
6. Dias Z, Dias U, Setubal JC (2012) SIS: a program to generate draft genome sequence scaffolds for prokaryotes. BMC Bioinformatics 13:96
7. Galardini M, Biondi EG, Bazzicalupo M, Mengoni A (2011) CONTIGuator: a bacterial genomes finishing tool for structural insights on draft genomes. Source Code Biol Med 6:1
8. Harrison PW, Lower RP, Kim NK, Young JPW (2010) Introducing the bacterial "chromid": not a chromosome, not a plasmid. Trends Microbiol 18:141–148
9. Kurtz S, Phillippy A, Delcher AL et al (2004) Versatile and open software for comparing large genomes. Genome Biol 5:R12
10. Medini D, Donati C, Tettelin H, Masignani V, Rappuoli R (2005) The microbial pan-genome. Curr Opin Genet Dev 15:589–594
11. Muñoz A, Zheng C, Zhu Q et al (2010) Scaffold filling, contig fusion and comparative gene order inference. BMC Bioinformatics 11:304
12. Richter DC, Schuster SC, Huson DH (2007) OSLay: optimal syntenic layout of unfinished assemblies. Bioinformatics 23:1573–1579
13. Rozen S, & Skaletsky H. (1999). Primer3 on the WWW for general users and for biologist programmers. In Bioinformatics methods and protocols (pp. 365–386). Humana Press
14. Silva GG, Dutilh BE, Matthews TD et al (2013) Combining de novo and reference-guided assembly with scaffold_builder. Source Code Biol Med 8:23
15. Tettelin H, Riley D, Cattuto C, Medini D (2008) Comparative genomics: the bacterial pan-genome. Curr Opin Microbiol 11:472–477
16. van Hijum SA, Zomer AL, Kuipers OP, Kok J (2005) Projector 2: contig mapping for efficient gap-closure of prokaryotic genome sequence assemblies. Nucleic Acids Res 33: W560–W566

Chapter 12

Gene Calling and Bacterial Genome Annotation with BG7

Raquel Tobes, Pablo Pareja-Tobes, Marina Manrique, Eduardo Pareja-Tobes, Evdokim Kovach, Alexey Alekhin, and Eduardo Pareja

Abstract

New massive sequencing technologies are providing many bacterial genome sequences from diverse taxa but a refined annotation of these genomes is crucial for obtaining scientific findings and new knowledge. Thus, bacterial genome annotation has emerged as a key point to investigate in bacteria. Any efficient tool designed specifically to annotate bacterial genomes sequenced with massively parallel technologies has to consider the specific features of bacterial genomes (absence of introns and scarcity of nonprotein-coding sequence) and of next-generation sequencing (NGS) technologies (presence of errors and not perfectly assembled genomes). These features make it convenient to focus on coding regions and, hence, on protein sequences that are the elements directly related with biological functions.

In this chapter we describe how to annotate bacterial genomes with BG7, an open-source tool based on a protein-centered gene calling/annotation paradigm. BG7 is specifically designed for the annotation of bacterial genomes sequenced with NGS. This tool is sequence error tolerant maintaining their capabilities for the annotation of highly fragmented genomes or for annotating mixed sequences coming from several genomes (as those obtained through metagenomics samples). BG7 has been designed with scalability as a requirement, with a computing infrastructure completely based on cloud computing (Amazon Web Services).

Key words Bacterial genomics, Genome annotation, Gene calling, Next-generation sequencing, Cloud computing, Metagenomics, Functional annotation, Gene prediction, Biographika, Massive parallel sequencing

1 Introduction

With the availability of new massive sequencing technologies, genome annotation becomes a crucial need in order to reach new findings within the amazing world of bacteria. Annotation is the basic central step in any sequence analysis pipeline, linking raw data and biological knowledge; it is the foundation on which further analysis builds upon.

Any efficient tool designed specifically to annotate bacterial genomes sequenced with massively parallel technologies has to

Alessio Mengoni et al. (eds.), *Bacterial Pangenomics: Methods and Protocols*, Methods in Molecular Biology, vol. 1231, DOI 10.1007/978-1-4939-1720-4_12, © Springer Science+Business Media New York 2015

consider the specific features, first, of bacterial genomes and, second, of next-generation sequencing (NGS) technologies. The absence of introns and the scarcity of nonprotein-coding space in bacterial genomes are critical differences with eukaryotic ones. These features make it convenient to focus on coding regions and, hence, on protein sequences that are the elements directly related with biological functions. The extreme (even intraspecies) diversity of bacteria makes also preferable to study bacterial genomes not using the "model organism" paradigm, but more flexible approaches that fit better with the plasticity, evolutionary changes, and gene flux that bacterial genomes possess.

The fast-evolving massive sequencing technologies also demand flexible algorithms, able to work with different technology-dependent error patterns and highly preliminary, heavily fragmented draft genome assemblies. Another challenge to face is the annotation of contigs coming from metagenomics samples of different non-culturable bacteria.

Classical gene prediction algorithms are based on statistical features of gene and non-gene sequences and on some specific signals and patterns present in the sequences flanking genic regions [1–3]. Many of them need a previous training phase with known annotated genes that are not always available as in the really interesting case of genomes very distant from known ones.

In contrast with methods that separate ORF prediction from their annotation [4–12], BG7 [13] predicts genes and infers their function mainly based on protein similarity, integrating ORF prediction and functional annotation in a single process. If the gene function is assigned based on protein sequence similarity, why not to predict genes based on this very same sequence similarity? Among other advantages, this provides more directly traceable gene annotations, since the similarity with a specific known gene product is responsible for both the gene prediction and the function assignment. Thus, the system is more flexible, tunable, and traceable and the sequence errors can be easily managed even if the pattern of errors changes with the introduction of a new sequencing technology.

In this protein-centered gene calling/annotation paradigm the set of reference proteins is the most determinant element when setting the annotation process; having an appropriate set of reference proteins is perfectly affordable for a biologist working in bacteria. BG7 predicts genes not only based on similarity but also analyzing sequence signals as stop and start codons and joining fragments of similarity that probably correspond to the same gene. Similar reference sequences compete for annotating a region of the bacterial genome and finally the best predicted and annotated genes are selected. The problem of small contigs, frequent in NGS assemblies, is also solved with BG7 that is able to detect fragmented genes or genes only partially sequenced. Given that BG7 carries out the functional annotation in the same step as the gene prediction,

the system is tolerant to the lack or gain of start/stop signals and able to annotate fragments of genes.

In this chapter we describe how to annotate a bacterial genome with BG7. It has been tested with data from most of the NGS technologies currently available (454, Illumina, IonTorrent, and PacBio), assembled with different assembly tools, yielding to high-quality annotations in all these cases. Due to how it is designed, BG7 is tolerant to the most frequent NGS errors like errors in homopolymeric regions or any other type of insertions, deletions, or substitutions.

BG7 has been designed with scalability as a requirement, with a computing infrastructure completely based on cloud computing (Amazon Web Services). It is a perfect fit for big annotation projects involving hundreds and thousands of genomes: BG7 makes possible to obtain their annotations in a time independent of the number of genomes, by adjusting the number of provisioned resources accordingly.

1.1 BG7 Algorithm

BG7 is designed from the ground up so as to deal with the special characteristics of both NGS data and bacterial genomes.

1.1.1 Selection of UniProt Reference Proteins and Reference RNAs

This selection has to be objective driven. Two common objectives are the prediction of all the genes and the assignment of their function as accurately as possible; however, in many cases the annotation is essentially focused on specific types of functionalities, as could be antibiotic resistance, enzymatic activities, metabolic pathways, or plasmidic genes. This can be accomplished through the selection of reference protein sets matching those needs.

1.1.2 Search of Similarities Between Contigs and Reference Proteins to Predict and Annotate the Coding Regions

This is carried out doing a tBLASTn search of the reference proteins against the contig sequences. As a result of this BLAST search we will have lots of BLAST hits of the proteins with the contigs, some of them with possibly lots of aligned fragments (HSPs: high-scoring segment pairs) of the reference proteins with the contigs.

1.1.3 Gene Prediction

First we need to define a single similarity region between the protein and the contig, by merging all the coherent HSPs from a hit. Then we look upstream and downstream for start and stop signals, and define preliminary genes accordingly. These just defined genes could suffer from a series of deficiencies: noncanonical start/stop codons, intragenic stop codons, and/or frameshifts. We check for all these possibilities, and mark the corresponding noncanonical genes with their deficiencies. This is one of the main reasons why this system is so robust to NGS sequencing errors since it is able to tolerate all the types of indels and substitutions covering the local errors common in each sequencing technology. Noncanonical stop or start codon, intragenic stops, and frameshifts are indicated in the annotation of each gene.

1.1.4 Selecting the Best Gene for Each Contig Region

At this point we have lots of preliminary genes predicted for each contig region; we thus need to select the best gene for each of them, solving overlapping conflicts between predicted genes. Each gene is predicted by similarity to one protein and logically the best gene for each genome region is that with higher similarity value in the alignment of the protein and the contig region. The rest of predicted genes are marked as dismissed genes.

1.1.5 RNA Prediction

The search for RNA genes is done in a very similar way, using BLASTn to face the reference RNA sequences against the contig sequences. At the final integration step, predicted RNA genes are always preferential over protein-coding genes.

2 Materials

Here we describe the inputs that you need for running BG7 (summarized in Table 1). In Subheading 3 we explain in detail how you could obtain them.

2.1 Genome Sequences to be Annotated

A FASTA file (*see* **Notes 1** and **2** for tips on how this file should be) containing a set of contigs comprising the (pan)genome you want to annotate.

2.2 Reference Proteins

A text file (*see* **Note 1**) containing a list of UniProt identifiers, one per line, (*see* **Note 3**) corresponding to the set of proteins that will be used as reference proteins for gene prediction and annotation.

Step 3 in Subheading 3 is dedicated to how you should choose your reference proteins, and how to obtain the corresponding file in this format.

2.3 Reference RNAs

A FASTA file (*see* **Note 2**) containing the sequence of RNAs that will be used as a reference RNA: *See* **step 4** in Subheading 3 for details about how to obtain them.

Table 1
Input files needed for the execution of BG7

Input files for BG7	
File name	**Content**
EC1_genome_contigs.fna	Sequences of DNA to be annotated in FASTA format
Reference_protein_ID_list.txt	List of reference protein UniProt IDs in text format
Reference_RNAs.FASTA	Reference RNA sequences in FASTA format
Configuration.scala	Genome metadata and values for BG7 parameters
AWS_keys.txt	Text file with AWS keys

2.4 Metadata for Generating GenBank and EMBL Format Files

Metadata of the genome you want to annotate like the species name, the complete taxonomic lineage, or a brief description of the sampling and sequenced genome/s. *See* **step 5** in Subheading 3.

2.5 AWS Keys

A text file (*see* **Note 1**) containing access keys (*see* **Note 5**) for an *AWS* (*A*mazon *W*eb *S*ervices) account (*see* **Notes 4** and **6**).

The user will need to create a new AWS account (if he or she does not have one); instructions for this are in Subheading 3, **step 1**.

3 Methods

3.1 Set Up the Environment

Before running the first annotation the user has to set up the environment. This should be done only once in each machine the user wants to use to run the annotations. The only requirements for running BG7 are a Java Virtual Machine, the Scala simple build tool (sbt), and the BG7 command line interface; up-to-date instructions for their installation can be found at the BG7 website: http://bg7.ohnosequences.com.

3.2 Create AWS Credentials

If the user does not have an AWS account (*see* **Note 6**) he or she needs to register there first; go to aws.amazon.com, click on "sign-up," and follow the instructions.

BG7 will create and manage all AWS resources automatically, but for that a set of valid keys with the right permissions are needed. The easiest and safest way to obtain them is by creating an IAM user (*see* **Note 8**) through the Amazon Web Services web console (*see* **Note 7**). The user is given the opportunity to download the aforementioned credentials only once, just after creating the IAM user (*see* **Note 8**).

3.3 Get the Sequences to be Annotated

BG7 works with genome assemblies, even still in draft status, or with any type of DNA sequences in FASTA format with a minimal length (over around 500 bp). In a typical bacterial genome project the user must assemble the genome before the annotation. There are many methods for obtaining genome assemblies from sequencing data (see the corresponding chapter in this book), but a thorough description of them would be, however, out of the scope of this chapter.

3.4 Select the Reference Protein Dataset

The user must provide a list of UniProt accession numbers (*see* **Note 3**) of the proteins that need to be used as reference proteins. The set of proteins can be composed using the UniProt search tools at the UniProt website (http://www.uniprot.org). The list of UniProt IDs can be obtained from the UniProt website in an easy way: once the user has the set of proteins that need to be used as reference just click on the "Download" button on the right and then click on the "List" option to obtain a text file with the

UniProt accession numbers of the reference proteins in the required text format; *see* **Note 9** for some guidelines on how you could choose this set of proteins.

It is possible to focus the annotation on a particular biological process of interest; *see* **Note 10**.

Internally, BG7 will use Biographika (*see* **Note 11**) to actually retrieve the UniProt protein sequences and their associated functional information; *see* **Note 12**. It is also possible to obtain reference proteins directly from Biographika; *see* **Note 13**.

3.5 Select the Reference RNA Dataset

The reference RNAs must be provided in a FASTA file with the headers format as NCBI provides them through .frn files. *See* **Note 14** for a possible selection strategy.

3.6 Create the Annotation Project

The next step is to create the annotation project. This is done very easily using the BG7 command line interface tool, just typing the following command:

- bg7 create

At this point the user will be asked some questions like a project name and an e-mail address for notifications. This command will create locally a folder called like the project name given by the user.

3.7 Fill Metadata for Your Annotation and Set the Parameters in the Configuration File

The next step is to fill the metadata for your annotation and set the parameters in the configuration file.

Genome metadata is provided in the configuration file called *configuration.scala* (*see* **Note 15**). Before running the annotation the user must edit the file and change the default values for these fields:

- Locus tag prefix; *see* **Note 16**.
- Organism.
- Complete taxonomic lineage; *see* **Note 17**.
- Genome definition.

Some BG7 parameters can be set in the configuration file *configuration.scala* (*see* **Note 15**), like the following:

- The maximum distance to search for start and stop codons at the ends of the preliminary gene regions predicted by one HSP or several merged HSPs.
- The maximum length allowed for gene overlapping.
- The maximum *dif-span* value allowed for merging two HSPs of the same BLAST hit. *dif-span* is the difference between the distance between two HSPs in the reference protein and the distance of the corresponding aligned fragments in the contigs. *dif-span* is evaluated for joining different HSPs to construct a gene with coherent fragments that probably belong to the same gene.

Setting these parameters is optional. All of them are provided with default values that have been proved to be appropriate for most scenarios.

3.8 Check Your Input Data

This step is not mandatory either, but we recommend the user to check the input data (*see* Table 1). The user should check he or she has:

- The file with the *AWS keys* as in **step 1**.
- The genome sequences to be annotated as detailed in **step 2**.
- The text file with the list of UniProt accession numbers for the *reference proteins*, from **step 3**.
- The FASTA file with the *reference RNAs* obtained in **step 4**.
- The configuration.scala file with the *metadata* and the correct values for the *parameters* (*see* **step 6**).

3.9 Launch the Annotation

For launching an annotation the user has just to follow these two steps:

1. Publishing the annotation project.
2. Running the annotation.
   For publishing the annotation project (*see* **Note 18**):
   - bg7 publish

   and for running the annotation:
   - bg7 run

   The user receives notifications and updates about the progress via e-mail. About the running time, *see* **Note 19**. BG7 execution costs depend on the time and type of AWS resources used; *see* **Note 20**.

3.10 Download the Output

Once the annotation is finished the user can download the output files (*see* Table 2) in two different ways:

- Using the Amazon console (*see* **Note 7**): the output files (*see* Table 2) are stored in an S3 bucket.
- Clicking on the link provided in the notification mail that is sent once the annotation is finished *see* **Note 21**.

4 Notes

1. Incorrect text file encoding can result in erroneous results and unexpected BG7 behavior. Make sure that all your text input files (*see* Table 1) are in UTF-8 without BOM. If you are using Windows as your operating system you can check this (and correct it if it is needed) with a good text editor such as Notepad++.

Table 2
Main output files with BG7 annotation results

Main BG7 output files	
File name	**Content**
EC1_sequences_header_fixed.fna	Annotated DNA sequences in FASTA format with corrected headers
EC1_protein_nucleotide_sequences.FASTA	Nucleotide sequences of predicted genes in FASTA format
EC1_protein_aminoacid_sequences.FASTA	Protein sequences of predicted genes in FASTA format
EC1_Intergenic.FASTA	Sequences of intergenic spaces in FASTA format
EC1_Annotation.gff	Annotation in gff format
EC1_Annotation.tsv	Annotation in tsv format compatible with Excel
EC1_Annotation.embl	Annotation in embl format
EC1_all.gbk (in GenBank folder)	Annotation in GenBank format for all the contigs
EC100000X.gbk (in GenBank folder)	Annotation in GenBank independent for each contig

2. FASTA files are just text files representing a set of sequences in a specific format described in http://www.ncbi.nlm.nih.gov/BLAST/blastcgihelp.shtml. This is an example of how the FASTA format looks like:

   >sequence id1231

   TACGAGGTAGATGCGAGTGCGAGAGGGGGCTGAGC GAGTGCGAGTGAGC

   TCGACCCGATCCCGTGAGGATGGGCGAGGAAAGT GAGAAAGCGTGTGTT

   TAAACTTACGCAGAAAATTTAA

   >sequence id2167

   TACGAGGTAGATGCAAGAGTGCGTTAGAGGGTTC ATCCTGCGAGTGAGCC

   TCGACCTGCGAGAGGGGAGGATGGGCGAGGAAAG TGAGCATCCCTGTGTT

   TCCGGC

3. The format of the file containing the protein IDs is as follows:

   P62552

   P62554

   P04737

   P03012

P14565

P10026

P08716

It is important to note that for the reference proteins BG7 works with the so-called UniProt primary accession number. The user should refer to the UniProt user manual site for more information about the accession number.

4. AWS, standing for *A*mazon *W*eb *S*ervices, is the biggest de facto standard cloud computing provider. BG7 uses the following services:
   - EC2 for providing the compute infrastructure.
   - S3 as a storage service for input and output data.
   - SQS for scheduling computations and in general for communication between components.
   - DynamoDB for managing the state of the different components.
5. Access keys are a pair of strings, the *access key ID* and *secret access key*, which are used to sign programmatic requests made to AWS. BG7 will use these keys to create a set of resources on your behalf, needed for executing the annotation process. The input file with the keys to be provided to BG7 looks like this:

   accessKey=DKIZI23W4SKMA4C7FL4A

   secretKey=Iq2F5xHV8aqTnEgS8bVcOzZSW3ZDcc3Wd1RzvlG
6. It is important not to confuse amazon.com accounts with AWS accounts. They are different entities; if in doubt follow the instructions in **step 1** and create a new AWS account.
7. The user can manage AWS services and resources through a graphical interface, the "Amazon Web Services Web Console," available at https://aws.amazon.com/console.
8. IAM, part of the AWS offer, is a service providing user and access control facilities to the rest of AWS services. The user can access it through the web console (*see* **Note 7**). When creating an IAM user through the web console, the user can grant him full administrative access if he or she does not want to deal with the complexity of fine-grained permissions. The user can copy the user AWS credentials or download them only once, just after creating them; however, the user can regenerate AWS credentials as many times as needed.
9. For a 5 Mb bacterial genome we recommend using around 200,000 proteins as reference proteins. We recommend including all the proteins from close species as well as additional proteins from more distant taxa involved in processes of interest

for the user, i.e., proteins involved in host interactions, in a particular metabolic pathway or plasmidic proteins.

A good strategy in some cases is to select representative proteins from UniRef100 or UniRef90. It allows covering a higher diversity of proteins and taxa maintaining a manageable number of reference proteins. The selection of UniRef100 representative proteins in the case of species with many available genomes causes a reduction in the protein number needed to cover one species of one order of magnitude, maintaining the same number of different sequences (all the proteins included in a UniRef100 cluster shared a sequence 100 % identical to the representative ones). This is the case, i.e., for *Escherichia coli* genomes. Using UniRef90 cluster representative proteins you can cover more taxa with the same number of proteins since each cluster groups all the proteins with 90 % of identity to the representative ones. If you want to select UniRef protein IDs the only modification that you have to do is to remove the prefix UniRef100 or UniRef90 to compose the definitive list of UniProt reference protein IDs for BG7 input.

10. It is possible to focus the annotation on a particular biological process, pathway, or any specific aspect of interest selecting the reference proteins in a proper way. For example, if the user is especially interested in the proteins involved in antibiotic resistance but he or she also wants to annotate the rest of proteins of the genome, he or she should simply *add* a set of specifically selected antibiotic resistance UniProt proteins to the set of reference proteins. Another possibility is that the user wants to annotate *only* the proteins related to antibiotic resistance. In that case he or she should include *only* resistance-related proteins in the set of reference proteins.
11. Biographika (www.biographika.com) is a high-performance biological data platform integrating most data available in UniProt KB (SwissProt + TrEMBL), Gene Ontology (GO), UniRef (50,90,100), RefSeq, NCBI taxonomy, and Expasy Enzyme DB (Pablo Pareja-Tobes et al. Manuscript in preparation). The graph data model is directly deployable to AWS. BG7 uses Biographika to access all data linked with proteins such as their sequence, and functional data (Gene Ontology annotations, keywords, enzymatic activity, etc.).
12. Internally BG7 uses Biographika for accessing the proteins defined by the UniProt identifiers provided as part of the input. Those input identifiers that correspond to proteins that are not included in Biographika will be discarded. Given that Biographika includes all the UniProt proteins and that it is updated very frequently if the user obtains the list of UniProt IDs for the reference protein set from the UniProt website probably no one protein will be dismissed.

13. It is possible to select reference proteins directly through Biographika in a programmatic way; this involves coding, but it can be a great option when the reference sets need to be extracted using complex consults to Biographika database. Graph databases offer new capabilities for complex querying and consulting.
14. We recommend retrieving the FASTA files of the reference RNAs from the NCBI FTP site. The FASTA files containing the RNA information are those with the extension .frn. This is the format required for the headers of the reference RNAs:

    >ref|NC_009925|:29248-29320|Arg tRNA| [locus_tag= AM1_0026

    It is possible to use any RNA sequences as reference if the file is in FASTA format and the header format is compatible with this NCBI format. Normally the RNAs from one close genome are enough for a proper annotation of the main RNAs of a genome.
15. The file configuration.scala is Scala code, a hybrid functional-object-oriented programming language with Java interoperability: Writing the equivalent of configuration files and parameters as code can look a bit strange at first, but it has a key set of advantages:
    (a) The configuration is thoroughly checked before launching anything, drastically reducing the amount of run-time errors. This is particularly important here, where BG7 will be instantiating tens of machines and millions of tasks in the course of the annotation process.
    (b) It makes much easier to run annotations programmatically, as the configuration the user needs to provide can be expressed directly as code.
16. The locus tag prefix should be a combination of letters and numbers no longer than four characters to be used as unique prefix to identify the contigs of the genome/s under analysis. EC1 could be an example of a proper locus tag prefix. Each unique contig ID will be composed by this prefix and by a number. EC1000001 would be an example of locus tag ID for a contig from a genome with a locus tag prefix EC1.
17. The complete taxonomic lineage for a given organism can be obtained pretty easily at the NCBI Taxonomy website just searching for the organism in the text search field and then clicking on the corresponding entry in the results. For example the complete taxonomic lineage of the organism *Escherichia coli* O17 str. K12a would be *cellular organisms; Bacteria; Proteobacteria; Gammaproteobacteria; Enterobacteriales; Enterobacteriaceae; Escherichia; Escherichia coli;* and *Escherichia coli* O17 obtained from its entry at the NCBI Taxonomy site.

18. Publishing the annotation project *does not* mean that the project data is public. It just means that the code and the files are accessible to all AWS resources that would perform the annotation but it does not mean that these files are public in any way.
19. The real BG7 running time depends on many factors but mainly on the number and type of machine/s launched; tBLASTn of the reference proteins against the contigs is usually the most time-consuming process. This BLAST computational time is directly dependent on the reference protein number and on the similarity of the proteins with the genome sequences. The total size of the genome sequences to be annotated also contributes to the computational cost, but at a minor level since the total size of reference sequences usually is much bigger than the total contig size. It is thus impossible to give a precise estimate for the running time of one BG7 annotation; experience shows though that a normal project is finished in less than 1 h. You can find some estimates of running time in specific conditions in the BG7 website.
20. Each BG7 execution incurs in some costs due to the use of AWS resources. Before launching your first BG7 annotation the user needs to consult the prices of each type of machine at the AWS site to design his or her project. Some figures about BG7 execution costs for specific genome annotation examples will be available through the BG7 website.
21. Once the annotation is finished the user receives a notification by e-mail with a temporary link to download the output files (*see* Table 2). It is important to note that this is a temporary link that will be accessible for a short period of time.

Acknowledgements

This work has been partially funded by the CDTI project NEXTMICRO (grant IDI-20120242). A.A. and E.K. are funded by the INTERCROSSING (Grant agreement no.: 289974) ITN European project.

Competing Interests

Era7 offers service of bacterial annotation based on BG7, but BG7 code is available at GitHub, https://github.com/bg7/, under the license AGPLv3. All authors work at the research group named Oh no sequences! within Era7 Bioinformatics company.

References

1. Salzberg SL, Delcher AL, Kasif S, White S (1998) Microbial gene identification using interpolated Markov models. Nucleic Acids Res 26(2):544–548
2. Besemer J, Lomsadze A, Borodovsky M (2001) GeneMarkS: a self-training method for prediction of gene starts in microbial genomes. Implications for finding sequence motifs in regulatory regions. Nucleic Acids Res 29(12): 2607–2618
3. Hyatt D, Chen GL, Locascio PF, Land ML, Larimer FW, Hauser LJ (2010) Prodigal: prokaryotic gene recognition and translation initiation site identification. BMC Bioinformatics 11:119
4. Aziz RK, Bartels D, Best AA, DeJongh M, Disz T, Edwards RA, Formsma K, Gerdes S, Glass EM, Kubal M, Meyer F, Olsen GJ, Olson R, Osterman AL, Overbeek RA, McNeil LK, Paarmann D, Paczian T, Parrello B, Pusch GD, Reich C, Stevens R, Vassieva O, Vonstein V, Wilke A, Zagnitko O (2008) The RAST server: rapid annotations using subsystems technology. BMC Genomics 9(1):75
5. Mavromatis K, Ivanova NN, Chen IA, Szeto E, Markowitz VM, Kyrpides NC (2009) The DOE-JGI standard operating procedure for the annotations of microbial genomes. Stand Genomic Sci 1(1):63–67
6. Borodovsky M, Mills R, Besemer J, Lomsadze A (2003) Prokaryotic gene prediction using GeneMark and GeneMark.Hmm. In: Andreas D. Baxevanis et al. (eds) Current protocols in bioinformatics (Chapter 4 (7), Unit 4.5)
7. Stewart AC, Osborne B, Read TD (2009) DIYA: a bacterial annotation pipeline for any genomics lab. Bioinformatics 25(7):962–963
8. Kumar K, Desai V, Cheng L, Khitrov M, Grover D, Satya RV, Yu C, Zavaljevski N, Reifman J (2011) AGeS: a software system for microbial genome sequence annotation. PloS One 6(3):e17469
9. Tanenbaum DM, Goll J, Murphy S, Kumar P, Zafar N, Thiagarajan M, Madupu R, Davidsen T, Kagan L, Kravitz S, Rusch DB, Yooseph S (2010) The JCVI standard operating procedure for annotating prokaryotic metagenomic shotgun sequencing data. Stand Genomic Sci 2(2):229–237
10. Hemmerich C, Buechlein A, Podicheti R, Revanna KV, Dong Q (2010) An Ergatis-based prokaryotic genome annotation web server. Bioinformatics 26(8):1122–1124
11. Van Domselaar GH, Stothard P, Shrivastava S, Cruz JA, Guo A, Dong X, Lu P, Szafron D, Greiner R, Wishart DS (2005) BASys: a web server for automated bacterial genome annotation. Nucleic Acids Res 33(Web Server issue):W455–W459
12. Lee D, Seo H, Park C, Park K (2009) WeGAS: a web-based microbial genome annotation system. Biosci Biotechnol Biochem 73(1): 213–216
13. Pareja-Tobes P, Manrique M, Pareja-Tobes E, Pareja E, Tobes R (2012) BG7: a new approach for bacterial genome annotation designed for next generation sequencing data. PLoS One 7(11):e49239

Chapter 13

Defining Orthologs and Pangenome Size Metrics

Emanuele Bosi, Renato Fani, and Marco Fondi

Abstract

Since the advent of ultra-massive sequencing techniques, the consequent drop-off in both price and time required made feasible the sequencing of increasingly more genomes from microbes belonging to the same taxonomic unit. Eventually, this led to the concept of *pangenome*, that is, the entire set of genes present in a group of representatives of the same genus/species, which, in turn, can be divided into *core genome*, defined as the set of those genes present in all the genomes under study, and a *dispensable genome*, the set of genes possessed only by one or a subset of organism.

When analyzing a pangenome, an interesting point is to measure its size, thus estimating the gene repertoire of a given taxonomic group. This is usually performed counting the novel genes added to the overall pangenome when new genomes are sequenced and annotated. A pangenome can be also classified as *open* or *close:* in an open pangenome its size increases indefinitely when adding new genomes; thus sequencing additional strains will likely yield novel genes. Conversely, in a close pangenome, adding new genomes will not lead to the discovery of new coding capabilities.

A central point in pangenomics is the definition of *homology relationships* between genes belonging to different genomes. This may turn into the search of those genes with similar sequences between different organisms (and including both *paralogous* and *orthologous genes*).

In this chapter, methods for finding groups of orthologs between genomes and for estimating the pangenome size are discussed. Also, working codes to address these tasks are provided.

Key words Bacterial genomics, Comparative genomics, Pangenome, Next-generation sequencing, Gene homology, Core genome, Pangenome size, Gene prediction, Ortholog finding

1 Introduction

The advent of parallel massive sequencing technologies has led to a great reduction of the experimental and economical efforts required for sequencing a genome. Indeed, the sequencing of genomes from multiple strains for each species has become ordinary [1–4].

The availability of hundreds of genomic sequences allowed comparative analyses of multiple genomes of individual species, which revealed an extensive genomic intraspecies diversity [5]. This has revolutionized the microbial evolution perception, shifting

Alessio Mengoni et al. (eds.), *Bacterial Pangenomics: Methods and Protocols*, Methods in Molecular Biology, vol. 1231, DOI 10.1007/978-1-4939-1720-4_13, © Springer Science+Business Media New York 2015

from a view of "stable" genomes to a more dynamic scenario, in which gene gain/loss and the mobilization of genetic elements have played and are still playing a major role in shaping microbial genomes, to the point that defining the genomics boundaries of a bacterial species is a hard task.

It has been argued [2] that a bacterial species may be described by its *pangenome*, i.e., the set of all the genes belonging to it [6], which can be split into a *core genome* (the set of genes shared by all the genomes and that likely encode functions related to the basic cellular biology), and a *dispensable genome*, which, in turn, can be subdivided into an *accessory genome* (the set of genes possessed by a subset of genomes) and a *unique genome* (i.e., genes embedded only in one genome). Particularly, the latter contribute to the diversity of the species and probably provide functions that are not essential for cell viability and surviving, even though they might confer some advantages under particular environmental conditions, such as adaptation to specific niches, antibiotic resistance, and the ability to colonize new hosts [7].

A common visualization used for representing a pangenome is the *Venn diagram*, in which each set stands for the collections of all the genes of a given genome, and the intersections among them represent the pangenome components, namely the *core* genome and the *accessory/unique* genomes (Fig. 1).

From an evolutionary viewpoint the *pangenome* model of a species can provide information about its genomic heterogenicity (in terms of gene content), and can be used to estimate the following:

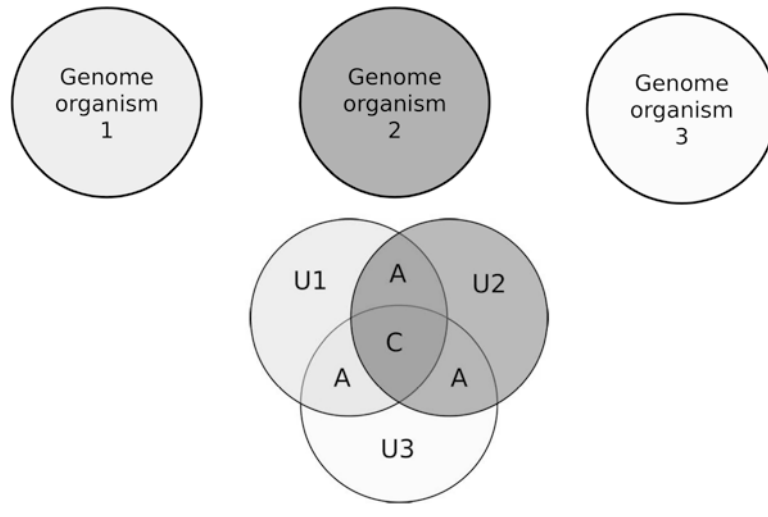

Fig. 1 Pangenome representation. This Venn diagram represents a hypothetical pangenome composed by three genomes (three labeled circles). The letters in the Venn diagram label different pangenome sections, i.e., the core genome (C), the accessory genome (A), and the unique genomes (U1, U2, U3)

(1) the extent of the global gene repertoire of that specific taxa (to which we will refer as *pangenome size*), (2) the size of the species *core* genome, and (3) the average number of novel genes added to the pangenome when new genomes are sequenced. The latter point is related to the concept of *open/close pangenome*: for closed pangenomes, completing the genome sequence of additional bacterial strains is unlikely to yield novel genes, whereas for open pangenomes, each new genome sequence usually reveals new members of the gene pool for that species [6].

The approach for estimating the pangenome size, the *core* genome, and novel gene discovery rate have been pioneered by Tettelin et al. [2]; intuitively, by starting from a small pangenome model (i.e., two genomes) and adding genomes to it, a high number of novel genes will be found, since the starting gene repertoire was small; conversely, the size of the *core* genome will decrease, since genes will be less likely to be shared by all the genomes. The greater the number of genomes added, the larger the pangenome, and the lesser the number of novel genes that will be disclosed; parallel to this, the size of the *core* genome will decrease. It is quite possible that a "saturation" point will be reached, in the sense that adding new genomes will not increase the size of the core genome, while the ratio of novel genes will be asymptotically stabilized on a certain value. For a closed pangenome, this value is zero and the pangenome size can be estimated; for an open pangenome, this value is nonzero, and the pangenome size cannot be estimated (i.e., it will probably grow "indefinitely").

Since the number of shared genes and the number of strain-specific genes of a pangenome depend on how many strains are taken into account, the approach used by Tettelin et al. consisted in using eight genomes of pathogenic *Streptococcus agalactiae* strains and computing all the possible comparisons between n genomes (i.e., eight possible combinations for pangenome of $n=2$ genomes).

Plotting the number of shared genes and the number of novel genes for every comparison as a function of the n strains considered, Tettelin et al. were able to fit exponential decaying function curves over the data which asymptotically reached the values of 1,806 shared genes and 33 novel genes, corresponding to the estimate of core genome size and novel gene discovery rate (Fig. 2a, b). The latter value was used for extrapolating the *S. agalactiae* pangenome size (Fig. 2c).

1.1 Ortholog Definition

A central step in comparative genomics is the definition of homology relationships between genes belonging to different genomes, that is, to infer whether two genes descend either from the speciation event (orthologs) or a duplication event (paralogs) of an ancestral sequence. This is usually achieved by means of sequence similarity between the genes (Fig. 3a). Often, orthologs are referred

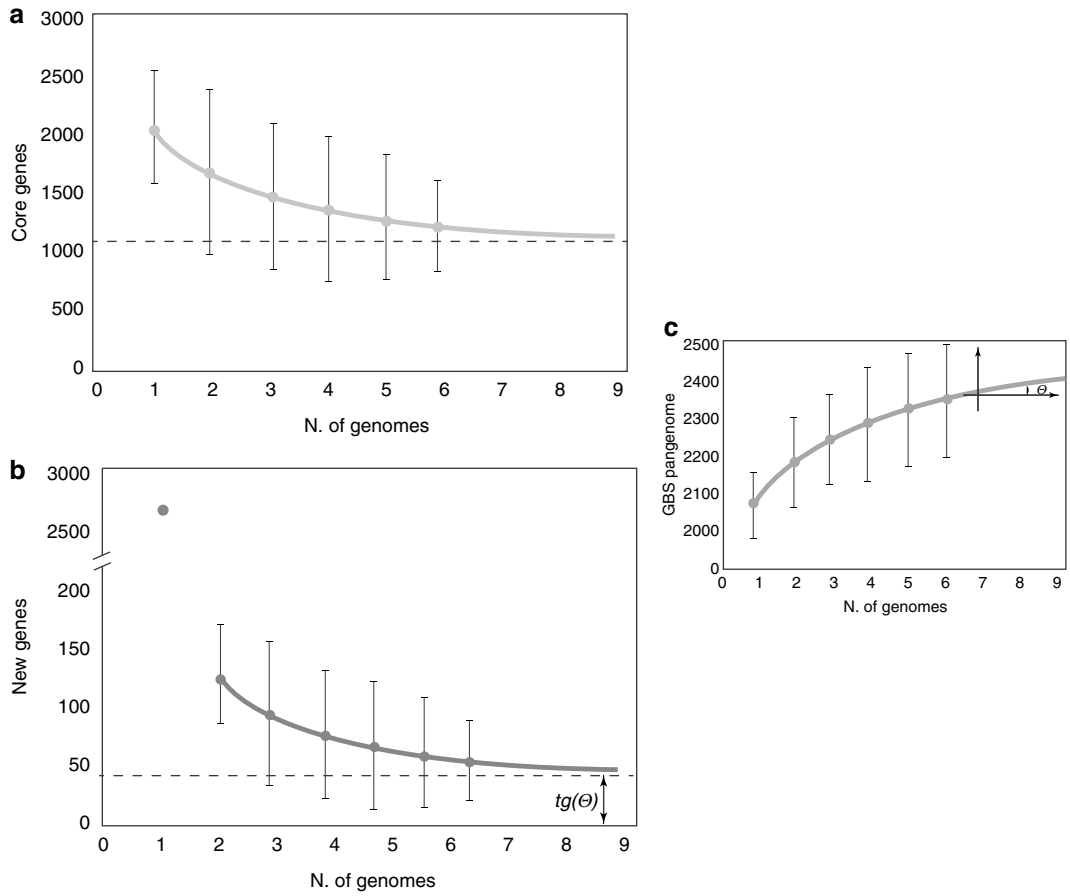

Fig. 2 Pangenome metrics trends. The three schematic curves (**a**, **b**, and **c**) represent, respectively, the core genome size, the novel gene discovery rate, and the pangenome size. The curves are fitted on points obtained from uniformly sampled pangenomes of increasing dimension (modified from ref. 7)

as corresponding genes or, in a more intuitive (yet less accurate) manner, the same genes in different species. The sequence similarity approach used for finding orthologous sequences (genes or proteins) relies on the assumption that they are more similar to each other than they are to any other sequence from the compared genome, or also, they are *bidirectional best hits* (BBHs) [8]. Thus, it can be assumed that BBHs are most likely to be composed of orthologs, justifying the use of this fast and simple method for the identification of gene families (*BBH approach*, Fig. 3a). However, this approach does not take into account the duplication event(s) that might have occurred after a speciation event, since it captures only one-to-one orthologous relationships. More in detail, defining as *inparalogs* those paralogous sequences resulting from a gene duplication event after a given speciation event, the BBH approach will likely fail to recognize the co-orthologous relationships

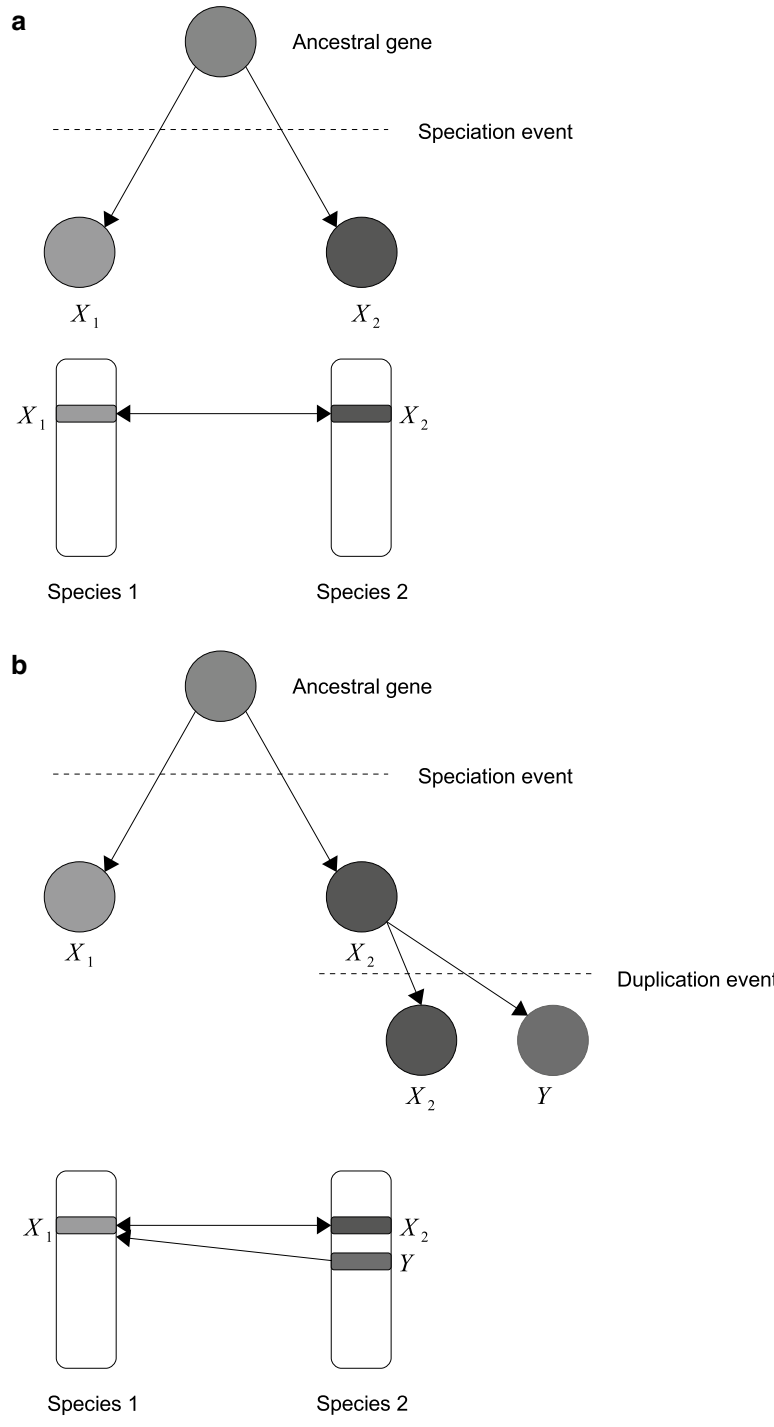

Fig. 3 Orthologous relationship. The figure shows the differences between ortholog (**a**) and inparalog (**b**) clusters. A double-edged arrow represents bidirectional best hits. It can be observed how a BBH approach would underestimate the size of the clusters of orthologous genes in case of in-paralogy

between the inparalogs (Fig. 3b). To overcome this issue, other approaches for ortholog identification can be used, relying on the concept of orthologous groups, which generalize and extend the notion of genome-specific best hit, by (1) abandoning the best hit reciprocity condition and (2) extending the notion of genome-specific best hit to multiple genomes such that clusters of consistent best hits are identified. Among the different approaches developed, it is worthy to mention *Cluster of Orthologous Genes proteins* (COGs) and InParanoid/MultiParanoid, which are, respectively, used to call orthologs in pairwise comparison and multiple genome comparison [9–11].

2 Materials

In this section the inputs needed for creating a pangenome model are listed; in Subheading 3 the programs will be discussed while in Subheading 4 a tutorial will guide you to the creation of a small *S. aureus* pangenome and to its metrics estimation.

2.1 Sequences

Sequences in FASTA format of genes (nucleotide) and/or the corresponding proteins (amino acid) are required. Usually the gene sequences can be found as GenBank annotations; however, when the annotations are not available, gene sequences might be predicted from the genomic sequence by using an appropriate tool (i.e., Prodigal, *see* Subheading 3.1).

3 Methods

3.1 Gene Calling

A gene calling tool may be used to obtain the gene/protein sequences from genomic sequences. Indeed, microbial gene prediction is a well-studied issue and currently there are a number of tools, like *GeneMarkHMM*, *Glimmer*, or *Prodigal* [10, 12, 13], that rely on statistical learning methods such as hidden Markov model (HMM) to address this task. Particularly, tools based on unsupervised learning (i.e., Prodigal) are fast and easy to use since they do not require additional datasets for the training phase, being able to infer the algorithm's parameters from the input genome.

3.2 Ortholog Identification

When the collection of the entire gene sequences for a given set of genomes is available, orthology relationships can be inferred, using *ad hoc*-designed software. Among most commonly used programs, it is worthy to cite (1) *OrthoMCL* [14], and (2) *InParanoid* [15]. Both the methods carry out genome pairwise comparisons using a similar BLAST-based approach to identify the orthologous relationships between two sequences. Then, the orthologous genes are clustered with one of the two abovementioned approaches.

OrthoMCL uses the Markov clustering algorithm [16], a method based on a graph flow theory, which, by simulating random walks on a graph using Markov matrices, determines the transition probabilities among the nodes in the graphs, eventually producing clusters of nodes representing groups of orthologous proteins between two or more species. InParanoid was initially designed for finding orthologous sequences in pairwise genome analysis only [15]; more recently, the algorithm called MultiParanoid [11] was set up to complement and extend the InParanoid approach by taking as input the collection of pairwise orthologous clusters and producing clusters of orthologous genes.

The comparison of the results obtained by using these different methods for ortholog identification revealed the existence of only small performance differences between them [17].

3.3 Pangenome Construction and Metrics

The identification of the orthologous genes in a group of related organisms allows generating a pangenome model. As previously stated, given a group of G input genomes and their corresponding groups of orthologs, we can define the *Core genome* as the set of the genes shared by all the input genomes, the *Dispensable genome* as the set of genes present only in some genomes, and the *Unique genome* as the set of genes present only in one genome. The *Pangenome size* can be defined as the total number of the gene groups, corresponding to the union of the sets of genes. Similarly, the *Core genome size* is the number of the group of genes present in the core genome.

Using an iterative approach, the shared and strain-specific gene pool size can be extrapolated, by simulating the sequential inclusion of (up to) G genomes in all possible combinations. The total number of independent measurement (N) for n genomes taken into account is

$$N = \frac{G!}{(n-1)!\times(G-n)!}$$

For n going from 1 to G (that means to consider a pangenome composed by 1,2,…,G genomes), for each of the N possible independent measured pangenomes, the numbers of shared and strain-specific genes, and the pangenome size as well, are assessed. The size of the species core genome and the number of strain-specific genes for a large number of sequenced genomes were extrapolated by fitting the exponential decaying functions:

$F_c = \kappa_c \exp[-n/\tau_c] + \Omega$ and $F_s = \kappa_s \exp[-n/\tau_s] + tg(\theta)$, respectively, to the amount of conserved genes and of strain-specific genes. In this formula (1) n is the number of sequenced strains, (2) κ_c, κ_s, τ_c, τ_s and Ω are free parameters, and (3) $tg(\theta)$ is a parameter representing the extrapolated rate of growth of the pangenome size, $P(n)$, as a greater number of independent genome sequences become available. The pangenome size can be written as function of n as follows:

$$P(n) = D + \sum_{j=2}^{n} \left\{ k_s \exp(-j\tau_s) + tg(\theta) \right\}$$

where D is the average number of genes of the input genomes.

From this equation, it derives that

$$\lim_{n \to \infty} P(n) \approx tg(\theta) \cdot n$$

By fitting the pangenome size and the number of shared and strain-specific genes, computed as function of n, to the exponential functions described above, the parameters corresponding to the best fitting and their associated correlation coefficient are found. In particular, the value $tg(\theta)$ corresponds to the inferred number of strain-specific genes for a pangenome of infinite size; that is, by sequencing new genomes, the number of novel genes found will asymptotically reach the value of $tg(\theta)$.

4 Notes

In this section scripts for the common tasks required for a pangenome construction, such as genomic sequence downloading, gene calling, and ortholog identification, are presented. Indeed, these tasks can be performed in a simple way by exploiting existent software and taking advantage of the UNIX shell. The scripts presented in this section are intended to be executed directly in a shell terminal.

4.1 Download Genomic Sequences

The sequences of complete and draft genomes can be found at the GenBank site (ftp://ftp.ncbi.nih.gov/genomes/). Even though the sequences can be manually downloaded, a high-throughput method is preferable, especially when dealing with a high number of sequences of the same species/genus. This task can be performed by using a combination of *curl* and *wget*. Curl is a tool to transfer data from or to a server, using a plethora of protocols (DICT, FILE, FTP, FTPS, GOPHER, HTTP, HTTPS, IMAP, IMAPS, LDAP, LDAPS, POP3, POP3S, RTMP, RTSP, SCP, SFTP, SMTP, SMTPS, TELNET, and TFTP). *wget* is a utility for non-interactive downloading of files, supporting HTTP, HTTPS, and FTP protocols, as well as retrievals through HTTP proxies. Given a genus/species (i.e., *Escherichia coli*), the list of the corresponding completely sequenced strains can be obtained as follows:

```
species=Escherichia_coli
strains=`curl   ftp://ftp.ncbi.nih.gov/genomes/
Bacteria/ -l -s | grep $species`
```

For each strain, the sequence of the replicons can be obtained using *wget*:

```
strain=Escherichia_coli_042_uid161985
files=`curl ftp://ftp.ncbi.nih.gov/genomes/Bacteria/
$strain/ -l -s | grep ".fna"`
for f in $files; do wget ftp://ftp.ncbi.nih.gov/
genomes/Bacteria/$strain/$f ;done
```

The final pipeline for obtaining all the complete genomes is

```
species=Escherichia_coli_042_uid161985
strains=`curl   ftp://ftp.ncbi.nih.gov/genomes/
Bacteria/ -l -s | grep $species`
for strain in $strains;
 do
mkdir $strain
cd $strain
files=`curl ftp://ftp.ncbi.nih.gov/genomes/Bacteria/
$strain/ -l -s | grep ".fna"`
for f in $files; do wget ftp://ftp.ncbi.nih.gov/
genomes/Bacteria/$strain/$f ;done
cd ..
done
```

Similarly, the draft sequences can be obtained by substituting ftp://ftp.ncbi.nih.gov/genomes/Bacteria/ with ftp://ftp.ncbi.nih.gov/genomes/Bacteria_DRAFT/ in the above commands.

4.2 Gene Calling

In this section the usage of Prodigal for calling genes from genomic sequences is described. The latest software versions are available at the Google code prodigal page (http://code.google.com/p/prodigal/downloads/list). Although the software does not require an installation step, it is necessary to make it executable with the *chmod* command (should be run with root privileges):

```
chmod+x ./prodigal.v2_60.linux
```

The tool is versatile and accepts up to 15 different options, but the most common usage is to output genes and proteins in quiet mode:

```
genome=Escherichia_coli_042_uid161985_genome.
fna
./prodigal.v2_60.linux -i $genome -d $genome.
genes -a $genome.prots -q
```

This can be done for a high number of genome (files tagged with the ".inp" extension):

```
inputs=`ls | grep .inp$`
for i in $inputs; do ./prodigal.v2_60.linux -i
$i -d $i.genes -a $i.prots -q; done
```

4.3 Ortholog Identification

In this section the ortholog identification task with the combination of InParanoid and MultiParanoid is reported. The two programs are designed to be used together, since MultiParanoid takes as input the output(s) of InParanoid.

The InParanoid and MultiParanoid software can be obtained, respectively, from http://software.sbc.su.se/cgi-bin/request.cgi?project=inparanoid and http://multiparanoid.sbc.su.se/download/.

InParanoid takes as input two protein files in multi-FASTA format, and performs different BLAST search [18] between the two files, eventually outputting (1) an *Output* file summarizing the InParanoid analysis; (2) a *table* file, which reports in a tabular format the homology relationships between the proteins; and (3) a *sqltable* file, which is a sql-computable equivalent of the table file.

The software usage is

```
inparanoid.pl<FASTAFILE with sequences of species A><FASTAFILE with sequences of species B>
```

As caveat, assuming that the software is correctly installed and the input files are in a same directory and tagged with the ".inp" extension, InParanoid can be used in a high-throughput fashion as follows:

```
inputs=`ls | grep .inp$`
set -- $inputs
for a
 do
shift;
for b;
 do
$(printf "./inparanoid.pl %s %s\n" "$a" "$b");
done;
done
```

For a high number of genomes this may take days of computation, also with high-performance machines. Once finished, a set of output files for each pair of genomes should be created, which may eventually be used as input from MultiParanoid.

Before using it, the MultiParanoid script (multiparanoid.pl) requires the proper setting of some variables within the script itself (*$inputdir* and *$output*, encoding the values of the inputs and output directory, respectively). To account for these changes, the script can be opened with a text editor of choice and then modified accordingly.

The MultiParanoid usage is as follows:

```
./multiparanoid.pl -species<LIST>
```

where <LIST> is the list of the species names, connected with "+" (i.e., mouse+cat+dog). This list can be produced from the directory containing the output files from InParanoid, with the following code:

```
all_species=`ls INPDIR/ | grep ^table | sed 's/
table.//g' | awk -F "-" '{print $1}' | uniq`
out=`echo $all_species | sed 's\ \+\g'`
```

where INPDIR is the directory containing the InParanoid output files.

Eventually, MultiParanoid will produce a single output file with the orthologous clusters and their compositions. The output file will contain a line for each protein, with seven tab-separated values: clusterID, species, gene, is_seed_ortholog, confidence_score, species_in_cluster, and tree_conflict.

From the software manual it can be found that "is_seed_orthologs" means that the protein was a seed ortholog in at least one InParanoid cluster, "confidence_score" is an average InParanoid score across the input clusters, while "tree_conflict" indicates that, from the point of view of different species, the number of inparalogs varied in at least one other species ("diff.numbers") or the numbers were the same, but the IDs differed ("diff.names").

References

1. Read TD, Salzberg SL, Pop M, Shumway M, Umayam L, Jiang L, Holtzapple E, Busch JD, Smith KL, Schupp JM, Solomon D, Keim P, Fraser CM. Comparative genome sequencing for discovery of novel polymorphisms in Bacillus anthracis. Science 296:2028–2033
2. Tettelin H, Masignani V, Cieslewicz MJ, Donati C, Medini D, Ward NL, Angiuoli SV, Crabtree J, Jones AL, Durkin AS, DeBoy RT, Davidsen TM, Mora M, Scarselli M, Ros IM, Peterson JD, Hauser CR, Sundaram JP, Nelson WC, Madupu R, Brinkac LM, Dodson RJ, Rosovitz MJ, Sullivan SA, Daugherty SC, Haft DH, Selengut J, Gwinn ML, Zhou L, Zafar N, Khouri H, Radune D, Dimitrov G, Watkins K, O'Connor KJB, Smith S, Utterback TR, White O, Rubens CE, Grandi G, Madoff LC, Kasper DL, Telford JL, Wessels MR, Rappuoli R, Fraser CM (2005) Genome analysis of multiple pathogenic isolates of *Streptococcus agalactiae*: implications for the microbial "pan-genome". Proc Natl Acad Sci U S A 10:13950–13955
3. Hayashi T, Makino K, Ohnishi M, Kurokawa K, Ishii K, Yokoyama K, Han C, Ohtsubo E, Nakayama K, Murata T, Tanaka M, Tobe T, Iida T, Takami H, Honda T, Sasakawa C, Ogasawara N, Yasunaga T, Kuhara S, Shiba T, Hattori M, Shinagawa H (2001) Complete genome sequence of enterohemorrhagic *Escherichia coli* O157: H7 and genomic comparison with a laboratory strain K-12. DNA Res 8:11–22
4. Kuroda M, Ohta T, Uchiyama I, Baba T, Yuzawa H, Kobayashi I, Cui L, Oguchi A, Aoki K, Nagai Y, Lian J, Ito T, Kanamori M, Matsumaru H, Maruyama A, Murakami H, Hosoyama A, Mizutani-Ui Y, Takahashi NK, Sawano T, Inoue R, Kaito C, Sekimizu K, Hirakawa H, Kuhara S, Goto S, Yabuzaki J, Kanehisa M, Yamashita A, Oshima K, Furuya K, Yoshino C, Shiba T, Hattori M, Ogasawara N, Hayashi H, Hiramatsu K (2001) Whole genome sequencing of meticillin-resistant *Staphylococcus aureus*. Lancet 357: 1225–1240
5. Pallen MJ, Wren BW (2007) Bacterial pathogenomics. Nature 449:835–842
6. Medini D, Donati C, Tettelin H, Masignani V, Rappuoli R (2005) The microbial pan-genome. Curr Opin Genet Dev 15:589–594
7. Tettelin H, Riley D, Cattuto C, Medini D (2008) Comparative genomics: the bacterial pan-genome. Curr Opin Microbiol 11: 472–477

8. Koonin EV (2005) Orthologs, paralogs, and evolutionary genomics 1. Annu Rev Genet 39:309–338
9. Tatusov RL, Koonin EV, Lipman DJ (1997) A genomic perspective on protein families. Science 278:631–637
10. Hyatt D, Chen GL, LoCascio PF, Land ML, Larimer FW, Hauser LJ (2010) Prodigal: prokaryotic gene recognition and translation initiation site identification. BMC Bioinformatics 11:119
11. Alexeyenko A, Tamas I, Liu G, Sonnhammer EL (2006) Automatic clustering of orthologs and inparalogs shared by multiple proteomes. Bioinformatics 22:e9–e15
12. Lukashin AV, Borodovsky M (1998) GeneMark. hmm: new solutions for gene finding. Nucleic Acids Res 26:1107–1115
13. Delcher AL, Harmon D, Kasif S, White O, Salzberg SL (1999) Improved microbial gene identification with GLIMMER. Nucleic Acids Res 27:4636–4641
14. Li L, Stoeckert CJ, Roos DS (2003) OrthoMCL: identification of ortholog groups for eukaryotic genomes. Genome Res 13: 2178–2189
15. O'Brien KP, Remm M, Sonnhammer EL (2005) Inparanoid: a comprehensive database of eukaryotic orthologs. Nucleic Acids Res 33:D476–D480
16. van Dongen SM (2000) Graph clustering by flow simulation
17. Galardini M, Mengoni A, Biondi EG, Semeraro R, Florio A, Bazzicalupo M, Benedetti A, Mocali S (2013) DuctApe: a suite for the analysis and correlation of genomes and Omnilog™ Phenotype Microarray data. Genomics
18. Altschul SF, Gish W, Miller W, Myers EW, Lipman DJ (1990) Basic local alignment search tool. J Mol Biol 215(3):403–410

Chapter 14

Robust Identification of Orthologues and Paralogues for Microbial Pan-Genomics Using GET_HOMOLOGUES: A Case Study of pIncA/C Plasmids

Pablo Vinuesa and Bruno Contreras-Moreira

Abstract

GET_HOMOLOGUES is an open-source software package written in Perl and R to define robust core- and pan-genomes by computing consensus clusters of orthologous gene families from whole-genome sequences using the bidirectional best-hit, COGtriangles, and OrthoMCL clustering algorithms. The granularity of the clusters can be fine-tuned by a user-configurable filtering strategy based on a combination of blastp pairwise alignment parameters, hmmscan-based scanning of Pfam domain composition of the proteins in each cluster, and a partial synteny criterion. We present detailed protocols to fit exponential and binomial mixture models to estimate core- and pan-genome sizes, compute pan-genome trees from the pan-genome matrix using a parsimony criterion, analyze and graphically represent the pan-genome structure, and identify lineage-specific gene families for the 12 complete pIncA/C plasmids currently available in NCBI's RefSeq. The software package, license, and detailed user manual can be downloaded for free for academic use from two mirrors: http://www.eead.csic.es/compbio/soft/gethoms.php and http://maya.ccg.unam.mx/soft/gethoms.php.

Key words Orthologs, Paralogs, Pan-genomics, Comparative genomics, Bacterial genomes, pIncA/C plasmids, Core-genome, Pan-genome, Software, Open-source

1 Introduction

The advent of next-generation sequencing (NGS) technology has recently boosted the number of genome sequencing projects publicly available [1]. This trend empowers comparative genomics and pan-genomics approaches to genome analysis, motivating the development of more and better software tools for these tasks. Early within-species genome comparisons, such as those performed by the group of Fred Blattner on three *Escherichia coli* strains with contrasting ecological niches (the commensal K12, the uropathogen CFT073, and enterohemorrhagic EDL933), revealed an extensive "mosaic" genome structure [2]. They determined that only 39 % of their combined proteomes were shared by all three strains.

Alessio Mengoni et al. (eds.), *Bacterial Pangenomics: Methods and Protocols*, Methods in Molecular Biology, vol. 1231, DOI 10.1007/978-1-4939-1720-4_14,

However, they found that the strains maintained remarkable synteny in the common, vertically inherited genome backbone, which is interrupted by the insertion of genomic islands that are acquired by horizontal gene transfer. Genes on these islands were found to be largely responsible for defining the lifestyles and niches of the strains. Three years after this landmark paper, Tettelin and colleagues were the first to introduce the concept of the pan-genome, the collective genetic repertoire of a certain species, developing first computational strategies to estimate its size [3]. Ever since, the microbial pan-genome has been a key topic in microbial genomics, as it has profound implication on how we understand bacterial evolution, niche adaptation, and population structure, with strong practical implications in areas such as epidemiology and vaccine development [4].

Here we present a detailed tutorial on the use of the open-source GET_HOMOLOGUES software package [5], demonstrating some of its bioinformatic, statistical, and graphical capabilities for microbial pan-genomics. Protocols are provided to define robust orthologous gene families, fit exponential and mixture models to estimate core- and pan-genome sizes, analyze and graphically represent the pan-genome structure, and identify lineage-specific gene families for the 12 complete pIncA/C plasmids currently available in NCBI's RefSeq [6].

The package is released under a GNU General Public License and is written mainly in Perl and R. GET_HOMOLOGUES is highly configurable; runs on MacOSX, and Linux operating systems; and was designed to take advantage of multiprocessor machines and computer clusters to distribute time-consuming blast+ [7] and HMMER3 [8] jobs. If constrained by RAM, the software implements the possibility to write data structures temporarily to disk using BerkeleyDB. Together these features make it possible to analyze large datasets of hundreds of microbial genomes on a dedicated server. Smaller sets up to ~50 bacterial genomes can be analyzed on a modern commodity desktop or laptop in reasonable time [5]. It automatically computes homologous gene families based on three alternative and well-established reciprocal BLAST hit algorithms (RBHAs): our own implementation of the bidirectional best-hit (BDHB) algorithm [5], COGtriangles [9], and OrthoMCL [10]. RBHAs are heuristic in nature [11, 12], but have been recently shown to produce highly accurate orthologous gene clusters when compared with tree-based methods, which are generally prohibitive for large datasets due to the computational burden they impose [13].

The GET_HOMOLOGUES package bundles several auxiliary scripts to facilitate the interrogation of homologous gene clusters, and computation of pan-genome sets and pan-genome trees based on the pan-genomic presence–absence matrix. A unique feature of the software is its capacity to compute consensus core- and

pan-genomes, that is, to define these genome sets based on the joint evidence of any combination of the three abovementioned clustering algorithms. This generates very robust, although conservative clusters. The tightness of the clusters generated by each algorithm can be fine-tuned by controlling key blast parameters such as percentage overlap and identity of pairwise alignments and E-score cutoff value. It is also possible to make orthologous gene clusters even more stringent by imposing a partial synteny criterion and/or by scanning the Pfam domain composition of the clusters using hmmscan of the HMMER3 package. Several auxiliary scripts are provided for the statistical and graphical analysis of core- and pan-genomes, which can fit both exponential and binomial mixture models to the data to estimate the sizes of the core- and pan-genomes [3, 14, 15]. The package also bundles an installation script that takes care of the installation of most external dependencies, including the downloading and formatting of the latest Pfam database required by hmmscan for domain-scanning of proteins. A detailed manual with >40 pages documenting all the software's features and options makes the use of GET_HOMOLOGUES reasonably user friendly.

To demonstrate some of the key features and capabilities of GET_HOMOLOGUES, we present detailed protocols on the use of the main script get_homologues.pl and several auxiliary scripts bundled with the package to compute robust core- and pan-genome sets of 12 large, broad-host-range bacterial resistance plasmids of the IncA/C incompatibility group (pIncA/C) [16–19], statistically estimate the size of their core- and pan-genomes, graphically visualize the structure of the pan-genome, and identify genes specifically found in the two plasmids containing the *bla*NDM-1 (New Delhi metallo-beta-lactamase-1) gene [20]. The encoded protein is one of the most recently reported metalloenzymes conferring resistance to all beta-lactams, including carbapenems, the last drug type in this class conferring nearly universal, anti-Gram-negative activity until the recent appearance of carbapenemases [21]. To make things worse, carbapenemase-producing bacteria are typically multidrug resistant (MDR) or even pan resistant [22], making the emergence and rapid spread of NDM a worldwide public health concern [18, 23]. Different plasmids, including those of the A/C incompatibility group, are largely involved in the rapid spread of NDM and other resistance genes such as $bla_{\text{CYM-2}}$, *tetA*, *flo*, and *sul* [16, 18].

2 Materials

1. The protocol depends on the installation of the GET_HOMOLOGUES software package (version 20140901 or later) [5] on a MacOSX, or Linux box. For larger datasets (>50 fully sequenced bacterial genomes), the software is best

run on a multiprocessor machine, with 8GB of RAM or more, or on a Linux computer cluster. For the demo dataset analyzed herein, a standard commodity laptop or desktop with 2 cores and 1GB of RAM will suffice. The package is freely available for academic purposes, but not for commercial or military use, as detailed in the license agreement, which can be found along with the software from two mirror servers: http://maya.ccg.unam.mx/soft/gethoms.php (Mexico) and http://www.eead.csic.es/compbio/soft/gethoms.php (Spain). Additionally the user will need to download the GenBank files for the selected pIncA/C plasmids, also available as a compressed tar file from the URL provided below: http://maya.ccg.unam.mx/soft/protocols_gethom/methMolBiol2014_get_homologues.tgz.

Subheading 3.1 provides detailed methods on how to unpack this file, which also contains all the code and auxiliary scripts used in this chapter.

2. Typographical conventions: `Monospaced text` will be used for all commands to be issued by the user, as well as directory names, program names, and output. The command prompt will be represented with the `$` symbol.

3 Methods

3.1 Downloading Selected pIncA/C Plasmid GenBank Files Using NCBI's Entrez System

The easiest way to get the GenBank files required for the protocols in this chapter is to download them from the URL provided below. Create a directory named `pIncAC/` to store the files, move into it, and use the following command to fetch the file:

```
# Make the directory, cd into it and save its path for easy access later on
$ mkdir pIncAC && top_dir=$(pwd) && cd pIncAC
$ gbk_dir=$(pwd)
$ wget -c
http://maya.ccg.unam.mx/soft/protocols_gethom/methMolBiol2014_get_homologues. tgz
```

Unpack the `*tgz` file and view the new directory's contents with the following command:

```
$ tar -xvzf methMolBiol2014_get_homologues.tgz && ls
```

3.2 Installing GET_HOMOLOGUES and Its External Dependencies

After downloading the package from the closest of the abovementioned mirrors you will have to unzip and unpack it, change into the `get_homologues/` directory, and launch the install script with the following command issued from your terminal:

```
$ tar xvfz get_homologues_X.Y.tgz; cd get_homologues_X.Y; ./install.pl
```

Note that _X.Y has to be changed to the actual distribution version you downloaded. Please follow the indications provided by the installation script in case some required dependency is missing. They should be enough to assist you with the installation of dependencies. The protocols presented below require a full installation of the external dependencies, which includes R and the latest version of the Pfam-A database [24]. Read "Subheading 2" of the manual (bundled with the distribution) if you need additional help on the installation process.

3.3 Computing Orthologous Gene Clusters for pIncA/C Plasmids Using the BDBH Algorithm Under Default Settings

We are now set to proceed with the actual calculations. The aim of this protocol is to compute orthologous gene clusters or families using the main script `get_homologues.pl` and its default clustering method (BDBH) under default parameter values. This is intentionally kept simple in order to focus the reader's attention on the basic computational steps involved in the whole process. Make sure that you are working in the parental directory (one directory above) of `pIncAC/`, the directory in which we stored the GenBank files (**step 1** of Subheading 2). To display the program's help menu simply type (*see* **Note 2**)

```
$ cd $top_dir
$ get_homologues.pl
```

Let us start by running a standard BDBH analysis with default parameter values (75 % pairwise alignment coverage [`-C 75`], E-value=1e-05 [`-E 1e-05`], using two threads or cores [`-n 2`] and retaining only clusters that contain at least one representative protein from each proteome analyzed [`-t number_of_proteomes`], running the analysis on the local machine [`-m local`]). This is as simple as issuing the following command from your terminal prompt:

```
$ get_homologues.pl -d pIncAC
```

The `get_homologues.pl` script will start extracting the CDSs from the GenBank files to generate replicon/genome-specific multi-FASTA files at the protein level (their proteomes; *see* **Note 3**), with sequences uniquely numbered to allow reusing of results if new proteomes are added. These are copied into a new directory named as the directory with the source GenBank files plus a "_homologues" suffix (`pIncAC_homologues/`in our example). The script will then use these FASTA-formatted proteomes to generate blast databases by automatically calling `makeblastdb` from the `blast`+package [7]. Next, `blastp` will be called to make an all-against-all blastp search, splitting jobs among the available threads. If your computer has more cores, you can use `-n <no_of_cores_to_use>` to speed up the process. In preparation for identifying bidirectional best-hits (BDBHs), the individual pairwise blast results are concatenated and sorted, so that all hits of a query are grouped together and ranked in terms of E-value.

Note that the BDBH algorithm requires a reference genome. If none is specified, `get_homologues.pl` will automatically select the smallest input file as the reference. The sorted blast table, which can be quite large, is then parsed in order to calculate alignment lengths, also managing hits with several multiple high-scoring segments. The resulting file is indexed for faster posterior data access, storing the first and last hits of every query. The algorithm starts by finding inparalogues [25] in the reference genome. These are operationally defined as bidirectional BDBHs found within the same genome from which the query protein derives, that is, better within-genome hits (obviously excluding the query protein itself) than those found in any other genomes included in the analysis. The inparalogues of a second proteome are labeled next, before identifying BDBHs between the reference genome and this second one. This process is repeated until all non-reference genomes were compared with the reference one, as depicted in Fig. 3 of the manual. All BDBHs found outside the reference genome for a particular protein are added to a cluster, labeled according to the reference protein name and written to disk. Note that these clusters will contain at least one representative of each proteome. A cluster that contains more members (proteins) than the number of proteomes compared indicates the presence of inparalogues in at least some non-reference proteomes. The BDBH clusters are all saved in a directory named in a fashion that makes it easy to identify the clustering algorithm and associated parameters used for that particular analysis. For example

```
EscherichiacolistrainSCEC2plasmidpSCEC2NC022377_f0_alltaxa_algBDBH_e0_
```

indicates the name of the reference genome, that no % length difference within cluster filtering was applied (_f0_), and that only clusters containing at least one member from all proteomes analyzed are considered. In our case that means that all clusters contain at least 12 protein sequences, one from each original proteome. Note that equivalent clusters of DNA sequences are also produced from input files if they are in GenBank format. These are therefore orthologous gene clusters, as defined by the BDBH algorithm. The flag _e0_ indicates that clusters with inparalogues were allowed (default behavior). How can we find out how many orthologous gene clusters were found and the number of protein sequences each one contains? This is easy to answer using basic shell filtering commands. Let us first change into the directory (`cd`) containing the blast results (`pIncAC_homologues/`) and explore its contents by issuing the following commands (lines preceded with a hash symbol are simply comments that are ignored by the shell command interpreter):

```
# cd into blast results directory and save its path in the variable $blast_dir
```

```
$ cd pIncAC_homologues
$ blast_dir=$(pwd)
# explore contents by file extension names
$ ls | cut -d\. -f2 | sort | uniq -c
   1 cluster_list
   1
EscherichiacolistrainSCEC2plasmidpSCEC2NC02
2377_f0_alltaxa_algBDBH_e0_
   216 gbk
   1 tmp
   1 txt
# find which of those files are directories
$ find . -type d
   ./tmp
   ./EscherichiacolistrainSCEC2plasmidpSCEC2NC
022377_f0_alltaxa_algBDBH_e0_
```

Take some time to explore the contents of the different files. Due to space constraints we cannot explain the contents of all the intermediary files herein, but more information can be found in the manual. So lets `cd` into the directory containing the BDBH orthologous clusters obtained by running `get_homologues.pl` under default settings to explore the results in greater detail. Note that the output of some of the commands is truncated or not shown, in order to save space and trees:

```
# cd into the BDBH clusters directory (default
BDBH clusters)
$                                          cd
EscherichiacolistrainSCEC2plasmidpSCEC2NC022
377_f0_alltaxa_algBDBH_e0_
# list contents (orthologous gene clusters) and
count them
$ ls
1238_repA.faa 1270_hypothetical_protein.faa
1241_putative_signal_peptide_peptidase_SppA.faa
1271_dsbc.faa
1242_DsbA-like_thioredoxin_domain_protein.faa
1287_protein_YbaA.faa
… output cut to save trees
$ ls | wc
   23 23 684
```

```
# how many genes does each orthologous cluster
contain?
$ grep -c '>' *faa
1238_repA.faa:12
1241_putative_signal_peptide_peptidase_SppA.
faa:12
1242_DsbA-like_thioredoxin_domain_protein.
faa:12
… output truncated
#which clusters contain inparalogues (in our
case > 12 sequences)?
$ grep -c '>' *faa | grep -v ':12'
1270_hypothetical_protein.faa:13
1271_dsbc.faa:13
```

The result of issuing these commands is that we found 23 clusters of orthologous proteins among the 12 plasmid proteomes, two of which contain 13 sequences (one cluster contains an inparalogue) and the remaining 21, twelve proteins, one from each source proteome. The question to answer now is the following: Which plasmids contain the loci with inparalogues?

```
# which plasmid proteome contains the locus with
inparalogues
#  for orthologous cluster 1270_hypothetical_
protein.faa?
$ grep '>' 1270_hypothetical_protein.faa | cut
-d\| -f2,3 | sort | uniq -c
   1 [Aeromonas hydrophila]|
   1 [Escherichia coli]|APEC1990_61
   1 [Escherichia coli]|AR060302
   1 [Escherichia coli]|H4H
   1 [Escherichia coli]|NDM-1 Dok01
   1 [Escherichia coli]|PG010208
   1 [Escherichia coli]|SCEC2
   1 [Escherichia coli UMNK88]|UMNK88
   1 [Klebsiella pneumoniae]|
   1 [Klebsiella pneumoniae]|Kp7
   2 [Salmonella enterica]|AM04528
   1  [Salmonella  enterica  subsp.  enterica
serovar Kentucky]|1643/10
```

That output reveals that the proteome of *Salmonella enterica* AM04528 is the one which contains two copies (inparalogues) for cluster 1270. Repeat the exercise for cluster 1271.

3.4 Computing Orthologous Gene Clusters for pIncA/C Plasmids Using the BDBH Algorithm Imposing Homogeneous Pfam Domain Composition on Cluster Members

The default parsing parameters for blast results are quite stringent, imposing 75 % pairwise alignment coverage [`-C 75`] and an E-value value cutoff=1e-05 [`-E 1e-05`]. Depending on the divergence of the dataset to be analyzed, these parameters may be relaxed (divergent set) or made more stringent (within species). A less arbitrary and very powerful means of selecting *bona fide* orthologous clusters is imposing the restriction that all members have the same Pfam domain composition [24]. Due to the relatively tight link that exists between protein domain architecture and function, this restriction makes the resulting clusters more likely to contain functionally equivalent proteins [26]. This can be easily performed calling the `get_homologues.pl` script with the `-D` option, as shown below:

```
# Generate BDBH clusters containing proteins with conserved Pfam domain composition
$ cd $top_dir
$ nohup get_homologues.pl -d pIncAC -D &> log.get_homologues_pIncAC_BDBH_C75D_allTaxa &
$ tail -f log.get_homologues_pIncAC_BDBH_C75D_allTaxa
```

For a brief explanation of the additional shell commands and syntax used in this command line *see* **Note 4**. The `-D` option calls the Pfam-based HMMER domain scanning function implemented in GET_HOMOLOGUES (*see* **Note 5**). Each protein from each source FASTA file will be scanned with hmmscan using the Pfam-A domain database [24]. The results are concatenated and parsed, generating a file containing strings of domain composition and order for each protein of all proteomes.

The `get_homologues.pl` script will notice that we are running a new analysis on the same input dataset and will therefore reuse as much of the previous calculations as possible. In this case, the script will reuse the all-versus-all `blastp` results from the previous run. However, the blast results are newly parsed, now taking into account the domain composition of the reciprocal best hits in order to construct the orthologous clusters. The new clustering results are saved in its own directory, named with a `_Pfam_` suffix, as shown below:

```
# cd into the Pfam-domain filtered BDBH cluster directory
$ cd $blast_dir
$ cd
EscherichiacolistrainSCEC2plasmidpSCEC2NC022377_f0_alltaxa_algBDBH_Pfam_e0_
# list contents (orthologous gene clusters) and count them
$ ls && ls | wc
```

```
1238_repA.faa    1259_N-6_DNA_Methylase_family_
protein.faa
1298_traF.faa
   1241_putative_signal_peptide_peptidase_SppA.
faa 1260_hypothetical_protein.faa 1299_traH.faa
… output truncated
   22 22 658
#which clusters contain inparalogues
$ grep -c '>' *faa | grep -v ':12'
1270_hypothetical_protein.faa:13
```

Repeating similar commands as shown in the previous section we find that this new BDBH analysis uncovers 22 orthologous clusters (vs. 23 in the previous one), only one of which has 13 proteins (i.e., contains an inparalog). So the question to answer now is the following: Which are the clusters from the standard BDBH analysis that do not contain a homogeneous Pfam domain composition? This can be easily answered with the following shell commands:

```
# generate two files listing the clusters found
by the standard and Pfam-domain filtered BDBH
clusters
$ ls *faa > Pfam_filtered_BDBH_clusters.list
$ ls
../EscherichiacolistrainSCEC2plasmid
pSCEC2NC022377_f0_alltaxa_algBDBH_e0_/*faa | \
sed
's#../EscherichiacolistrainSCEC2plasmi
dpSCEC2NC022377_f0_alltaxa_algBDBH_e0_/##' \
> standard_BDBH_clusters.list
# find the difference between the two lists
$ diff standard_BDBH_clusters.list
Pfam_filtered_BDBH_clusters.list | grep '<'
< 1262_topB.faa
< 1271_dsbc.faa
< 1293_site-specific_recombinase-_
phage_integrase_family.faa
```

This result demonstrates the higher stringency of the Pfam domain-composition filtering strategy. It also suggests that 1270_hypothetical_protein.faa may be a true inparalogue that has recently been duplicated, without changing its Pfam domain composition and ordering.

If the user wishes to obtain orthologous BDBH gene clusters containing only single-copy genes, any of the previous `get_homologues.pl` commands could have been expanded with the `-e` flag, which excludes clusters with inparalogues. We leave this exercise for the reader.

3.5 Computing a Robust Strict Core Genome and the Corresponding Clusters of Orthologous Gene Clusters with Homogeneous Pfam-Domain Composition for pIncA/C Plasmids Using the Intersection Between BDBH, COG, and OrthoMCL Gene Families

We have recently shown that the definition of orthologous clusters and their composition are variable depending on the clustering method used [5, 27]. Technical details aside, it is clear that the most robust orthologous gene clusters would be those recognized by all three clustering algorithms currently implemented in GET_HOMOLOGUES. We will now run `get_homolgues.pl` sequentially, to obtain the COG and OrthoMCL clusters of any size by using the `-t 0` option (only valid for these two algorithms, but not for BDBH, since the latter requires that the reference genome is always present in the clusters). This option is required when we are interested in computing pan-genome sizes and the frequency distribution of pan-genomic cluster sizes, the pan-genome structure (note that by default `-t` is set to the number of all proteomes). The auxiliary script `compare_clusters.pl` can then be used to produce intersection pan-genome matrices, including the computation of consensus core genomes. We will also use the `-c` flag for genome composition analysis, that is, to obtain tables of re-sampled core- and pan-genome sizes which can be used by the auxiliary script `plot_pancore_matrix.pl` to fit Tettelin [3] or Willenbrock [28] exponential decay models to estimate core genome sizes, and the exponential Tettelin model [28] to get estimates and graphical plots of the pan-genome size. The next code snippets show the use of `get_homologues.pl` to call the three clustering algorithms combined with `compare_clusters.pl` to parse them in order to obtain consensus clusters. Make sure that you are just above the `pIncAC/` directory holding the GenBank files and issue the following command:

```
$ cd $top_dir
$ nohup get_homologues.pl -d pIncAC -G -D -t 0 -c &> log.get_homologues_pIncAC_GDt0c && get_homologues.pl -d pIncAC -M -D -t 0 -c &> log.get_homologues_pIncAC_MDt0c && get_homologues.pl -d pIncAC -D -c &> log.get_homologues_pIncAC_BDBH_Dc &
```

This command will sequentially call the main script `get_homologues.pl` to run the COG, OrthoMCL, and BDBH algorithms under stringent conditions of homogeneous Pfam-domain composition (`-D`), reporting core- and pan-genome composition (`-c`), and in the case of the former two clustering methods, reporting clusters of all sizes (`-t 0`) (*see* **Note 6**). Note that running two jobs simultaneously on the same input directory might produce unexpected results, so it is not encouraged.

This will run very quickly, as we have already performed all `blastp` runs and Pfam-based `hmmscan` searches for domain composition. We are now ready to use the auxiliary `compare_clusters.pl` script that will read the contents of the three directories containing the BDBH, COG, and OrthoMCL clustering results to compute the consensus single-copy orthologous gene families, by using the following code snippet:

```
# generate the consensus single-copy orthologous gene clusters with compare_clusters.pl
$ cd $blast_dir
$ compare_clusters.pl -d
EscherichiacolistrainSCEC2plasmidpSCEC2NC022377_f0_0taxa_algCOG_Pfam_e0_,EscherichiacolistrainSCEC2plasmidpSCEC2NC022377_f0_0taxa_algOMCL_Pfam_e0_,EscherichiacolistrainSCEC2plasmidpSCEC2NC022377_f0_alltaxa_algBDBH_Pfam_e0_ -o intersect_core_BCM_Dt12 -t 12 -m
```

The `-d` option is used to pass the script the names of the three directories containing the source clusters. Option `-o` is required to provide an output directory to hold the resulting cluster information, the corresponding FASTA files, and a PDF file with a Venn diagram showing the results of the parsing analysis. Option `-t 12` tells the script to report only the clusters with the indicated number of proteomes (all in our case). The following code snippets show how to explore the contents of the newly generated results directory which we have named `intersect_core_BCM_Dt12/`

```
# cd into the intersect_core_BCM_Dt12 directory and explore its contents
$ cd intersect_core_BCM_Dt12 && ls && ls *faa | wc
1238_repA.faa1297_uvrD-REP_helicase_N-terminal_domain_protein.faa
1241_putative_signal_peptide_peptidase_SppA.faa
1298_traF.faa
1242_DsbA-like_thioredoxin_domain_protein.faa
1299_traH.faa
… output truncated
   18 18 553
# confirm that all 18 clusters contain only one sequence from each plasmid/proteome
$ grep '>' *faa | cut -d\| -f2,3 | sort | uniq -c
   18 [Aeromonas hydrophila]|
   18 [Escherichia coli]|APEC1990_61
   18 [Escherichia coli]|AR060302
   18 [Escherichia coli]|H4H
```

```
18 [Escherichia coli]|NDM-1 Dok01
18 [Escherichia coli]|PG010208
18 [Escherichia coli]|SCEC2
18 [Escherichia coli UMNK88]|UMNK88
18 [Klebsiella pneumoniae]|
18 [Klebsiella pneumoniae]|Kp7
18 [Salmonella enterica]|AM04528
18 [Salmonella enterica subsp. enterica serovar Kentucky]|1643/10
```

This quick analysis shows that there are 18 consensus orthologous clusters, each having a single sequence from each plasmid/proteome. Figure 1a shows the results of a Venn analysis of the composition of the clusters generated by each of the three clustering algorithms. This figure shows that only the BDBH algorithm detected an additional cluster, as we have learned in previous sections.

3.6 Computing Robust Consensus Pan-Genome Clusters as the Intersection of Homologous Gene Clusters Generated by the COG and OrthoMCL Algorithms, with Pfam-Based Domain Scanning

This exercise is similar to the previous one, except that here we are interested in defining a consensus pan-genome, that is, the set of clusters of any size consistently detected by the COG and OrthoMCL algorithms with Pfam-based domain scanning. To do so we will call the auxiliary `compare_clusters.pl` script with the `-t 0` option, which, as stated before, can only be used with these clustering algorithms, which do not require a reference genome to be included in each cluster. For this very reason they are better suited for computing the pan-genome cluster composition and hence statistically estimate its theoretical size. Issue the

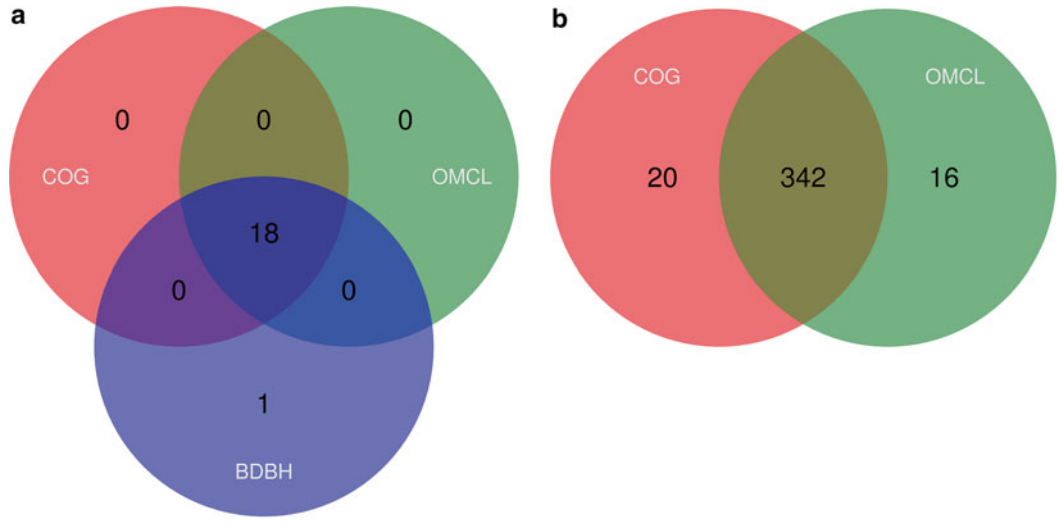

Fig. 1 Venn analyses of the consensus core- (**a**) and pan-genomes (**b**) computed from the intersection of the clusters found by the indicated algorithms

following command from the `pIncAC_homologues/` directory to get the results:

```
$ cd $blast_dir
[$ compare_clusters.pl -d
EscherichiacolistrainSCEC2plasmidpSCEC2NC022377_f0_0taxa_algCOG_Pfam_e0_,EscherichiacolistrainSCEC2plasmidpSCEC2NC022377_f0_0taxa_algOMCL_Pfam_e0_ -o intersect_pan_CM_Dt0 -t 0 -m -T &> log.comp_clusters_intersect_pan_CM_Dt0 &
```

Note that here we are redirecting the script's output to a file named `log.comp_clusters_intersect_pan_CM_Dt0` for later inspection. Notice also the use of the `-m` flag to tell the script that we want it to compute the pan-genome matrix. This is a table containing the presence–absence data for each gene (columns) and proteome/genome (rows). If R [29] is installed on the system, the script will run a Venn analysis and generate the corresponding Venn diagram, shown in Fig. 1b.

From the output saved in `log.comp_clusters_intersect_pan_CM_Dt0` we can see that the COG algorithm yielded 362 pan-genomes clusters, OrthoMCL 358, and 342 were predicted by both as graphically represented in Fig. 1b (*see* **Note 7**). The pan-genome matrix is also provided in PHYLIP format, which can then be used by `parse` (bundled with the GET_HOMOLOGUES package) from the PHYLIP package [30] to compute pan-genomic parsimony trees, as we have shown previously [5, 27] and detailed in the GET_HOMOLOGUES manual. Using the `-T` flag will do this automatically. Figure 2 shows such a pan-genomic parsimony tree depicting the relationships among the 12 pIncA/C plasmids based on the presence–absence matrix of homologous gene clusters. That is, this phylogeny depicts the phylogenetic relationships among plasmids based on their gene content.

3.7 Statistical Estimation of the Theoretical Core- and Pan-Genome Sizes by Fitting Exponential Models (Tettelin and Willenbrock)

Other features of the GET_HOMOLOGUES package that we want to demonstrate herein are its graphical and statistical capabilities, which are based on the powerful statistical and graphical computing environment R [29]. You may recall that in Subheading 3.5 we ran `get_homologues.pl` with the `-c` option enabled. As we will show now, this had the effect of generating three tab-delimited text files called `core_genome*tab` and `pan_genome*tab` found in the `pIncAC_homologues/` directory, where `*` stands for the clustering algorithm used to generate them. These files contain the results of ten sampling experiments, in which genomes are randomly ordered and sequentially added to the pan-genome pool, keeping track of novel genes contributed by each genome (pan) and those already found in previous clusters (core), a strategy first introduced by Tettelin and colleagues in their seminal work on

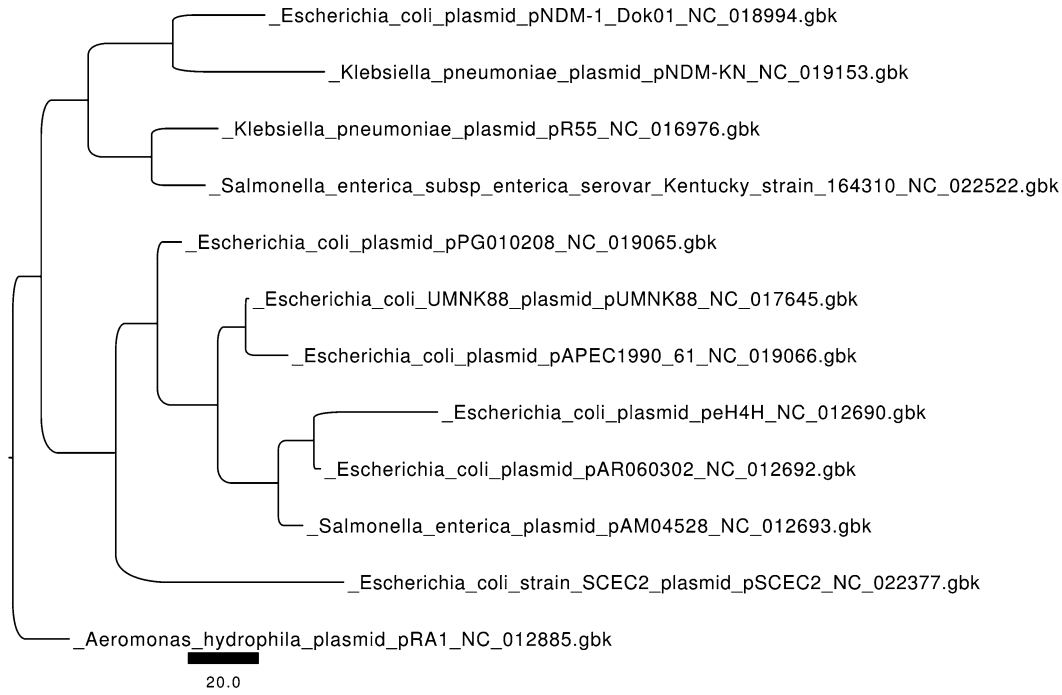

Fig. 2 Pan-genome tree depicting the relationships among pIncA/C plasmids based on the presence–absence pan-genome matrix. The phylogeny was recovered under standard Fitch parsimony and rooted in the reference pRA1 plasmid found in *Aeromonas hydrophila*, a non-enteric gamma-proteobacterium (*Aeromonadales*, *Aeromonadaceae*) strain recovered as a fish pathogen

Streptococcus pan-genomics [3]. These tables can be read by the auxiliary script `plot_pancore_matrix.pl`, which will convert them to R data frames to fit the exponential models of Tettelin et al. [15] and Willenbrock et al. [28]. These models are used to estimate the theoretical size of the core and pan-genomes. The following commands will fit the models and generate the files corresponding to the core- and pan-genome graphs, which are shown in Fig. 3a, b:

```
# find the names of the pancore tab files in pIncAC_homologues/
$ ls *tab
core_genome_algBDBH_Pfam.tab core_genome_algOMCL_Pfam.tab pan_genome_algCOG_Pfam.tab
core_genome_algCOG_Pfam.tab  pan_genome_algBDBH_Pfam.tab pan_genome_algOMCL_Pfam.tab
# visualize the contents of the core and pan-genome size files
# obtained by randomly sampling 10 genomes based on OMCL clustering
```

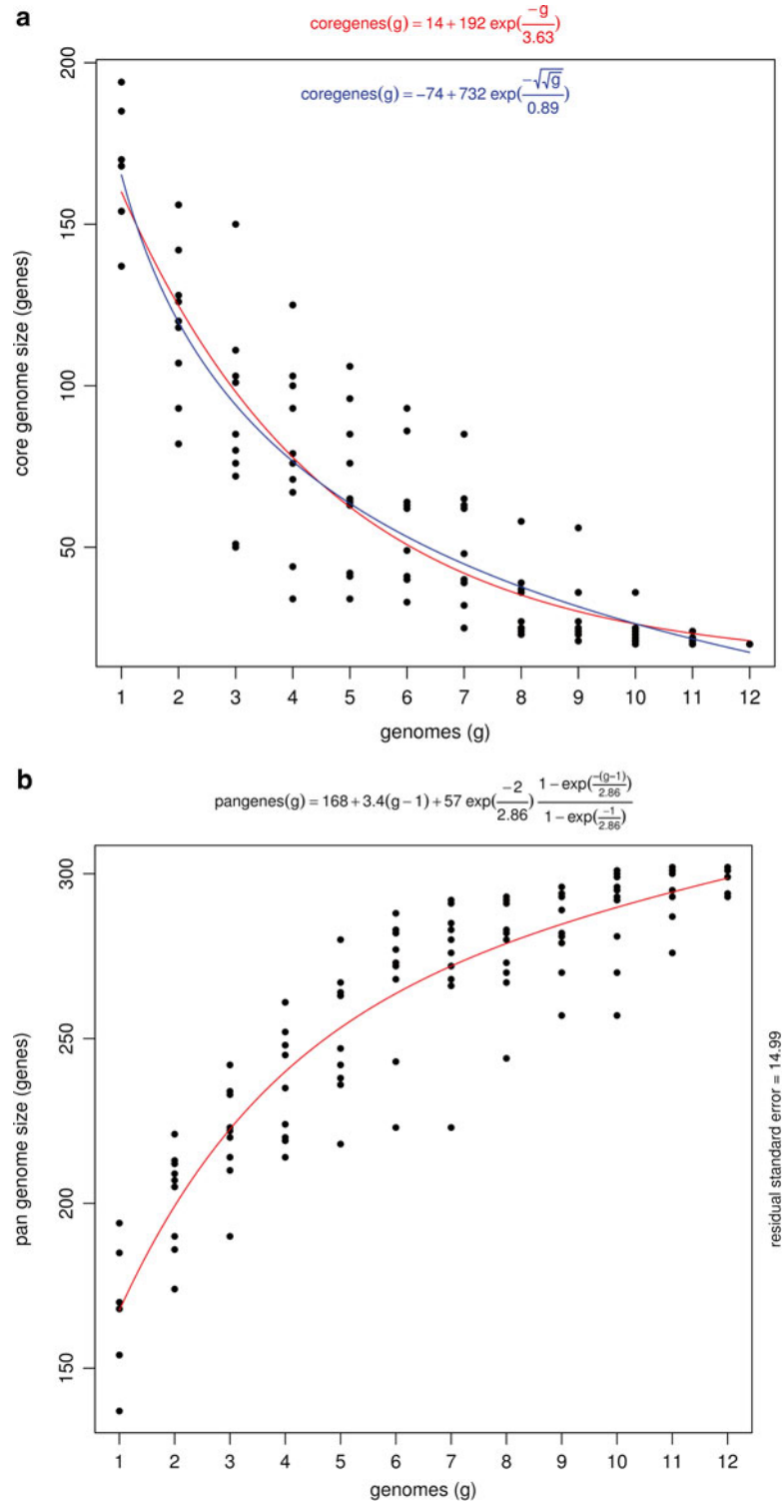

Fig. 3 Statistical estimation and graphical display of core-genome (**a**) and pan-genome (**b**) sizes obtained by fitting exponential functions [3, 28] to resamplings of the core- and pan-genome clusters

```
$ for file in *OMCL*Pfam.tab; do echo "# $file";
cat $file; echo; echo; done
# core_genome_algOMCL_Pfam.tab
g1    g2    g3    g4    g5    g6    g7    g8
g9    g10   g11   g12
154   107   101   100   96    93    65    25
21    21    20    20
168   129   87    61    53    39    38    38
38    38    21    20
161   65    57    55    41    41    24    24
22    22    21    20
… output truncated
# pan_genome_algOMCL_Pfam.tab
g1    g2    g3    g4    g5    g6    g7    g8
g9    g10   g11   g12
154   205   210   219   267   272   272   273
279   293   293   299
168   245   252   256   268   287   287   288
296   297   298   301
… output truncated
# use the *tab files computed based on the OrthoMCL
clustering results to fit
# both the Tettelin and Willenbrock exponential
decay functions to the core genome
# resampling data.
$ plot_pancore_matrix.pl -i core_genome_algOMCL_
Pfam.tab -f core_both
```

Note that due to the random sampling of the proteomes performed to compute the core- and pan-genome sizes, the actual output you get may be somewhat different, particularly in the first columns. The script also generates log files with the details of the statistical analysis. As an example, let us inspect one such file:

```
$ less core_genome_algOMCL.tab_core_both.log
# core_Tettelin fit converged
# residual standard error = 18.33
~ coregenes(g) == "14" + "192" exp(frac(-g,
"3.63"))…
output truncated
# core_Willenbrock fit converged
# residual standard error = 17.88
```

```
~ coregenes(g) == "-74" + "732" exp(frac(-
sqrt(sqrt(g)), "0.89"))
... output truncated
```

Based on the residual standard error, these results show that the Willenbrock model has a slightly better fit than the Tettelin model for this dataset.

3.8 Fitting Mixture Models to Estimate Pan-Genome Sizes and Graphical Analysis of the Pan-Genome Structure

The exponential models fitted to the core- and pan-genome resampling data demonstrated in Subheading 3.7 have been criticized by some authors [14], based on two objections: (1) Exponential models implicitly assume an infinite size for "open" pan-genomes [3, 15] and (2) they also imply that the pan-genome structure basically consists of two "compartments," the universally distributed core-genome genes and the less conserved "accessory genes" that conform the "flexible genome." Although the gene pool available to species with open pan-genomes is certainly impressively large [31], it is not realistic to assume that it is infinite [14]. Further, large-scale comparative genomics studies have consistently revealed that the structure of the microbial pan-genome has certainly more classes than just the core and flexible components [32]. In the latter class the frequency distribution of the taxa in homologous gene clusters varies strongly, but in their seminal work, Koonin and Wolf [32] show that on a coarse scale, the flexible components can be grouped in the shell and cloud components, the latter corresponding to genes present in very few proteomes/genomes of those analyzed.

The auxiliary script `parse_pangenome_matrix.pl` was designed to analyze the structure of the pan-genome, computing and plotting the strict core, relaxed core, shell, and cloud components of the pan-genome. The command lines shown below will illustrate the usage of the `parse_pangenome_matrix.pl` script to graphically explore the structure of the pan-genome of pIncA/C plasmids using the consensus COG and OrthoMCL clusters with Pfam-domain filtering computed in Subheading 3.6. We move into the `intersect_pan_CM_Dt0/` directory and issue the following command:

```
#first cd into the dir holding the consensus COG-
OrthoMCL pangenome
$ cd intersect_pan_CM_Dt0
# Fit mixture model and plot core-cloud-shell
pan-genome composition graphics
# saving the output to the file pan-genome_struc-
ture_analysis.out
$ parse_pangenome_matrix.pl -m pangenome_matrix_
t0.tab -s &> pan-genome_structure_analysis.out
```

The script returns the following files:

```
# find the output files just generated by the script
$ ls -ltr
pangenome_matrix_t0__softcore_list.txt
pangenome_matrix_t0__shell_list.txt
pangenome_matrix_t0__shell_input.txt
pangenome_matrix_t0__core_list.txt
pangenome_matrix_t0__cloud_list.txt
pangenome_matrix_t0__shell_estimates.tab
pangenome_matrix_t0__shell_circle.png
pangenome_matrix_t0__shell_circle.pdf
pangenome_matrix_t0__shell.png
pangenome_matrix_t0__shell.pdf
pan-genome_structure_analysis.out
```

Let us explore the `pangenome_matrix_t0_*_list.txt` files to find both conserved and plasmid-specific genes. In the first category we would expect for example to find the plasmid replication and mobilization genes (*rep* and *tra*). The following code will do the job:

```
# inspect the pangenome_matrix_t0__*_list.txt for
the presence plasmid replication and mobilization
genes
$ egrep 'mob|tra|rep' pangenome_matrix_t0*txt |
egrep -v 'transpo|transcr|trans' | grep core.list
pangenome_matrix_t0__core_list.txt:1238_repA.faa
pangenome_matrix_t0__core_list.txt:1295_DNA_
replication_terminus_site-binding-_Ter_protein.
faa
pangenome_matrix_t0__core_list.txt:1298_traF.faa
pangenome_matrix_t0__core_list.txt:1299_traH.faa
pangenome_matrix_t0__core_list.txt:1300_traG.faa
pangenome_matrix_t0__softcore_list.txt:1263_
traI.faa
pangenome_matrix_t0__softcore_list.txt:1264_
traD.faa
pangenome_matrix_t0__softcore_list.txt:1268_
traB.faa
pangenome_matrix_t0__softcore_list.txt:1269_
traV.faa
pangenome_matrix_t0__softcore_list.txt:1272_
traC.faa
```

```
pangenome_matrix_t0__softcore_list.txt:1273_
traF.faa
pangenome_matrix_t0__softcore_list.txt:1274_
traW.faa
pangenome_matrix_t0__softcore_list.txt:1276_
traU.faa
pangenome_matrix_t0__softcore_list.txt:1277_
traN.faa
… output cut.
```

As expected, most of these genes are part of the core-genome, although some are also part of the shell-genome. There are practical implications for defining such a set of bona fide core-genome sequences. They could for example be used (at the DNA level) to design degenerate PCR primers for the detection, typing, and phylogenetic analysis of pIncA/C plasmids. This task could be very easily performed with the `primers4clades` web server [33, 34]. Another key use of this set of proteins is for phylogenetic analysis

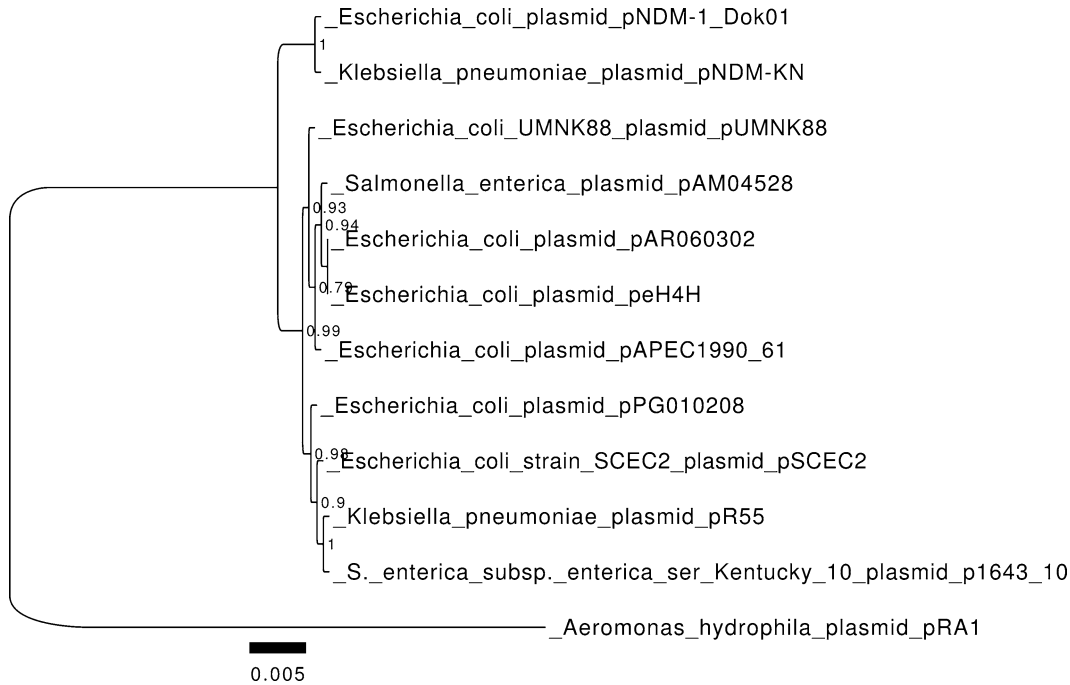

Fig. 4 Maximum likelihood phylogeny of pIncA/C plasmids based on the concatenation of the 18 consensus core-genome computed from the intersection of BDBH, COGtriangles, and OMCL clusters and Pfam domain-scanning enabled. The tree search was performed under the LG matrix with empirical frequencies + proportion of invariant sites + gamma correction of among-site rate variation using the BEST move in PhyML3

to unravel the evolutionary relationships between the plasmids under study and infer the evolutionary pathways that have shaped the final replicons, including the gain and loss of gene clusters. Figure 4 shows a maximum likelihood phylogeny inferred from the concatenation of the 18 strict core loci (*see* **Note 8**).

Now let us interrogate the lists to search for some interesting and famous antimicrobial resistance genes, like beta-lactamases and tetracycline resistance genes (*bla* and *tet* genes):

```
# inspect the pangenome_matrix_t0__*_list.txt for the presence of bla or tet genes
$ egrep 'bla|lactamase|tet|tetracycline' pangenome_matrix_t0__*_list.txt
pangenome_matrix_t0__cloud_list.txt:546_tetA.faa
pangenome_matrix_t0__cloud_list.txt:867_blaNDM-1.faa
pangenome_matrix_t0__cloud_list.txt:870_blaTEM-1.faa
pangenome_matrix_t0__cloud_list.txt:1611_blaOXA-21.faa
pangenome_matrix_t0__shell_list.txt:1252_tetA.faa
pangenome_matrix_t0__shell_list.txt:1253_tetR.faa
```

As expected, the antibiotic resistance genes are part of the cloud and shell gene pools.

Let us now inspect the output from the script, which was redirected to the `pan-genome_structure_analysis.out` file. Files in Linux or Unix systems can be viewed for example with `less pan-genome_structure_analysis.out`. We will focus on the mixture-model analysis section, which is displayed below:

```
# pan-genome size estimates (Snipen mixture model PMID:19691844): pangenome_matrix_t0__shell_estimates.tab
Core.size Pan.size BIC LogLikelihood
2 components 19 343 2836.97081559449 -1409.73319169165
3 components 12 401 1600.53932337316 -785.682634843925
4 components 0 475 1516.28702925272 -737.721677046643
5 components 0 484 1528.05926969358 -737.772986530008
6 components 0 500 1540.93353594771 -738.37530892001
```

```
7 components 0 482 1551.33188934561 -737.7396748819
8 components 0 478 1563.60197249279 -738.039905718426
9 components 0 472 1574.81631547103 -737.812266470482
10 components 0 434 1596.0860933087 -742.612344652257
```

Based on the Bayesian Information Criterion (BIC) of the different components (second column from the right), this analysis shows that the best fit corresponds to a model with 4 components (as it has the lowest BIC value), followed by that with 5 components, at a distance of 11.7 AIC units (*see* **Note 9**). This analysis therefore strongly suggests that there are more than just two pan-genome components, which is consistent with the graphical analysis of cluster-size frequency distribution shown in Fig. 5a, b. The size of the pan-genome is estimated to be around 475 genes. The consensus core-genome size is estimated to be much smaller, around 0 genes, which clearly seems a strong underestimation. These results highlight the importance of refining all models to find more realistic and useful core- and pan-genome size estimates.

3.9 Identification of Lineage-Specific Genes in Consensus Pan-Genome Matrices Using parse_pangenome_matrix.pl

The `parse_pangenome_matrix.pl` script was designed to perform basic comparative genomics tasks. It can be used to compare two pan-genome sets to identify lineage-specific genes and lineage-specific gene expansions in one subset (A), as compared to the other one (B). From the inspection of the `pangenome_matrix_t0__cloud_list.txt` file we did in the previous section, we found that the bla_{NDM} genes were part of the cloud-genome. It is trivial to find the plasmids that contain them, using the following grep command:

```
# find the plasmids containing the NDM-1 genes
$ grep '>' 867_blaNDM-1.faa
>GI:410502926 |[Escherichia coli]|NDM-1 Dok01|blaNDM-1|NA|NC_018994(195560):139825-140637:-1 ^,GeneID:13876866^ Escherichia coli plasmid pNDM-1_Dok01, complete sequence.|neighbours:GI:410502925(-1),GI:410502927(-1)|neighbour_genes:hypothetical protein,IS903 transposase|
>GI:410656145 |[Klebsiella pneumoniae]|Kp7|NDM-1|NA|NC_019153(162746):108108-108920:-1 ^,GeneID:13914405^ Klebsiella pneumoniae plasmid pNDM-KN, complete sequence.|neighbours:GI:410656144(-1),GI:410656146(-1)|neighbour_genes:bleMBL,insertion element ISKpn14|
```

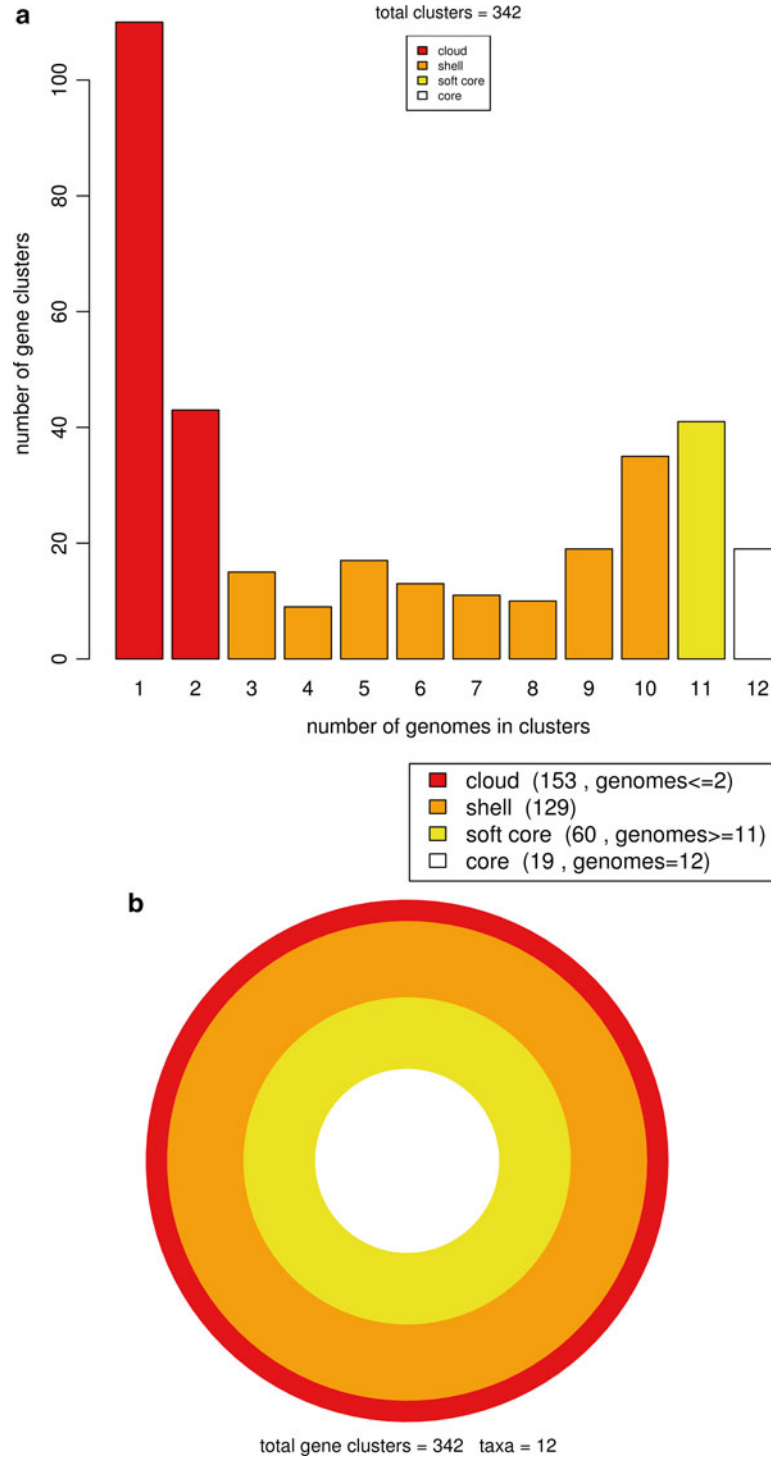

Fig. 5 Graphical analysis of the structure of the pIncA/C pan-genome protein space. Panel **a** depicts a bar plot showing the absolute size frequencies of orthologous clusters as predicted by the OMCL algorithm. Panel **b** shows a circle plot depicting the relative sizes (cluster numbers) contained in the core, soft-core, shell, and cloud genomes

This makes clear that only two plasmids contain the genes. We can now generate two lists of plasmid genomes: list A will contain the names of the GenBank files containing the *bla*$_{\text{NDM-1}}$ genes, and list B the names of the rest of the files. Generate such lists with the following code, working within the directory holding the `*gbk` files (`pIncAC/`):

```
# 1. Generate the lists of genomes to be compared
for lineage specific genes (in list A vs. B)
panGmat_dir=$(pwd)
cd $gbk_dir
$ ls *gbk | grep pNDM > listA_pNDB
$ ls *gbk | grep -v pNDM > listB_nonNDB
$ cd $panGmat_dir
```

Now we are ready to run to find the genes specific to the "A" list of plasmids:

```
# 2. parse the pangenome matrix file to find the
listA-specific genes
$ parse_pangenome_matrix.pl -A $gbk_dir/listA_
pNDM -B $gbk_dir/listB_nonNDB -g -m pangenome_
matrix_t0.tab   -p   _Escherichia_coli_plasmid_
pNDM1_Dok01_NC_018994
```

Now we can inspect the output file's content to see how many and which are the genes that are found only in the pIncA/C plasmids containing the blaNDM genes:

```
$ cat
pangenome_matrix_t0__Escherichia_coli_plasmid_
pNDM1_Dok01_NC_018994_pangenes_list.txt
# genes present in set A and absent in B (19):
846_armA.faa
862_groES.faa
863_hypothetical_protein.faa
864_hypothetical_protein.faa
865_trpF.faa
866_hypothetical_protein.faa
867_blaNDM-1.faa
879_Rhs_family_protein.faa
883_Tn7-like_transposition_protein_A.faa
884_Tn7-like_transposition_protein_B.faa
885_Tn7-like_transposition_protein_C.faa
886_hypothetical_protein.faa
888_type_I_site-specific_deoxyribonuclease-
_HsdR_family.faa
```

```
889_hypothetical_protein.faa
890_hypothetical_protein.faa
891_putative_type_I_restriction-modification_
system_restriction_subunit.faa
892_hypothetical_protein.faa
893_type_I_restriction-modification_system-_M_
subunit.faa
894_hypothetical_protein.faa
```

The sequential numbering of several genes (862–867 and 883–894) suggests that most of the list "A"-specific genes are clustered in two regions. The first one, containing the blaNDM-1 gene, also contains the well-known proteins GroES and TrpF. The first one is a component of the GroEL-GroES chaperonin complex. The *groS* gene is one of a network of 93 genes believed to play a role in promoting the stress-induced mutagenesis (SIM) response of *E. coli* K-12 (for more details see http://ecocyc.org/ECOLI/NEW-IMAGE?type=GENE&object=EG10600). TrpF (synonym of TrpC) is a bifunctional phosphoribosylanthranilate isomerase/indole-3-glycerol phosphate synthase. It carries out the third and fourth steps in the tryptophan biosynthesis pathway (for more details see http://ecocyc.org/ECOLI/NEW-IMAGE?type=GENE&object=EG11026). It is certainly somewhat surprising to find these two genes on a resistance plasmid. Readers interested in more details about these interesting findings are referred to the original publications describing the two NDM plasmids used in this chapter [16, 19].

3.10 Conclusions and Perspectives

In this chapter we have demonstrated some of the capabilities of the GET_HOMOLUGUES software, focusing in the detection of orthologs, the statistical evaluation and graphical analysis of the core- and pan-genome compartments, and the detection of lineage-specific genes in the pan-genome matrix. These features demonstrate the flexibility and robustness of the software, and highlight its ease of use. There are several other interesting features, such as the analysis of syntenic intergenic regions, the use of the synteny criterion to define orthologs, and the use of the BerkeleyDB system to trade speed for RAM when analyzing very large genomic datasets, which are well documented in the manual and have been published elsewhere [5, 27]. Altogether these features make GET_HOMOLOGUES a useful, versatile, flexible, and powerful piece of software that allows nonspecialists to make rigorous and detailed analyses of microbial pan-genomics and comparative genomics. Future development of the software will focus on including more statistical analyses and expanding its graphical capabilities.

4 Notes

1. The set of 12 GenBank files used in this chapter were downloaded from NCBI's RefSeq database [6] and further processed using the following protocol. Point your browser to the URL http://www.ncbi.nlm.nih.gov/nuccore/ and type the following query string into the text box: "`incA/C[text] AND plasmid[titl] AND complete sequence[titl] AND 90000[SLEN]: 200000[SLEN] AND srcdb_refseq _known[PROP]`." This will search for pIncA/C plasmids in NCBI's RefSeq database. The results are displayed in the summary format. In the upper right corner click "Send to ->File; Format ->Accesion List" and save the list of RefSeq accession numbers to the working directory on your hard drive with the name accNo.list. To fetch the actual GenBank files cd into the directory holding your `accNo.list` file (we will use the directory name `pIncAC/`herein) and type the following shell one-liner on your command prompt (all in one line):

   ```
   $ for acc in $(cat accNo.list); do accBase=$(cut -d\. -f1); wget -c

   ftp://ftp.ncbi.nlm.nih.gov/genomes/Plasmids/${accBase}.gbk; done
   ```

   This should fetch the desired GBK files. If you wish, you can rename those files with the file's DEFINITION line using the auxiliary shell script `rename_gbk_files_with_DEFINITION_line.sh`

   These simple scripts are bundled with the `*tgz` file mentioned in **step 1** of Subheading 2.

2. This is assuming that you have added the directory containing the distribution to your PATH variable (as explained in the manual bundled with the package). Otherwise you will need to precede the program name with the full path, like $HOME/path/to/get_homologues_XXX/get_homologues.pl.

3. Get_homologues.pl can also work with the genome's faa or ffn files, that is, the fasta files in for the CDSs in protein or nucleotide version, respectively. Please check the manual for all accepted combinations of input formats. It should be noted that specialized functionality like the extraction of orthologous intergenic spacers or the use of the synteny criterion to filter orthologs will not work here, as the software relies on the GenBank annotations to determine the identity of the neighboring genes. See the manual for more details.

4. The nohup (no hang-up) command allows a second command provided as argument to be executed even after you exit from a shell session. This is very useful when you are running large jobs on a server. You issue your command and can log out of the session

without killing your process. The `&> log.get_homologues_pIncAC_BDBH_C75D_allTaxa &` syntax tells the shell to redirect the standard output and standard error streams to the log.get_homologues_pIncAC_BDBH_C75D_allTaxa file, while the last ampersand asks the shell to run the whole process in the background. Finally, the `log.get_homologues_pIncAC_BDBH_C75D_allTaxa` command allows us to continuously follow the last ten lines of the growing log file. A CTRL-C will close (kill) the tail command to exit from it. Then execute the file instructions calling bash with the following command: `bash get_homol_batch_pIncAC.cmd`. After issuing this command, you can log out of your session, if you wish. The script will run in the background, calling get_homologues.pl sequentially to run the three clustering algorithms.

5. The latest version of Pfam-A domain database can be downloaded from the Sanger ftp site during the package installation process. The database will be automatically formatted with hmmpress during the installation process, making it ready to use (see the db/directory within your get_homologues.X.Y./directory).

6. It is convenient to save complex command lines like this to a file for later reference or even use them as a template to create similar commands for other datasets. Open an editor and type or paste the code reproduced below

```
nohup get_homologues.pl -d pIncAC -G -n 2 -t 0 &> log.get_homologues_pIncAC_Gn2t0 && get_homologues.pl -d pIncAC -M -n 2 -t 0 -c &> log.get_homologues_pIncAC_Mn26t0 && get_homologues.pl -d pIncAC -n 2 &> log.get_homologues_pIncAC_BDBHn2 &
```

   and name the file `get_homol_batch_pIncAC.cmd`. The command file can then be executed with this simple line:

```
$ bash get_homol_batch_pIncAC.cmd.
```

7. We have found that the COGtriangles clustering algorithm will consistently generate a larger number of unique clusters than the OMCL algorithm [5]. Most of these COG-specific clusters are actually singletons, consisting of single or pairs of proteins that were not merged into a proper cluster because at least three proteins from distinct organisms/proteomes are required to form a COG triangle [9, 35].

8. The individual clusters were aligned using `muscle` as in Subheading 3.8 [36] under default parameter values with the following command (assumes that muscle is installed on the system and in `PATH`):

```
$ for file in *faa; do muscle < $file > ${file%faa}_musAln.FAA; done
```

   The original ordering of the strains in the alignments was reestablished and the alignments concatenated. The concatenated

alignment was then subjected to a maximum-likelihood tree search using PhyML3 [37] under the LG model, estimating amino-acid frequencies, proportion of invariant sites, and the shape parameter of the gamma distribution to model among-site rate variation. The search was started from a BioNJ tree using the BEST moves algorithm. The tree was visualized and edited with FigTree [38].

9. With the R package "qpcR" it is very easy to compute Akaike weights. Simply generate a vector of AIC values, here called AIC.vals, and pass it to the function `akaike.weights()`. For more information, see for example http://www.inside-r.org/packages/cran/qpcR/docs/akaike.weights

   The R commands and output are shown below:

```
# call library qpcR
> library(qpcR)
# create a vector with the AIC values, in this case the three best ones (those with 3, 4 and 5 components, respectively) from the mixture model analysis in section 3.8
> AIC.vals<-c(1600.53932337316, 1516.28702925272, 1528.05926969358)
# pass the vector AIC.vals to the akaike.weights function.
> akaike.weights(AIC.vals)
$deltaAIC
[1] 84.25229 0.00000 11.77224
$rel.LL
[1] 5.068119e-19 1.000000e+00 2.777733e-03
$weights
[1] 5.054080e-19 9.972300e-01 2.770038e-03
```

   The output on the last line shows that the model with four classes has a relative weight of >99 % and second best (five components) a marginal 0.027 %, making the four-class model the clear winner.

Acknowledgements

We thank Romualdo Zayas, Víctor del Moral, and Alfredo J. Hernández at CCG-UNAM for technical support. We also thank David M. Kristensen and the development team of OrthoMCL for permission to use their code in our project. Funding for this work was provided by the Fundación ARAID, Consejo Superior de Investigaciones Científicas (grant 200720I038), DGAPA-PAPIIT UNAM-México (grant IN211814), and CONACyT-México (grant 179133).

References

1. Pagani I, Liolios K, Jansson J et al (2012) The Genomes OnLine Database (GOLD) v. 4: status of genomic and metagenomic projects and their associated metadata. Nucleic Acids Res 40:D571–D579
2. Welch RA, Burland V, Plunkett G 3rd et al (2002) Extensive mosaic structure revealed by the complete genome sequence of uropathogenic *Escherichia coli*. Proc Natl Acad Sci U S A 99:17020–17024
3. Tettelin H, Masignani V, Cieslewicz MJ et al (2005) Genome analysis of multiple pathogenic isolates of *Streptococcus agalactiae*: implications for the microbial "pan-genome". Proc Natl Acad Sci U S A 102:13950–13955
4. Mira A, Martin-Cuadrado AB, D'Auria G et al (2010) The bacterial pan-genome: a new paradigm in microbiology. Int Microbiol 13:45–57
5. Contreras-Moreira B, Vinuesa P (2013) GET_HOMOLOGUES, a versatile software package for scalable and robust microbial pangenome analysis. Appl Environ Microbiol 79:7696–7701
6. Tatusova T, Ciufo S, Fedorov B et al (2014) RefSeq microbial genomes database: new representation and annotation strategy. Nucleic Acids Res 42:D553–D559
7. Camacho C, Coulouris G, Avagyan V et al (2009) BLAST+: architecture and applications. BMC Bioinformatics 10:421
8. Eddy SR (2009) A new generation of homology search tools based on probabilistic inference. Genome Inform 23:205–211
9. Kristensen DM, Kannan L, Coleman MK et al (2010) A low-polynomial algorithm for assembling clusters of orthologous groups from intergenomic symmetric best matches. Bioinformatics 26:1481–1487
10. Li L, Stoeckert CJ Jr, Roos DS (2003) OrthoMCL: identification of ortholog groups for eukaryotic genomes. Genome Res 13:2178–2189
11. Altenhoff AM, Dessimoz C (2012) Inferring orthology and paralogy. Methods Mol Biol 855:259–279
12. Kristensen DM, Wolf YI, Mushegian AR et al (2011) Computational methods for gene orthology inference. Brief Bioinform 12:379–391
13. Wolf YI, Koonin EV (2012) A tight link between orthologs and bidirectional best hits in bacterial and archaeal genomes. Genome Biol Evol 4:1286–1294
14. Snipen L, Almoy T, Ussery DW (2009) Microbial comparative pan-genomics using binomial mixture models. BMC Genomics 10:385
15. Tettelin H, Riley D, Cattuto C et al (2008) Comparative genomics: the bacterial pan-genome. Curr Opin Microbiol 11:472–477
16. Carattoli A, Villa L, Poirel L et al (2012) Evolution of IncA/C blaCMY-(2)-carrying plasmids by acquisition of the blaNDM-(1) carbapenemase gene. Antimicrob Agents Chemother 56:783–786
17. Fricke WF, Welch TJ, McDermott PF et al (2009) Comparative genomics of the IncA/C multidrug resistance plasmid family. J Bacteriol 191:4750–4757
18. Johnson TJ, Lang KS (2012) IncA/C plasmids: an emerging threat to human and animal health? Mob Genet Elements 2:55–58
19. Sekizuka T, Matsui M, Yamane K et al (2011) Complete sequencing of the bla(NDM-1)-positive IncA/C plasmid from *Escherichia coli* ST38 isolate suggests a possible origin from plant pathogens. PLoS One 6:e25334
20. Poirel L, Hombrouck-Alet C, Freneaux C et al (2010) Global spread of New Delhi metallo-beta-lactamase 1. Lancet Infect Dis 10:832
21. Nordmann P, Poirel L, Walsh TR et al (2011) The emerging NDM carbapenemases. Trends Microbiol 19:588–595
22. Poirel L, Bonnin RA, Nordmann P (2011) Analysis of the resistome of a multidrug-resistant NDM-1-producing *Escherichia coli* strain by high-throughput genome sequencing. Antimicrob Agents Chemother 55:4224–4229
23. Moellering RC Jr (2010) NDM-1 – a cause for worldwide concern. N Engl J Med 363: 2377–2379
24. Finn RD, Tate J, Mistry J et al (2008) The Pfam protein families database. Nucleic Acids Res 36:D281–D288
25. Sonnhammer EL, Koonin EV (2002) Orthology, paralogy and proposed classification for paralog subtypes. Trends Genet 18: 619–620
26. Forslund K, Pekkari I, Sonnhammer EL (2011) Domain architecture conservation in orthologs. BMC Bioinformatics 12:326
27. Vinuesa P, Contreras-Moreira B (2014) Pangenomic analysis of the *Rhizobiales* using the GET_HOMOLOGUES software package. In: De Bruijn FJ (ed) Biological nitrogen fixation 7. Wiley/Blackwell, Hoboken, NJ
28. Willenbrock H, Hallin PF, Wassenaar TM et al (2007) Characterization of probiotic

Escherichia coli isolates with a novel pan-genome microarray. Genome Biol 8:R267

29. R Development Core Team (2012) R: a language and environment for statistical computing. http://www.R-project.org. Vienna, Austria
30. Felsenstein J (2004) PHYLIP (phylogeny inference package). In: Distributed by the author. Department of Genetics, University of Washington, Seattle
31. Kaas RS, Friis C, Ussery DW et al (2012) Estimating variation within the genes and inferring the phylogeny of 186 sequenced diverse *Escherichia coli* genomes. BMC Genomics 13:577
32. Koonin EV, Wolf YI (2008) Genomics of bacteria and archaea: the emerging dynamic view of the prokaryotic world. Nucleic Acids Res 36: 6688–6719
33. Contreras-Moreira B, Sachman-Ruiz B, Figueroa-Palacios I et al (2009) primers4clades: a web server that uses phylogenetic trees to design lineage-specific PCR primers for metagenomic and diversity studies. Nucleic Acids Res 37:W95–W100
34. Sachman-Ruiz B, Contreras-Moreira B, Zozaya E et al (2011) Primers4clades, a web server to design lineage-specific PCR primers for gene-targeted metagenomics. In: de Bruijn FJ (ed) Handbook of molecular microbial ecology I: metagenomics and complementary approaches. Wiley/Blackwell, Hoboken, NJ, pp 441–452
35. Tatusov RL, Koonin EV, Lipman DJ (1997) A genomic perspective on protein families. Science 278:631–637
36. Edgar RC (2004) MUSCLE: multiple sequence alignment with high accuracy and high throughput. Nucleic Acids Res 32:1792–1797
37. Guindon S, Dufayard JF, Lefort V et al (2010) New algorithms and methods to estimate maximum-likelihood phylogenies: assessing the performance of PhyML 3.0. Syst Biol 59: 307–321
38. Rambaut A (2009) FigTree v1.4.0. Available from http://tree.bio.ed.ac.uk/software/figtree/

Chapter 15

Genome-Scale Metabolic Network Reconstruction

Marco Fondi and Pietro Liò

Abstract

Bacterial metabolism is an important source of novel products/processes for everyday life and strong efforts are being undertaken to discover and exploit new usable substances of microbial origin. Computational modeling and in silico simulations are powerful tools in this context since they allow the exploration and a deeper understanding of bacterial metabolic circuits. Many approaches exist to quantitatively simulate chemical reaction fluxes within the whole microbial metabolism and, regardless of the technique of choice, metabolic model reconstruction is the first step in every modeling pipeline. Reconstructing a metabolic network consists in drafting the list of the biochemical reactions that an organism can carry out together with information on cellular boundaries, a biomass assembly reaction, and exchange fluxes with the external environment. Building up models able to represent the different functional cellular states is universally recognized as a tricky task that requires intensive manual effort and much additional information besides genome sequence. In this chapter we present a general protocol for metabolic reconstruction in bacteria and the main challenges encountered during this process.

Key words Metabolic model reconstruction, Flux balance analysis, Metabolic modeling

1 Introduction

One of the most important drawbacks derived from the booming of genomics resides in the possibility to (almost) automatically derive the potential metabolic landscape of a strain, given its genome. This is of particular importance when dealing with biotechnologically or clinically relevant strains since metabolism represents a key factor for understanding their physiology. In general, living organisms possess complex metabolic networks, ranging from hundreds to thousands of chemical reactions and conferring them the capability to synthesize and/or catabolize the building blocks of their cells. The sum of these chemical reactions represents the core of any living organism and the coordination of these processes results in the physiology we associate to each organism, from bacteria to humans [1]. Bacteria, in particular, continuously provide industry with novel products/processes based on the use of their metabolism and numerous efforts are being undertaken

Alessio Mengoni et al. (eds.), *Bacterial Pangenomics: Methods and Protocols*, Methods in Molecular Biology, vol. 1231, DOI 10.1007/978-1-4939-1720-4_15,

worldwide, with an ultimate goal to deliver new usable substances of microbial origin to the marketplace [2], including pharmaceuticals, biofuels, and bioactive compounds in general. Classical examples of industrial bio-based production of valuable compounds include vitamin C [3], xanthan (E425) [4], isopropanol, butanol and ethanol mixture [5], and succinate [6].

The importance of bio-based products in everyday life has tremendously boosted research on microbial metabolic processes and understanding the basic functioning of the biosynthetic circuits of living cells has become a crucial issue in systems microbiology. In this context, computational modeling and in silico simulations are often adopted by metabolic engineers to quantitatively simulate chemical reaction fluxes within the whole microbial metabolism [7, 8]. Among possible approaches, the so-called constraint-based methods (e.g., flux balance analysis, FBA, [9]) can be applied to large (genome-scale) biochemical systems since they require only the information on metabolic reaction stoichiometry and mass balances around the metabolites under pseudo-steady-state assumption [10]. Thus, according to this methodology, detailed information on the chemical equations of the studied system is not required. Genome-scale metabolic modeling has become an important tool in the study of metabolic networks in pathogens [11], and chemical [3] and environmental [11] research areas. Methods and tools for in silico metabolic modeling have been recently reviewed in [12], [13], and [14, 15], respectively.

To exploit computational approaches, cellular metabolic networks are transformed into a model by drafting the list of the biochemical reactions that an organism can carry out together with the boundaries of the system, a biomass assembly reaction, and exchange fluxes with the environment [16]. These reconstructions account for the functions of hundreds to thousands of genes, and are ideally intended to incorporate all known metabolic reactions for a particular organism into a standardized format, enabling the generation of a computational model that can be analyzed with a variety of emerging mathematical techniques [8]. Constraint-based modeling framework can be used to automatically compute the resulting balance of all the chemical reactions predicted to be active in the cell and, in turn, to bridge the gap between knowledge of the metabolic network structure and observed metabolic phenotypes.

The process of reconstructing and validating a metabolic model is a complex task. Currently, about 4,000 complete genome sequences are available in public databases (www.genomesonline.org); conversely, only around 100 reconstructions of microbial metabolic systems can be retrieved (see http://systemsbiology.ucsd.edu/InSilicoOrganisms/OtherOrganisms for an updated list). This gap is the most evident consequence of the difficulties in

reconstructing "working" metabolic models starting from genome annotations and is a key challenge for future systems microbiology.

Drafting a metabolic model of an organism nowadays is almost straightforward since many tools able to make this step automatic are available [17–21] (described in details below); however, turning these reconstructions into models capable of fully representing the functional states of a given organisms is not trivial. In particular, most of the draft metabolic available to date are incomplete because (1) they often do not include essential metabolic steps for sustaining in silico cellular growth (metabolic gaps) and (2) key issues of embedded reactions such as stoichiometry, directionality, and charge are sometimes missing or erroneous. Moreover, since draft models are mainly reconstructed on the basis of sequence homology in respect to other (closely related) microorganisms, they will not include organism-specific metabolic pathways (often responsible for key phenotypic features). To overcome these difficulties and guide model revision, Thiele and Palsson [22, 23] have built a protocol including (at least) 4 stages and 94 different steps necessary for the reconstruction of reliable, high-quality, metabolic models. Importantly, a large fraction of these steps cannot be performed in an automated fashion, thus requiring intense and time-consuming manual effort/curation. Speeding up some of the steps represents one of the most important achievements required for accelerating the overall process of metabolic modeling and engineering of microbial strains.

This chapter is intended as a general protocol for bacterial metabolic model reconstruction and will describe the main steps encountered during this process (using FBA as the modeling framework). Further details and specific challenges of each step herein described can be found in previous (and more detailed) papers [22–24].

2 Materials

In this section we describe what is needed for starting to reconstruct the metabolic model of a strain under study.

2.1 Genome Sequence

Genome sequence is nowadays the most widely adopted resource for drafting the metabolic model of an organism. So, to start the reconstruction you will need a FASTA file embedding a set of contigs or coding sequences of the genome you want to analyze.

2.2 Online Reconstruction Tools

A number of tools exist for drafting the metabolic model of a given organism (Table 1).

1. The coupling of *RAST* and *Model SEED* pipelines provides a fully automated annotation and model reconstruction service

Table 1
List of software/methods for automatic metabolic reconstructions

Name	Reference	Website	Standalone version	Free (F)/ commercial (C)
RAST/Model Seed	[17, 25, 50]	http://rast.nmpdr.org/, http://www.theseed.org/	Not available	F
MicrobesFlux	[19]	http://tanglab.engineering.wustl.edu/static/MicrobesFlux.html	Not available	F
FAME	[51]	http://f-a-m-e.org/	Not available	F
Pathway Tools Software	[20, 28]	http://bioinformatics.ai.sri.com/ptools/	Available	F/C
COPABI	[21]	–	Not available	F
CARMEN	[18]	http://carmen.cebitec.uni-bielefeld.de		F
Kbase	–	www.kbase.us	Available	F
GEMSiRV	[34]	http://sb.nhri.org.tw/GEMSiRV/en/GEMSiRV	Available	F
RAVEN	[35]	http://www.sysbio.se/BioMet	Available	F
Metashark	[33]	http://bioinformatics.leeds.ac.uk/shark/	Available	F
SuBliMinaL Toolbox	[36]	http://www.mcisb.org/resources/subliminal/	Available	F

for archaeal and bacterial genomes. The service seeks to rapidly produce high-quality assessments of gene functions and, most importantly in the context of the present chapter, an initial (draft) metabolic reconstruction [17]. Each preliminary model network includes all reactions associated with one or more enzymes encoded in the organism's genome as well as a set of spontaneous reactions that do not require enzymatic catalysis [17, 25]. Importantly, Model SEED also provides tools for preliminary analysis of reconstructed metabolic networks, including auto-filling of metabolic gaps and FBA of the model. Overall, about ~48 h is necessary to reconstruct a metabolic model from an assembled genome sequence.

2. *MicrobesFlux* is a platform to build metabolic models for all the organisms whose completely sequenced genome is present in the KEGG (Kyoto Encyclopedia of Genes and Genomes) database. Indeed, this tool is able to automatically download the metabolic network (including enzymatic reactions and metabolites) of ~1,200 species from the KEGG database and then convert it to a metabolic model draft.

3. The Flux Analysis and Modeling Environment (*FAME*) is a web-based modeling tool that allows the reconstruction of metabolic models. In addition this tool also includes other tasks, such as editing, running, and analyzing/visualizing stoichiometric models. FAME allows users to either upload their own preexisting model or to build a new model. To perform this latter task, this software takes advantage of the metabolic models stored in the KEGG database [26]. Importantly, FAME is quite flexible, allowing any stoichiometric model to be loaded into FAME, provided it is encoded in the Systems Biology Markup Language (SMBL, see below).
4. Similarly, the software tool *CARMEN* [18] performs in silico reconstruction of metabolic networks to help translate genomic data into functional ones. CARMEN also enables the visualization of automatically derived metabolic networks based on pathway information from the KEGG database [26] or from user-defined SBML templates.
5. *Pathway Tools* [20, 27, 28] is a software environment for management, analysis, and visualization of integrated collections of genome, pathway, and regulatory data. This tool can be used for de novo genome-scale model generation and also for other post-processing tasks such as interactive editing, visualization, and comparative analyses. Recently, PathwayTools has been used for a systematic comparison between KEGG and MetaCyc [29, 30] databases, revealing differences in the two repositories in that KEGG contains significantly more compounds than does MetaCyc, whereas MetaCyc contains more reactions and pathways than does KEGG; in particular KEGG modules are quite incomplete [31].
6. A Computational Platform for the Access of Biological Information (*COPABI*) has been recently developed by Reyes et al. [21]. This platform allows the automation of a methodology for the reconstruction of genome-scale metabolic models for any organism. The algorithm comprises several steps including (1) the information compilation from free-access biological databases, (2) interaction of the user with the platform in order to properly select the parameters for the probabilistic criteria and choices for the biomass components and restrictions, and finally (3) application of unicity and completeness criteria and production of the output. Unicity criterion aims at identifying reactions that appear more than once and also identifies their enzymes. Repeated reactions are then eliminated following the criterion according to which the enzyme that appears less frequently in the model is not eliminated. Completeness aims at adding novel reactions to the model in order to fill the gaps that are commonly found in the draft reconstruction process.

7. The metabolic SearcH And Reconstruction Kit (*metaSHARK*) [32, 33] is a new fully automated software package for the detection of enzyme-encoding genes within unannotated genome data and their visualization in the context of the surrounding metabolic network. Unlike most of the previously described reconstruction tools that start with a set of predicted proteins from an annotated genome and that, by a variety of text mining and/or sequence comparison methods, construct a list of the enzymatic functions that are asserted to be present, metaSHARK only requires a set of DNA sequences [finished chromosomes, contigs, genome survey sequences, or expressed sequence tags (ESTs)] as input, and hence can be applied to extract new knowledge of metabolic capabilities from preliminary data produced by unannotated and ongoing genome sequencing projects.
8. The recently proposed *KBase* (http://kbase.science.energy.gov/) is a software environment designed to enable researchers to reconstruct, optimize, and analyze genome-scale metabolic models. Genome-scale metabolic models are reconstructed starting from an annotated genome object using the DOE Systems Biology Knowledgebase tools.
9. The metabolic network reconstruction module of *GEMSIRV* (GEnome-scale Metabolic model Simulation, Reconstruction and Visualization) [34] allows editing/updating the content of a model that has been previously imported (in SBML or spreadsheet format), using other models of closely related species as a guide. Alternatively, a draft reconstruction can be generated by mapping a blank reconstruction (containing gene information only) to a reference reconstruction.
10. *Raven* (Reconstruction, Analysis, and Visualization of Metabolic Networks) [35] is a tool for automatic reconstruction of GEMs based on protein orthology and (optionally) on a set of already available genome-scale metabolic models. The method takes advantage of the KEGG Orthology (KO) IDs for inferring gene-protein-reaction association.
11. *SuBliMinaL* toolbox [36] is a collection of methods enabling the automated reconstruction of genome-scale metabolic models, exploiting both KEGG and MetaCyc resources. In the generated model, all the metabolic pathways described in each resource are merged and can be used for the successive pipeline step (annotation), in which already existing reconstructions can be used for improving the de novo-reconstructed model.

2.3 SBML-Formatted Metabolic Model

SBML (Systems Biology Markup Language) is a software-independent language for describing different biological processes and is nowadays considered the standard medium for representation

and exchange of biochemical network models [37]. In its general formulation, it resembles the basic features of the XML data stream [38] and allows representing all the elements accounting for biochemical reactions, including (1) the cellular compartment(s) in which the reaction occurs, (2) chemical species involved (substrates and products), (3) the reversibility (or irreversibility) of each reaction, and (4) unit definition (according to which quantities of substrates and/or products that are consumed and/or produced are expressed). A simple model (seven compounds, one reaction) together with its SBML counterpart is reported in Fig. 1. This model represents the ATP-dependent transport of (periplasmic) D-glucose into the cellular cytoplasm according to the iAF1260 metabolic reconstruction of *Escherichia coli* [39]. As shown in Fig. 1b, this SBML representation can be divided into three main sections: the first part (black font) includes general details on the reconstructed model such as the organism (*E. coli* iAF1260), the unit definition (mmol/gDW h), and the model compartments (extracellular, periplasm, cytosol). The second section (blue font) includes the list of the chemical species that the model will be able to recognize and handle (together with their name, formula, charge and boundary condition). The last section of this small model (red font) lists all the possible biochemical transformations

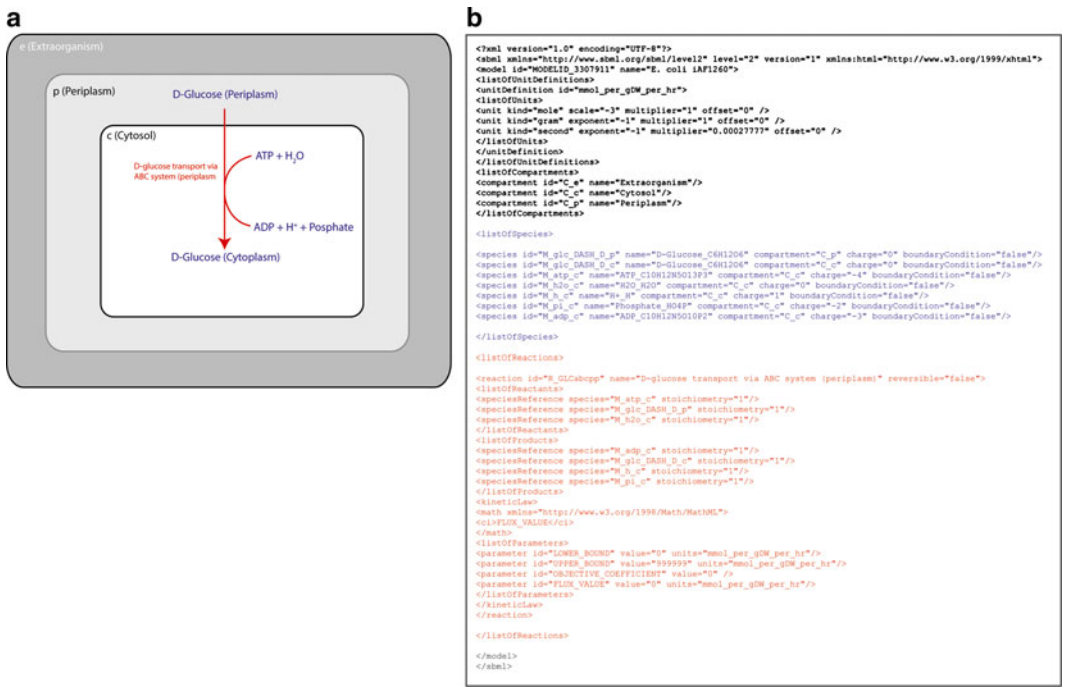

Fig. 1 Schematic representation of a one-reaction metabolic model in a compartmentalized cell (**a**) and the corresponding SMBL code (**b**)

(only "D-glucose transport via ABC system" in this case) together with information on the reversibility of the reactions, the stoichiometric coefficients of substrates/products, and reaction bounds (upper and lower, UB and LB, respectively).

2.4 Modeling Framework

There are several available tools for performing constraint-based metabolic modeling (reviewed in [24]). SBML-formatted models are generally recognized by these tools and can be imported/converted for successive computation. Among them, COBRA toolbox [40] is probably the most widely adopted. The original version of this package is to be used within the Matlab (The Mathworks Inc.) numerical computation and visualization environment although, recently, a version exploiting Python programming language has been developed (COBRApy, [41]). Command lines reported in this protocol refer to the Matlab-based version of COBRA toolbox. When available, their COBRApy counterpart is also reported. Also needed are libSMBL (an API library for manipulation of systems biology models) [42] and a Linear Programming (LP) solver supported by the COBRA Toolbox as, for example, gkpl (http://www.gnu.org/software/glpk) or Gurobi (Gurobi Optimization, http://www.gurobi.com). Please refer to specific literature/manuals/websites for information on the installation and configuration of these tools.

3 Methods

3.1 Obtain a Draft Metabolic Model

Most of the tools in Table 1 allow uploading a draft genome (or a set of coding sequences, CDS) and return an SBML-formatted metabolic model. RAST, for example, can generate a draft metabolic model just by selecting the "*Build metabolic model*" option before starting genome annotation process. MicrobesFlux allows creating metabolic models from all organisms present in KEGG database and extract them in SBML format by clicking the "*Get SBML*". With KBase one can generate an SBML-formatted model either selecting a genome that is already in the KBase Central Data Store (CDS) or a genome that has been already annotated; metabolic models are reconstructed through the function "*genomeTO_to_reconstructionTO*".

Regardless of the tool used, the output of this preliminary step is a draft SBML metabolic reconstruction (Fig. 2) that still needs manual curation to be turned into a functional model.

3.2 Model Evaluation

3.2.1 Missing Reactions and Alternative Pathways

At this stage, the reconstructed model may be incomplete and lack metabolic genes and/or functions. Thus, before starting modeling procedures, it is important to check possible sources of errors. To do this, revise the reconstructed metabolic model in a pathway-by-pathway manner to highlight, for example, potential missing

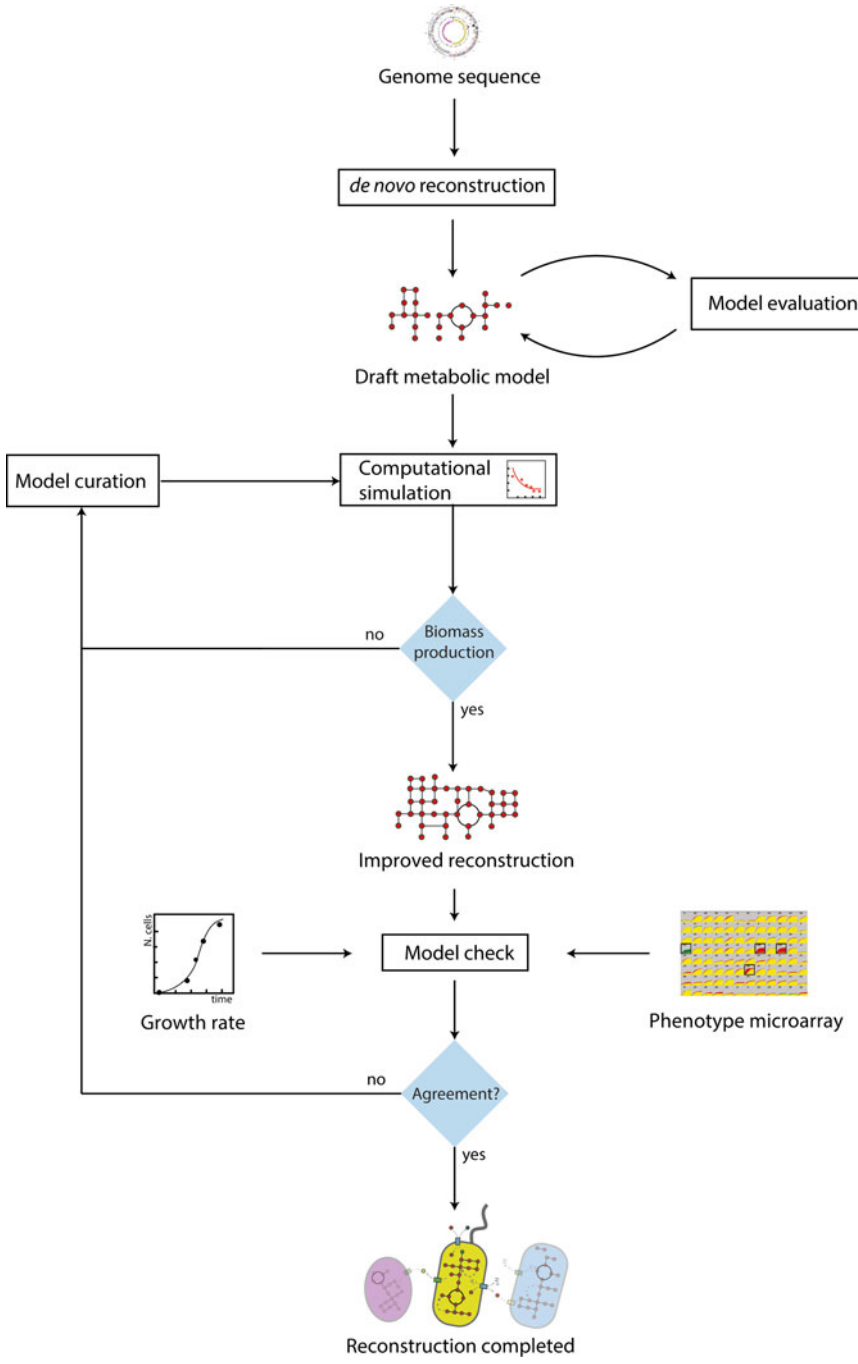

Fig. 2 Schematic representation of a pipeline for metabolic model reconstruction and checking

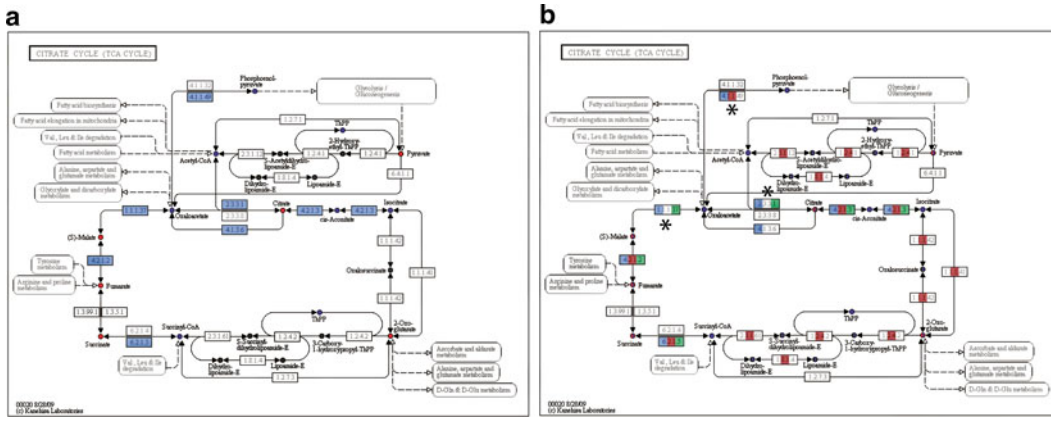

Fig. 3 Reconstructions of the citrate cycle from three Gamma-proteobacteria have been compared [*E. coli* K12 (*red boxes*), *P. haloplanktis* TAC125 (*blue boxes*) and *A. baylyi* ADP1 (*green boxes*)]. Reactions marked with "*" represent paradigmatic examples of how comparative genomics can be exploited for gap-filling metabolic models

reactions. To accomplish this task, graphical visualization of metabolic pathways is highly recommended. On the SEED user page one can display the different metabolic maps for the model under study and browse them just by clicking on the biosynthetic route name. In Fig. 3a citrate cycle is shown for the iAF1260 reconstruction of *E. coli*. As shown in this figure, reactions are colored according to their presence within the studied reconstruction. By doing so information on each metabolic step can be obtained just by clicking on the reaction E.C. code and possible gaps (interrupted pathways) and/or alternative metabolic steps can be easily identified. Record every potential missing reaction or any other unusual metabolic reaction in a spreadsheet and store as much information as possible [e.g., enzyme E.C. number, reaction code (starting with *rxn* in SEED model viewer), metabolic pathway] for each of them. Once all the pathways have been examined, this list should be carefully analyzed and integrated with as much information as possible. In particular, in case of potentially missing reactions one should:

1. *Check available scientific literature and metabolic databases* for the microbe under study since alternative metabolic steps may have been previously described for a given pathway. Also, data on metabolic auxotrophies of the strain of interest may be of interest (e.g., results from Biolog Phenotype Microarray experiments) in this phase.
2. *Use comparative genomics of closely related microorganisms.* SEED model viewer is of great help in this stage since multiple reconstructions can be simultaneously displayed over the same map. To do this, select other models of closely related organisms

from the initial (log-in) page and then click on the biosynthetic pathway you want to examine in detail. A possible output of this procedure is shown in Fig. 3b, in which reconstructions of the citrate cycle from three Gamma-proteobacteria have been compared (*E. coli* K12, *Pseudoalteromonas haloplanktis* TAC125 and *Acinetobacter baylyi* ADP1). Reactions marked with "*" represent paradigmatic examples of how comparative genomics can be exploited for gap-filling metabolic models. Indeed, the presence of those reactions in two organisms out of three might suggest potential errors during the metabolic reconstruction of the other strain. To validate this indication retrieve the sequence of the enzyme encoding for that reaction in one of the organisms possessing it and perform a BLAST search in the genome of the organism missing it. The presence of a hortologous sequence in the probed genome is a strong indication for gap filling the corresponding metabolic step. Other databases can be explored for retrieving information on the metabolic features of closely related microorganisms including KEGG [26] and MetaCyc [43].

The possibility to introduce a confidence score for each reaction added in this stage has been proposed [22], accounting for the type of evidence used for including the reactions within the reconstruction and ranging from 5 (in case biochemical data is available for that specific step) to 1 (in case that reaction has been included only for modeling purposes and no experimental evidence has been provided). These codes are particularly useful during model curation since low-confidence reactions can easily be identified.

3.2.2 Check Reaction Consistency

Each reaction present in the metabolic model at this stage should be carefully inspected in order to check (at least):

1. Substrate and cofactor usage
2. Charged formula for each metabolite
3. Reaction stoichiometry
4. Reaction directionality
5. Information for gene and reaction localization
6. Gene-protein-reaction (GPR) associations

See [22] for detailed instruction on how to accomplish each of these sub-steps.

3.3 Define Biomass Reaction

Reconstructed models usually embed an "artificial" reaction accounting for the assembly of all known biomass components (e.g., DNA, RNA, lipids, proteins, peptidoglycan) and their relative contributions to the overall cellular biomass. As an example, the biomass assembly reaction from the iAbaylyiV4 reconstruction of *A. baylyi* ADP1 is expressed as follows (according to Model SEED, *see* **Note 1** for compound name):

```
<reaction  id="rxn12832"  name="Biomass assembly"  reversible="false">
<notes>
<html:p>GENE_ASSOCIATION:UNKNOWN</html:p>
</notes>
<listOfReactants>
<speciesReference  species="cpd00001_c"  stoichiometry="40"/>
<speciesReference  species="cpd00002_c"  stoichiometry="40"/>
<speciesReference  species="cpd11461_c"  stoichiometry="0.032"/>
<speciesReference  species="cpd11462_c"  stoichiometry="0.2"/>
<speciesReference  species="cpd11463_c"  stoichiometry="0.633"/>
<speciesReference  species="cpd11649_c"  stoichiometry="0.003"/>
<speciesReference  species="cpd11677_c"  stoichiometry="0.002"/>
<speciesReference  species="cpd16601_c"  stoichiometry="0.002"/>
<speciesReference  species="cpd16653_c"  stoichiometry="0.032"/>
<speciesReference  species="cpd16661_e"  stoichiometry="0.028"/>
<speciesReference  species="cpd16662_c"  stoichiometry="0.041"/>
<speciesReference  species="cpd16663_c"  stoichiometry="0.021"/>
<speciesReference  species="cpd16669_c"  stoichiometry="0.006"/>
</listOfReactants>
<listOfProducts>
<speciesReference  species="cpd00008_c"  stoichiometry="40"/>
<speciesReference  species="cpd00009_c"  stoichiometry="40"/>
<speciesReference  species="cpd11416_c"  stoichiometry="1"/>
</listOfProducts>
<kineticLaw>
       <math xmlns="http://www.w3.org/1998/Math/MathML">
                    <ci> FLUX_VALUE </ci>
       </math>
       <listOfParameters>
              <parameter id="LOWER_BOUND" value="0" name="mmol_per_gDW_per_hr"/>
              <parameter id="UPPER_BOUND" value="10000" name="mmol_per_gDW_per_hr"/>
              <parameter id="OBJECTIVE_COEFFICIENT" value="0.0"/>
              <parameter id="FLUX_VALUE" value="0.0" name="mmol_per_gDW_per_hr"/>
       </listOfParameters>
</kineticLaw>
</reaction>
```

Conventionally, the biomass reaction is expressed in h^{-1}, since precursor fractions are converted to mmol/gDW. The biomass assembly reaction sums the mole fraction of each precursor necessary to produce 1 g dry weight of cells [22].

So, at this point, scan available literature for biomass composition of the strain under study. In case available experimental data is not enough, you may derive missing pieces of information from the biomass composition from (more studied) closely related strains. Store information on biomass components in a spreadsheet and then add the assembly reaction into the draft model.

As shown for iAbaylyiV4 reconstruction, biomass assembly reaction should also account for the energy (in the form of ATP) necessary for cell replication. This is usually referred to as GAM (growth-associated ATP maintenance) reaction and can be calculated experimentally. In case no experimental information is available, one can approximate growth-associated costs from the GAM reaction of a closely related strain or deriving it from an estimation of total amount of ATP required to synthesize cellular macromolecules (protein, DNA, and RNA) whose amount can be derived from databases. This latter step is fully described in [22].

3.4 Define Additional Exchange Reactions

Exchange reactions (conventionally labeled with "EX_") allow defining the composition of the in silico growth medium and environmental conditions during simulations. In other words, these reactions define the range of compounds that can be imported into the cellular model and metabolized to form biomass constituents or other cellular products. At this stage, the draft model should already include a minimal set of exchange reactions. As an example, a typical exchange reaction allowing the model to use glucose (ModelSEED code `cpd00027`) can be represented as follows:

```
<reaction  id="EX_cpd00027_e"  name="EX_D-Glucose_e"  reversible="true">
        <notes>
                <html:p>GENE_ASSOCIATION: </html:p>
                <html:p>PROTEIN_ASSOCIATION: </html:p>
                <html:p>SUBSYSTEM: S_</html:p>
                <html:p>PROTEIN_CLASS: </html:p>
        </notes>
        <listOfReactants>
                <speciesReference  species="cpd00027_e" stoichiometry="1.000000"/>
        </listOfReactants>
        <listOfProducts>
                <speciesReference  species="cpd00027_b" stoichiometry="1.000000"/>
        </listOfProducts>
        <kineticLaw>
                <math xmlns="http://www.w3.org/1998/Math/MathML">
                                <ci> FLUX_VALUE </ci>
                </math>
                <listOfParameters>
                        <parameter id="LOWER_BOUND" value="-10000"
units="mmol_per_gDW_per_hr"/>
                        <parameter id="UPPER_BOUND" value="10000"
units="mmol_per_gDW_per_hr"/>
                        <parameter id="OBJECTIVE_COEFFICIENT" value="0.0"/>
                        <parameter id="FLUX_VALUE" value="0.000000"
units="mmol_per_gDW_per_hr"/>
                </listOfParameters>
        </kineticLaw>
</reaction>
```

This particular reaction allows transforming `cpd00027_b` into `cpd00027_e` that will be then used by the other reactions of the model (for example by a transport reaction that will convert `cpd00027_e` into its cytoplasmic counterpart `cpd00027_c`).

So, in this step, check every exchange reaction in respect to specific growth requirements of the strain under study. Information on commonly growth media is highly useful in this phase. Check the composition of every known growth medium for the strain under analysis and include one exchange reaction for each constituent of the growth medium. These reactions can also be added afterwards as they will be used in the next steps to predict cellular growth in specific nutritional conditions.

3.5 Validation of the SBML Model

Validate your model and search for potential formatting errors using the online tool SBML Validator at http://sbml.org/Facilities/Validator/. If no errors are issued it means that your reconstruction is a valid SBML model and you can use it for the next step.

3.6 Import Model into Modeling Framework

The reconstruction is now ready for being imported into COBRA toolbox and start metabolic modeling procedures using FBA. We will assume that Matlab, COBRA toolbox, libSBML, and a valid LP solver (together with their dependencies) have been successfully installed and initialized on the workstation. Assuming that the SBML-formatted model is stored in a file called *draft_model.xml*, import the reconstruction into Matlab with

```
model = readCbModel('draft_model.xml')
```

In Cobrapy run `model=create_cobra_model_from_sbml('draft_model.xml')`

If the model has been correctly imported, you should see something similar to this (within the Matlab console):

```
model =
                rxns: {1284x1 cell}
                mets: {1108x1 cell}
                   S: [1108x1284 double]
                 rev: [1284x1 double]
                  lb: [1284x1 double]
                  ub: [1284x1 double]
                   c: [1284x1 double]
          metCharge: [1108x1 int32]
               rules: {713x1 cell}
               genes: {713x1 cell}
         rxnGeneMat: [1284x713 double]
            grRules: {1284x1 cell}
         subSystems: {1284x1 cell}
    confidenceScores: {1284x1 cell}
       rxnReferences: {1284x1 cell}
        rxnECNumbers: {1284x1 cell}
            rxnNotes: {1284x1 cell}
            rxnNames: {1284x1 cell}
            metNames: {1108x1 cell}
         metFormulas: {1108x1 cell}
          metChEBIID: {1108x1 cell}
           metKEGGID: {1108x1 cell}
        metPubChemID: {1108x1 cell}
      metInChIString: {1108x1 cell}
                   b: [1108x1 double]
         description: 'draft_model.xml'
```

3.7 Check for Model Consistency

COBRA toolbox allows evaluating the imported reconstruction in a global fashion. So, before starting modeling procedures use it to check for:

1. Mass, charge, and stoichiometrically unbalanced reactions: For this you can use the COBRA function:

   ```
   [massImbalance,imBalancedMass,imBalancedCharge,imBalancedBool,Elements] =
   checkMassChargeBalance(model)
   ```

   (In Cobrapy: `unbalanced_rxns=[r for r in model.reactions if r.check_mass_balance() != []]`)

   Fix reactions listed in `imBalancedMass` and `imBalancedCharge` adding, for example, missing protons or proton donors.

2. Gaps in reconstruction (dead-end metabolites, i.e., metabolites that are produced but not consumed): For this you can use the COBRA function:

```
[allGaps,rootGaps,downstreamGaps] = gapFind
(model)
```

Fill gaps in `allGaps` by searching for possible reactions involved in the consumption/production of identified dead-end metabolites. You may use the same approach described in Subheading 3.2.1.

3.8 Set In Silico Medium Composition

The composition of the growth medium can be defined tuning lower and upper bounds (LB and UB, respectively) of exchange reactions in the model (*see* Subheading 3.4). Indeed, through LB and UB it is possible to define the maximum utilization rate for each of the compounds to be imported into the model through exchange reactions. Conventionally, uptake (utilization) rates for a given compound are defined by tuning LB values of the corresponding exchange reaction. Water and inorganic ions are usually considered to be present in non-limiting concentrations and LBs of their corresponding exchange reactions are set to very high values (e.g., 1,000 mmol/g h). Conversely, setting the LB of the exchange reaction regulating the utilization rate of the carbon source(s) present in the medium requires much more attention. Defining a wrong value here would reveal in unreal prediction of cellular growth rate (*see* **Note 2** for details on uptake ration calculation). The LBs of all the other exchange reactions present in the model must be set to "0".

Hence, according to the growth medium in which you want to test the model:

1. For each of the exchange reactions of inorganic ions (in this case `cpd00048`, sulfate) present in the growth medium, set LB to "1,000 mmol/g h", using the `changeRxnBounds` COBRA toolbox function:

```
model = changeRxnBounds(model, 'EX_cpd00048
(e)', -1000, 'l')
```

In Cobrapy run

```
rxn= model.reactions.get_by_id(EX_cpd00048(e))
rxn.lower_bound=-1000.0
rxn.upper_bound=1000.0
```

where `EX_cpd00048(e)` is the exchange reaction for sulfate. Alternatively, you can first define a Matlab list embedding all the exchange reactions of inorganic ions present in the model with

```
IonExchangeReactions={'EX_cpd00048(e)', 'EX_
cpd00067(e)', …}
```

and then set their LB value with

```
model = changeRxnBounds(model, IonExchange
Reactions, -1000, 'l')
```

2. Set the LB of the exchange reaction for the carbon source to some realistic value. In this example we will use glucose (`'cpd00027'`) as the carbon source and we will set the LB of the corresponding exchange reaction to 18 mmol/g h [a value that has been calculated for *E. coli* during fed batch growth [7]]:

```
model = changeRxnBounds(model, 'EX_cpd00048
(e)', -18, 'l')
```

3. Set the LBs of all the remaining exchange reactions to "0". We assume that these reactions have been stored in a list called `RemainingEXreactions`:

```
model = changeRxnBounds(model, Remaining
EXreactions, 0, 'l')
```

3.9 Optimize Model for Biomass Production

The model is now ready for optimization. First of all, identify one reaction of the model as the optimization objective function, i.e., the reaction of the model you want to maximize during simulations. By doing this, linear programming can be used to infer the flux distribution that maximizes (or minimizes) the output of that specific reaction. At this stage of the reconstruction, biomass production should be set as the model objective function. In this way, one can test whether all the compounds involved in biomass assembly (*see* Subheading 3.3) can be synthesized or not. In the first case, the flux out of the biomass assembly (`f`) will be greater than 0; conversely, in case one (or more) biomass constituent(s) cannot be produced, `f` will be equal to 0. Use the following command to define the objective function of the model with COBRA:

```
model = changeObjective(model, 'rxn12832')
```

In Cobrapy run

```
rxn=model.reactions.get_by_id('rxn12832')
rxn.objective_coefficient = 1.0
```

where `rxn12832` is the biomass assembly reaction as defined by the model in this specific case. Then, to derive the flux distribution that optimizes the flux through objective reaction (exploiting FBA), use the following command:

```
FBAsolution = optimizeCbModel(model,'max')
```

In Cobrapy run

```
model.optimize(solver='gurobi')
print model.solution)
```

In a Matlab console, the output of this command should look like this:

```
FBAsolution =
           x: [1284x1 double]
           f: 0
           y: [1108x1 double]
           w: []
        stat: 1
    origStat: -99
      solver: 'gurobi5'
        time: 0.0131
```

In this example, the value of `FBAsolution.f` is 0. This means that one (or more) substrate(s) of the biomass assembly reaction cannot be produced, most likely because of missing reactions (gaps) in the model.

3.10 Manual Curation

To identify which of the biomass precursor(s) cannot be synthesized, repeat the following points for each of them:

1. Add an artificial exchange reaction to the model, accounting for the extrusion of that compound, as shown here for compound `cpd00155_c` of the biomass assembly reaction of Subheading 3.3:

```
<reaction  id="EX_ cpd00155_e"  name="EX_Glycogen"  reversible="true">
       <listOfReactants>
              <speciesReference  species=" cpd00155_c "
stoichiometry="1.000000"/>
       </listOfReactants>
       <listOfProducts>
              <speciesReference  species=" cpd00155_e"
stoichiometry="1.000000"/>
       </listOfProducts>
       <kineticLaw>
              <math xmlns="http://www.w3.org/1998/Math/MathML">
                           <ci> FLUX_VALUE </ci>
              </math>
              <listOfParameters>
                     <parameter id="LOWER_BOUND" value="-10000"
units="mmol_per_gDW_per_hr"/>
                     <parameter id="UPPER_BOUND" value="10000"
units="mmol_per_gDW_per_hr"/>
                     <parameter id="OBJECTIVE_COEFFICIENT" value="0.0"/>
                     <parameter id="FLUX_VALUE" value="0.000000"
units="mmol_per_gDW_per_hr"/>
              </listOfParameters>
       </kineticLaw>
</reaction>
```

2. Set this newly added reaction as the model objective function:

   ```
   model = changeObjective(model, 'EX_ cpd00155 (e)')
   ```

3. Optimize the model for this objective function:

   ```
   FBAsolution = optimizeCbModel(model,'max')
   ```

4. If `FBAsolution.f` is greater than 0, it means that the compound under analysis can be synthesized and you can move to the next one. In case the flux value (`f`) across this reaction is 0, then one (or more) metabolic gap is present along the biosynthetic pathway leading to the production of that specific biomass precursor. In order to trace them back, repeat **steps 1–3** for each of the metabolic reactions that are involved in the biosynthesis of the biomass component (and its precursors) that cannot be synthesized until you find the missing reaction(s). Once identified, you can use comparative genomics and organism-specific databases (*see* Subheading 3.1) to fill the gap.

Once you have repeated these steps for all the biomass constituents and gap-filled the model, the model should be able to produce biomass on the growth medium defined by the LBs of exchange reactions.

3.11 Validate Model Against Experimental Data

Besides being able to produce biomass, metabolic reconstructions are also required to fit as much as possible with experimental data. To check their reliability in predicting growth phenotypes, metabolic models can be compared against large-scale growth tests (e.g., Biolog Phenotype Microarray, *see* Chapter 7) or experimentally calculated growth rates. Comparing in silico-predicted growth against data from high-throughput phenomics gives indication on the overall capability of the model to correctly predict growth on a large set of known carbon sources. As already done for other reconstructions [44–46], Biolog data and model optimization outcomes can be easily compared. To do this:

1. Collect Biolog information on known carbon sources in a spreadsheet. This should include (a) compound names, (b) KEGG compound codes, and (c) growth/non-growth phenotypes. A simple example of a valid reference file for this analysis is shown in Table 2.

Table 2
Schematic tabular representation of processed Biolog results used for comparing in vivo data with model predictions

KEGG code	Substrate	Growth
C00025	L-Glutamic acid	Yes
C00026	a-Keto-glutaric acid	No
C00031	a-D-Glucose	No
C00033	Acetic acid	Yes

2. For each of the compounds listed in the Biolog-derived table
   (a) If the compound is not present in the model, add it. Use KEGG reference code to univocally identify shared compounds between Biolog dataset and metabolic reconstruction.
   (b) Add an exchange reaction accounting for its utilization by the model (*see* Subheading 3.4).
   (c) Use LB of this reaction to set its uptake rate (if not known then use an arbitrary value, e.g., 10 mmol/g h).
   (d) Ensure that all the other LBs of exchange reactions of carbon source compounds are set to 0.
   (e) Optimize the model for biomass production and record `f` value (`f` > 0 or `f` = 0).
   (f) Check if Biolog data and model prediction agree (both growth or both non-growth phenotypes) or if they do not. In this latter scenario, two alternatives are possible, i.e., (1) Biolog records growth whereas model predicts no growth or, conversely, (2) Biolog does not record growth whereas model predicts growth. To fix point (1):
      - Check if a transport reaction for the carbon source under analysis is present within the model. If this is not the case, check the genome for a gene putatively encoding a transporter able to import the carbon source under analysis. TCDB (Transporter Classification Data Base, http://www.tcdb.org/, [47]) can be a valuable resource in this sense. An artificial transport reaction (i.e., without any associated coding genes) can be added at this point for debugging purposes.
      - If biomass is not produced (`f` = 0) even after adding the transport reaction to the model, search for possible metabolic gaps in the model by repeating **steps 1–4** of Subheading 3.10 (setting biomass production as the objective function).
      - Include a further column to the table shown above, including model growth prediction for each compound and highlight possible incongruences (*see* Table 3 for an example).

Fixing case (2) is trickier and it may involve the removal of one (or more) reaction(s) erroneously added to the model during the reconstruction process (with the risk to remove reactions that are crucial under other growth conditions) and/or an accurate revision the substrate specificity of the transporters in model.

As a rule of thumb, 80/90 % agreement between Biolog data and model predictions can be considered satisfactory; this is usually found for most of the reconstructions that have been validated against high-throughput Phenomics to date [44, 45].

Table 3
Comparison between model prediction and in vivo (Biolog) data

KEGG code	Substrate	Growth	Model
C00025	L-Glutamic acid	Yes	Yes
C00026	a-Keto-glutaric acid	No	Yes
C00031	a-D-Glucose	No	No
C00033	Acetic acid	Yes	Yes

Model-predicted growth rate (`f` value) can also be compared against experimentally determined growth rates (both expressed as h^{-1}). Specific solutions for fixing erroneous in silico predictions (either too fast or too slow predicted growth) can be found in [22].

Although the scope and the purpose of the reconstruction define whether the iterative reconstruction process can be considered "finished" [22], the model capability of synthesizing all the components of the biomass and an overall agreement between model prediction and experimental data are usually considered a first achievement in the overall reconstruction process and a reliable base for further analyses. However, since the reconstruction will not likely embed information on more than 20–30 % of the encoded enzymes, continuous effort is necessary to periodically update and revise the metabolic model and to include as much information as possible (e.g., gene-protein relationships, organism-specific reactions, experimental data).

3.12 Dynamic Flux Balance Analysis (dFBA)

Among all the possible modeling strategies, dFBA has been gaining increasing interest. Basically, dFBA combines extracellular dynamics with intracellular pseudosteady states and thus may be suitable for the simulation of metabolic behavior under dynamic conditions. dFBA provides a framework for analyzing the transience of metabolism due to metabolic reprogramming and for obtaining insights for the design of metabolic networks [48]. This technique has been widely adopted for predicting different cellular metabolic states, including the diauxic shift of *E. coli* growth [48] and the effect of genetic manipulations on ethanol production [49].

dFBA is basically an iteration of the FBA method where at each (user-defined, see below) time step FBA is used to compute the cellular growth rate together with nutrient utilization rate and (eventual) by-product efflux. These outputs are then used, at the following time point, to recompute biomass production, nutrient uptake, and by-product secretion. This iterative procedure continues until the last time point is reached. Resulting biomass, nutrients, and by-products can then be plotted in a graph accounting for the values of each of these quantities at the different time points.

dFBA is implemented in COBRA toolbox and to run it on your reconstructed model in the COBRA toolbox use

```
dynamicFBA(model,SubstrateUptake,InitialConcentration,InitialBiomass, TimeStep, NSteps,RxnsToPlot);
```

where

`model` is the metabolic model as imported into COBRA.

`SubstrateUptake` embeds the list of the reactions accounting for the uptake of the nutrients.

`InitialConcentration` is the concentration of the nutrient source at the beginning of the dFBA run.

`InitialBiomass` is the initial amount of cellular biomass.

`TimeStep` defines the size of each time step during the iteration.

`NSteps` defines how many steps will be performed during the iteration.

`RxnsToPlot` defines the list of reactions whose values will be used for plotting the results of the dFBA simulation.

4 Notes

1. Compound names of iAbaylyiV4 biomass assembly reaction are

```
<species id="cpd00001_c" name="H2O_H2O" compartment="c" charge="0" boundaryCondition="false"/>
<species id="cpd00002_c" name="ATP_C10H13N5O13P3" compartment="c" charge="-3" boundaryCondition="false"/>
<species id="cpd11461_c" name="DNA_C15H23O13P2R3" compartment="c" charge="-2" boundaryCondition="false"/>
<species id="cpd11462_c" name="mRNA_" compartment="c" charge="10000000" boundaryCondition="false"/>
<species id="cpd11463_c" name="Protein_C4H5N2O3R2" compartment="c" charge="-1" boundaryCondition="false"/>
<species id="cpd11469_c" name="(2E)-Dodecenoyl-[acp]_C23H41N2O8PRS" compartment="c" charge="-1" boundaryCondition="false"/>
<species id="cpd11677_c" name="Triglyceride_C6H5O6R3" compartment="c"
charge="0" boundaryCondition="false"/>
<species id="cpd16601_c" name="generic_fatty acid chain for free molecules (mass)_"
```

```
compartment="c" charge="10000000" boundary
Condition="false"/>
<species id="cpd16653_c" name="generic_cofac-
tor molecule (mass)_" compartment="c" charge=
"10000000" boundaryCondition="false"/>
<species id="cpd16661_c" name="generic_pep-
tidoglycan (mass)_" compartment="c" charge=
"10000000" boundaryCondition="false"/>
<species id="cpd16662_c" name="generic_phos-
pholipid (mass)_" compartment="c" charge=
"10000000" boundaryCondition="false"/>
<species id="cpd16663_c" name="generic_free
polysaccharide (mass)_" compartment="c" charge=
"10000000" boundaryCondition="false"/>
<species id="cpd16669_c" name="generic_wax
esters (mass)_" compartment="c" charge=
"10000000" boundaryCondition="false"/>
<species id="cpd00008_c" name="ADP_
C10H13N5O10P2" compartment="c" charge="-2"
boundaryCondition="false"/>
<species id="cpd00009_c" name="Phosphate_
HO4P" compartment="c" charge="-2" boundary
Condition="false"/>
<species id="cpd11416_c" name="Biomass_"
compartment="c" charge="0" boundaryCondition=
"false"/>
```

2. The enzymatic capacity (EC) for carbon source utilization is determined as the ratio of the growth rate (μ) to the biomass yield in batch experiments (biomass yield) [7]:

$$\mathrm{EC} = \frac{\mu}{\text{Biomass yield}}$$

Biomass yield is defined as the ratio of the amount of biomass produced to the amount of substrate consumed:

$$\text{Biomass yield} = \frac{g \text{ of biomass}}{g \text{ of substrate utilized}}$$

Note that, since in FBA all reaction fluxes are expressed as mmol/g h, biomass yield should be converted taking into consideration mmol of substrate provided before calculating EC with

$$\text{mmol of substrate} = \frac{g \text{ of substrate}}{\text{MW of substrate}}$$

where MW is the molecular weight of the substrate.

References

1. Downs DM (2003) Genomics and bacterial metabolism. Curr Issues Mol Biol 5(1):17–25
2. Beloqui A, de Maria PD, Golyshin PN, Ferrer M (2008) Recent trends in industrial microbiology. Curr Opin Microbiol 11(3):240–248
3. Zou W et al (2012) Reconstruction and analysis of a genome-scale metabolic model of the vitamin C producing industrial strain Ketogulonicigenium vulgare WSH-001. J Biotechnol 161(1):42–48
4. Garcia-Ochoa F, Santos VE, Casas JA, Gomez E (2000) Xanthan gum: production, recovery, and properties. Biotechnol Adv 18(7):549–579
5. George HA, Johnson JL, Moore WE, Holdeman LV, Chen JS (1983) Acetone, isopropanol, and butanol production by Clostridium beijerinckii (syn. Clostridium butylicum) and Clostridium aurantibutyricum. Appl Environ Microbiol 45(3):1160–1163
6. Lee SJ et al (2005) Metabolic engineering of Escherichia coli for enhanced production of succinic acid, based on genome comparison and in silico gene knockout simulation. (Translated from eng). Appl Environ Microbiol 71(12):7880–7887
7. Varma A, Palsson BO (1994) Stoichiometric flux balance models quantitatively predict growth and metabolic by-product secretion in wild-type Escherichia coli W3110. Appl Environ Microbiol 60(10):3724–3731
8. Oberhardt MA, Palsson BO, Papin JA (2009) Applications of genome-scale metabolic reconstructions. Mol Syst Biol 5:320 (in eng)
9. Orth JD, Thiele I, Palsson BO (2010) What is flux balance analysis? Nat Biotechnol 28(3): 245–248
10. Oberhardt MA, Chavali AK, Papin JA (2009) Flux balance analysis: interrogating genome-scale metabolic networks. Methods Mol Biol 500:61–80
11. Fang X, Wallqvist A, Reifman J (2012) Modeling phenotypic metabolic adaptations of Mycobacterium tuberculosis H37Rv under hypoxia. PLoS Comput Biol 8(9):e1002688
12. Park JM, Kim TY, Lee SY (2009) Constraints-based genome-scale metabolic simulation for systems metabolic engineering. Biotechnol Adv 27(6):979–988 (in eng)
13. Maarleveld TR, Khandelwal RA, Olivier BG, Teusink B, Bruggeman FJ (2013) Basic concepts and principles of stoichiometric modeling of metabolic networks. Biotechnol J 8(9): 997–1008
14. Moura M, Broadbelt L, Tyo K (2013) Computational tools for guided discovery and engineering of metabolic pathways. Methods Mol Biol 985:123–147
15. Copeland WB et al (2012) Computational tools for metabolic engineering. Metab Eng 14(3):270–280
16. Durot M, Bourguignon PY, Schachter V (2009) Genome-scale models of bacterial metabolism: reconstruction and applications. FEMS Microbiol Rev 33(1):164–190 (in eng)
17. Aziz RK et al (2008) The RAST Server: rapid annotations using subsystems technology. (Translated from eng). BMC Genomics 9:75
18. Schneider J et al (2010) CARMEN - comparative analysis and in silico reconstruction of organism-specific MEtabolic networks. Genet Mol Res 9(3):1660–1672
19. Feng X, Xu Y, Chen Y, Tang YJ (2012) MicrobesFlux: a web platform for drafting metabolic models from the KEGG database. BMC Syst Biol 6:94
20. Karp PD et al (2010) Pathway Tools version 13.0: integrated software for pathway/genome informatics and systems biology. Brief Bioinform 11(1):40–79
21. Reyes R et al (2012) Automation on the generation of genome-scale metabolic models. (Translated from eng). J Comput Biol 19(12): 1295–1306
22. Thiele I, Palsson BO (2010) A protocol for generating a high-quality genome-scale metabolic reconstruction. Nat Protoc 5(1):93–121
23. Thiele I, Palsson BO (2010) Reconstruction annotation jamborees: a community approach to systems biology. Mol Syst Biol 6:361
24. Dandekar T, Fieselmann A, Majeed S, Ahmed Z (2012) Software applications toward quantitative metabolic flux analysis and modeling. Brief Bioinform 15:91
25. Henry CS et al (2010) High-throughput generation, optimization and analysis of genome-scale metabolic models. (Translated from eng). Nat Biotechnol 28(9):977–982
26. Kanehisa M (2002) The KEGG database. Novartis Found Symp 247:91–101, discussion 101–103, 119–128, 244–152
27. Karp PD, Paley S (1996) Integrated access to metabolic and genomic data. J Comput Biol 3(1):191–212 (in eng)
28. Karp PD, Paley S, Romero P (2002) The pathway tools software. Bioinformatics 18(Suppl 1): S225–S232
29. Caspi R et al (2012) The MetaCyc database of metabolic pathways and enzymes and the BioCyc collection of pathway/genome databases.

Nucleic Acids Res 40(Database issue): D742–D753
30. Karp PD, Riley M, Paley SM, Pellegrini-Toole A (2002) The MetaCyc database. Nucleic Acids Res 30(1):59–61
31. Altman T, Travers M, Kothari A, Caspi R, Karp PD (2013) A systematic comparison of the MetaCyc and KEGG pathway databases. BMC Bioinform 14(1):112 (in Eng)
32. Hyland C, Pinney JW, McConkey GA, Westhead DR (2006) metaSHARK: a WWW platform for interactive exploration of metabolic networks. Nucleic Acids Res 34(Web Server issue):W725–W728
33. Pinney JW, Shirley MW, McConkey GA, Westhead DR (2005) metaSHARK: software for automated metabolic network prediction from DNA sequence and its application to the genomes of Plasmodium falciparum and Eimeria tenella. Nucleic Acids Res 33(4): 1399–1409
34. Liao YC, Tsai MH, Chen FC, Hsiung CA (2012) GEMSiRV: a software platform for GEnome-scale metabolic model simulation, reconstruction and visualization. Bioinformatics 28(13):1752–1758
35. Agren R et al (2013) The RAVEN toolbox and its use for generating a genome-scale metabolic model for Penicillium chrysogenum. PLoS Comput Biol 9(3):e1002980
36. Swainston N, Smallbone K, Mendes P, Kell D, Paton N (2011) The SuBliMinaL Toolbox: automating steps in the reconstruction of metabolic networks. J Integr Bioinform 8(2):186
37. Hucka M et al (2003) The systems biology markup language (SBML): a medium for representation and exchange of biochemical network models. Bioinformatics 19(4):524–531
38. Bray T, Paoli J, Sperberg-McQueen CM (1998) Extensible markup language (XML) 1.0. Available from: http://www.w3.org/TR/1998/REC-xml-19980210
39. Feist AM et al (2007) A genome-scale metabolic reconstruction for Escherichia coli K-12 MG1655 that accounts for 1260 ORFs and thermodynamic information. Mol Syst Biol 3:121
40. Schellenberger J et al (2011) Quantitative prediction of cellular metabolism with constraint-based models: the COBRA Toolbox v2.0. Nat Protoc 6(9):1290–1307
41. Ebrahim A, Lerman JA, Palsson BO, Hyduke DR (2013) COBRApy: COnstraints-based reconstruction and analysis for python. BMC Syst Biol 7:74
42. Bornstein BJ, Keating SM, Jouraku A, Hucka M (2008) LibSBML: an API library for SBML. Bioinformatics 24(6):880–881
43. Karp PD et al (2000) The EcoCyc and MetaCyc databases. Nucleic Acids Res 28(1):56–59
44. Oberhardt MA, Puchalka J, Fryer KE, Martins dos Santos VA, Papin JA (2008) Genome-scale metabolic network analysis of the opportunistic pathogen Pseudomonas aeruginosa PAO1. J Bacteriol 190(8):2790–2803
45. Fang K et al (2011) Exploring the metabolic network of the epidemic pathogen Burkholderia cenocepacia J2315 via genome-scale reconstruction. BMC Syst Biol 5:83
46. Oberhardt MA, Puchalka J, Martins dos Santos VA, Papin JA (2011) Reconciliation of genome-scale metabolic reconstructions for comparative systems analysis. PLoS Comput Biol 7(3): e1001116
47. Saier MH Jr, Tran CV, Barabote RD (2006) TCDB: the Transporter Classification Database for membrane transport protein analyses and information. Nucleic Acids Res 34(Database issue):D181–D186
48. Mahadevan R, Edwards JS, Doyle FJ 3rd (2002) Dynamic flux balance analysis of diauxic growth in *Escherichia coli*. Biophys J 83(3): 1331–1340 (in eng)
49. Lisha KP, Sarkar D (2014) Dynamic flux balance analysis of batch fermentation: effect of genetic manipulations on ethanol production. Bioprocess Biosyst Eng 37(4):617–627
50. Overbeek R et al (2005) The subsystems approach to genome annotation and its use in the project to annotate 1000 genomes. Nucleic Acids Res 33(17):5691–5702
51. Boele J, Olivier BG, Teusink B (2012) FAME, the flux analysis and modeling environment. BMC Syst Biol 6:8

Chapter 16

From Pangenome to Panphenome and Back

Marco Galardini, Alessio Mengoni, and Stefano Mocali

Abstract

The ability to relate genomic differences in bacterial species to their variability in expressed phenotypes is one of the most challenging tasks in today's biology. Such task is of paramount importance towards the understanding of biotechnologically relevant pathways and possibly for their manipulation. Fundamental prerequisites are the genome-wide reconstruction of metabolic pathways and a comprehensive measurement of cellular phenotypes. Cellular pathways can be reliably reconstructed using the KEGG database, while the OmniLog™ Phenotype Microarray (PM) technology may be used to measure nearly 2,000 growth conditions over time. However, few computational tools that can directly link PM data with the gene(s) of interest followed by the extraction of information on gene–phenotype correlation are available.

In this chapter the use of the DuctApe software suite is presented, which allows the joint analysis of bacterial genomic and phenomic data, highlighting those pathways and reactions most probably associated with phenotypic variability. A case study on four *Sinorhizobium meliloti* strains is presented; more example datasets are available online.

Key words Phenotype microarray, Metabolic pathways, Genomic variability, Phenotypic variability

1 Introduction

Addressing the genetic determinants of the phenotypic variability observed in or between bacterial species is one of the most ambitious and challenging tasks of today's biology. Several seminal studies have shown that this correlation is indeed exploitable either through experimental and/or computational approaches [1–3]. The fundamental prerequisites to achieve this goal are the reconstruction of the cellular metabolic pathways and the collection of rich and comprehensive phenotypic data.

The reconstruction of bacterial metabolic pathways is a challenging task by itself. In fact, a large fraction of the genes from bacterial genomes has no ascribed function or its function is merely based on sequence homology, with little support from experimental data [4]. Nevertheless, a series of methods for the joint automatic and curated reconstruction of cellular metabolic models have been developed, such as those applied in the KEGG, MetaCyc, and

Alessio Mengoni et al. (eds.), *Bacterial Pangenomics: Methods and Protocols*, Methods in Molecular Biology, vol. 1231, DOI 10.1007/978-1-4939-1720-4_16,

SEED databases [5–7], as well as organism specific curated models [3]. Such models may also include regulatory layers and other modules not strictly related to the cellular metabolism, possibly improving the adherence of the model to the observed phenotypes [8].

The effort in reconstructing—and most importantly, refining—a cellular model relies also on the collection of rich and comprehensive phenotypic data. Such measurements are of great importance to highlight the phenotypic differences between strains or species, which may be then related to the genetic differences observed in the metabolic models. Moreover, the recording of the temporal dimension may also be important in highlighting subtle but possibly important differences. A technology able to efficiently perform such task is the Phenotype Microarray (PM), based on the OmniLog™ platform. The system is able to record the cellular metabolism on roughly 2,000 growth and stress conditions; the cell respiration is used as a reporter of cellular active metabolism. In fact, when the metabolism is active, the flow of electrons will be directed towards the production of NADH; the PM technology then records the change of its concentration over time using a tetrazolium dye that develops a purple color once reduced by NADH. The color intensity is then proportional to the cellular metabolic levels and is recorded by a camera every 15 min, thus allowing a rich time-course experiment [9]. Since the introduction of this technology, several software have been developed to store and analyze PM data, such as the PhD database [10], RetroSpect™, PheMaDB [11] and the opm package [12,13]. Even though this tools are of great help in the interpretation of phenotypic data coming from the PM technology, the ability to relate phenotypic differences to genomic metabolic reconstructions was still missing, even though several works have been published with attempt to link between genomic and PM data [14–17] or improving genome annotation [18]. To fill this gap, the DuctApe suite has been developed; the cellular metabolism is reconstructed based on KEGG metabolic pathways, in which the genes are mapped to single reactions, while single PM experiments are mapped to metabolic compounds [19]. The DuctApe suite provides various network statistics to help predict which parts of the metabolic network may be more related to the utilization of a specific compound. The suite also allows the analysis of several kinds of experimental setups: (a) a single strain experiment, (b) mutational experiments with one reference strain and one or more mutants, and (c) pangenomic experiments, having more organisms simultaneously.

In this chapter an example application of DuctApe on four *Sinorhizobium meliloti* strains is showed, comprising both genomic and PM phenotypic data.

1.1 Note on This Chapter

In this chapter we use some typesetting conventions. We use:

`this format`

in order to refer to command line input or output, but also to refer to external text.

2 Materials

The materials needed to perform a full DuctApe analysis are divided into two categories: genomic and phenotypic data. The example dataset analyzed in this chapter, together with the list of commands can be found online (https://github.com/combogenomics/ductape_data/tree/master/smeliloti).

2.1 Environment

The DuctApe suite can be used in any UNIX-like shell, such as bash, zsh or cygwin. This chapter assumes that the bash shell is been used in a Linux operating system such as Ubuntu 13.10. This chapter is based on DuctApe version 0.16.4.

2.1.1 Software Dependencies

The DuctApe suite has several software dependencies, listed in Table 1. All the dependencies can be installed following the instructions provided in the project website (http://combogenomics.github.io/DuctApe/); however if a package manager is present in the operating system (such as the `apt-get` command in Ubuntu), most of the dependencies can be installed directly from the package manager.

Table 1
DuctApe software dependencies

Name	Source	Version
DuctApe	http://combogenomics.github.io/DuctApe/	>=0.16.4
Python	http://www.python.org/	>=2.7
BioPython	http://biopython.org	>=1.59
NumPy	http://www.numpy.org/	>=1.8.1
SciPy	http://www.scipy.org/	>=0.14.0
matplotlib	http://matplotlib.org/	>=1.3.1
Scikits learn	http://scikit-learn.org/	>=0.14.0
NetworkX	http://networkx.github.io/	>=1.8.1
PyYaml	http://pyyaml.org/	>=3.11
BLAST+	http://blast.ncbi.nlm.nih.gov	Any

2.2 Genomic Data

2.2.1 Genomic Sequences

The `dgenome` program (the part of the DuctApe suite that handles the genomic data) needs the sequences and IDs of all the proteins that belong to the analyzed genomes, in FASTA format. The FASTA format looks as follows:

```
>sequence_ID description
MRMNLATAPGGFQAGSN…
>sequence2_ID description
MTDTGWIDLALVSARPQAMGA…
```

Depending on the project type (single, mutant or pangenome) one, or more files have to be provided. In particular, in the case of a mutational experiment, the FASTA file of the mutant should contain only those sequences that have been deleted or added with respect to the wild-type strain. To avoid confusion, it is suggested that each file is named with the strain identifier used when invoking the `dape add` command.

2.2.2 KEGG Annotations

To perform the metabolic network reconstruction, either a full KEGG database proteome FASTA file or a series of KAAS annotation files must be provided [20]. Since the KEGG FASTA file is beyond a paywall since 2010, the most probable source of KEGG annotation would be a series of KAAS annotation files, one for each input strains. The file looks as follows:

```
sequence_ID K02313
sequence2_ID
sequence3_ID     K03088
…
```

Where the two columns are separated by a "tab" character. The first column should contain the protein sequences identifiers s in the genomic FASTA files, while the second column should contain the KEGG orthology (KO) identifiers. The lines where no KO identifier is shown can be deleted, as those proteins are not annotated by KAAS and won't be considered in the metabolic reconstruction.

2.3 Phenotypic Data

The `dphenome` program (the part of the DuctApe suite that analyzes PM data) only needs the Phenotype Microarray plates data, either encoded as csv files (as provided by the Omnilog™ platform) or as YAML/JSON files (as provided by the opm package or the DuctApe suite).

The csv format looks as follows:

```
Data File , C:\Program Files\Biolog\TEST.OKA
Set up Time ,Sep 03 2010 3:24 PM
Position , 1-A
Plate Type ,PM 1-
Strain Type ,---
```

```
Sample Number,1
Strain Name ,BL225C
Strain Number,12
Other ,
Hour, A01, A02, A03, A04, …
0.000, 0.00, 4.00, 0.00, 0.00, 0.00, …
0.250, 0.00, 0.00, 0.00, 0.00, 0.00, …
0.500, 0.00, 0.00, 0.00, 0.00, 0.00, …
…
```

Please note that the field "Strain name" should contain the strain identifier, as indicated in the project setup.

The YAML format (similar to JSON) looks as follows:

```
csv_data:
Data file: ''
File: ''
Other: ''
Plate Type: PM01
Position: ''
Sample Number: 1
Setup Time: ''
Strain Name: BL225C
Strain Number: 12
Strain Type: ''
measurements:
A01:
- 0.0
- 0.0
- 0.0
- 0.0
…
```

Please note that the field "Strain name," similarly to the csv format, should contain the strain identifier, as indicated in the project setup.

3 Methods

The DuctApe suite contains three distinct programs, each with a specific scope: `dape` for project setup and the joint genomic and phenotypic analysis, `dgenome` for the metabolic reconstruction

from genomic data and `dphenome` for PM data analysis and integration in the metabolic maps. The three programs act on a common file, by default called `ductape.db`, which holds all the input data and from which all the analysis are drawn.

3.1 Installation

- From a terminal, just type:

```
sudo apt-get install DuctApe
```

All the software dependencies will be downloaded, except for the NCBI BLAST+ software, which can be installed using the operating system package manager. In Ubuntu and other Debian-based Linux distributions, type in a terminal:

```
sudo apt-get install ncbi-blast+
```

- Verify the successful by typing in a terminal

```
dape --version
```

- Download the test dataset from (https://github.com/combogenomics/ductape_data/tree/master/smeliloti)
- Move into the downloaded directory, using the `cd` command

3.2 Project Setup

The test dataset contains genomic and phenotypic data for four *Sinorhizobium meliloti* strains. The first step of the analysis consists in creating a DuctApe project and indicating the strains identifiers.

- Type the following commands in a terminal to initialize the project file:

```
dape init
dape add Rm1021 -c red
dape add BL225C -c green
dape add AK83 -c blue
dape add AK58 -c orange
```

These commands will create a file called `ductape.db`. The program will understand that it needs to compute a pangenome analysis, since than more than one strain have been set up. The `-c` option assigns a color to each strain; if it is not provided, a random color will be assigned to each strain.

3.3 Genomic Analysis

- Run the following command to import the genomic FASTA files

```
dgenome add-dir genome
```

3.3.1 Import the Genomic Data

Please note that this command expects that each file with a `.faa` extension in the `genome` directory will have a name that is equal to the strain identifiers that we provided when initializing the project.

3.3.2 Annotate the Genomes Using KAAS

The annotation of each genome to the KEGG orthology database has to be provided, in order for DuctApe to reconstruct the reaction content of each KEGG pathway. The easiest way to obtain

such annotation is to use the KAAS web server (http://www.genome.jp/kaas-bin/kaas_main). The four FASTA files have to be submitted as separate runs, indicating the BBH method and the prokaryotes gene set.

- Go to http://www.genome.jp/kaas-bin/kaas_main
- Select one of the four FASTA files of the "genome" directory
- Select the "prokaryotes" gene set
- Select the "BBH" assignment method
- Press "compute"
- Once the analysis is completed, download the KO annotation file (`query.ko`)

For this dataset, all the KAAS annotation files are already provided in the `kegg` directory

- Run the following command to import the KAAS annotation files

```
dgenome add-ko kegg/*
```

3.3.3 Pangenome Estimation and Pathways Reconstruction

Once the genomic sequences and the KAAS annotations have been saved to the project file the program can perform the pangenome estimation (using the built-in BBH algorithm) and the KEGG pathways reconstruction. This operation can be parallelized using multiple cores, to speed-up the analysis.

- Type the following in a terminal to compute the pangenome using 4 cores and perform the metabolic network reconstruction:

```
dgenome start -n 4 -x SinMel_
```

Please note that this command needs a stable internet connection to access the KEGG API, needed for the metabolic network reconstruction. Also note that this command may need some time to complete.

Since the KAAS annotation is an automated method, it may, it may miss some KEGG annotations; the `dgenome annotate` command can transfer KEGG annotations between orthologous genes, thus improving the metabolic network reconstruction.

- Type the following to transfer KEGG annotations between orthologs:

```
dgenome annotate
```

Please note that this operation can be reverted with the `dgenome deannotate` command.

3.3.4 Generate Statistics and Export Genomic Data

The analysis on the genomic data is then concluded, and summary statistics can be drawn and relevant data can be exported.

- Type the following to obtain statistics on the pangenome and the metabolic reconstruction:

```
dgenome stats
```

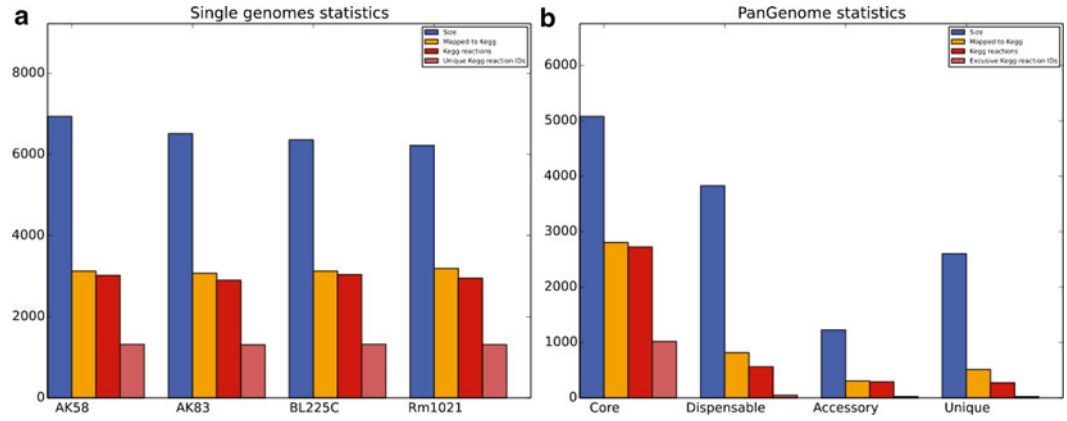

Fig. 1 Some of the `dgenome stats` output plots. (**a**) Single genomes sizes and number of proteins mapped to KEGG pathways and reactions. (**b**) Sizes and number of orthologous groups mapped to KEGG reactions in each pangenome compartment

Some of the plots that are produced by this command are reported in Fig. 1. The dispensable genome (which contains genes that are variable among the input strains) has a lower number of orthologs mapped to the KEGG pathways as compared to the conserved core genome; the reaction encoded by the dispensable genome may be related to the phenotypic differences that will be highlighted using the `dphenome` program. Further information can be obtained using the `dgenome export` command.

- Type the following to export the pangenome reconstruction and the KEGG reactions lists:

```
dgenome export
```

Among the files produced by this command, the `pangenome.tsv` file contains the information of which protein identifiers participate in each orthologous group of the pangenome.

```
#orth_id prot_id
SinMel_24 SinmeB_5617
SinMel_25 SinmeB_1216
SinMel_26 gi|15963768|ref|NP_384121.1|
SinMel_27 gi|16264270|ref|NP_437062.1|
...
```

Another interesting file is `reactions_exclusive_AK83.tsv` which indicates that this strain has some reactions that are not mapped in the other three strains, three of which are part of the map00260 pathway (glycine, serine, and threonine metabolism).

```
#re_id name description pathway(s)
R06979 Ectoine hydro-lyase N-gamma-Acetyldiaminobutyrate <=> H2O + Ectoine map00260,map01100,map01120,map01210
```

```
R01290 L-serine hydro-lyase (adding homocysteine; L-cystathionine-forming) L-Serine + L-Homocysteine <=> L-Cystathionine + H2O map00260,map00270,map01100,map01230
R00891 L-serine hydro-lyase (adding hydrogen sulfide, L-cysteine-forming) L-Serine + Hydrogen sulfide <=> L-Cysteine + H2O map00260
...
```

The analysis of a PM experiment may indicate whether these exclusive reactions are related to a phenotypic variability between this strain and the others.

3.4 Phenotypic Analysis

3.4.1 Import PM Data

- Run the following command to import PM experiments data in the project file:

```
dphenome add-dir phenome
```

Please note that this command expects that each file with a `.csv` extension in the `phenome` directory will have a name that is equal to the strain identifiers that we provided when initializing the project. Also note that the command automatically detects the replicates of each plate.

3.4.2 Subtract the Control Wells (Optional)

Some PM plates have one or more wells with no metabolite, to act as a negative control; the signal of these wells can be subtracted from the other wells of the same plate, thus providing a way to normalize the other signals of the plate.

- Run the following command to remove the control signals from the plates that have at least one control well:

```
dphenome zero
```

Please note that the above command also accepts "blank plates," that are plates with no inoculants; the use of such plates can reduce the biases due to well compounds that cause a spontaneous reduction of the tetrazolium dye.

3.4.3 Calculate PM Curve Parameters

In order to rank each respiration curve and thus compare the metabolic capabilities of each strain of the experiment, a series of parameters from the PM curves are extracted. To allow an easier comparison between PM curves, a k-means clusterization on these parameters is then performed, assigning to each curve an Activity Index (AV), that corresponds to the k-means cluster; since the single clusters are ranked by the area under the respiration curve, lower AV values indicate lower metabolic activity. The optimal number of clusters (and therefore the maximum AV value) is determined using an elbow test.

- Run the following command to calculate the PM curve parameters and perform the elbow test:

```
dphenome start -e -g
```

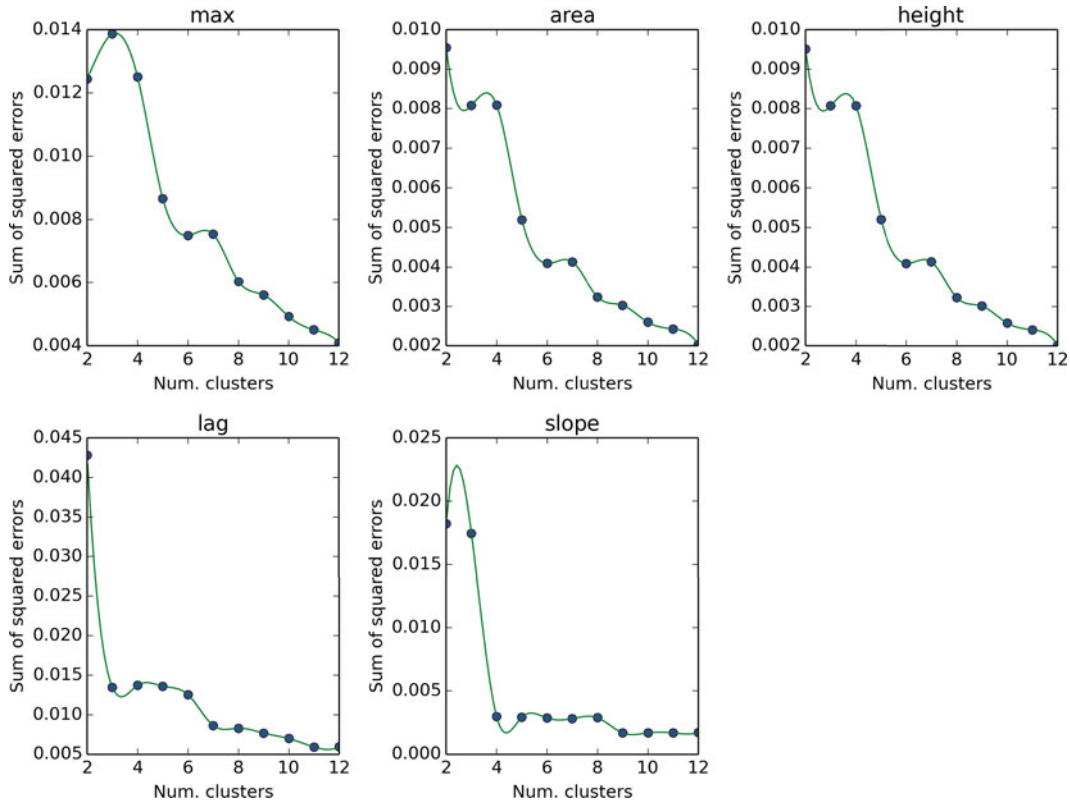

Fig. 2 The elbow test plot obtained through the `dphenome start` command

- Analyze the `elbow.png` file (Fig. 2) and choose the number of clusters that maximizes the reduction of errors

In the dataset presented in this chapter the number of clusters that maximizes the reduction of error is five on three parameters over five.

- Run the following command to clusterize the PM curves

```
dphenome start -n 5
```

The above command also maps each PM compound to the KEGG pathways; therefore this command also needs an internet connection to operate and terminate the metabolic reconstruction.

3.4.4 Plot PM Curves

The overall PM experiment can be summarized in a single ring visualization, with a color code directly related to the AV value that has been calculated using the `dphenome start` command. The single curves can also be plotted, allowing a direct comparison between the four strains of the experiment.

- Run the following command to generate activity rings, showing the differences in metabolic activity between the Rm1021 reference strain and the other strains (Fig. 3a)

```
dphenome rings -o Rm1021 -d 2
```

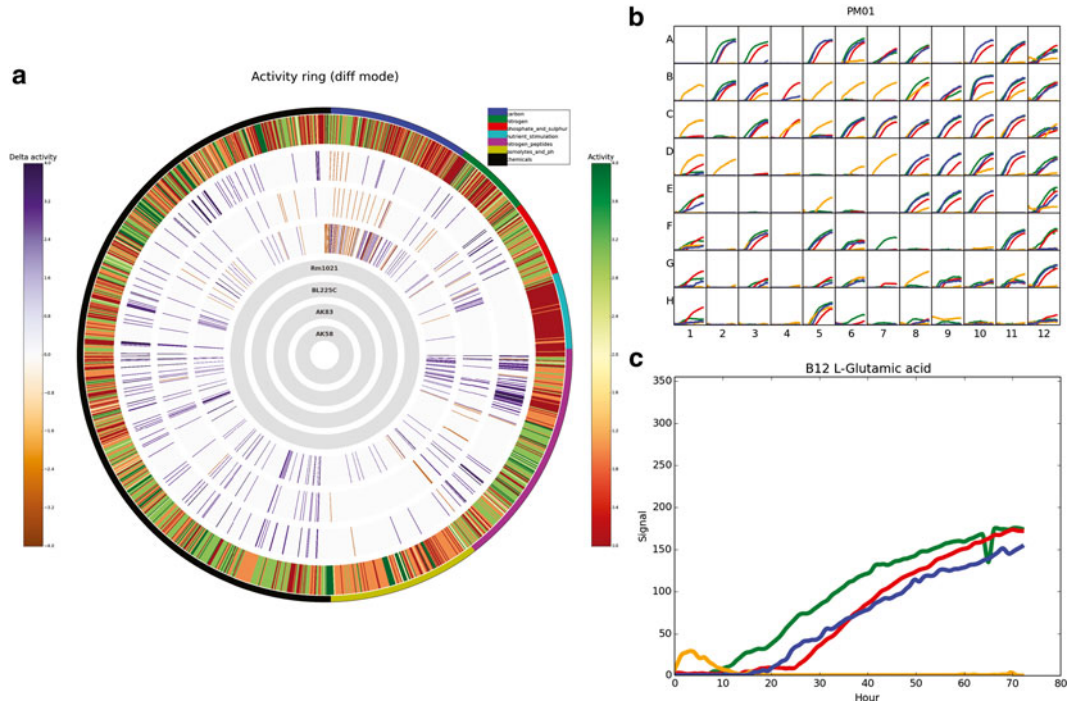

Fig. 3 PM experiments plots. (**a**) Activity rings: each *line* in the *outer rings* represents a single PM curve; the *color* in the *outer ring* is proportional to the AV value in the Rm1021 strain, the *color* of the other *three rings* represents the difference in the AV value between each strain and the Rm1021 strain, *purple* if the AV value is greater than the Rm1021 strain, *orange* otherwise. (**b**) Plate-wise plot of the curves belonging to plate PM01. (**c**) Curve plot for well B12 in plate PM01. Color codes: *red* Rm1021, *green* BL225C, *blue* AK83, *orange* AK58

- Run the following command to plot the single PM curves and plate-wise plots (Fig. 3b, c)

```
dphenome plot
```

As showed in Fig. 3a, strain Rm1021 appears to be less metabolically versatile than the other three strains. By looking at the curve plot in Fig. 3b, strain AK58 is incapable of growing on L-Glutamate as a carbon source. Since this compound is mapped to the KEGG compound database (KEGG ID C00025) it may be possible to relate this difference to some genetic variability.

3.4.5 Generate Statistics and Export Phenotypic Data

- Type the following to obtain statistics on the phenotypic data:

```
dphenome stats -a 3 -d 2
```

- Run the following command to export PM data in YAML/JSON format

```
dphenome export
```

Table 2
Proportion (%) of PM curves with AV>=2, divided for each PM compounds category

Category	AK58	AK83	BL225C	Rm1021
Carbon	32.29	29.69	38.54	34.90
Nitrogen	59.38	21.88	41.67	38.54
phosphate_and_sulphur	51.04	75.00	77.08	56.25
nutrient_stimulation	6.25	7.29	5.21	1.04
nitrogen_peptides	68.75	29.86	61.81	49.31
osmolytes_and_ph	29.69	16.67	22.40	20.31
Chemicals	51.25	49.17	56.15	44.17

As can be observed by looking at the `active_stats.tsv` file (Table 2), strain AK58 and BL225C seem to be more metabolically active on nitrogen sources as compared to the other strains.

3.5 Combined Analysis

3.5.1 Combine Genetic and Phenotypic Variability

The dape program is used to highlight the pathways that most probably contribute to the genetic and phenotypic variability in the four analyzed strains. The genetic variability of a KEGG pathway is defined as the ratio between the number of reactions that are differentially present in the input strains and the number of total reactions mapped to the pathway. The phenotypic variability of each KEGG compound that participates in a pathway is measured by looking at the differences in the AV value measured in each strain. The intersection of genetic and phenotypic variability can highlight probable causal links between genetic and phenotypic variability.

- Run the following command to run the combined analysis, using a threshold of 2 AV:

```
dape start -t 2
```

The analysis of the `combined_matrix.tsv` file (graphically represented in Fig. 4a), indicates that the previously analyzed L-GLUTAMATE participates in several pathways where a significant amount of genetic variability is present. Further inspection of these pathways may highlight which reactions are responsible for these observed phenotypic differences.

3.5.2 Generate Pathway Maps

- Run the following command to generate interactive KEGG pathway maps for each strain:

```
dape map -a
```

As shown in Fig. 4b, the Butanoate metabolism pathway (file `map00650.png`) shows an interesting difference between strain

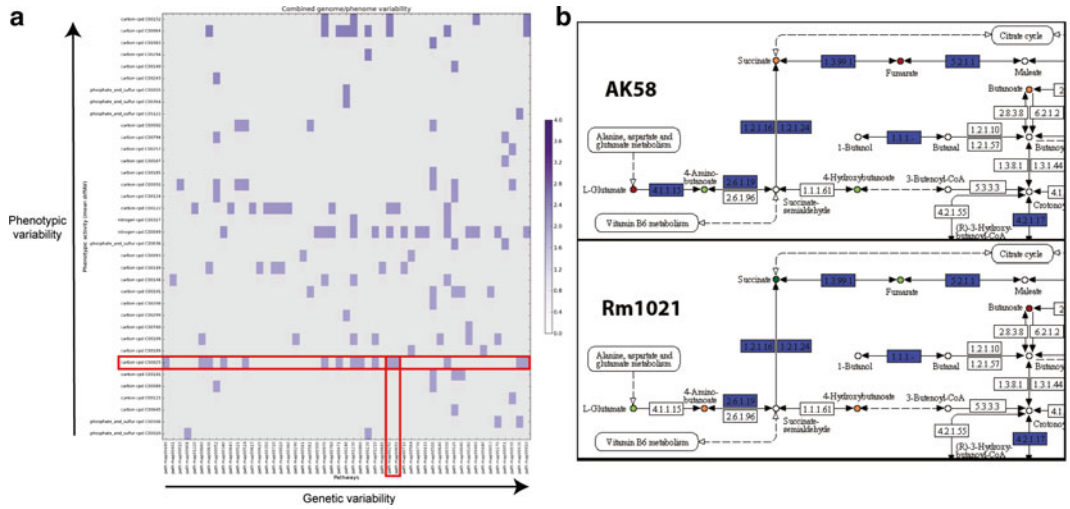

Fig. 4 Genotype/phenotype combined analysis. (**a**) Combined heatmap: the KEGG compounds are reported on the *y*-axis ordered by their phenotypic variability, KEGG pathways are reported on the *x*-axis ordered by their genotypic variability; *purple dots* are present when a compound participates in a given pathway, the color level being proportional to the phenotypic variability. (**b**) KEGG pathway maps of Butanoate metabolism in strain AK58 and Rm1021; *blue boxes* represent reactions present in the strain genome, while compounds are colored according to their AV value

AK58 (incapable of growing on L-Glutamate as a Carbon source) and strain Rm1021, as suggested by the output of the `dape start` command. The reaction that converts L-Glutamate into 4-Aminobutanoate (EC number 4.1.1.15) is indicated as present only in strain AK58. Since one molecule of gaseous carbon dioxide is removed from L-Glutamate by this enzyme, this reaction is irreversible, thus reducing the L-Glutamate pool available for the other reactions/pathways. This can be a possible explanation for the lack of growth of strain AK58 on this carbon source. Other putative genotype/phenotype links may be found after the inspection of the combined analysis.

References

1. Bayjanov JR, Molenaar D, Tzeneva V, Siezen RJ, van Hijum SA (2012) PhenoLink-a web-tool for linking phenotype to ~omics data for bacteria: application to gene-trait matching for Lactobacillus plantarum strains. BMC Genomics 13:170
2. Harper MA, Chen Z, Toy T et al (2011) Phenotype sequencing: identifying the genes that cause a phenotype directly from pooled sequencing of independent mutants. PLoS One 6:e16517
3. Karr JR, Sanghvi JC, Macklin DN et al (2012) A whole-cell computational model predicts phenotype from genotype. Cell 150:389–401
4. Mardis ER (2008) The impact of next-generation sequencing technology on genetics. Trends Genet 24:133
5. Caspi R, Foerster H, Fulcher CA et al (2008) The MetaCyc Database of metabolic pathways and enzymes and the BioCyc collection of Pathway/Genome Databases. Nucleic Acids Res 36:D623–D631
6. Kanehisa M, Goto S (2000) KEGG: kyoto encyclopedia of genes and genomes. Nucleic Acids Res 28:27–30
7. Overbeek R, Begley T, Butler RM et al (2005) The subsystems approach to genome annotation

and its use in the project to annotate 1000 genomes. Nucleic Acids Res 33:5691–5702
8. Zhang W, Li F, Nie L (2010) Integrating multiple 'omics' analysis for microbial biology: application and methodologies. Microbiology 156:287–301
9. Bochner BR, Gadzinski P, Panomitros E (2001) Phenotype microarrays for high-throughput phenotypic testing and assay of gene function. Genome Res 11:1246–1255
10. Li JL, Li MX, Deng HY, Duffy P, Deng HW (2005) PhD: a web database application for phenotype data management. Bioinformatics 21:3443–3444
11. Chang W, Sarver K, Higgs B et al (2011) PheMaDB: a solution for storage, retrieval, and analysis of high throughput phenotype data. BMC Bioinformatics 12:109
12. Vaas LA, Sikorski J, Hofner B et al (2013) opm: an R package for analysing OmniLog® phenotype microarray data. Bioinformatics 29(14):1823–1824
13. Vaas LAI, Sikorski J, Michael V, Göker M, Klenk HP (2012) Visualization and curve-parameter estimation strategies for efficient exploration of phenotype microarray kinetics. PLoS One 7:e34846
14. Biondi EG, Tatti E, Comparini D et al (2009) Metabolic capacity of *Sinorhizobium* (*Ensifer*) *meliloti* strains as determined by phenotype microarray analysis. Appl Environ Microbiol 75:5396–5404
15. Henry CS, DeJongh M, Best AA et al (2010) High-throughput generation, optimization and analysis of genome-scale metabolic models. Nat Biotechnol 28:977–982
16. Peleg AY, de Breij A, Adams MD et al (2012) The success of Acinetobacter species; genetic, metabolic and virulence attributes. PLoS One 7:e46984
17. Viti C, Decorosi F, Mini A, Tatti E, Giovannetti L (2009) Involvement of the *oscA* gene in the sulphur starvation response and in Cr(VI) resistance in *Pseudomonas corrugata* 28. Microbiology 155:95–105
18. Reed JL, Famili I, Thiele I, Palsson BO (2006) Towards multidimensional genome annotation. Nat Rev Genet 7:130–141
19. Galardini M, Mengoni A, Biondi EG et al (2014) DuctApe: a suite for the analysis and correlation of genomic and OmniLog™ phenotype microarray data. Genomics 103:1–10
20. Moriya Y, Itoh M, Okuda S, Yoshizawa AC, Kanehisa M (2007) KAAS: an automatic genome annotation and pathway reconstruction server. Nucleic Acids Res 35: W182–W185

Chapter 17

Genome-Wide Detection of Selection and Other Evolutionary Forces

Zhuofei Xu and Rui Zhou

Abstract

As is well known, pathogenic microbes evolve rapidly to escape from the host immune system and antibiotics. Genetic variations among microbial populations occur frequently during the long-term pathogen–host evolutionary arms race, and individual mutation beneficial for the fitness can be fixed preferentially. Many recent comparative genomics studies have pointed out the importance of selective forces in the molecular evolution of bacterial pathogens. The public availability of large-scale next-generation sequencing data and many state-of-the-art statistical methods of molecular evolution enable us to scan genome-wide alignments for evidence of positive Darwinian selection, recombination, and other evolutionary forces operating on the coding regions. In this chapter, we describe an integrative analysis pipeline and its application to tracking featured evolutionary trajectories on the genome of an animal pathogen. The evolutionary analysis of the protein-coding part of the genomes will provide a wide spectrum of genetic variations that play potential roles in adaptive evolution of bacteria.

Key words Sequence alignment, Positive selection, Intragenic homologous recombination, Adaptive evolution, Bacteria

1 Introduction

In the microbial evolution, natural selection and homologous recombination are two important driving forces for species innovation and adaptation to new niches. Genetic variations exerted by these evolutionary forces are often linked with special phenotypic alterations of microorganisms. A number of recent microbial studies on evolutionary genomics have highlighted the crucial roles of selection and recombination in the adaptive evolution of pathogenic bacteria and viruses, such as *Escherichia coli* [1], *Streptococcus* [2], Campylobacter [3], and Influenza A virus [4]. These studies have pointed out the immune and defense-associated genes are usually evolving more rapidly with obviously molecular evidence for natural selection pressure [1]. Particularly, some positively selected amino acid sites on these genes have been experimentally validated

Alessio Mengoni et al. (eds.), *Bacterial Pangenomics: Methods and Protocols*, Methods in Molecular Biology, vol. 1231, DOI 10.1007/978-1-4939-1720-4_17,

to be involved in antibiotic resistance and other pathogen–host interactions. Genome-wide detection of selection can provide microbiologists valuable insights into the molecular mechanism of adaptive evolution in bacteria.

To detect featured genetic alterations on protein-coding DNA sequences, currently, we can access to many computational methods and tools developed for the single-gene analysis. The test of positive selection is an alignment-based statistical approach that can provide convincing molecular evidence for Darwin's theory of natural selection on protein-coding sequences [5]. For the analysis of positive selection, there is a key indicator—ω, the ratio of nonsynonymous nucleotide substitutions (d_N) to that of synonymous substitutions (d_S). Estimation of the ratio ω is a powerful statistical approach to test evolutionary selective pressure acting on protein coding genes: $\omega = 1$, <1, >1 indicate neutral evolution, purifying (negative) selection, and positive (adaptive) selection, respectively [6]. Combined with the codon models of variable ω ratio among sites, this approach can further infer a small fraction of amino acid sites which are evolving under strong selective pressure and involved in the evasion of host immunity [1].

With the wide application and low cost of next generation sequencing technologies (i.e. Illumina, 454, SOLiD, and Ion Torrent), a huge amount of genome sequencing data are being released from the main data repositories. Undoubtedly, it provides us sufficient genetic information to characterize the mechanisms of bacterial adaptive evolution in a new dimension. For example, there are 1,606 whole genome shotgun sequencing projects of *Escherichia coli* deposited in the GenBank database (updated in November 2013). However, it's a great challenge for microbiologists with limited computational knowledge to explore genome-wide evolutionary characterization and uncover functional genes evolving rapidly using their sequenced and publicly available data resources.

In this chapter, we will present an automatically computational pipeline to investigate the evidence for positive selection and homologous recombination on the protein-coding genomes of bacteria. It implements a series of operational analysis tasks: gathering of genome sequences, clustering of orthologous genes, multiple sequence alignment, estimation of alignment quality, test for intragenic recombination, reconstruction of maximum likelihood phylogenetic tree, detection of genes subjected to positive selection, and biological interpretation of positively selected genes.

2 Materials

All software tools used in this computational protocol can be downloaded for free. To implement the automation of phyloge-

netic analysis of large data sets, for example, 1,000+ sequence alignments, you need custom wrapper scripts to prepare input files in correct data formats and call relevant stand-alone computational programs in batch mode. The scripts mentioned in the chapter are written in Perl and some of them require pre-installation of the BioPerl modules. All scripts and example sequence files are available on GitHub (https://github.com/tigerxu/GWDSR).

Typographical conventions: `Monospaced text` will be used for all commands to be issued by the user. The command prompt will be represented with the `$` symbol.

2.1 Prodigal

To date, a number of gene finding programs for microbial genomes have been developed, such as Glimmer [7], GenemarkHMM [8], MED [9], and Prodigal [10]. We will use Prodigal (Prokaryotic Dynamic Programming Genefinding Algorithm) in the example. Prodigal is a fast microbial (bacterial and archaeal) gene recognition tool which has three improved advantages on prediction of gene structure, recognition of translation initiation site, and reduced false positives [10]. The web server of Prodigal is available at http://code.google.com/p/prodigal/. The Prodigal program can be run locally under operating systems Linux, Windows, and Mac. DNA sequences in FASTA, GenBank, and EMBL formats can be used as input file for Prodigal.

2.2 CD-HIT

CD-HIT can cluster similar protein/DNA sequences into clusters that meet a user-defined similarity threshold [11]. The program is included in the CD-HIT package, which is freely available from http://weizhong-lab.ucsd.edu/cd-hit/. The CD-HIT package is a command-line software and can be run on Linux systems or other systems that support C++.

2.3 T-Coffee

T-Coffee is a multiple alignment package for DNA, RNA, and amino-acid sequences [12]. T-Coffee application online can be accessed at http://www.tcoffee.org/Projects/tcoffee/. The stand-alone program of T-Coffee is also available on the above webpage. T-Coffee can be run on UNIX-like operating platforms (i.e. Linux, MacOSX, and Cygwin on Windows system).

2.4 PAL2NAL

The PAL2NAL program is designed to convert a multiple sequence alignment of proteins and the corresponding DNA sequences into a codon-based DNA alignment [13]. The output codon alignments can be used to identify evolutionary evidence for homologous recombination and positive selection acting on genes. In the case of a large-scale sequence analysis, the stand-alone program of PAL2NAL, which is written in Perl, is available at http://www.bork.embl.de/pal2nal/.

2.5 Gblocks

To improve the performance of positive selection scanning, we should detect and remove unreliable regions in the inferred multiple sequence alignments. The alignment confidence method Gblocks can filter incorrectly aligned columns and divergent regions in a DNA or protein sequence alignment [14]. The Gblocks program is available at http://molevol.cmima.csic.es/castresana/Gblocks.html and can be run on Linux, MacOSX, and Windows operating systems.

2.6 GARD

GARD is a genetic algorithm for detecting evidence of recombination breakpoints in a multiple sequence alignment [15]. Running GARD in parallel is computationally intensive and requires a distributed computing environment (message passing interface, MPI) in a computer cluster. GARD is implemented in the HyPhy package and also on the Datamonkey webserver (http://www.datamonkey.org/) [16]. The web server can allow an input alignment file per submission and thus not suitable for recombination detection of hundreds of alignments. However, it is possible to download and install the latest HyPhy version at http://www.hyphy.org and run GARD via command line for many data sets of DNA alignments on the Linux cluster.

2.7 PhyML

The PhyML program can support fast reconstruction of maximum likelihood (ML) phylogenetic trees [17]. You can upload multiple data sets in PHYLIP format to the PhyML web server http://atgc.lirmm.fr/phyml/ for tree reconstruction. The binary file of PhyML can be downloaded from the above webpage and installed on Linux, MacOS, and Windows systems. The command-line PhyML interface is well-suited for running program in batch mode.

2.8 PAML

PAML (Phylogenetic Analysis by Maximum Likelihood) is a program package for phylogenetic analysis of DNA or protein sequences using ML [18]. The *codeml* program in the PAML package is a commonly used analytical tool for test of adaptive molecular evolution and identification of amino acid sites under diversifying selection. *codeml* can estimate variable ω ratios (nonsynonymous/synonymous, or d_N/d_S) among amino acid sites in a protein based on codon-substitution models [6]. PAML executables for Linux/Mac OSX/Windows are available for academic use at http://abacus.gene.ucl.ac.uk/software/paml.html.

2.9 Phyre2

Phyre2 (Protein Homology/analogy Recognition Engine V2.0) is a widely used web server for prediction of protein structure [19]. The Phyre2 server is freely available for academic use (http://www.sbg.bio.ic.ac.uk/phyre2). You need to copy and paste the tested protein sequence on the web server. The server can automatically predict a three-dimensional (3D)

structure model for a complete protein sequence if you choose the option "Intensive" as modeling mode. A PDB formatted model of your protein can be returned in the specified email address and visualized by an academic version of PyMol (http://pymol.org/educational/).

3 Methods

Here we show a step by step computational protocol for genome-wide detection of intragenic homologous recombination and positive Darwinian selection of a single bacterial species. Of course, the analysis pipeline is readily applicable to the protein-coding genomes of those closely related prokaryotic organisms within the same genus. The datasets of sequenced genomes used in the following example are from a Gram-negative animal pathogen *Actinobacillus pleuropneumoniae* and contain 12 genomes [20].

3.1 Download Genome Datasets

The genome nucleotide sequences of 12 *A. pleuropneumoniae* strains are first downloaded from NCBI (http://www.ncbi.nlm.nih.gov/nucleotide) and saved in individual files. The GenBank accession numbers are as follows: CP000569 (L20), CP000687 (JL03), CP001091 (AP76), ADOD00000000 (4074), ADOE00000000 (S1536), ADOF00000000 (M62), ADOG00000000 (Femφ), ADOI00000000 (CVJ13261), ADOJ00000000 (D13039), ADOK00000000 (56153), ADOL00000000 (1096), ADOM00000000 (N273) (*see* **Note 1**). Strain name is shown in parentheses. For the records with complete genome sequences, click on the "Send:" button at the top of the webpage to open the save options for this genome. Under "Choose Destination" we select the "File". In the Format box, we choose "FASTA". Then click the "Create File" button to download the sequences to a text file on your computer. For the draft genome assemblies, click on the WGS link at the end of the page and then click on "Download" button to download a compressed file with the extension ".fsa.1.gz". After uncompressing, sequence data must be in FASTA format and optionally using the uniform filename extension ".fna".

3.2 Gene Calling

Next we apply the Prodigal program to locate all protein-coding sequences (CDSs) on the collected genome sequences.

```
$ prodigal -i input_genome.fna -a orfs.faa -d
orfs.ffn -o orfs.gbk -m -q
```

The set of the above command-line arguments creates three output files. The default Prodigal output specified by the argument -o is a Genbank-like feature table of predicted CDSs. Nucleotide and translated amino acid sequences of predicted genes are output to the files in FASTA format by the arguments -d and -a, respectively.

To meet the styles of sequence data required by the subsequent phylogenetic analysis, the FASTA headers of sequence records are simplified using fasta_header_rename.pl. We also need to delete the "*" character denoting the stop codon (i.e. TAG, TGA, TAA) per protein in the corresponding output files. We can optionally remove the short genes with less than 50 amino acids using aaseq_filtering.pl. The threshold of the minimum sequence length is set by the -t option. The usage of the perl wrapper scripts is showed below (*see* **Note 2**).

```
$ perl fasta_header_rename.pl -i orfs.faa -d genome_id -o renamed_header.fasta
$ perl aaseq_filtering.pl -i renamed_header.fasta -t 50 -o genome_id.faa
```

A few simplified header lines of FASTA sequence records are shown below:

>genome_id_0001

>genome_id_0002

3.3 Clustering of Orthologous Genes

To obtain orthologous gene groups among multiple stains within a single species, the protein sequences of all CDSs per genome (12 .faa files) are merged into a single text file in FASTA format. We then apply the program CD-HIT for sequence clustering. A preliminary set of orthologous genes is defined by the following criteria: an amino acid sequence identity of >80 % over at least 80 % of the representative sequence (i.e. longest sequence) in a group.

```
$ cat *.faa > all_genome_gene.faa
$ cd-hit -i all_genome_gene.faa -o group -c 0.8 -aL 0.8 -g 1 -d 0
```

In the command-line options of *cd-hit*, the -c option is used to set a sequence identity threshold. The -aL option denotes the alignment must cover 80 % of the representative sequence within a group. The -g 1 option is recommended as it can assign a sequence into the most similar group that meets the cut-off values. CPU time in this case is ~20 min. We can obtain three output files: group, group.bak.clstr, and group.clstr. All the representative sequences of each cluster are stored in the output file group. The resultant file with the extension .clstr can be used to produce a list of orthologous genes that are single-copy conserved genes present in all genomes using a perl script ortholog_list.pl. Gene identifiers affiliated to an orthologous group are displayed in a line and exported in a tabular form. The genes belonging to the bacterial core genome often play the roles in the fundamental metabolic activities of cells. We subsequently extract the protein sequences of each orthologous group and separately save the sequences within a group into FASTA format files in a newly created directory using

extract_ortholog_seq.pl. Consistently, a directory composed of files containing the DNA or protein sequences of orthologous gene groups can be created, respectively.

```
$ perl ortholog_list.pl -i group.clstr -t 12 -o ortholog_list.txt
$ perl extract_ortholog_cluster_seq.pl -i all_genome_gene.faa -l ortholog_list.txt -o protein_cluster_dir
$ perl extract_ortholog_cluster_seq.pl -i all_genome_gene.ffa -l ortholog_list.txt -o gene_cluster_dir
```

3.4 Functional Annotation of Orthologous Group

To provide biological interpretation of the individual orthologous group detected later, we can extract the representative protein sequences of orthologous groups for functional annotation and classification. Since gene annotation is beyond the scope of this chapter, we will only show general approaches but might be slower than the other commonly used methods. The *blastall* program within the BLASTALL package (ftp://ftp.ncbi.nlm.nih.gov/blast/) [21] is run locally to search the set of representative protein sequences against two commonly used databases NCBI NR (ftp://ftp.ncbi.nlm.nih.gov/blast/db/) and COG (ftp://ftp.ncbi.nih.gov/pub/COG/COG/) [22] with an E_value cutoff of 1e-10, respectively. For the BLAST output report based on NCBI NR, the query representative gene is annotated by adding the functional description of the best hit. On the other hand, for the BLAST output report based on COG, the query is classified into a functional category in which the best hit is affiliated.

```
$ perl extract_representative_seq.pl -i group -l ortholog_list.txt -o reference.faa
$ blastall -p blastp -d path_to_NCBI_NR_database -i reference.faa -o gene2NCBINR.blastp.txt -e 1e-10 -v 5 -b 5
$ blastall -p blastp -d path_to_COG_database -i reference.faa -o gene2COG.blastp.txt -e 1e-10
$ perl parse_annotation_ncbiNR.pl -i gene2NCBINR.blastp.txt -o gene.annotation.tab
$ perl parse_category_COG.pl -i gene2COG.blastp.txt -l path_to_COG_category_file -o gene_category.tab
$ join -t $'\t' gene_annotation.tab gene_category.tab > gene.combined.tab
```

Based on the BLAST commands for sequence similarity searching and the perl scripts that parse the BLAST output report, we can then combine functional annotation and classification of orthologous groups and create a joint tabular form. For example:

OG_id	Function	COG_id	COG_category
OG0007	potassium efflux protein KefA	M	Cell membrane
OG0011	Beta-galactosidase	G	Carbohydrate
OG0012	ribonuclease E	J	Translation

3.5 Multiple Sequence Alignment

In the preceding Subheading 3.3, we have obtained nucleotide and amino acid sequences of all orthologous gene groups. To reduce the effect of incorrect insertions/deletions (indels) on the codon alignments, multiple sequence alignments are initially carried out by using amino acid sequences of each ortholog group. The protein sequence file per ortholog group can be directly used as input for the program T-Coffee and the default output is a sequence alignment in the ClustalW format. We can also use a perl wrapper script below to execute the command of T-Coffee for the alignments of a lot of sequence files on the terminal.

```
$ perl run_tcoffee.pl -i protein_cluster_dir
```

Running the above script will export three output files per ortholog group in the current directory. The output files with extension .aln contain the resulting sequence alignments in the ClustalW format. The aligned amino acid sequences together with the corresponding nucleotide sequences of each ortholog group are converted into DNA alignments at the codon level using the script run_pal2nal.pl (*see* **Note 3**). Place the executable PAL2NAL script in the same directory as the wrapper script and two subdirectories composed of files of protein sequence alignments and the relevant DNA sequences. Alternatively, you can add the full path of the PAL2NAL script on your computer to the wrapper script. The output files in the newly created directory should be the aligned DNA sequences at the codon level.

```
$ perl run_pal2nal.pl -p protein_alignment_
dir -d gene_cluster_dir -c output_dir
```

3.6 Removal of Unreliable Alignment Regions

To further improve alignment quality for reducing the false-positive rate of the positive selection analysis, we should remove very divergent regions that are useless for phylogenetic reconstruction by applying the command-line program Gblocks. The sequence type being codon (the option -t=c) and the default relaxed settings as defined by Talavera [23] are adopted in this example. We show the command-line arguments below for a single job and also a perl wrapper script for running Gblocks in batch mode.

```
$ Gblocks input_alignment_file -t=c -e=.fa
-b2=9 -b3=10 -b4=5 -b5=h
$ perl run_gblocks.pl -i directory_codon_alignments
```

The high quality sequence alignment is output to a FASTA format file with the extension .fa (*see* **Note 4**). You will also obtain an .html file visualizing the original alignment with the selected reliable positions highlighted and a description of the parameters employed. An example of reliable blocks in the resulting multiple sequence alignment is shown on Fig. 1. Based on the above steps, we now get a correct alignment format as input for the next test of recombination.

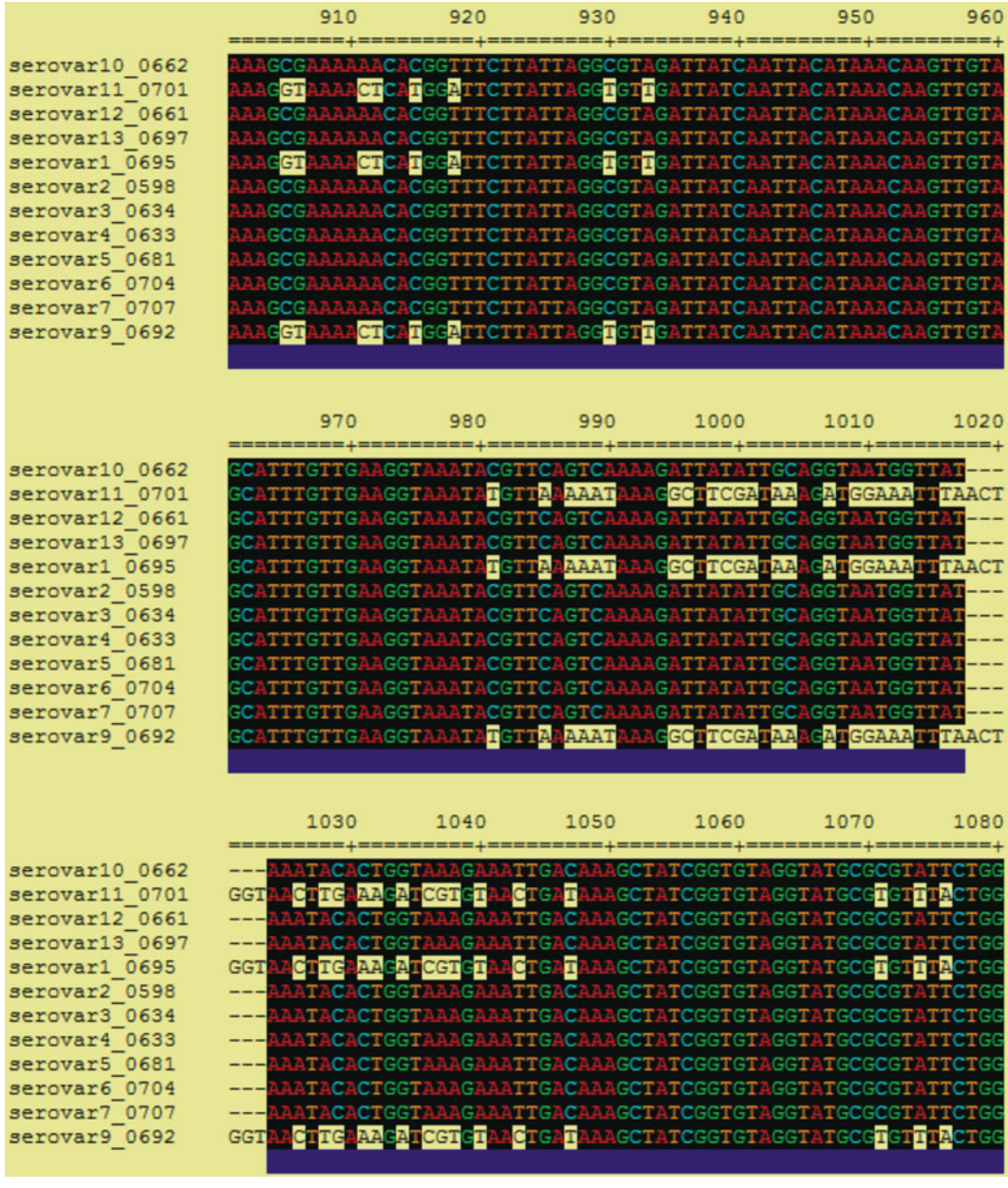

Fig. 1 An example of conserved alignment blocks detected by Gblocks

3.7 Detection of Intragenic Homologous Recombination

As recombining can lead to phylogenic incongruence and have a negative effect on the selection detection, we should first identify the recombination breakpoints in the alignment and then reconstruct a single phylogeny for each nonrecombinant fragment [24]. We will apply GARD on the alignments obtained in the above step through a MPI-enabled command line version of HyPhy. You need to prepare a HyPhy wrapper batch file (run_GARD.bf) that includes a set of input parameters required by running GARD.bf on a large number of DNA alignment files. The set of the recommended default options are as follows: A general time-reversible (GTR) model of nucleotide substitution; general discrete with three rate classes [24]. Then you should create a text file named as input_path.txt which should contain the full path to each tested alignment file per line. Place both newly created files in the same directory as the alignment files and execute the following command on the terminal.

```
$ mpirun -np 4 HYPHYMPI run_GARD.bf
```

You will be prompted to type the filename input_path.txt to run the GARD test. Four parallel processors are set by the -np option. For each alignment, GARD generated four output files with the extensions .html, _finalout, _splits, and _ga_details, respectively, in the same directory as the inputs. Each .html file gives us a summary of the analysis for each alignment. For example, in Fig. 2, GARD reports the best model with three breakpoints at the positions 291, 498, and 735 in the alignment of OG0539. In addition, the HTML page also presents the log-likelihood, c-AIC, estimated rate distribution, and nucleotide substitution rate matrix.

To further confirm whether a recombination breakpoint is significant or not, we carry out a post-processing statistical analysis by applying the batch file GARDProcessor.bf in HyPhy. The significant breakpoints are inferred based on the Shimodaira-Hasegawa (SH) test, with a p-value cutoff of 0.05. To implement this statistical approach, we should create a template batch file GARDProcessor_temp.bf specifying full paths to the tested alignment and the output file _splits generated by the above GARD test. Run a perl script run_GARDProcessor.pl under the same directory as the directory containing input and output files by GARD. You also need to set the -p option by obtaining the full path to the current working directory.

```
$ perl run_GARDProcessor.pl -i gard_directory -t
GARDProcessor_temp.bf   -p   full_path_to_working_
directory
```

The output files with extension .SH.txt store the information generated by GARDProcessor.bf. At the end of the output report, raw and adjusted p-values using the SH test are shown for the left

Processed alignment **/home/fedora/data/tcoffee/output_dir/gard/OG0539.improved.aln** with 12 sequences and 1047 sites.

Processor time taken: 209 seconds.

Results

GARD found evidence of 3 breakpoints

BPs	c-AIC	Δ c-AIC	Segments
0	4312.23		1-1047
1	4232.46	79.7702	498
2	4192.31	40.1517	291 498
3	4168.51	23.8025	291 498 735
4	4168.51	0	291 498 735

Substitution process parameters

Nucleotide Model (012345) Fit Results

Log likelihood : -2122.01
c-AIC : 4312.23

Using general discrete distribution of rates across sites
Rate : 0.274 Weight : 0.897
Rate : 4.405 Weight : 0.102
Rate : 313.655 Weight : 0.001

Nucleotide substitution rate matrix

	A	C	G	T
A	*	0.210083	1	0.195922
C	-	*	0.269098	1.43648
G	-	-	*	0.370778
T	-	-	-	*

Fig. 2 A HTML-format example report of the GARD analysis

and right segment of each breakpoint inferred by GARD. We can summarize all recombining genes with their confirmed breakpoints into a tabular form using summary_breakpoint.pl.

```
$ perl summary_breakpoint.pl -i gard_dir -o breakpoint.tab
```

A few lines from the output breakpoint.tab are shown below:

OG_ID	Length (bp)	Breakpoint sites	Number of Breakpoint
OG0500	1,098	None	0
OG0501	1,095	None	0
OG0502	1,095	291	1
OG0503	1,095	None	0
OG0504	1,092	None	0
OG0505	1,089	None	0
OG0506	1,083	389 729	2
OG0507	1,089	426 681	2
OG0508	1,089	None	0
OG0509	1,080	171	1

Each gene alignment showing significant evidence of homologous recombination will be partitioned based on the inferred breakpoints in the third column using the wrapper script alignment_partition.pl. Meanwhile, the nonrecombinant fragment alignments should be written into the PHYLIP format required by the programs PhyML and PAML. The newly created files of nonrecombinant gene/fragment alignments in the PHYLIP sequential format will be exported to another output directory (*see* **Note 5**).

```
$ perl alignment_partition.pl -i gard_dir -t
breakpoint.tab -o phylip_dir
```

3.8 Reconstruction of Phylogenetic Tree

A tree topology will be made for each of the alignments of complete genes or gene segments. The command-line PhyML interface is employed for estimating ML phylogenies. A general time-reversible (GTR) model of nucleotide substitutions with the ML estimates for gamma distributed rate heterogeneity of four categories (Γ_4) and a proportion of invariable sites is set for reconstruction of all trees. To run PhyML with these parameters in the terminal, a set of command-line arguments is specified: -m GTR -f m -t e -v e -c 4. We use a wrapper script runPhyML.pl to make trees for a number of alignments in the PHYLIP format.

```
$ perl runPhyML.pl -i phylip_dir -o MLtree_dir
```

For each of the input DNA alignments, PhyML will generate two output files: the ML tree file and the model parameter file with the extensions _tree.txt and _stats.txt, respectively. The output ML tree is in standard Newick format. The estimates of branch support in the tree should be removed and the new tree files are exported to a specified output directory. An example ML tree is shown below. The numbers followed by the colon are branch lengths that are allowed as the input tree topology for the PAML analysis.

```
((((((seq1:0.0001,(seq2:0.0001,(seq3:0.0001,s
eq4:0.0001):0.0001):0.0001):0.0001,seq5:0.0001):0
.1055,(seq6:0.0056,seq7:0.0009):0.0018):0.0009,se
q8:0.0145):0.0672,seq9:0.0667):0.0009,seq10:0.
0001):0.00095,seq11:0.0001,seq12:0.0001);
```

3.9 Detection of Selection

Before this step, we should have obtained an alignment file and the corresponding tree file for positive selection scanning. Next we will apply the codeml program within the PAML v4.3 package in the example. We should create a "control file" including specific codeml's variables for each gene/fragment. Based on the topology of the ML tree per gene (or nonrecombinant fragment) alignment, we can apply two site-specific models that allow variable ω ratios among codons: M1a (NearlyNeutral) and M2a (PositiveSelection). M1a is a null hypothesis model which specifies two classes of sites: conserved sites with $\omega < 1$ and neutral sites with $\omega = 1$. The model M2a adds an extra site class for a fraction of positively selected amino acid sites with $\omega > 1$. A likelihood ratio test (LRT) compares M1a with M2a to test for the sites subject to positive selection and calculates the likelihood statistic ($2\Delta l$) with the χ^2 distribution with two degrees of freedom (d.f.). The Bayes empirical Bayes approach is employed to identify positively selected sites under the likelihood framework [25]. We show the script commands below to run the codeml analysis on multiple datasets and extract parameter estimates from the output files with extensions .M1 and .M2 using custom scripts.

```
$ perl run_codeml_M1.pl -s phylip -t MLtree_
dir -o M1a
$ perl run_codeml_M2.pl -s phylip -t MLtree_
dir -o M2a
$ perl parse_codeml_M1a.pl -i M1a -o M1a.tab
$ perl parse_codeml_M2a.pl -i M2a -o M2a.tab
```

The parsing information of the null model M1a and alternative model M2a is merged into a tabular form and a few lines are shown as an example in the Table 1. For the positively selected genes with highly significant LRT (p-value < 0.01), the statistic ($2\Delta l$) should be less than 9.21 that is 1 % χ^2 critical value with d.f. = 2. We can also observe the proportion (p) of the amino acid sites under positive selective pressure and the related d_N/d_S ratio (ω) among these adaptive evolving sites. The positively selected sites are identified if their posterior probability is greater than 95 %.

3.10 Biological Interpretation of PS Genes

In the Table 1, two genes (OG0517 and OG0524) are detected to be under strongly positive selected pressure with low p-values ($p < 0.001$). The gene OG0517 encodes an outer membrane P2-like protein OmpP2 that is a beta barrel porin [26], and OG0524 encodes for a TDP-Fuc4NAc:lipid II Fuc4NAc transferase that is involved in the synthesis of an enterobacterial common

Table 1
Summary of parameter estimates and identified positively selected sites in the genes tested

OG_ID	*l* (M1a)	*l* (M2a)	2Δ*l*	*p*	ω	Positively selected sites
OG0500	−1,587.36	−1,587.36	0	0	1	
OG0501-1	−663.21	−663.21	0	0	1	
OG0501-2	−417.89	−420.56	−5.33	0	1	
OG0501-3	−795.07	−796.98	−3.83	0	58.583	
OG0517	−1,706.18	−1,697.59	**17.18**	0.089	8.631	327, 334, 338, 341
OG0524	−1,658.35	−1,647.01	**22.68**	0.035	14.226	70, 92, 182, 183, 289
OG0542	−1,492.40	−1,488.34	8.12	0.003	119.296	

antigen-like glycoconjugate [27]. According to the function annotation, both bacterial genes play potential roles in the interactions with the host immune and defense systems. For instance, OmpP2 of *A. pleuropneumoniae* has been experimentally confirmed to be essential for in vivo survival by signature-tagged mutagenesis and also an immunogenic surface antigen by the immunoproteomic approach [28, 29]. Consistently, four amino acid residues (327, 334, 338, and 341) of *A. pleuropneumoniae* OmpP2 are subject to intense positive selective pressure. To visualize spatial confirmation of these particular residues, you could predict the 3D structural model of the OmpP2 protein by submitting the amino acid sequence to the Phyre2 server. For this special case, we can also predict the trans-membrane structure of the beta porin OmpP2 by applying the web server of PRED-TMBB (http://bioinformatics.biol.uoa.gr/PRED-TMBB/). The protein structure model in the PDB format can be visualized by PyMol (Fig. 3). Herein, we highlighted four positively selected sites in the orange spheres. Combined with the prediction generated by PRED-TMBB, all these four residues were found to be located on an extracellular loop in the C-terminus of OmpP2, perhaps associated with potential antigenic epitope. Detection of these adaptive sites and the relevant functional genes of bacteria should provide a genetic context for further research into the mechanisms of immune invasion and the pathogen–host interaction.

4 Notes

1. Gene calling and annotation of bacterial genome sequencing projects are usually carried out by different research groups around the world. For the prediction of protein-coding genes,

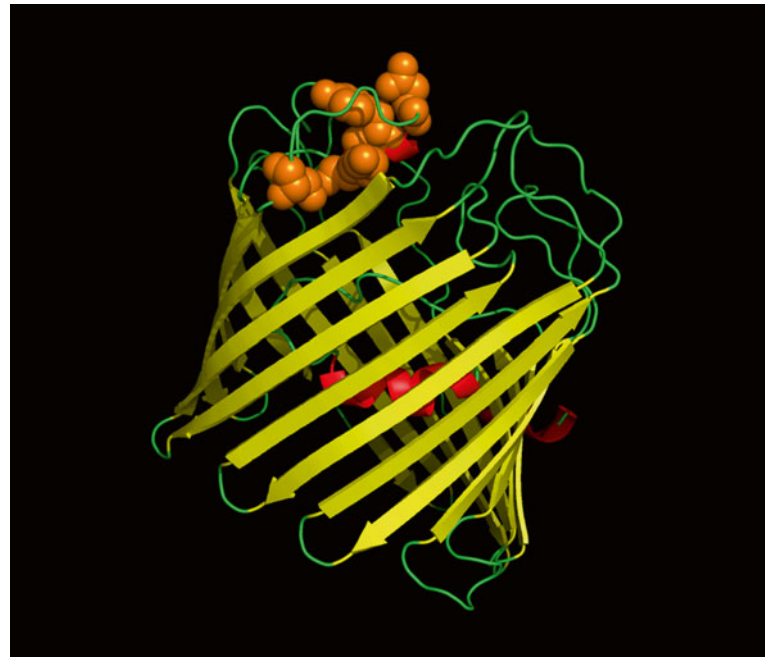

Fig. 3 Three-dimensional structural models of *A. pleuropneumoniae* beta barrel porin OmpP2. Secondary structural elements are colored: helix in red, sheet in yellow, and loop in green. Orange spheres denote amino acid sites that are subject to strong positive selection (posterior probability > 95 %)

distinct computational approaches probably generate inconsistent coordinates of translation initiation sites and also the number of genes [10]. In such situation, we recommend you to download the original genome nucleotide sequences for all the subsequent analyses but not those existing gene boundaries recorded in the GenBank database.

2. To create correct input data format required by PhyML and PAML, the sequence name must start with a letter and has no more than 20 characters. In addition, very short sequences will affect the power of the likelihood ratio test (LRT) when positive selection scanning is performed [30]. Thus, we recommend you to remove the genes less than 50 codons using a perl script aaseq_filtering.pl.
3. The quality of sequence alignment is a major factor to interfere with positive selection scanning, especially estimation of positively selected sites. For most currently used alignment programs, it's possible to place nonhomologous amino acids into the same column [31]. No sequence aligner is perfect. We could use T-Coffee to combine results generated by several alignment methods, e.g. PRANK, MUSCLE, MAFFT, and ClustalW, to obtain a high quality alignment.

4. Identification and removal of incorrectly aligned regions can increase the accuracy of positive selection inference [32]. It's highly recommended to filter unreliable columns by the alignment confidence methods, such as GBLOCKS. For GBLOCKS used in this case, the output file with an extension defined by the -e option is not a standard FASTA format. You should remove the blank characters per line from each file. Finally, we advise to manually check the resulting codon alignments using the visualization tools for sequence alignment, e.g. MEGA4 (http://www.megasoftware.net/mega4/mega.html).
5. We have paid attention to the position of the resulting breakpoints when we partition the sequence alignment using the script alignment_partition.pl. As improper indels present in the alignment will mislead to the inference for positive selected genes or amino acid sites, the extracted codon alignment partitions should be consistent with the original reading frame in a protein.
6. We recommend you to install and test the programs and wrapper scripts mentioned in this case on a Linux-like operating system. If you do not know how to install or configure the PATH environment variables of the program binary files, ask your system administrator.

Acknowledgments

This work was supported by the National Basic Research Program of China (973 Program, 2012CB518802).

References

1. Petersen L, Bollback JP, Dimmic M et al (2007) Genes under positive selection in *Escherichia coli*. Genome Res 17:1336–1343
2. Lefébure T, Stanhope MJ (2007) Evolution of the core and pan-genome of *Streptococcus*: positive selection, recombination, and genome composition. Genome Biol 8:R71
3. Lefébure T, Stanhope MJ (2009) Pervasive, genome-wide positive selection leading to functional divergence in the bacterial genus *Campylobacter*. Genome Res 19:1224–1232
4. Lam TT, Hon CC, Pybus OG et al (2008) Evolutionary and transmission dynamics of reassortant H5N1 influenza virus in Indonesia. PLoS Pathog 4:e1000130
5. Yang Z, Bielawski JP (2000) Statistical methods for detecting molecular adaptation. Trends Ecol Evol 15:496–503
6. Yang Z, Nielsen R, Goldman N et al (2000) Codon-substitution models for heterogeneous selection pressure at amino acid sites. Genetics 155:431–449
7. Delcher AL, Bratke KA, Powers EC et al (2007) Identifying bacterial genes and endosymbiont DNA with Glimmer. Bioinformatics 23:673–679
8. Lukashin AV, Borodovsky M (1998) GeneMark.hmm: new solutions for gene finding. Nucleic Acids Res 26:1107–1115
9. Zhu H, Hu GQ, Yang YF et al (2007) MED: a new non-supervised gene prediction algorithm for bacterial and archaeal genomes. BMC Bioinform 8:97
10. Hyatt D, Chen GL, Locascio PF et al (2010) Prodigal: prokaryotic gene recognition and translation initiation site identification. BMC Bioinform 11:119
11. Li W, Godzik A (2006) Cd-hit: a fast program for clustering and comparing large sets of protein or nucleotide sequences. Bioinformatics 22:1658–1659

12. Notredame C, Higgins DG, Heringa J (2000) T-Coffee: a novel method for fast and accurate multiple sequence alignment. J Mol Biol 302:205–217
13. Suyama M, Torrents D, Bork P (2006) PAL2NAL: robust conversion of protein sequence alignments into the corresponding codon alignments. Nucleic Acids Res 34:W609–W612
14. Castresana J (2000) Selection of conserved blocks from multiple alignments for their use in phylogenetic analysis. Mol Biol Evol 17:540–552
15. Kosakovsky Pond SL, Posada D, Gravenor MB et al (2006) GARD: a genetic algorithm for recombination detection. Bioinformatics 22:3096–3098
16. Pond SL, Frost SD, Muse SV (2005) HyPhy: hypothesis testing using phylogenies. Bioinformatics 21:676–679
17. Guindon S, Gascuel O (2003) A simple, fast, and accurate algorithm to estimate large phylogenies by maximum likelihood. Syst Biol 52:696–704
18. Yang Z (2007) PAML 4: phylogenetic analysis by maximum likelihood. Mol Biol Evol 24:1586–1591
19. Kelley LA, Sternberg MJ (2009) Protein structure prediction on the Web: a case study using the Phyre server. Nat Protoc 4:363–371
20. Xu Z, Chen X, Li L et al (2010) Comparative genomic characterization of *Actinobacillus pleuropneumoniae*. J Bacteriol 192: 5625–5636
21. Altschul SF, Gish W, Miller W et al (1990) Basic local alignment search tool. J Mol Biol 215:403–410
22. Tatusov RL, Fedorova ND, Jackson JD et al (2003) The COG database: an updated version includes eukaryotes. BMC Bioinform 4:41
23. Talavera G, Castresana J (2007) Improvement of phylogenies after removing divergent and ambiguously aligned blocks from protein sequence alignments. Syst Biol 56:564–577
24. Pond SL, Murrell B, Poon AF (2012) Evolution of viral genomes: interplay between selection, recombination, and other forces. Methods Mol Biol 856:239–272
25. Yang Z, Wong WS, Nielsen R (2005) Bayes empirical bayes inference of amino acid sites under positive selection. Mol Biol Evol 22:1107–1118
26. Xu Z, Chen H, Zhou R (2011) Genome-wide evidence for positive selection and recombination in *Actinobacillus pleuropneumoniae*. BMC Evol Biol 11:203
27. Banks KE, Fortney KR, Baker B et al (2008) The enterobacterial common antigen-like gene cluster of *Haemophilus ducreyi* contributes to virulence in humans. J Infect Dis 197:1531–1536
28. Chung JW, Ng-Thow-Hing C, Budman LI et al (2007) Outer membrane proteome of *Actinobacillus pleuropneumoniae*: LC-MS/MS analyses validate *in silico* predictions. Proteomics 7:1854–1865
29. Sheehan BJ, Bossé JT, Beddek AJ et al (2003) Identification of *Actinobacillus pleuropneumoniae* genes important for survival during infection in its natural host. Infect Immun 71:3960–3970
30. Anisimova M, Bielawski JP, Yang Z (2001) Accuracy and power of the likelihood ratio test in detecting adaptive molecular evolution. Mol Biol Evol 18:1585–1592
31. Fletcher W, Yang Z (2010) The effect of insertions, deletions, and alignment errors on the branch-site test of positive selection. Mol Biol Evol 27:2257–2267
32. Privman E, Penn O, Pupko T (2012) Improving the performance of positive selection inference by filtering unreliable alignment regions. Mol Biol Evol 29:1–5

Chapter 18

The Integrated Microbial Genome Resource of Analysis

Alice Checcucci and Alessio Mengoni

Abstract

Integrated Microbial Genomes and Metagenomes (IMG) is a biocomputational system that allows to provide information and support for annotation and comparative analysis of microbial genomes and metagenomes. IMG has been developed by the US Department of Energy (DOE)-Joint Genome Institute (JGI). IMG platform contains both draft and complete genomes, sequenced by Joint Genome Institute and other public and available genomes. Genomes of strains belonging to Archaea, Bacteria, and Eukarya domains are present as well as those of viruses and plasmids. Here, we provide some essential features of IMG system and case study for pangenome analysis.

Key words Genome database, Metagenome database, Integrated Microbial Genomes and Metagenomes, Joint Genome Institute, Bioinformatics, Genome comparison

1 Introduction

Integrated Microbial Genomes and Metagenomes (IMG, URL: https://img.jgi.doe.gov) is a biocomputational system that allows to provide information and support for annotation and comparative analysis of microbial genomes and metagenomes [1, 2]. IMG has been developed by the US Department of Energy (DOE)-Joint Genome Institute (JGI) and is one of the JGI database resources belonging to its Genome Portal (http://genome.jgi.doe.gov). The Genome Portal has a "Tree of Life" data organization, where the sequenced genomes are arranged by domains (and kingdom, phylum, class, or order) and metagenomes by the niche.

IMG platform contains both draft and complete genomes, sequenced by Joint Genome Institute and other public and available genomes. Genomes of strains belonging to Archaea, Bacteria, and Eukarya domains are present as well as those of viruses and plasmids. On December 31, 2013, IMG stored more than 18,000 genomes, 13,334 of which are bacterial.

The genome storage can be investigated by comparisons on single or multiple genes, at the genome scale and by single or

Alessio Mengoni et al. (eds.), *Bacterial Pangenomics: Methods and Protocols*, Methods in Molecular Biology, vol. 1231, DOI 10.1007/978-1-4939-1720-4_18,

multiple functions. The system is therefore composed by three kinds of genome analysis: primary sequences, genic model (annotation) and biocomputational predictions, and functional connection and pathway information.

Finally, IMG provides users some linked tools to support comparative microbial genes, genomes, and metagenomics analysis, including COG, KEGG, Pfam, InterPro, and the Gene Ontology. Consequently, thanks also to the graphical user interface IMG is particularly suited for **nonexperienced bioinformaticians** which want to perform comparative genome analyses.

The two main functions available in the platform are

- **Exploration of data**
- **Genome comparison analyses**

2 Exploring Data on IMG

With the buttons "Find", it is possible to start the data scanning of genomes, genes, and then functions and metabolic pathways according to various biocomputational tools, as Blast, COG, KOG, Pfam, TIGRfam, and KEGG.

2.1 Find Genomes

One of the most used functions is the **Genome Browser**, where all the genomes filed in the platform are listed *alphabetically* or as *phylogenetic tree*. Every genome is described for domain, status, study name, sequencing center, size of the genome, and number of genes found.

The **Genome Search** function is instead used to search for that particular genome which is of interest of the user. Search filters as the simple name or metadata ("data about data") categories. Concerning metadata values as phenotype, habitat, disease, relevance, geographic location and host can be used. *Individual genomes* can be examined with o*rganism details* page that can be accessed by clicking on a genome name in every list of genomes collected in IMG.

The information page of every organism contains four main sections:

- The *Overview* that includes genome information as sequencing, taxonomic classification, metadata, metabolism, publications, and NCBI ID.
- The *Genome statistics* provides information about DNA sequence, as GC content, annotation, scaffold, and cluster gene according to the main tools (COG, KOG, Pfam, TIGRfam).
- The *Viewer* section shows linear or circular chromosome map of the organism, its scaffolds, and contigs.
- The *Export* section allows to move and save genome sequence or data in a variety of formats detectable using Excel.

Other interesting functions collect specific data of the genome, as *phylogenetic distribution of genes*, that allow to observe the distribution of the genes using Blast on the IMG dataset genome, or *horizontally transferred genes*, that gives statistics about gene or sequence moving during the evolution.

2.2 Find Genes

With this function it is possible to search a single gene or a group of genes (as for instance an entire operon) in selected genomes by using keywords and a variety of filters, like "Gene Product name", "Locus Tag", "IMG", or "GenBank ID". In particular genes can be retrieved through a Blast search [3, 4] or performing a phylogenetic profiling.

- **Blast** (Basic Local Alignment Search Tool) functions (blastp, blastx, tblastn, blastn) allow to find matches of the selected gene sequence (with a "copy and paste" simple operation in the text box) in one or more genomes choosing the favorite e-value cutoff.
- **Phylogenic Profiler** gives to the users the possibility to analyze the phylogenetic position and the presence of homologs of a single gene or of an operon.

Single genes can also be examined with a specific function, the *Gene details* page, that includes gene, protein and pathway information, and functional predictions.

To manage every function or activity in IMG that involves **more than one gene or genome**, the user can add the genomes to the cart. For example, the Gene List (created with the addition to cart) allows the user to maintain a list of all the genes resulted from IMG analysis. After the generation of the directory (the cart), the user can upload one or more genes in the list, or export some of them in FASTA format or their information in tab Excel format.

2.3 Find Function

Functional gene study and comparisons in IMG can be performed with the button "find function". Genes can be selected using **Search item** and **Pathways** or direct links also to external browsers for functional assignment as COG, KOG, Pfam, TIGRfam, KEGG, IMG network, enzyme, phenotype, and protein family comparison.

- **Functional item** and **pathway** investigations permit to find functions in selected genomes using keywords and definite filters. In this way it is possible to restrict the search to one or few genomes that contain one gene function or metabolic pathway according to the selected functional classification.

For each available bioinformatics tool, three operations are available: *Browser*, *List*, and *List with Stats.*

Between all the available browser analysis, **IMG Networks** is also placed. Through this tool it is possible to accede to Browser

and List areas. Here, one of the most interesting functions is *IMG Pathways*, where every pathway detailed in IMG is listed. Choosing "pathway ID" button, the user can display the detail page for each one, enzymatic reactions related to that pathway, and the genomes that have at least one gene associated with the pathway (with the corresponding phylogenetic distribution).

3 Comparing Genomes on IMG

Other than being a system storage of microbial genomes, IMG is also a platform to perform **comparative analysis** of genomes. It is provided by a variety of tools that allow to compare genomes in terms of gene content, sequence conservations, clustering, synteny analysis, and distance tree. The access to the comparative analysis functions is possible from the menu options. Below some of the genome comparison tools available are presented.

- **Genome statistics** includes *summary* and *general* statistics: The *summary* comprises a variety of DNA characteristics for the selected genome, such as GC content, number of protein-coding genes, and various functional annotations, and can be summarized and split up according to COG and KEGG categories; the user can select the COG or KEGG classification links listed in the summary table, and in this way, display all the itemized data according to the selected tool. Instead, the *general* shows all the statistics for all the genomes in IMG.
- With the function **Synteny Viewers**, it is possible to visualize the DNA conservation (specifically, gene loci co-localization in different organisms) through three comparative analysis tools: *VISTA*, *Dotplot*, and *Artemis ACT*. *VISTA* is preferably used to compare sequence alignments of compared genomes to explore and study the conservation sequences. In IMG platform, a variety of pre-alignments are available for use; selecting one of the possible choices, the user can display the data alignment. *Dotplot* can generate diagrams to prospect the similarity between two or more genomes. Finally, *Artemis ACT* is used for pairwise genome DNA sequence comparisons.
- **Abundance profiles** tool permits to compare genomes in terms of abundance of protein functions and families (according to COG, Pfams, and TIGRfams). In the *Overview*, the user can view abundance for all functions of selected genomes, and can select the output would; *heat map* shows the proteins/families abundant with different colors: the red one is the most abundant. *Matrix* displays the output in tabular format. In the Search section, it is possible to research one function based on its abundance in different genomes.

- **Distance** and **Radial Tree** permit to select a minimum of three or five genomes in IMG platform ad visualize the phylogenetic tree that correlates them.
- **Genome Clustering** permits the user to clusterize genomes based on similar function profiles. During the analysis it is possible to choose the clustering method, besides genome status and upload selected sequences; the types of cluster are sorted by function (COG, Pfam, KO, TIGRfam) and by taxonomy (class, family, genus); instead the cluster methods are based on hierarchical clustering, correlation clustering, and analysis of the principal components. Most used in the analysis are *hierarchical clustering*, that shows by a tree the phylogenetic distance between compared genomes, and *correlation clustering*, that gives the possibility to display by matrix the correlation coefficient.

In the **Analysis Cart** of Genes, Functions, Genomes, and Scaffolds (incomplete genomes), the user can find all the items that he or she selected during IMG analysis. The genomes that the user want to analyze and compare can be also uploaded and, if necessary, exported and saved on personal platform.

MyIMG allows users to set preferences for platform use, and to upload and manage their genomes.

4 An Example of Pangenome Analysis with IMG

As an example of pangenome study with IMG we provide an example of comparison among genomes of the nitrogen fixing symbiotic rhizobia of genus *Ensifer*.

Browser Requirements: Java should be installed on your local OS and Java applets should be enabled.

- After accessing the IMG genome website (https://img.jgi.doe.gov) go to "Find Genome"—"Genome Search" menu.
- Type the genus name you are searching for. In this case type "Ensifer", by using as filter "Genome Name". Click the "Go" button.
- A list containing all genomes which contain the word "Ensifer" is now displayed. For each genome the taxonomic domain, the Status of the genome, the Genome Name, the Proposal Name, the Sequencing Center, the Genome size (in bp), and gene counts are reported. This view can also be customized by selecting additional search field.
- Select all genomes in the page and then click on "Add Selected to Genome cart" button.
- Go in the "Compare genome" menu and scroll down to select the different menu options.

- Select "Distance Tree" from "Compare genome" menu. The list of genomes will appear to select the total number ("Select All") or a subset of genomes (at least three).
- Click on "Select All" and then on "Go" button. A distance tree is now displayed. The tree menu allows to change fonts and the graphics of the tree. Moreover, the tree can be saved as pdf file ("Tools" - "Save as pdf") or files in phyloXML, Newick, NHX, and Nexus formats can be displayed and then copied and saved in a separate file, allowing to redraw the dendrogram with other software, as Mega [5], or TreeView (http://taxonomy.zoology.gla.ac.uk/rod/treeview.html).
- It is possible also to proceed with a comparative analysis of functions. Firstly, you have to select one or more functions. Go to the "Find functions" menu and select one of the options, as for instance "Function Search". Here you can search for functions based on function name or on different codes (as COG, Pfam, KEGG definitions, Enzyme codes, MetaCyc). Select "Function Profile" from the "Compare Genomes" menu.
- Type "nitrite reductase" as gene product name. A list of the different nitrite reductases, their number (gene count), and the number of genomes containing each nitrite reductase is displayed. Select the copper-containing nitrite reductase. By clicking on the number of genome it is possible to visualize the genomes containing the selected gene. Cu-containing nitrite reductase is known to be part of the dispensable genome fraction of *Ensifer* sp. and confer tolerance to sodium nitrite [6–8].
- Another possibility to compare functions it to proceed with an overview of all functions. Go on the "Compare Genomes" menu and select the "Abundance Profiles" - "Overview (all functions)" option. Here you can proceed with the drawing of a heat map showing the abundance (absolute or normalized to genome size) of all functions. Functions can be chosen as COG, Enzyme, KO, Pfam, and TIGfam. For instance select "COG" and then select the genomes to be compared by searching in the list or browsing the phylogenetic tree. For instance browse the phylogenetic tree on "*Proteobacteria*", then *Alphaproteobacteria*, then *Rhizobiales*, then *Rhizobiaceae*, and finally "*Ensifer*". Select "*Ensifer*" and click the "Go" button. A heat map will be displayed with color indicating the abundance (red, high abundance; blue, low abundance) for each COG. Here functions present in all genomes (core genome) with respect to dispensable or differentially occurring functions can be identified.

5 Conclusions

This chapter has shown some key functions of IMG platform, which can be applied by also nonexperienced bioinformaticians to analyze genome data and perform several basic and advanced analyses on comparative bacterial genomics. For further information and utilities please consult the related publications [1, 9] and the manual ("Using IMG" menu on IMG webpage).

References

1. Grigoriev IV, Nordberg H, Shabalov I, Aerts A, Cantor M, Goodstein D, Kuo A, Minovitsky S, Nikitin R, Ohm RA, Otillar R, Poliakov A, Ratnere I, Riley R, Smirnova T, Rokhsar D, Dubchak I (2011) The Genome Portal of the Department of Energy Joint Genome Institute. Nucleic Acids Res 40:D26–D32
2. Markowitz VM, Chen IMA, Palaniappan K, Chu K, Szeto E, Pillay M, Ratner A, Huang J, Woyke T, Huntemann M, Anderson I, Billis K, Varghese N, Mavromatis K, Pati A, Ivanova NN, Kyrpides NC (2014) IMG 4 version of the integrated microbial genomes comparative analysis system. Nucleic Acids Res 42(D1): D560–D567
3. Altschul SF, Madden TL, Schaffer AA, Zhang J, Zhang Z, Miller W, Lipman DJ (1997) Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. Nucleic Acids Res 25:3389–3402
4. Johnson M, Zaretskaya I, Raytselis Y, Merezhuk Y, McGinnis S, Madden T (2008) NCBI BLAST web site NCBI BLAST: a better web interface. Nucleic Acids Res 36:W5–W9
5. Tamura K, Peterson D, Peterson N, Stecher G, Nei M, Kumar S (2011) MEGA5: molecular evolutionary genetics analysis using maximum likelihood, evolutionary distance, and maximum parsimony methods. Mol Biol Evol 28: 2731–2739
6. Galardini M, Mengoni A, Brilli M, Pini F, Fioravanti A, Lucas S, Lapidus A, Cheng J-F, Goodwin L, Pitluck S, Land M, Hauser L, Woyke T, Mikhailova N, Ivanova N, Daligault H, Bruce D, Detter C, Tapia R, Han C, Teshima H, Mocali S, Bazzicalupo M, Biondi EG (2011) Exploring the symbiotic pangenome of the nitrogen-fixing bacterium *Sinorhizobium meliloti*. BMC Genomics 12:235
7. Biondi EG, Tatti E, Comparini D, Giuntini E, Mocali S, Giovannetti L, Bazzicalupo M, Mengoni A, Viti C (2009) Metabolic capacity of *Sinorhizobium* (*Ensifer*) *meliloti* strains as determined by phenotype microarray analysis. Appl Environ Microbiol 75:5396–5404
8. Galardini M, Mengoni A, Biondi EG, Semeraro R, Florio A, Bazzicalupo M, Benedetti A, Mocali S (2013) DuctApe: a suite for the analysis and correlation of genomic and OmniLog™ phenotype microarray data. Genomics. doi:10.1016/j.ygeno.2013.11.005
9. Markowitz VM, Mavromatis K, Ivanova NN, Chen IMA, Chu K, Kyrpides NC (2009) IMG ER: a system for microbial genome annotation expert review and curation. Bioinformatics 25:2271–2278

Index

Alessio Mengoni et al. (eds.), *Bacterial Pangenomics: Methods and Protocols*, Methods in Molecular Biology, vol. 1231, DOI 10.1007/978-1-4939-1720-4, © Springer Science+Business Media New York 2015

N

O

P

R

S