Chapter 21

Bacterial Genomic Data Analysis in the Next-Generation Sequencing Era

Massimiliano Orsini, Gianmauro Cuccuru, Paolo Uva, and Giorgio Fotia

Abstract

Bacterial genome sequencing is now an affordable choice for many laboratories for applications in research, diagnostic, and clinical microbiology. Nowadays, an overabundance of tools is available for genomic data analysis. However, tools differ for algorithms, languages, hardware requirements, and user interface, and combining them as it is necessary for sequence data interpretation often requires (bio)informatics skills which can be difficult to find in many laboratories. In addition, multiple data sources, as well as exceedingly large dataset sizes, and increasingly computational complexity further challenge the accessibility, reproducibility, and transparency of the entire process. In this chapter we will cover the main bioinformatics steps required for a complete bacterial genome analysis using next-generation sequencing data, from the raw sequence data to assembled and annotated genomes. All the tools described are available in the Orione framework (http://orione.crs4.it), which uniquely combines in a transparent way the most used open source bioinformatics tools for microbiology, allowing microbiologist without any specific hardware or informatics skill to conduct data-intensive computational analyses from quality control to microbial gene annotation.

Key words Microbiology, Sequence analysis, Genome assembly, Next-generation sequencing, Galaxy, Computational biology, Genomics, Bioinformatics

1 Introduction

High-throughput sequencing is now fast and cheap enough to be considered part of standard analysis in microbiology. This allows clinicians, environmental microbiologists, epidemiologists, and public health operators to have available new tools for their researches. But even if the technology behind the production of sequence data is growing fast, providing higher throughputs, longer sequences, and lower costs, the *dry* side of next-generation sequencing (NGS) analysis is still in the cradle with new and better computational methods and analysis tools appearing all the time.

In essence, end-to-end NGS microbiology data analysis requires chaining a number of analysis tools together to form computational analysis pipelines. Due to high data volumes and sophisticated

Oliviero Carugo and Frank Eisenhaber (eds.), *Data Mining Techniques for the Life Sciences*, Methods in Molecular Biology, vol. 1415, DOI 10.1007/978-1-4939-3572-7_21,

computational methods, NGS analysis pipelines can be extremely compute-intensive. Integrating new algorithms into those pipelines using traditional scripting languages can be laborious and time consuming due to the variety of interfaces, input and output formats, and deployment requirements. Furthermore, there are emerging requirements that have to be properly addressed in this context, namely, interoperability, reproducibility, and transparency [1].

On this way, Galaxy [2–4] is a well-known open platform for reproducible data-intensive computational analysis in many diverse biomedical research environments. It provides a web-based interface that permits users to bind together computational tools that have been prewrapped and provides developers a simple way to encapsulate computational tools and datasets in a graphical user interface.

One of the most appreciated aspects of Galaxy, by nonprogrammers users, is the possibility to access complex workflows without the need to learn the implementation details of every single tool involved. While this feature is extremely useful for biologists, other advanced users may have a need for a programmatic access to single tools or a way to automate bulk processing. To deal with those tasks, Galaxy includes a RESTful API that allows programmatic access to a consistent subset of its workflow management infrastructure. A Python library, called BioBlend [5], provides a high-level interface for controlling operations performed with Galaxy. For example, loading a dataset and run a Galaxy workflow on it can be accomplished with just a few lines of code [6].

Leveraging on Galaxy, we developed Orione (http://orione.crs4.it) [7], a specialized domain server for integrative analysis of NGS microbial data, which covers the whole life cycle of microbiology research data, bringing together all the tools to perform steps such as quality check, alignment, assembly, scaffolding, and annotation. Integration into Galaxy permits the analysis results to be documented, shared, and published guaranteeing transparency and reproducibility.

Orione complements the modular Galaxy environment, consolidating publicly available research software and newly developed tools and workflows to build complex, reproducible pipelines for "straight on target" microbiological analysis. Furthermore, Orione is part of an integrated infrastructure at CRS4 for automated NGS data management and processing (Fig. 1), and as such it provides seamless integration to computing and advanced data facilities and resources [8].

Orione adds to a number of Galaxy servers developed in the last few years by the Galaxy Community, see http://wiki.galaxyproject.org/PublicGalaxyServers for a complete, updated list. Many of these servers are specialized in a particular type of analysis, like ChIP-Seq analysis (Cistrome [9], Nebula [10]), adaptive divergence in prokaryotes (OdoSE [11]), metagenomic taxonomy (MGTAXA [12]),

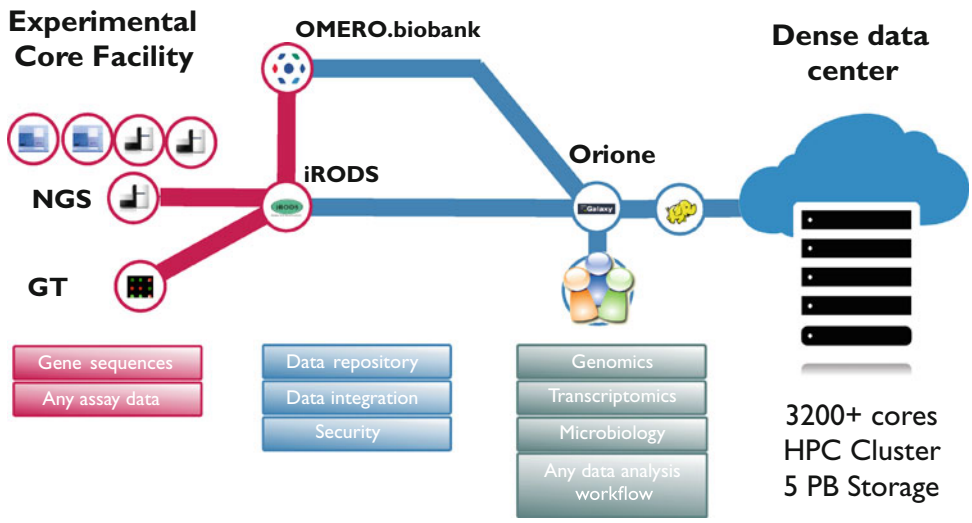

Fig. 1 Orione is a key component of the fully automated infrastructure to support the analysis of the DNA sequencing data generated by the CRS4 NGS facility, currently the largest in Italy by throughput, number of samples processed, and amount of data generated. Such infrastructure includes iRODS [63] for efficient inter-institutional data sharing, OMERO.biobank [64] to model biomedical data and the chain of actions that connect them, and Hadoop-based tools to provide scalable computing [65]. *GT* genotyping arrays

microbiome, metabolome, and immunome data analysis (MBAC Metabiome Portal [13]) or microbial communities comparison (Fast UniFrac [14]).

The remainder of this chapter is structured as follows. Throughout this chapter, we use Orione as our reference. We begin by describing the main steps of the bacterial NGS data analysis, namely, pre-processing, alignment, de novo assembly, scaffolding, post-assembly, variant calling, and annotation. We then continue illustrating a selection of pipelines we implemented that summarize the current best practices in data pre-processing, genome re-sequencing, and de novo assembly. A description of the sequencing technologies is out of the scope of this chapter. We refer the reader to Ref. [15] for a recent review on this topic.

2 Delving into Microbiology NGS Data Analysis

Sequencing of microbial genomes is now a widely used strategy in microbiology research, with applications in a wide range of topics such as pathogenicity, drug resistance, and evolutionary and epidemiological studies. Despite impressive technological advances that currently enable microbiological laboratories to routinely perform bacterial whole genome sequencing [15], the bioinformatics analysis of bacterial genome data is still a challenging task. The data analysis workflow has been divided into seven logical sections:

pre-processing, alignment, de novo assembly, scaffolding, post-assembly, variant calling, and annotation. Each section includes a list of freely available programs, recommendation on how to use them, and references to more detailed technical information.

2.1 Pre-Processing

During the past decade, the overall quality of NGS data has significantly improved and it is still growing, thanks to the progress being made in NGS technology. However, mapping/assembly artifacts can still arise from errors in base calling, sequence contamination (e.g., primer or adapter retention, DNA contamination), and low-quality reads. Some recent software for NGS data analysis can partially compensate for noisy data and improve the quality of the final results of the sequencing experiment, e.g., low-quality read tails will be automatically clipped by the BWA-MEM aligner [16], but will strongly reduce the sensitivity of other programs such as BWA-backtrack [17] and Bowtie [18] which perform an end-to-end alignment. For these reasons we always recommend readers to perform an accurate quality control of reads before any alignment or assembly steps. We note that different NGS platforms share several sources of error such as the presence of homopolymers/low-complexity regions, with an impact on the identification of sequence variants and the genome assembly, while other quality issues are platform specific [19]. The following metrics should be considered to assess the read quality: percentage of reads filtered by the software supplied with the sequencing machines, per read and per base sequence quality, per base sequence content, percentage of duplicated reads (PCR artifacts), and presence of overrepresented sequences. Once a quality issue is detected, possible actions include the trimming of the low-quality reads (i.e., progressive removal of bases at 5′ and 3′ of the read), the removal of poor quality reads, or a combination of both strategies.

Orione integrates tools for read quality control, such as the widely adopted FastQC software [20] which computes several quality statistics and programs for trimming/filtering specifically developed for Orione such as *FASTQ positional and quality trimming* and *Paired-end compositional filtering*. *FASTQ positional and quality trimming* trims FASTQ files by position, minimum Phred quality score, average Phred score using sliding windows (bases will be trimmed one-by-one until the average read quality reaches this value), and filters reads by residual minimum length after trimming. *Paired-end compositional filtering* filters low-complexity sequences by frequency of monomers, dimers, and trimers. They both accept paired-end FASTQ files as input and preserve mate integrity. Unpaired reads after filtering are kept in separated files.

Subheading 3 describes a general NGS quality control workflow, which should enable researchers to detect and remove low-quality sequences and ensure that biological conclusions are not plagued by sequencing quality issues.

2.2 Alignment

Once the raw data have been filtered for low-quality reads and artifacts have been removed, the next step is to align the sequences against a suitable reference genome. Programs for read alignment have been developed to optimize the trade-off between accuracy of the alignment and speed and to exploit the specific features of the different sequencing technologies, namely, long reads (Roche 454, Ion Torrent, PacBio), short reads (Illumina), and color space encoding (SOLiD). The selection of software for short read alignment available in Orione is far from being exhaustive. To date, more than 100 NGS aligners have been developed [21], and benchmarks have been published comparing the aligners alone [22] or combinations of aligners with software for downstream analyses (e.g., variant calling [23]). Notwithstanding the plethora of aligners, they can be grouped based on the underlying algorithms in hashed-seed and suffix tree methods. Members of the hashing-based category share the seed-and-extend algorithm, which starts with an exact match of a seed sequence against the reference, and then tries to extend the alignment. These include the classical BLAST program (slow, not well suited for large NGS datasets) [24] and BLAT (fast, for closely related species as it requires multiple perfect matches in close proximity, enabling the detection of small indels within homologous regions) [25]. Other options include LASTZ [26] which has been developed for large-scale genome alignment and that natively handles long sequences as those produced by Roche 454, but can be adapted to align short reads, and MOSAIK [27] which support reads of different lengths, being part of a suite to produce reference-guided assemblies with gapped alignments. Suffix tree-based methods are faster and require a lower memory usage than hashing-based methods but are less accurate. Members of this class are Bowtie (supports ungapped alignments only), Bowtie 2 [28] (performs gapped alignments, designed for sequences longer than 50 bp), BWA-backtrack (for sequences up to 100 bp), BWA-MEM (for sequences longer than 70 bp), and SOAP2 [29] (robust for closely related species with small numbers of SNPs and indels). We refer to [30, 31] for a comprehensive description of the algorithms used by the different programs. We suggest to first align short reads by using the suffix tree-based methods, while longer reads are better mapped with software supporting higher number of mismatches/indels. Then, if the mapping percentage is low, multiple programs should be tested. Fortunately, running multiple aligners in Orione is straightforward.

The output of short read aligners is often in SAM/BAM format, ready to be processed by downstream applications. Where the format is different, e.g., alignments produced by SOAP2, tools for format conversion are available in Orione.

It is important to remark the limits of the mapping-to-reference approach for the re-sequencing of bacterial genomes. If the divergence between the target species and the reference genome is high,

this approach will not align a large portion of the reads; hence the user should opt for a de novo assembly strategy or a combination of both approaches. In some cases even for different strains of the same bacteria, a de novo approach may be preferred.

2.3 De Novo Assembly

A crucial step in bacterial genomics is to obtain a whole chromosome sequence directly from sequencing reads, without the bias of choosing a reference genome as a guide. This is particularly true when the genome of the organism being studied is not particularly stable, and it is known to exhibit high intraspecies variability. De novo assembly is the process of obtaining a whole genome sequence from short reads by finding common subsequences and assembling overlapping reads in longer sequences (contigs) supposing that they have been generated by the same genomic location.

Due to the complexity of this task, a plethora of genome assemblers have been implemented based on different algorithms. In general, most current assembly algorithms can be assigned to one of three classes based on their underlying data structure: De Bruijn graph, overlap layout consensus, and read layout (or greedy approach) assemblers. While the latter is based on a self-aligning algorithm, the two former approaches utilize a graph structure built upon the sequencing reads and algorithms for graph walking to derive overlapping sequences. We refer to [32, 33] for a detailed description of the algorithms and to [34] for a comparison between de novo assemblers.

Different software for de novo genome assembly are available in Orione. These include Velvet and ABySS [35] which assemble *k*-mers using a de Bruijn graph, EDENA [36] which is based on the overlap-layout-consensus algorithm, and the greedy assembler SSAKE [37]. Long reads as those produced by Ion Torrent, Roche 454, and PacBio technologies are well suited for the MIRA assembler [38], which relies on a modified Smith-Waterman algorithm and generates hybrid assemblies using a mixture of reads from different technologies, when available.

The depth of coverage and read length drive the appropriate *k*-mer selection of de Bruijn graph assemblers. The *VelvetOptimiser* [39] program can assist in selecting the optimal *k*-mer size to achieve a trade-off between the specificity of long *k*-mers and the sensitivity of shorter ones by running a number of *Velvet* [40] steps at different *k*-mer sizes.

2.4 Scaffolding

Both de novo and re-sequencing approaches return contigs, but small-sized contigs limit the applicability of whole genome sequences for genetic analysis.

To enhance the quality of de novo sequence assemblies, contigs have to be elongated or joined and correctly orientated to build scaffolds, i.e., an ordered sequence consisting of contigs and gaps of known sizes. If read pairs with a known insert size are

available, i.e., mate-pair or paired-end reads, this information can be used to scaffold contigs. This strategy is useful to span gaps due to misassembled regions containing long repetitive elements which are hard to resolve solely by overlapping reads of limited length. Using paired-read sequencing data, it is also possible to assess the order, distance, and orientation of contigs and combine them. Although the latter process is a crucial step in finishing genomes, scaffolding algorithms are often built-in functions in de novo assembly tools and cannot be independently controlled. This led us to include in Orione several scaffolders, such as SSPACE [41], SSAKE, SEQuel [42], and SOPRA [43]. Similarly to de novo assemblers, scaffolders' performance is affected by sequencing platform and read quality.

2.5 Post-assembly

Obtaining a genome as complete as possible is crucial for successive genomic analysis and strain comparison. We present here a selection of tools to perform assembly evaluation, integration of multiple assemblies produced with different approaches, and contigs ordering against a reference genome, once de novo or reference-based assemblies have been obtained.

For a preliminary evaluation of the assembly, we implemented the *Check bacterial contigs* and *Check bacterial draft* tools which compute metrics such as the number of assembled nucleotides, the average coverage, N50, NG50 and contigs length statistics. Genomic regions corresponding to high-quality segments and contigs longer than a given threshold can be extracted from genome drafts by running *Extract contigs* tool.

Contigs coming from different assemblies can be merged by *CISA contigs integrator* [44] which improves the accuracy of the final assembly by extending contigs and by removing the misassembled ones.

Contigs may be ordered against a reference genome, usually the most closely related bacterium with a "finished" genome, under the hypothesis that the two organisms share synteny. Ordering of contigs can be achieved using tools such as MUMmer [45], Mugsy [46], or BLAST and then processing the results. However the easiest way is to run the contig ordering tool in the program Mauve [47].

At the end of the post-processing procedure, draft genomes and scaffolds can still include errors, gaps, and misassembled regions due to technical artifacts, evolutionary differences, the presence of clustered regularly interspaced short palindromic repeats (CRISPRs), and prophages. In fact, as demonstrated during the Genome Assembly Gold-standard Evaluations (GAGE) [34], all the assemblies contained errors. An accurate estimate of the error rate can be only calculated if a closely related reference genome is available, e.g., by aligning the contigs against the reference with Mauve or MUMmer and then counting the number of miscalled based, missing calls, and missing and extra segments.

2.6 Variant Calling

Nucleotide polymorphisms can be directly identified from the alignment of assembly-based contigs and scaffolds against the reference genome using MUMmer and Mauve. However, for closely related species, we suggest to align the preprocessed reads with a short read aligner, and once the alignment has been obtained, genetic variants can be identified with the SAMtools-BCFtools pipeline [48] (wrapped as *BAM to consensus* in Orione), FreeBayes [49], GATK Unified Genotyper (UG), and GATK Haplotype Caller (HC) [50]. When comparing output from multiple variant callers, differences emerge [23] which reflect the differences between algorithms: SAMtools-BCFtools and GATK variant callers report variants based on the alignments of the sequence reads against the reference genome, while GATK HC and FreeBayes perform an additional local realignment of reads (haplotype-based callers).

All these tools have been developed for diploid organisms, but their use with haploid genomes has been described in literature [51–53]. GATK HC/UG and FreeBayes have an option for explicitly setting the ploidy when executed on bacterial genomes (default value is 2). The full list of variants can be further filtered based on variant and genotype quality values using the *Filter a VCF file* tool or alternatively can be converted with *VCF to tabular converter* and opened with any spreadsheet program.

2.7 Annotation

Once obtained a FASTA sequence for the assembled genome, most researchers will be interested in identifying all the genes and other relevant features of the sequence such as ribosomal and transfer RNAs, other noncoding RNAs, and the presence of signal peptides. Orione includes Glimmer (Gene Locator and Interpolated Markov ModelER) [54], which uses interpolated Markov models for finding genes, and it is best suited for the genomes of bacteria, archaea, and viruses; tRNAscan-SE [55], which combines multiple tRNA search methods for the identification of transfer RNA; and Prokka [56], a software that combines multiple BLAST searches and a suite of feature prediction tools (Prodigal [57] for coding sequence (CDS), RNAmmer [58] for ribosomal RNA genes (rRNA), Aragorn [59] for transfer RNA and tmRNA genes, SignalP [60] for signal peptides (at N-term of CDS), and Infernal [61] for noncoding RNA) to provide the most complete set of annotations, from the translated coding genes to the annotated files with the predicted features in multiple formats, ready for submission to public repositories such as NCBI. The prediction of the effect of genetic variants (e.g., amino acid change) can be assessed by SnpEff [62].

2.8 Complementary Tasks

A collection of additional tools and utilities complete the Orione framework with the aim of providing an accessible toolkit to facilitate the dataflow and ultimately support the creation of analysis workflows. Orione makes available to the users various tools and

scripts that can be used to get data from external repositories, manipulate FASTQ, SAM, or FASTA files, as well as to convert files from one format to another, filter, join, or parse complex data.

3 Advanced Workflow Examples

Galaxy workflows allow the user to combine different tools into reproducible processing pipelines that can run automatically over different set of data, without the need of recall single tools or resetting parameters. In the following, we illustrate a set of workflows that summarize the current best practice in NGS-based bacterial genome analysis: pre-processing, bacterial re-sequencing, and de novo assembly. All these pipelines are available as public workflows in the shared data section of Orione and can be used as starting points, which can then be further tailored. For the sake of simplicity, all the workflows described in this section refer to paired-end datasets.

3.1 Workflow #1: Pre-processing

The workflow "W1—Pre-processing|Paired-end" (Fig. 2 and Table 1) proposes nine steps to improve the overall paired-end dataset quality. To emphasize the outcome of the process, a quality report from FastQC has been placed before and after the editing steps.

Input

- Raw FASTQ paired-end reads

Output

- Processed FASTQ paired-end reads

3.2 Workflow #2: Bacterial Re-sequencing

We designed the workflow "W2—Bacterial re-sequencing|Paired-end" (Fig. 3 and Table 2) with the aim of assembling genomes of well-known or already characterized isolates. The primary task is to identify variants rather than the genome assembly itself. The

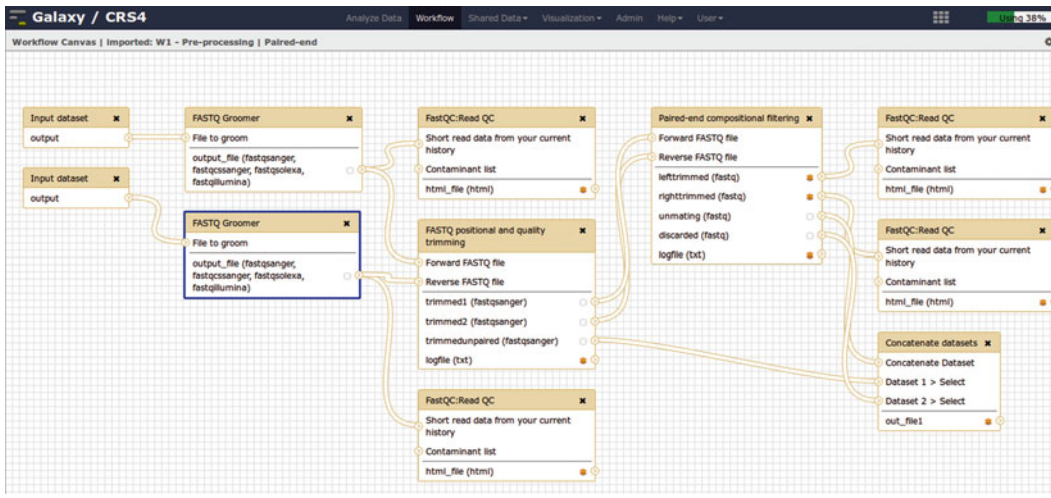

Fig. 2 Workflow "W1—Pre-processing|Paired-end" canvas

Table 1
Tools used in workflow "W1—Pre-processing|Paired-end"

Steps	Tools	References
Convert forward reads to fastqsanger encoding	FASTQ groomer	[66]
Convert reverse reads to fastqsanger encoding	FASTQ groomer	[66]
Quality control of unaltered forward reads	FastQC	[20]
Quality control of unaltered reverse reads	FastQC	[20]
Trimming/filtering based on sequence quality and length	FASTQ positional and quality trimming	[7]
Filter reads based on frequency of monomers, dimers, and trimers	Paired-end compositional filtering	[7]
Quality control of filtered forward reads	FastQC	[20]
Quality control of filtered reverse reads	FastQC	[20]
Concatenate filtered reads	Concatenate datasets	[4]

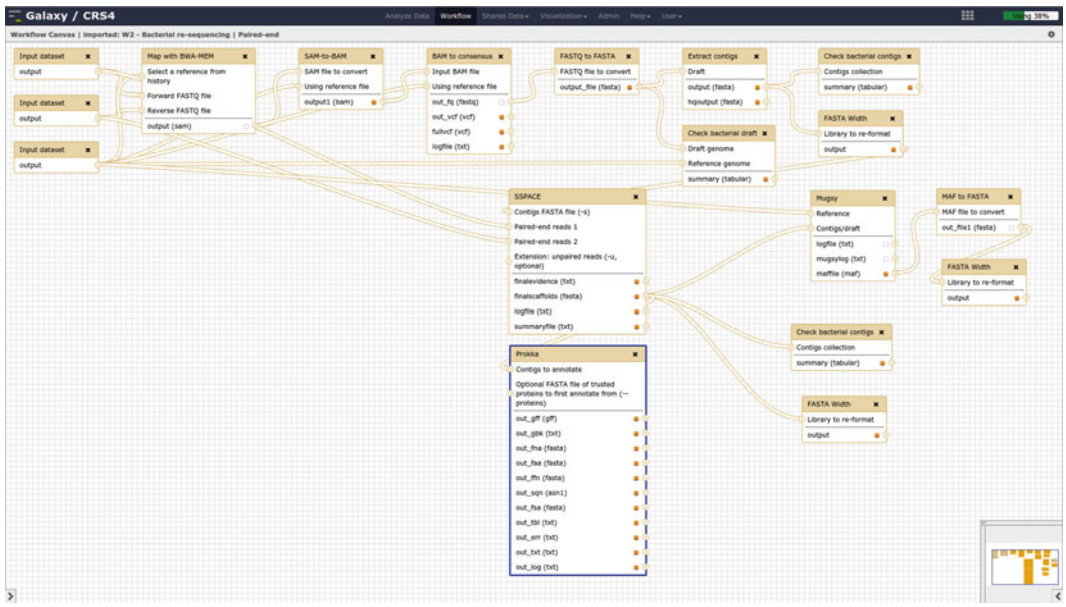

Fig. 3 Workflow "W2—Bacterial re-sequencing|Paired-end" canvas

workflow uses the BWA-MEM aligner since it permits gapped alignment. We highlight that recent versions of BWA include three different algorithms optimized for different read lengths (backtrack, MEM, SW) allowing users to customize the workflow according to the sequencing platform used for generating data. Users can easily customize the workflow. As an example, to align long reads with LASTZ instead of BWA-MEM, the first step can be replaced by

Table 2
Tools used in workflow "W2—Bacterial re-sequencing|Paired-end"

Steps	Tools	References
Align against a reference genome with	BWA-MEM	[16]
Convert alignment from SAM to BAM format	SAM-to-BAM	[48]
Extract a draft consensus sequence	BAM to consensus	[48]
Convert the draft from FASTQ to FASTA	FASTQ to FASTA	[66]
Evaluate draft quality	Check bacterial draft	[7]
Extract contigs (longer than a given threshold) from draft	Extract contigs	[7]
Evaluate contigs quality	Check bacterial contigs	[7]
Contigs scaffolding	SSPACE	[41]
Scaffolds evaluation	Check bacterial contigs	[7]
Align scaffolds against reference	Mugsy	[46]
Convert MUMmer output to FASTA	MAF to FASTA	[67]
Annotate draft/contigs	Prokka	[56]

FASTQ to FASTA conversion and *LASTZ mapping*. The alignment file is used to derive a consensus draft sequence and a list of variants. Contigs are extracted from the draft genome and submitted to SSPACE scaffolder. Scaffolds are subsequently width formatted, realigned to the reference genome using MUMmer for SNP detection, and finally annotated by Prokka. Basic statistics are calculated in each key step (draft, contigs, scaffolds) by the appropriate *Check bacterial draft/contigs* tool. A simpler workflow, where Prokka directly annotates the draft sequence, can be extracted by skipping the last steps.

In addition, Mauve can replace Mugsy for the alignment of the scaffolds against the reference genome, and the scaffolds can be eventually integrated with the scaffolds generated by de novo assembly using CISA.

Input

- Processed FASTQ reads
- Reference genome

Output

- Contigs sequences (FASTA)
- Scaffolds sequences (FASTA)
- Scaffolds annotations (multiple formats available)
- Report with draft/contigs/scaffolds quality
- Variants with respect to the reference genome

3.3 Workflow #3: Bacterial De Novo Assembly

The workflow "W3—Bacterial de novo assembly|Paired-end 15.06" (Fig. 4 and Table 3) executes multiple de novo assemblers: VelvetOptimiser at different *k*-mer values, SPADES, which also runs a scaffolding step, and ABySS. Contigs obtained with the different tools are then integrated using CISA. Basic statistics are calculated on the combined contigs using the *Check bacterial contigs* tool. Finally, sequences are annotated using Prokka.

Input

- Processed FASTQ reads

Output

- Contigs/scaffolds from each assembler (FASTA)
- Integrated contig sequences (FASTA)

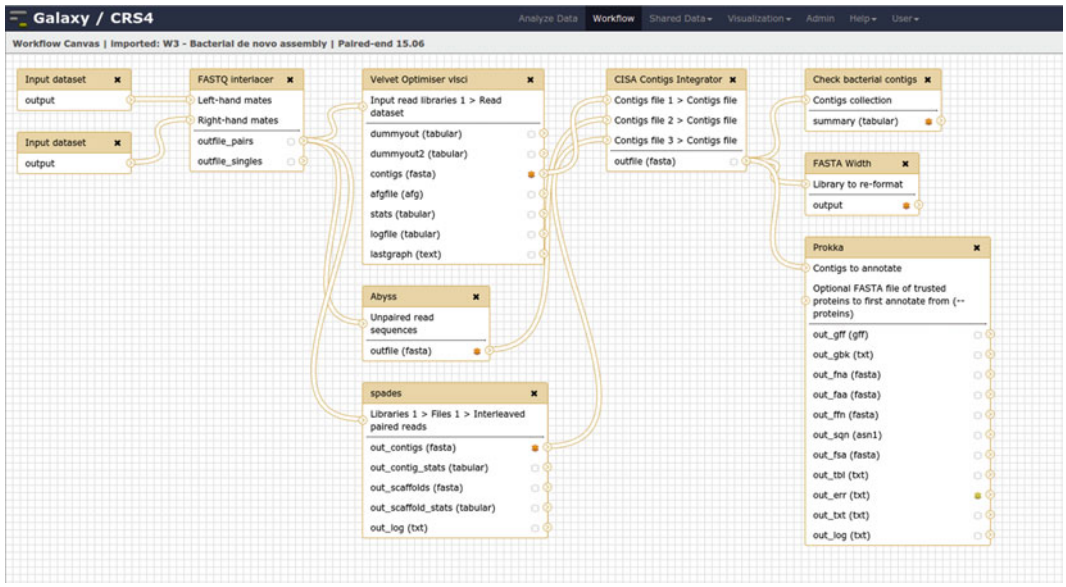

Fig. 4 Workflow "W3—Bacterial de novo assembly|Paired-end 15.06" canvas

Table 3
Tools used in workflow "W3—Bacterial de novo assembly|Paired-end 15.06"

Steps	Tools	References
Prepare reads for assemblers	FASTQ interlacer	[68]
De novo assembly	VelvetOptimiser	[69]
De novo assembly	ABySS	[35]
De novo assembly	SPAdes	[70]
Integrates contigs by	CISA	[44]
Evaluate contigs/scaffolds quality	Check bacterial contigs	[7]
Annotate sequences	Prokka	[56]

- Sequence annotations (multiple formats available)
- Report with de novo assembly statistics

4 Conclusions

Next-generation sequencing microbiology data analysis requires a diversity of tools from bacterial re-sequencing, de novo assembly to scaffolding, bacterial RNA-Seq, gene annotation, and metagenomics. Sophisticated frameworks are needed to integrate state-of-the-art software to build computational pipelines and complex workflows and, more importantly, to cope with the lack of interoperability, reproducibility, and transparency.

Leveraging on the Galaxy framework, Orione provides an integrated web-based environment that enables microbiology researchers to conduct their own custom NGS analysis and data manipulation without software installation or programming. Providing microbiologist with many different tools, workflows, and options for bacterial genomics analysis—for applications ranging from bacterial genome assembling to emerging fields (e.g., differential transcriptional or microbiome analysis)—Orione supports the whole life cycle of microbiology research data, from creation, annotation to publication and reuse. Orione is available at http://orione.crs4.it.

Acknowledgments

This work was partially supported by the Sardinian Regional Authorities.

References

1. Sandve GK, Nekrutenko A, Taylor J, Hovig E (2013) Ten simple rules for reproducible computational research. PLoS Comput Biol 9:e1003285
2. Giardine B, Riemer C, Hardison RC, Burhans R, Elnitski L, Shah P, Zhang Y, Blankenberg D, Albert I, Taylor J, Miller W, Kent WJ, Nekrutenko A (2005) Galaxy: a platform for interactive large-scale genome analysis. Genome Res 15: 1451–1455
3. Goecks J, Nekrutenko A, Taylor J (2010) Galaxy: a comprehensive approach for supporting accessible, reproducible, and transparent computational research in the life sciences. Genome Biol 11:R86
4. Blankenberg D, Von Kuster G, Coraor N, Ananda G, Lazarus R, Mangan M, Nekrutenko A, Taylor J (2010) Galaxy: a web-based genome analysis tool for experimentalists. Curr Protoc Mol Biol Chapter 19: Unit 19.10.1–21
5. Sloggett C, Goonasekera N, Afgan E (2013) BioBlend: automating pipeline analyses within Galaxy and CloudMan. Bioinformatics 29:1685–1686
6. Leo S, Pireddu L, Cuccuru G, Lianas L, Soranzo N, Afgan E, Zanetti G (2014) BioBlend.objects: metacomputing with Galaxy. Bioinformatics 30:2816–2817. doi:10.1093/bioinformatics/btu386
7. Cuccuru G, Orsini M, Pinna A, Sbardellati A, Soranzo N, Travaglione A, Uva P, Zanetti G, Fotia G (2014) Orione, a web-based framework for NGS analysis in microbiology. Bioinformatics 30:1928–1929. doi:10.1093/bioinformatics/btu135

8. Cuccuru G, Leo S, Lianas L, Muggiri M, Pinna A, Pireddu L, Uva P, Angius A, Fotia G, Zanetti G, Bioinformatics H (2014) An automated infrastructure to support high-troughput bioinformatics. In: Smari, Waleed W, Zeljkovic V (eds) Proc. IEEE Int. Conf. High Perform. Comput. Simul. (HPCS 2014). IEEE. pp 600–607
9. Liu T, Ortiz JA, Taing L, Meyer CA, Lee B, Zhang Y, Shin H, Wong SS, Ma J, Lei Y, Pape UJ, Poidinger M, Chen Y, Yeung K, Brown M, Turpaz Y, Liu XS (2011) Cistrome: an integrative platform for transcriptional regulation studies. Genome Biol 12:R83. doi:10.1186/gb-2011-12-8-r83
10. Boeva V, Lermine A, Barette C, Guillouf C, Barillot E (2012) Nebula--a web-server for advanced ChIP-seq data analysis. Bioinformatics 28:2517–2519. doi:10.1093/bioinformatics/bts463
11. Vos M, te Beek TAH, van Driel MA, Huynen MA, Eyre-Walker A, van Passel MWJ (2013) ODoSE: a webserver for genome-wide calculation of adaptive divergence in prokaryotes. PLoS One 8:e62447. doi:10.1371/journal.pone.0062447
12. Williamson SJ, Allen LZ, Lorenzi HA, Fadrosh DW, Brami D, Thiagarajan M, McCrow JP, Tovchigrechko A, Yooseph S, Venter JC (2012) Metagenomic exploration of viruses throughout the Indian Ocean. PLoS One 7:e42047. doi:10.1371/journal.pone.0042047
13. MBAC metabiome portal. Accessed 15 Jun 2015 from http://mbac.gmu.edu:8080
14. Hamady M, Lozupone C, Knight R (2010) Fast UniFrac: facilitating high-throughput phylogenetic analyses of microbial communities including analysis of pyrosequencing and PhyloChip data. ISME J 4:17–27
15. Loman NJ, Constantinidou C, Chan JZM, Halachev M, Sergeant M, Penn CW, Robinson ER, Pallen MJ (2012) High-throughput bacterial genome sequencing: an embarrassment of choice, a world of opportunity. Nat Rev Microbiol 10:599–606
16. BWA-MEM. Accessed 15 Jun 2015 from http://bio-bwa.sourceforge.net/bwa.shtml
17. Li H, Durbin R (2009) Fast and accurate short read alignment with Burrows-Wheeler transform. Bioinformatics 25:1754–1760. doi:10.1093/bioinformatics/btp324
18. Langmead B (2010) Aligning short sequencing reads with Bowtie. Curr Protoc Bioinformatics Chapter 11: 11–7
19. Quail MA, Smith M, Coupland P, Otto TD, Harris SR, Connor TR, Bertoni A, Swerdlow HP, Gu Y (2012) A tale of three next generation sequencing platforms: comparison of Ion Torrent, Pacific Biosciences and Illumina MiSeq sequencers. BMC Genomics 13:341
20. Andrews S FastQC a quality control tool for high throughput sequence data. Accessed 15 Jun 2015 from http://www.bioinformatics.babraham.ac.uk/projects/fastqc/
21. SeqAnswers. Accessed 15 Jun 2015 from http://seqanswers.com/wiki/Software/list
22. Hatem A, Bozdağ D, Toland AE, Çatalyürek ÜV (2013) Benchmarking short sequence mapping tools. BMC Bioinformatics 14:184. doi:10.1186/1471-2105-14-184
23. Cornish A, Guda C (2015) A comparison of variant calling pipelines using genome in a bottle as a reference. Biomed Res Int 2015:456479
24. Camacho C, Coulouris G, Avagyan V, Ma N, Papadopoulos J, Bealer K, Madden TL (2009) BLAST+: architecture and applications. BMC Bioinformatics 10:421
25. Kent WJ (2002) BLAT—the BLAST-like alignment tool. Genome Res 12:656–664. doi:10.1101/gr.229202
26. Harris RS (2007) Improved pairwise alignment of genomic DNA. Pennsylvania State University, State College, PA
27. Lee W-P, Stromberg MP, Ward A, Stewart C, Garrison EP, Marth GT (2014) MOSAIK: a hash-based algorithm for accurate next-generation sequencing short-read mapping. PLoS One 9:e90581
28. Langmead B, Salzberg SL (2012) Fast gapped-read alignment with Bowtie 2. Nat Methods 9:357–359. doi:10.1038/nmeth.1923
29. Li R, Yu C, Li Y, Lam T-W, Yiu S-M, Kristiansen K, Wang J (2009) SOAP2: an improved ultrafast tool for short read alignment. Bioinformatics 25:1966–1967. doi:10.1093/bioinformatics/btp336
30. Li H, Homer N (2010) A survey of sequence alignment algorithms for next-generation sequencing. Brief Bioinform 11:473–483. doi:10.1093/bib/bbq015
31. Mielczarek M, Szyda J (2015) Review of alignment and SNP calling algorithms for next-generation sequencing data. J Appl Genet (in press)
32. Wajid B, Serpedin E (2012) Review of general algorithmic features for genome assemblers for next generation sequencers. Genomics Proteomics Bioinformatics 10:58–73
33. El-Metwally S, Hamza T, Zakaria M, Helmy M (2013) Next-generation sequence assembly: four stages of data processing and computational challenges. PLoS Comput Biol 9:e1003345
34. Salzberg SL, Phillippy AM, Zimin A, Puiu D, Magoc T, Koren S, Treangen TJ, Schatz MC, Delcher AL, Roberts M, Marçais G, Pop M,

Yorke JA (2012) GAGE: a critical evaluation of genome assemblies and assembly algorithms. Genome Res 22:557–567

35. Simpson JT, Wong K, Jackman SD, Schein JE, Jones SJM, Birol I (2009) ABySS: a parallel assembler for short read sequence data. Genome Res 19:1117–1123
36. Hernandez D, François P, Farinelli L, Osterås M, Schrenzel J (2008) De novo bacterial genome sequencing: millions of very short reads assembled on a desktop computer. Genome Res 18:802–809
37. Warren RL, Sutton GG, Jones SJM, Holt RA (2007) Assembling millions of short DNA sequences using SSAKE. Bioinformatics 23:500–501. doi:10.1093/bioinformatics/btl629
38. The MIRA assembler. Accessed 15 Jun 2015 from http://sourceforge.net/projects/mira-assembler/
39. Gladman S, Seemann T VelvetOptimiser. Accessed 15 Jun 2015 from http://bioinformatics.net.au/software.velvetoptimiser.shtml
40. Zerbino DR, Birney E (2008) Velvet: algorithms for de novo short read assembly using de Bruijn graphs. Genome Res 18:821–829
41. Boetzer M, Henkel CV, Jansen HJ, Butler D, Pirovano W (2011) Scaffolding pre-assembled contigs using SSPACE. Bioinformatics 27:578–579. doi:10.1093/bioinformatics/btq683
42. Ronen R, Boucher C, Chitsaz H, Pevzner P (2012) SEQuel: improving the accuracy of genome assemblies. Bioinformatics 28:i188–i196. doi:10.1093/bioinformatics/bts219
43. Dayarian A, Michael TP, Sengupta AM (2010) SOPRA: scaffolding algorithm for paired reads via statistical optimization. BMC Bioinformatics 11:345. doi:10.1186/1471-2105-11-345
44. Lin S-H, Liao Y-C (2013) CISA: contig integrator for sequence assembly of bacterial genomes. PLoS One 8:e60843. doi:10.1371/journal.pone.0060843
45. Kurtz S, Phillippy A, Delcher AL, Smoot M, Shumway M, Antonescu C, Salzberg SL (2004) Versatile and open software for comparing large genomes. Genome Biol 5:R12
46. Angiuoli SV, Salzberg SL (2011) Mugsy: fast multiple alignment of closely related whole genomes. Bioinformatics 27:334–342
47. Darling AE, Mau B, Perna NT (2010) progressiveMauve: multiple genome alignment with gene gain, loss and rearrangement. PLoS One 5:e11147
48. Li H, Handsaker B, Wysoker A, Fennell T, Ruan J, Homer N, Marth G, Abecasis G, Durbin R (2009) The Sequence Alignment/Map format and SAMtools. Bioinformatics 25:2078–2079. doi:10.1093/bioinformatics/btp352
49. Garrison E, Marth G (2012) Haplotype-based variant detection from short-read sequencing. arXiv Prepr arXiv12073907 342:9. doi: arXiv:1207.3907 [q-bio.GN]
50. McKenna A, Hanna M, Banks E, Sivachenko A, Cibulskis K, Kernytsky A, Garimella K, Altshuler D, Gabriel S, Daly M, DePristo MA (2010) The Genome Analysis Toolkit: a MapReduce framework for analyzing next-generation DNA sequencing data. Genome Res 20:1297–1303. doi:10.1101/gr.107524.110
51. Lukens AK, Ross LS, Heidebrecht R, Javier Gamo F, Lafuente-Monasterio MJ, Booker ML, Hartl DL, Wiegand RC, Wirth DF (2014) Harnessing evolutionary fitness in Plasmodium falciparum for drug discovery and suppressing resistance. Proc Natl Acad Sci U S A 111:799–804
52. Veenemans J, Overdevest IT, Snelders E, Willemsen I, Hendriks Y, Adesokan A, Doran G, Bruso S, Rolfe A, Pettersson A, Kluytmans JAJW (2014) Next-generation sequencing for typing and detection of resistance genes: performance of a new commercial method during an outbreak of extended-spectrum-beta-lactamase-producing Escherichia coli. J Clin Microbiol 52:2454–2460
53. Al-Shahib A, Underwood A (2013) snp-search: simple processing, manipulation and searching of SNPs from high-throughput sequencing. BMC Bioinformatics 14:326
54. Delcher AL, Bratke KA, Powers EC, Salzberg SL (2007) Identifying bacterial genes and endosymbiont DNA with Glimmer. Bioinformatics 23:673–679
55. Lowe TM, Eddy SR (1997) tRNAscan-SE: a program for improved detection of transfer RNA genes in genomic sequence. Nucleic Acids Res 25:955–964
56. Seemann T (2014) Prokka: rapid prokaryotic genome annotation. Bioinformatics 30:2068–2069. doi:10.1093/bioinformatics/btu153
57. Hyatt D, Chen G-L, Locascio PF, Land ML, Larimer FW, Hauser LJ (2010) Prodigal: prokaryotic gene recognition and translation initiation site identification. BMC Bioinformatics 11:119
58. Lagesen K, Hallin P, Rødland EA, Staerfeldt H-H, Rognes T, Ussery DW (2007) RNAmmer: consistent and rapid annotation of ribosomal RNA genes. Nucleic Acids Res 35:3100–3108
59. Laslett D (2004) ARAGORN, a program to detect tRNA genes and tmRNA genes in nucleotide sequences. Nucleic Acids Res 32:11–16
60. Petersen TN, Brunak S, von Heijne G, Nielsen H (2011) SignalP 4.0: discriminating signal peptides from transmembrane regions. Nat Methods 8:785–786

61. Nawrocki EP, Eddy SR (2013) Infernal 1.1: 100-fold faster RNA homology searches. Bioinformatics 29:2933–2935
62. Cingolani P, Platts A, Wang L, Coon M, Nguyen T, Land S, Lu X, Ruden D (2012) A program for annotating and predicting the effects of single nucleotide polymorphisms, SnpEff: SNPs in the genome of Drosophila melanogaster strain w1118; iso-2; iso-3. Fly 6:80–92. doi:10.4161/fly.19695
63. Rajasekar A, Moore R, Hou C-Y, Lee CA, Marciano R, de Torcy A, Wan M, Schroeder W, Chen S-Y, Gilbert L, Tooby P, Zhu B (2010) iRODS primer: integrated rule-oriented data system. Synth Lect Inf Concepts, Retrieval, Serv 2:1–143. doi:10.2200/S00233ED1V01Y200912ICR012
64. Allan C, Burel J-M, Moore J, Blackburn C, Linkert M, Loynton S, MacDonald D, Moore WJ, Neves C, Patterson A, Porter M, Tarkowska A, Loranger B, Avondo J, Lagerstedt I, Lianas L, Leo S, Hands K, Hay RT, Patwardhan A, Best C, Kleywegt GJ, Zanetti G, Swedlow JR (2012) OMERO: flexible, model-driven data management for experimental biology. Nat Methods 9:245–253. doi:10.1038/nmeth.1896
65. Leo S, Pireddu L, Zanetti G (2012) SNP genotype calling with MapReduce, Proc. third Int. Work. MapReduce its Appl. Date - MapReduce'12. ACM, New York, NY, p 49
66. Blankenberg D, Gordon A, Von Kuster G, Coraor N, Taylor J, Nekrutenko A (2010) Manipulation of FASTQ data with Galaxy. Bioinformatics 26:1783–1785
67. Blankenberg D, Taylor J, Nekrutenko A (2011) Making whole genome multiple alignments usable for biologists. Bioinformatics 27:2426–2428
68. FASTQ paired-end interlacer. Accessed 15 Jun 2015 from https://toolshed.g2.bx.psu.edu/view/devteam/fastq_paired_end_interlacer/b89bdf6acb6c
69. VelvetOptimizer. Accessed 15 Jun 2015 from https://github.com/tseemann/VelvetOptimiser
70. Bankevich A, Nurk S, Antipov D, Gurevich AA, Dvorkin M, Kulikov AS, Lesin VM, Nikolenko SI, Pham S, Prjibelski AD, Pyshkin AV, Sirotkin AV, Vyahhi N, Tesler G, Alekseyev MA, Pevzner PA (2012) SPAdes: a new genome assembly algorithm and its applications to single-cell sequencing. J Comput Biol 19:455–477. doi:10.1089/cmb.2012.0021