Chapter 7

A Practical Guide for Comparative Genomics of Mobile Genetic Elements in Prokaryotic Genomes

Danillo Oliveira Alvarenga, Leandro M. Moreira, Mick Chandler, and Alessandro M. Varani

Abstract

Mobile genetic elements (MGEs) are an important feature of prokaryote genomes but are seldom well annotated and, consequently, are often underestimated. MGEs include transposons (Tn), insertion sequences (ISs), prophages, genomic islands (GEIs), integrons, and integrative and conjugative elements (ICEs). They are intimately involved in genome evolution and promote phenomena such as genomic expansion and rearrangement, emergence of virulence and pathogenicity, and symbiosis. In spite of the annotation bottleneck, there are so far at least 75 different programs and databases dedicated to prokaryotic MGE analysis and annotation, and this number is rapidly growing. Here, we present a practical guide to explore, compare, and visualize prokaryote MGEs using a combination of available software and databases tailored to small scale genome analyses. This protocol can be coupled with expert MGE annotation and exploited for evolutionary and comparative genomic analyses.

Key words Transposons, Insertion sequences, Prophages, Clustered regularly interspaced short palindromic repeats, Genomic islands, Integrons, Integrative conjugative elements, Evolution, Genomics

1 Introduction

Most genome annotation pipelines developed to date are dedicated to the prediction and characterization of coding regions and their putative products, signal peptides, pseudogenes, and noncoding RNAs. Surprisingly, annotation of ubiquitous genomic features such as Mobile Genetic Elements (MGEs) is generally not fully addressed and/or is neglected in most of these pipelines, and thus MGEs are seldom well characterized and annotated. Consequently, they are generally underestimated, impacting the quality of the annotation and potentially affecting downstream genome analyses.

Transposons (Tn), Insertion Sequences (ISs), Prophages, Clustered Regularly-Interspaced Short Palindromic Repeats (CRISPRs), Genomic Islands (GEIs), Integrons and Integrative and Conjugative Elements (ICEs) are the most common MGEs found in prokaryote genomes and their properties and behavior are important in lateral gene transfer events (LGT). These elements are known to promote genomic expansion and rearrangements, and the emergence of virulence, pathogenicity, and symbiosis. Over the last 15 years, several reviews have focused on the impact of LGT mediated by MGEs [1–13]. It is now generally recognized that MGEs are central to bacterial and archaeal genome evolution.

To begin an analysis of the MGE content of a given genome, it is first necessary to identify and annotate ordinary genome features such as coding sequences (CDSs), rRNAs, and tRNAs. There are several useful genome annotation pipelines that use a combination of distinct methods based on computer predictions for CDS detection and product assignment. Putative gene product attribution for each identified CDS is generally based on sequence similarity obtained by using BLAST [14] and Hidden Markov Model [15] searches against trusted databases such as RefSeq [16], Interpro [17], Pfam [18], and TIGRFAMs [19]. However, few sequencing projects include manual curation and validation, mostly relying on the annotation results of automatic computer predictions. Indeed, this generally results in quite variable annotation qualities [20], impacting downstream analyses.

For MGE detection and annotation it is essential to use a variety of approaches including additional specialized pipelines and software which are normally not implemented in a regular annotation pipeline. Most MGE de novo-detection software are not integrated into the most popular genome annotation pipelines, such as Prokka [21], the NCBI Prokaryotic Genome Annotation Pipeline (PGAP), the Rapid Annotation using Subsystem Technology (RAST) [22, 23], or the Integrated Microbial Genomes System/Expert Review (IMG/ER) [24]. The absence of MGE predictions is therefore an important example of the deficiencies in expert genome annotation. The resulting incomplete annotation files are regularly deposited in public databases and used by the scientific community for detailed evolutionary and comparative genomic analyses. Identification, annotation, and classification of each MGE type are becoming central in the prokaryote genomics field. A correct MGE annotation will ultimately lead to new insights into how prokaryotes are genetically tailored to their lifestyles [25].

A considerable number of software and databases have been developed for MGE detection and analysis in the last decades. This makes it tiresome and time consuming to evaluate the programs that are most suited for a given work-flow. Some MGE software and databases provide very precise and curated information, together with gold standard methods and protocols for annotation. These include the ISfinder Database [26] and ISsaga2 [25], for IS analysis and annotation, respectively, while other databases such as ICEberg [27], INTEGRALL [28], and PHAST [29] for ICEs, integrons and prophages, respectively, provide valuable information concerning other MGEs. Conversely, there are a number of predictors that provide de novo methods for MGE detection, giving the user the opportunity to investigate the detailed role of MGE in novel complete and draft genomes.

Tables 1 and 2 provide a list of software and databases published and currently available for MGE research and some of their features. There are several software and/or databases for each MGE, and therefore there is no single solution to deal with all MGE types at once. This makes it difficult to choose the most reliable and appropriate software and databases. In view of the number of possibilities available (as shown in Table 1), a novice in this research area will probably spend a considerable time switching between programs and deciding which should be included in their projects and methodologies. Figure 1 illustrates the canonical mechanisms of known lateral gene transfer events (transduction, conjugation, and transformation) in a prokaryotic cell, and the databases and software for detection and analysis of each MGE type used by this protocol. It is worth mentioning that it is not unusual to find chimeras between different MGE types (Fig. 1, dashed boxes), making it difficult to distinguish their boundaries. These concepts are discussed in this protocol.

To provide an easier introduction to this subject, we will present a standard MGE bioinformatics work-flow for prediction and annotation using established methods and tools and provide tips for MGE investigation and analysis. This chapter will show how to run de novo MGE predictors, how to extract and interpret results, and how to visualize these results in circular or linear genome maps by using public bioinformatics software. Overall, the results of the analyses generated by this protocol can be exploited for expert annotation and comparative genomics analysis with genome browsers and annotation tools, such as Gview [96] and Artemis [97].

2 Materials

For de novo MGE prediction and comparative genomics analysis, a modern computer containing a 64-bit processor, with at least 4 GB of RAM, a 500 GB hard drive, and broadband Internet access are usually enough for meeting the minimum requirements of most software. Additionally, as presented in Table 1, a Unix-like operating system (OS) is required by the majority of the available MGE analysis software.

In these examples, we use Ubuntu 16.04, a free and open source Linux distribution that can be downloaded from https://www.ubuntu.com. In most hardware, the 64-bit version of Ubuntu can be installed either as your sole OS, alongside a preexisting OS (such as Microsoft Windows or Apple OS X) or in a virtual machine

Table 1
Features from available software and databases for prokaryotic MGEs. Operating system was assumed as including both Linux and Apple Mac OS X when none was mentioned in documentation. When no license was specified, the software was assumed to be proprietary unless source code is available

Tool	Use	Availability	License	OS	References
A: Software and databases for transposon (insertion sequence) analysis					
ESSENTIALS	Transposons insertion analysis	Web server, Perl/R scripts	AGPL 3	Linux, Mac, Windows (browser-based)	[30]
ISbrowser	Insertion sequences visualization	Web server	Proprietary	Linux, Mac, Windows (browser-based)	[31]
IScan	Insertion sequences identification	Perl/C package	GPL 2	Linux	[32]
ISCR elements	Insertion sequences common region (ISCR) database	Web database	Proprietary	Linux, Mac, Windows (browser-based)	[7]
ISfinder	Insertion sequences database	Web server	Proprietary	Linux, Mac, Windows (browser-based)	[26]
ISMapper	Transposase insertion sites identification	Python package	Modified BSD	Linux, Mac	[33]
ISQuest	Insertion sequences identification	C/C++ package	GPL 3	Linux, Windows	[34]
ISsaga	Insertion sequences identification and annotation	Web server	Proprietary	Linux, Mac, Windows (browser-based)	[25]
OASIS	Insertion sequences identification and annotation	Python scripts	Unspecified (source available)	Linux, Mac	[35]
PRAP	Transposons identification	Perl scripts	GPL 3	Linux, Mac	[36]
Recon	Transposons identification and classification	C/Perl package	GPL 2	Linux	[37]
Red	Transposons detection	C++ package	Public domain	Linux, Mac	[38]
RepeatFinder	Transposons detection	C++/Perl package	Artistic	Linux, Mac	[39]
RepeatMasker	Transposons detection	C++/Perl package	OSL 2.1	Linux	[40]

(continued)

Table 1
(continued)

Tool	Use	Availability	License	OS	References
Repseek	Transposons detection	C package	LGPL 2	Linux, Mac	[41]
RISCI	Transposons identification	Perl scripts	Unspecified (source available)	Linux	[42]
TnpPred	Insertion sequences prediction	Web server	Proprietary	Linux, Mac, Windows (browser-based)	[43]
B: Software and databases for prophage and CRISPR analysis					
ACLAME	Prophages database	Web server	Unspecified (available for download)	Linux, Mac, Windows (browser-based)	[44]
Phage_Finder	Prophages identification	Perl scripts	GPL 2	Linux, Mac	[45]
PHAST	Prophages identification and annotation	Web server	Unspecified (available for download)	Linux, Mac, Windows (browser-based)	[29]
PhiSpy	Prophages prediction	Python/R/C++ package	Unspecified (source available)	Linux, Mac	[46]
Prophage database	Prophages database	Web server	Proprietary	Linux, Mac, Windows (browser-based)	[47]
Prophinder	Prophages prediction	Web server	Proprietary	Linux, Mac, Windows (browser-based)	[48]
VirSorter	Prophages prediction	Perl/C package	GPL 2	Linux	[49]
CRASS	CRISPR assembly in metagenomic data	C++/Shell package	GPL 3	Linux, Mac	[50]
CRISPI	CRISPR database	Web server	Proprietary	Linux, Mac, Windows (browser-based)	[51]
CRISPRcompar	CRISPR comparison	Web server	Proprietary	Linux, Mac, Windows (browser-based)	[52]

(continued)

Table 1
(continued)

Tool	Use	Availability	License	OS	References
CRISPRdb	CRISPR database and prediction	Web server	Proprietary	Linux, Mac, Windows (browser-based)	[53]
CRISPResso	CRISPR analysis	Web server, Python/Java/C package	AGPL 3	Linux, Mac, Windows (browser-based)	[54]
CRISPRfinder	CRISPR prediction	Web server	Proprietary	Linux, Mac, Windows (browser-based)	[55]
CRISPRmap	CRISPR classification	Web server	Proprietary	Linux, Mac, Windows (browser-based)	[56]
CRISPR recognition tool	CRISPR prediction	Java package	Public domain	Linux, Mac, Windows	[57]
CRISPRstrand	CRISPR prediction	Web server	Proprietary	Linux, Mac, Windows (browser-based)	[58]
CRISPR target	CRISPR prediction and analysis	Web server	Proprietary	Linux, Mac, Windows (browser-based)	[59]
MINCED	CRISPR prediction	Java package	GPL 3	Linux, Mac, Windows	[60]
PILER-CR	CRISPR prediction	C package	Public domain	Linux, Mac	[61]
C: Software and databases for genomic and pathogenicity island analysis					
EGID	Genomic islands prediction	Perl/Java/C++ package	Custom (source available)	Linux	[62]
GIDetector	Genomic islands classification	Perl/Java package	GPL 2	Windows	[63]
GIHunter	Genomic islands prediction	Perl/Java/C++ package	Custom (source available)	Linux	[64]
GIPSY	Genomic islands prediction	Java package	GPL	Linux, Windows	[65]
GIST	Genomic islands prediction	Perl/Java/C++ package	Custom (source available)	Linux	[66]

(continued)

Table 1 (continued)

Tool	Use	Availability	License	OS	References
GIV	Genomic islands visualization	C++ package	Proprietary	Linux, Mac	[67]
IGPIT	Genomic islands prediction	Web server	Proprietary	Linux, Mac, Windows (browser-based)	[68]
Islander	Genomic islands identification and database	Perl scripts, web server	Unspecified (source available)	Linux, Mac; Linux, Mac, Windows (browser-based)	[69]
IslandHunter	Genomic islands prediction	Java package	Proprietary	Linux, Mac, Windows	[70]
IslandPath	Genomic islands prediction	Perl scripts	GPL 2	Linux	[71]
IslandPick	Genomic islands comparison and identification	Perl scripts	GPL 2	Linux	[72]
IslandViewer	Genomic islands identification and visualization	Perl scripts	GPL 2	Linux	[73]
Mobilome-FINDER	Genomic islands detection	Web server	Proprietary	Linux, Mac, Windows (browser-based)	[74]
MSGIP	Genomic islands prediction	Java package	Unspecified (source available)	Linux, Mac, Windows	[75]
OligoWords	Genomic islands identification	Python script	CC-BY	Linux, Mac	[76]
SeqWord genome browser	Genomic islands identification and visualization	Web server, python scripts	Unspecified (source available)	Linux, Mac, Windows (browser-based)	[77]
Colombo/SIGI-HMM	Genomic islands prediction	Java package	Proprietary	Linux, Mac, Windows	[78]
tRNAcc	Genomic islands identification	C++/Perl package	Proprietary (source available)	Windows	[79]
Alien_Hunter	Pathogenicity islands prediction	Perl/Java package	GPL 2	Linux, Mac	[80]
PAIDB	Pathogenicity islands database	Web server	Proprietary	Linux, Mac, Windows (browser-based)	[81]
PIPS	Pathogenicity islands prediction	Perl scripts	GPL 3	Linux, Mac	[82]

(continued)

Table 1
(continued)

Tool	Use	Availability	License	OS	References
PredictBias	Genomic and pathogenicity islands prediction	Web server	Proprietary	Linux, Mac, Windows (browser-based)	[83]
D: Databases for integron and integrative conjugative elements (ICE)					
ACID	Integrons database	Web server	Proprietary	Linux, Mac, Windows (browser-based)	[84]
ICEberg	Integrative and conjugative elements database	Web server (available for download)	Proprietary	Linux, Mac, Windows (browser-based)	[27]
INTEGRALL	Integrons database	Web server	Proprietary	Linux, Mac, Windows (browser-based)	[28]

Table 2
Currently unavailable software for prokaryotic MGE analysis. Some of these programs might be obtained upon request from authors

Tool	Target MGE	Reference
Centroid	Genomic islands	[85]
GI-POP	Genomic islands	[86]
PAI-IDA	Genomic islands	[87]
SIGI	Genomic islands	[88]
DOASIS/NOASIS	Insertion sequences	[89]
ISA	Insertion sequences	[90]
Prophage finder	Prophages	[91]
Unamed script	Prophages	[3]
SSFinder	CRISPR	[92]
IRs search	Transposable elements	[93]
MUST	Transposable elements	[94]
Profile HMM search	Transposable elements	[93]
Repeats search	Transposable elements	[93]
Transposon express	Transposable elements	[95]

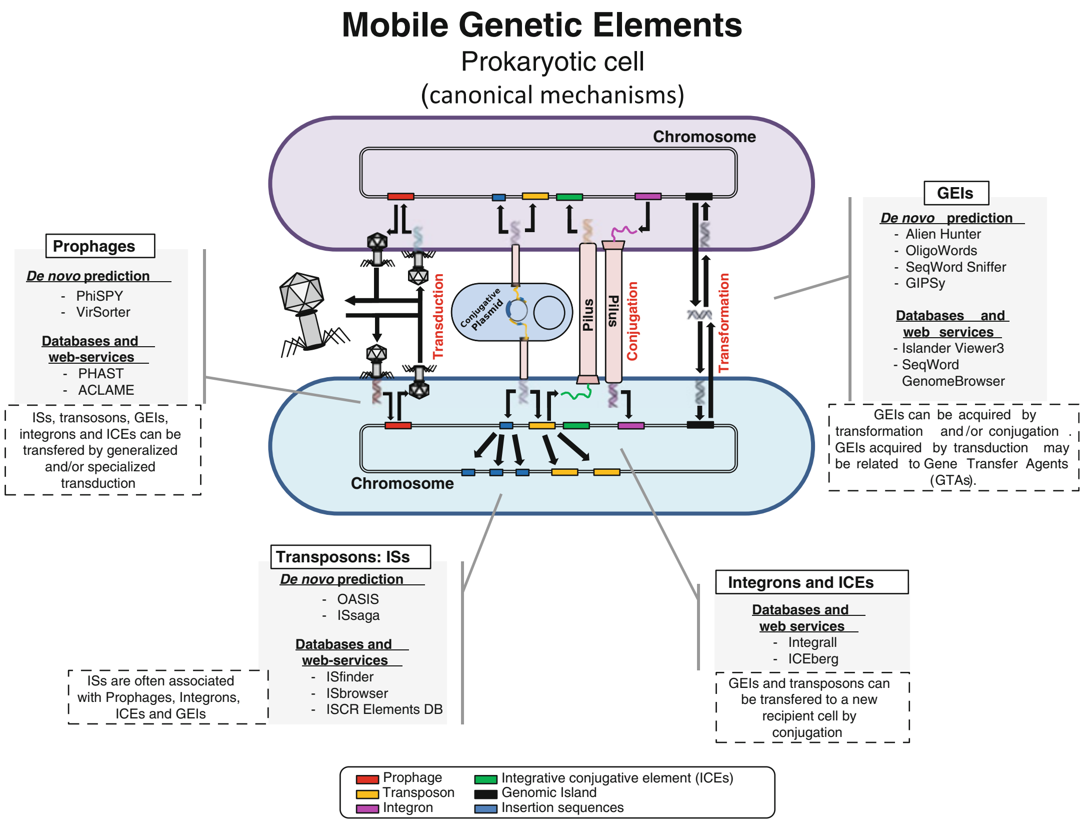
Mobile Genetic Elements
Prokaryotic cell
(canonical mechanisms)
Chromosome
Chromosome
Transduction
Conjugative Plasmid
Pilus
Pilus
Conjugation
Transformation
Prophages
De novo prediction
- PhiSPY
- VirSorter
Databases and web-services
- PHAST
- ACLAME
ISs, transosons, GEIs, integrons and ICEs can be transfered by generalized and/or specialized transduction
GEIs
De novo prediction
- Alien Hunter
- OligoWords
- SeqWord Sniffer
- GIPSy
Databases and web services
- Islander Viewer3
- SeqWord GenomeBrowser
GEIs can be acquired by transformation and/or conjugation. GEIs acquired by transduction may be related to Gene Transfer Agents (GTAs).
Transposons: ISs
De novo prediction
- OASIS
- ISsaga
Databases and web-services
- ISfinder
- ISbrowser
- ISCR Elements DB
ISs are often associated with Prophages, Integrons, ICEs and GEIs
Integrons and ICEs
Databases and web services
- Integrall
- ICEberg
GEIs and transposons can be transfered to a new recipient cell by conjugation
🟥 Prophage
🟨 Transposon
🟪 Integron
🟩 Integrative conjugative element (ICEs)
⬛ Genomic Island
🟦 Insertion sequences

Fig. 1 Canonical lateral gene transfer mechanisms present in a prokaryotic cell and their respective software and databases for analyses included in this guide. The dashed box shows chimeras between different MGE types which may occur

following instructions from the Ubuntu community help wiki (https://help.ubuntu.com/community/installation). However, for better performance, it is best to avoid virtual machines and install Ubuntu directly on the computer instead. To follow this guide a basic knowledge in management and analysis of directories, files and genomic data in a command line interface is also necessary.

We used 21 genomes from different bacterial and archaeal classes bearing distinct MGE types and numbers as examples for the methods described in this protocol. Final results for the analyses of these genomes can be obtained from https://github.com/danillo-alvarenga/mogece/blob/master/Analyses.tar.gz. To download these results, open the GitHub webpage, click on the "Download ZIP" link located on the right of your screen, and save it to an appropriate directory. We advise visualizing these files in addition to reading this chapter for a better understanding of what results the following methods should produce.

3 Methods

3.1 Preparing Your Dataset: Obtaining Prokaryote Genomes

For most MGE software and databases, either unannotated genome sequences in the plain fasta format (".fasta", ".fas", ".fa", ".fna", ".fnt" and ".fnn" file extensions) or annotated sequences in the GenBank format (".gbk" and ".gb" file extensions) or the new GenBank flatfile format (".gbff" file extension) are mandatory (*see* **Notes 1** and **2**). Thus, target genomes should be available in both formats so that a broad range of programs can be used. For novel genomes, SPAdes [98] is currently among the best options for assembling prokaryotic genomes from Illumina and Ion Torrent short reads, and Platanus [99] can be used as an additional step for scaffolding and gap-closing sequences.

To generate GenBank files there are several available genome annotation pipelines capable of providing CDS assignment and their putative products, signal peptides, non-coding RNAs, and other features. We recommend Prokka [21], an automated pipeline for annotating prokaryotic genomes, which runs on local hardware and can completely annotate a typical bacterial genome in under 10–15 min on a desktop or laptop computer. Conversely, three other satisfactory annotation pipelines are available on the following web-servers: RAST, IMG/ER, and NCBI Pipeline PGAP (for which further details are found in the NCBI handbook chapter at https://www.ncbi.nlm.nih.gov/books/NBK174280). All these pipelines generate standard-compliant output files.

If the user is not interested in analyzing novel genomes or wants to compare them to available genomes, two useful sources of bacterial and/or archeal genomes are the NCBI ftp server and the Genomes OnLine Database (GOLD) [100]. Available genomes can be retrieved by visiting ftp://ftp.ncbi.nlm.nih.gov/genomes/ on a web browser and navigating to *genbank/* at the bottom of the page, or browsing the GOLD website at https://gold.jgi.doe.gov/. Both databases include updated files on virtually every prokaryotic genome publicly released thus far.

3.2 Preparing Your Computer for MGE Analysis: Installing Dependencies

While some MGE analysis software provides web servers with browser-based user interfaces or graphical user interfaces, most available programs are based on command-line interfaces and need to be installed on a local computer running a Linux-based OS before analysis can be carried out (*see* **Note 3**). In some cases this can be difficult even with bioinformatics experience. Out-of-the-box Ubuntu lacks a few software elements and libraries on which these programs depend. Nevertheless, several of these dependencies can be easily installed by typing a few commands in a terminal emulator, a program that allows users to interact with computer programs by a text-based interface.

Ubuntu's terminal emulator can be accessed by typing "terminal" in the dash (the first button from top on the left-sided launcher bar) or pressing the <ctrl>+<alt>+<T> key combination. Although there are alternative terminal implementations, all commands described in this chapter are assumed to be typed in the standard Ubuntu terminal emulator. Before starting the analyses, follow the procedures below using this terminal emulator to prepare the system for the software described in the subsequent sections (*see* **Notes 4** and **5**).

First, it is important to install essential software for building programs and make sure the system has all that is needed for compiling sources. For this purpose, issue the following command to the apt-get package manager (*see* **Note 6**): *sudo apt-get install build-essential.*

Navigation in the terminal is achieved by the command *cd* followed by a space and the directory name. Since it takes some time to familiarize oneself with this method, it is useful to install a shortcut for opening a directory in the terminal directly from the Ubuntu file manager also using the apt-get manager: *sudo apt-get install nautilus-open-terminal.*

Software written in the Java programming language is commonly distributed as .jar files. These files depend on a runtime environment that does not come pre-installed in Ubuntu systems. To run Java-based applications, install the Open Java Development Kit: *sudo apt-get install openjdk-7-jdk.*

R is an open-source computing language widely used for statistics on which some bioinformatics tools rely. Most R packages are included in the Comprehensive R Archive Network (CRAN), but the Ubuntu repositories include some of these packages, which can also be installed with the apt-get manager for an easier process. Install the basic R files and the randomForest package with the following command: *sudo apt-get install r-base r-cran-randomforest.*

Bioinformatics software written in the Python programming language may sometimes depend on Biopython, a library containing several bioinformatics tools, or TkInter, an interface to the GUI toolkit Tk. Both libraries can be easily installed with the apt-get manager: *sudo apt-get install python-biopython python-tk.*

A different approach to dependencies is implemented by Docker, which is able to install a program and all its dependencies and create containers for this software. For bioinformatics programs that use this distribution method, reboot the operating system after installing Docker and adding your username to the Docker group with the following command: *sudo apt-get install docker.io; sudo usermod -aG docker $USER.*

The NCBI BLAST suite is another popular dependency for bioinformatics programs. At present there are two BLAST versions in the Ubuntu repositories. Some programs rely on the legacy version while others depend on the current version (*see* **Note 7**). To avoid incompatibility issues, both versions can be installed concomitantly: *sudo apt-get install blast2 ncbi-blast+*.

3.3 Insertion Sequence Prediction

ISs are the smallest and simplest autonomous mobile genetic elements, ranging from 0.7 to 3.5 kbp, and generally include a transposase gene encoding the enzyme that catalyzes IS movement. They are classified into about 28 different families on the basis of the relatedness of transposases and overall genetic organization [13]. The ISfinder database (https://www-is.biotoul.fr/) is the reference center for IS analysis, providing a gold standard method and protocol for IS identification, name attribution, annotation, and classification [25]. We strongly recommend following the ISfinder annotation protocol, using a manual expert annotation of each predicted IS, with the ISsaga web-service. These instructions can be found on the ISsaga Manual in the "About" section of the ISsaga main webpage (http://issaga.biotoul.fr/ISsaga2/issaga_manual.pdf). Other available software, such as OASIS [35] and ISQuest [59], are dedicated to de novo predictions, providing valuable resources for IS discovery in a given genome. This guide focuses on ISsaga and OASIS, but can be exploited with ISQuest and other IS software and databases shown in Table 1-A (*see* **Note 8**).

3.3.1 IS Detection with the ISsaga2 Web-Service

1. Open the ISsaga2 webpage at http://issaga.biotoul.fr.
2. Log-in into the system, or create a new account.
3. On the "ISsaga public section", click on Start Annotation.
4. Fill-in the forms, and choose which type of genome file (".fasta" or ".gbk") is to be analyzed. Click on the "Send/Run" button, and wait for the results.
5. Follow the ISfinder/ISsaga gold standards for IS annotation, naming and validating each predicted IS, according to the ISsaga manual (optional).
6. On the "Annotation Report" webpage of your submitted genome, click on the pull-down menu "Annotation," and choose the "Extract Annotation" option.
7. Click on the "Excel annotation file" button, and download the ".xls" file to your genome folder.
8. Open the ".xls" file in your favorite spreadsheet program, convert it to the comma-separated values format (".csv") and save this new file in your genome folder.
9. If the ISfinder/ISsaga gold standards for IS annotation were followed to the end, extract the final annotation results in the "Annotation" tab, and choose "Extract Annotation" option. On the new webpage, click on the "GenBank annotation file", and use this new GenBank file to visualize the results in the GView or Artemis genome viewers (optional).

3.3.2 IS Detection with OASIS

1. Download the OASIS source code from https://github.com/dgrtwo/OASIS.
2. Go to the Downloads folder in the file manager, right-click on the downloaded file and select "extract here."
3. Move the extracted folder to the directory where software and database files will be stored. In this and the following examples, we will use */home/bioinfo/software* as the default path for the installed software and */home/bioinfo/genomes* for genome files. Replace this path with your own directory paths whenever necessary (*see* **Note 9**).
4. Go to the folder where a *setup.py* file can be found, right-click on a blank space and select "open in terminal." In the opened terminal, install the software:

   ```
   python setup.py build; sudo python setup.py install
   ```
5. Find the data subdirectory inside the src directory and modify the *data.cfg* file to the following:

   ```
   [BLAST]
   BLAST_EXE=/usr/bin/blastall
   FORMAT_EXE=/usr/bin/formatdb
   ```
6. On these examples, we also use *TargetGenome* as proxy for the genome filename and *AnalysisDirectory* for the results directory, which should be replaced with the names of your file and directory. Open a terminal from your genome folder and run OASIS with the appropriate file and directory names:

   ```
   OASIS -g TargetGenome.gb -o OASISAnalysisDirectory
   ```
7. Repeat the previous step for each genome you wish to analyze.
8. OASIS will produce two files – a fasta file containing both nucleotide and amino acid sequences of each detected IS, and a General Feature Format file (".gff" extension) containing information about the predicted insertion sequences. Please note that, unlike ISsaga, OASIS does not classify ISs into families. The data obtained from OASIS must be parsed through ISfinder for this purpose (*see* **Note 10**).

3.4 Prophage and Clustered Regularly Interspaced Short Palindromic Repeat Prediction

Temperate or lysogenic phages generally integrate their genomes into the bacterial chromosome as prophages, which replicate passively with the host chromosome until conditions favor their reactivation. Integration occurs by site-specific recombination often into tRNA/tmRNA genes [46] (*see* **Note 11**). Prophages can encode virulence-related genes converting non-pathogenic bacteria into pathogens. It is well known that prophage identification in bacterial genomes is a difficult process [46]. Indeed, the longer and more complex the MGE the more difficult it is to correctly identify it. Prophages can also be confounded with Gene Transfer Agents (GTAs), which also resemble phages in a subtle way

[101]. GTAs are widespread in bacteria and may also be involved in virulence and adaptation [102].

CRISPRs are specific loci which provide acquired immunity against viruses and plasmids [103]. Therefore, in addition to the prophage prediction and analysis, CRISPR detection also provides valuable information concerning the history of phage infection of the analyzed genome (*see* **Note 12**). CRISPR analysis should use both de novo approaches, such as MinCED and CRISPRdb (http://crispr.u-psud.fr/crispr/), one of the most complete databases for CRISPR analysis.

3.4.1 Prophage Detection with PhiSpy 2.3

1. Download the PhiSpy source code from https://sourceforge.net/projects/phispy/, extract the package and move it to your software directory.
2. Go to the folder where a makefile can be found, open a terminal from the right-click menu, and compile the executable files:

   ```
   make
   ```
3. Make all Python scripts executable:

   ```
   chmod +x *.py */*.py
   ```
4. Add the directory to your global environment variables (*see* **Note 13**):

   ```
   echo 'export PATH=$PATH:'$(pwd) >> ~/.bashrc; source ~/.bashrc
   ```
5. Add the Python interpreter path to the main PhiSpy scripts:

   ```
   sed -i '1i#!/usr/bin/env python' phiSpy.py; sed -i '1i#!/usr/bin/env python' genbank_to_seed.py
   ```
6. Go to the directory where the target genomes are hosted:

   ```
   cd /home/bioinfo/genomes
   ```
7. Transform the target genome file from the GenBank file format to a SEED directory:

   ```
   genbank_to_seed.py TargetGenome.gb GenomeSEEDDirectory
   ```
8. Run PhiSpy on the newly-created SEED directory:

   ```
   phiSpy.py -i GenomeSEEDDirectory -o PhiSpyAnalysisDirectory
   ```
9. Repeat **steps 9** and **10** for the other target genomes.
10. After the analysis is finished, the output directory will host several files, including a subdirectory named “results” containing a file named *prophage.tbl*. This file contains the location of the predicted prophages.

3.4.2 Prophage Detection with VirSorter 1.0.3

1. Download VirSorter with Docker:

   ```
   docker.io pull discoenv/virsorter:v1.0.3
   ```
2. Download the VirSorter database from http://mirrors.iplantcollaborative.org/browse/iplant/home/shared/imicrobe/VirSorter/virsorter-data.tar.gz.

3. Extract the downloaded file and move the VirSorter database folder to your software directory.
4. Create a VirSorter analysis folder in your genomes folder and copy the fasta files for your target genomes there.
5. Open a terminal and run VirSorter from Docker (*see* **Note 14**):

   docker.io run -v /home/bioinfo/software/virsorter-data:/data -v/home/bioinfo/genomes/VirSorterAnalysisDirectory:/wdir -w/wdir --rm. discoenv/virsorter:v1.0.3 --db 2 --fna/wdir/TargetGenome.fasta
6. Claim ownership on the created files and change permissions:

   *sudo chown -R $USER: *; chmod -R 755 **
7. Repeat **steps 5** and **6** for each target genome.
8. VirSorter main results will be output to a comma-separated table named VIRSorter_global_phage_signal.csv, containing possible viral and prophage sequences divided into categories according to the certainty of prediction. However, this table does not provide genomic coordinates for theses prophages. These coordinates can be found in the headers from fasta sequences written to the Predicted_viral_sequences directory.

3.4.3 CRISPR Detection with MinCED

1. Download the MinCED source code from https://github.com/ctSkennerton/minced.
2. Go to the Downloads folder in the file manager, right-click on the downloaded file and select "extract here."
3. Move the extracted folder to the directory where you wish to keep software and databases files.
4. Go to the folder where a makefile can be found, right-click on a blank space, and select "open in terminal." In the opened terminal, compile the executable files:

   make
5. Export the MinCED directory by issuing the following command:

   echo 'export PATH=$PATH:'$(pwd) >> ~/.bashrc; source ~/.bashrc
6. Still in the terminal, change to your genomes directory:

   cd /home/bioinfo/genomes
7. Run MinCED on your target genome:

   minced -gff TargetGenome.fasta MinCEDResults.gff
8. Repeat the previous step for each genome you wish to analyze.
9. By default, MinCED outputs a tab-separated values table to the terminal. However, by adding the *-gff* parameter to its command and pointing a filename for writing, MinCED will produce a ".gff" file containing information about the predicted CRISPRs.

3.5 Genomic Island Prediction

GEIs are central to the dissemination and evolution of a broad spectrum of bacteria [8]. GEIs can be correlated with pathogenicity, virulence, fitness, symbiosis, and cell metabolism, depending on the gene composition of the given GEIs. Therefore, the better the CDS annotation, the better the GEI classification. There are several available GEI detection software packages. Genomic Island Prediction software (GIPSy) [65] is one of the most complete and user-friendly GEI detection software currently available, and it is capable of distinguishing pathogenicity islands, metabolic islands, resistance islands, and symbiotic islands. We recommend downloading and following the GIPSy annotation through their interface (http://www.bioinformatics.org/groups/?group_id=1180). The instructions below show how to use Alien_Hunter and SeqWord Gene Island Sniffer software for GEI prediction (*see* **Note 15**).

3.5.1 GEI Prediction with Alien_Hunter 1.7

1. Download the Alien_Hunter source code from http://www.sanger.ac.uk/science/tools/alien-hunter, extract the package and move it to your software folder.
2. Go to the folder where an *alien_hunter* file can be found, right-click on a blank space, and select "open in terminal." In the opened terminal, add the directory to your global environment variables:

   echo 'export PATH=$PATH:'$(pwd) >> ~/.bashrc; source ~/.bashrc
3. Change to your genome directory in the terminal:

   cd /home/bioinfo/genomes
4. Run Alien_Hunter on your fasta-formatted genome:

   alien_hunter TargetGenome.fasta AlienHunterOutputFile
5. You may opt to view the resulting prediction file on the Artemis genome browser. To do so, follow the procedure for installing Artemis described in the first step of Subheading 3.4.2. To open the resulting prediction file on Artemis after analysis, add the *-a* parameter to the command illustrated in the **step 3**:

   alien_hunter TargetGenome.fasta AlienHunterOutputFile -a
6. Alien_Hunter will generate three result files: two feature tables and a plot file. Look for the ".sco" file, which indicates coordinates and scores for the predicted genomic islands.

3.5.2 GEI Prediction with OligoWords 1.2 and SeqWord Gene Island Sniffer 1.0

1. Download OligoWords and SeqWord Gene Island Sniffer from http://www.bi.up.ac.za/SeqWord/downloads/OligoWords_1.2.1_linux.zip and http://www.bi.up.ac.za/SeqWord/downloads/SWGIS_1.0.zip, respectively, extract the packages, and move them to your software folder.
2. Open a terminal in the directory where the OligoWords and the SeqWord Sniffer scripts can be found and add their paths to your environment variables:

echo 'export PATH=$PATH:'$(pwd) >> ~/.bashrc; source ~/.bashrc

3. Add the Python interpreter path to the main scripts in their corresponding directories:

   sed -i '1i#!/usr/bin/env python' OligoWords1.2.1.1.py
   sed -i '1i#!/usr/bin/env python' SeqWordSniffer.py

4. Change executable permissions for the scripts:

   *chmod +x OligoWords1.2.1.1.py ; chmod +x */**
   *chmod +x SeqWordSniffer.py; chmod +x */**

5. Find the *input* folder inside the extracted directories and copy target genomes in the fasta file format, renaming extensions from ".fasta" to ".fna."

6. Open a terminal in the scripts directories and run the main scripts:

   OligoWords1.2.1.1.py
   SeqWordSniffer.py

7. When prompted, type *Y* to analyze the genomes using the default parameters or follow the instructions for changing the settings according to your preferences.

8. For each genome, a ".out" file with the results will be written to the *output* folder inside the software directories.

9. SeqWord Gene Island Sniffer 1.0 generates a circular genomic map showing GEI location and nucleotide composition using different metrics. This map is useful to visualize the detected island, and must be compared with the GIPSy results for high-quality GEI annotation.

3.6 ICE and Integron Analysis and Prediction

ICEs are self-transmissible integrative elements found in both Gram-positive and Gram-negative bacteria encoding conjugation-related genes [27], whereas integrons show a diverse genetic organization, containing a number of combinations of gene cassettes, mostly related to antibiotic resistance and often associated with transposons and conjugative plasmids [28, 104, 105]. GEI prediction software, such as Alien_Hunter, may detect these anomalous regions; however, they are not capable of distinguishing GEIs, ICEs, and integrons. Therefore, in addition to the IS, prophage, and GEI predictions, a complete MGE analysis should include searches against the ICE and integron reference databases, such as ICEberg (http://db-mml.sjtu.edu.cn/ICEberg), and INTEGRALL (http://integrall.bio.ua.pt).

3.7 Visualizing Prediction Results in Circular and Linear Genomic Maps

GView is a program for visualizing genomes. Although not developed specifically for MGE analyses, it can be used for viewing the results obtained in the previous steps in a broader genomic context, either for complete genomes or single contigs. It is also possible to visualize the previous results with the Artemis genome browser, which additionally allows users to perform manual curation of annotations, thus making it very useful for refining automated predictions (*see* **Note 16**). However, most MGE prediction software does not generate results in formats compatible with these programs. To integrate predictions with these viewers, we have developed Mobile Genetic Element Coordinates Extractor (MoGECE), which extracts mobile genetic element locations in prokaryotic genomes from outputs created by the MGE analysis software described in the previous section and generates files compatible with GView and Artemis. The following instructions show how to draw linear and circular genomic maps and visualize CDS annotations coupled with the MGE predictions.

3.7.1 Formatting Prediction Results with MoGECE 1.0.1

1. Download the MoGECE package from https://www.github.com/danillo-alvarenga/mogece/, extract it and move it to a folder in your software directory.
2. Enter a terminal from the MoGECE folder and add it to your environment:

   echo 'export PATH=$PATH:'$(pwd) >> ~/.bashrc; source ~/.bashrc
3. Go to the directories containing the relevant output files described in the previous subsections and run the program indicating the output file, the program in which this file was created, and which program visualization should be used. For Alien_Hunter, for example, issue the following command to the terminal in the outputs directory:

   MoGECE.py -f alienhunter.sco -l -a -g
4. MoGECE will output either a feature table for visualization in Artemis, a comma-separated values table for GView, or both according to the parameters you have chosen when running the program. The files produced will be named according to the program output in which they were based, e.g., an Alien_Hunter output file used as input for MoGECE with the *-a* and *-g* parameters will be processed and two files will be created: alienhunter.csv and alienhunter.ft.

3.7.2 Visualizing Results on Artemis 16.0.0

1. Download and extract the Artemis compressed file on your software folder from ftp://ftp.sanger.ac.uk/pub/resources/software/artemis/artemis.tar.gz, open the directory in the terminal, and enter the following command:

   echo 'export PATH=$PATH:'$(pwd) >> ~/.bashrc ; source ~/.bashrc
2. Start Artemis from a terminal by typing *art*.
3. In the “Options” menu, select the bacterial and plant plastid Genetic Code Table 11.

4. In the "File" menu, select "Open" and open the target genome in the GenBank file format.
5. In the genome annotation window, go to the "File" menu and click on "Read an Entry." Navigate to the directory containing the feature table created with MoGECE, change file types to "all files," and select the table (*see* **Note 17**).

3.7.3 Visualizing Results on GView 1.7

1. Download the Ubuntu/Debian GView package from https://www.gview.ca/wiki/GViewDownload/.
2. Double-click on the downloaded ".deb" file and install it from the Ubuntu Software Center.
3. Open GView by searching for its name in the dash.
4. In the "Open Files" window, go to the folder where your target genome is stored and select the ".gb" genome file as your sequence data. You may also indicate a previous GView Style Sheet in this window. Click on "Build Map" for visualizing the target genome.
5. GView can provide additional plots from ranges represented in comma-separated values. To create the necessary csv files from outputs produced by the software described in the previous steps, move the csv files created with MoGECE and the GView-StyleSheet.gss file included with it to the same directory where the GenBank-formatted file is stored and indicate both the genome file and the style sheet in the "Open Files" window (*see* **Notes 18** and **19**).
6. The circular and linear genomic maps can be exported to ". svg," ".png," and ".jpg" figures format.

3.8 How to Interpret the Results

Figures 2 and 3 show examples of analyses generated by this protocol. These circular genomic maps were produced by the GView software, and are used as a template for interpreting the results obtained. GView can generate a circular or linear genomic representation which can be zoomed-in or zoomed-out with point-and-click capabilities, showing CDS annotations. This is very useful to analyze each prediction, as described below. Linear and circular layouts can be interchanged in the "View" drop-down menu.

Figure 2 shows a global representation of all predicted MGEs together with CDS annotations of the *Escherichia coli* O26-H11 11368 genome. This genome is known to carry a large number of ISs and prophages and dozens of regions with potential anomalous nucleotide composition. Four examples of anomalous regions marked with dashed boxes were further explored. Figure 2-region A illustrates a typical laterally acquired island. This region was detected by both Alien_Hunter and SeqWord Sniffer, showing an anomaly in the GC content. In this example, the anomalous regions show lower GC content when compared to the genome average.

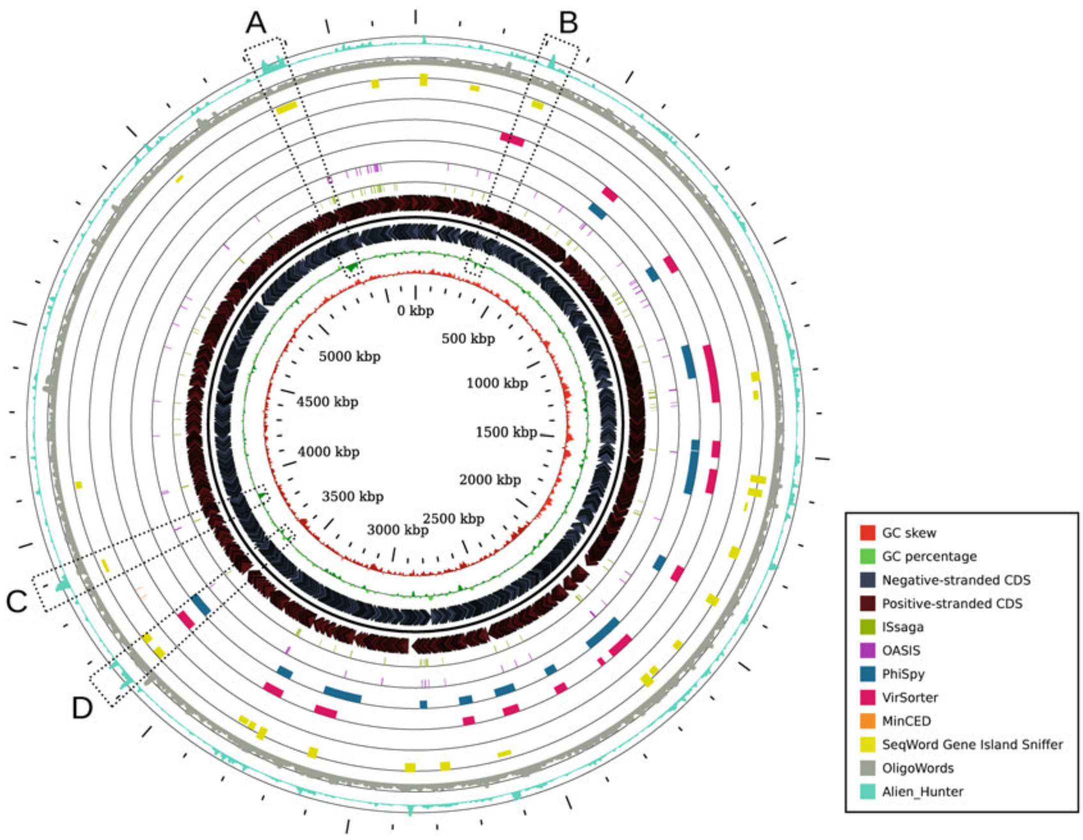

Fig. 2 Circular genomic representation of CDS and MGEs. GView circular representation of the *Escherichia coli* O26-H11 11368 genome illustrating from the inner to the outer circle in the following order: GC skew, GC content, CDS annotation, ISsaga, OASIS, PhiSpy, VirSorter, MinCED, SeqWord Gene Island Sniffer, OligoWords, and Alien_Hunter predictions. Dashed boxes A–D illustrate anomalous regions identified by different software

Depending on the analyzed genome, the laterally acquired island may show either higher or lower GC content when compared to the genome average. Abrupt changes in the GC content and GC skew are well-characterized landmarks of laterally acquired regions. However, this should not be taken as a rule, as GC characteristics are not unique parameters to identify laterally acquired islands. For instance, other cellular mechanisms such as intra-molecular recombination may generate similar patterns, and thus result in false positive predictions. Therefore, further manual inspection of the gene content in these regions is mandatory, and explained below. The CDS content of this region shows open reading frames (ORFs) encoding several hypothetical genes, IS transposases, chaperones, a lytic transglycosylase, and several genes from the type III secretion system, suggesting that it is a pathogenicity island. Figure 2-region B illustrates a typical GEI region, showing an integrase gene located on the border of the GEI. In addition, other ORFs, such as phage-

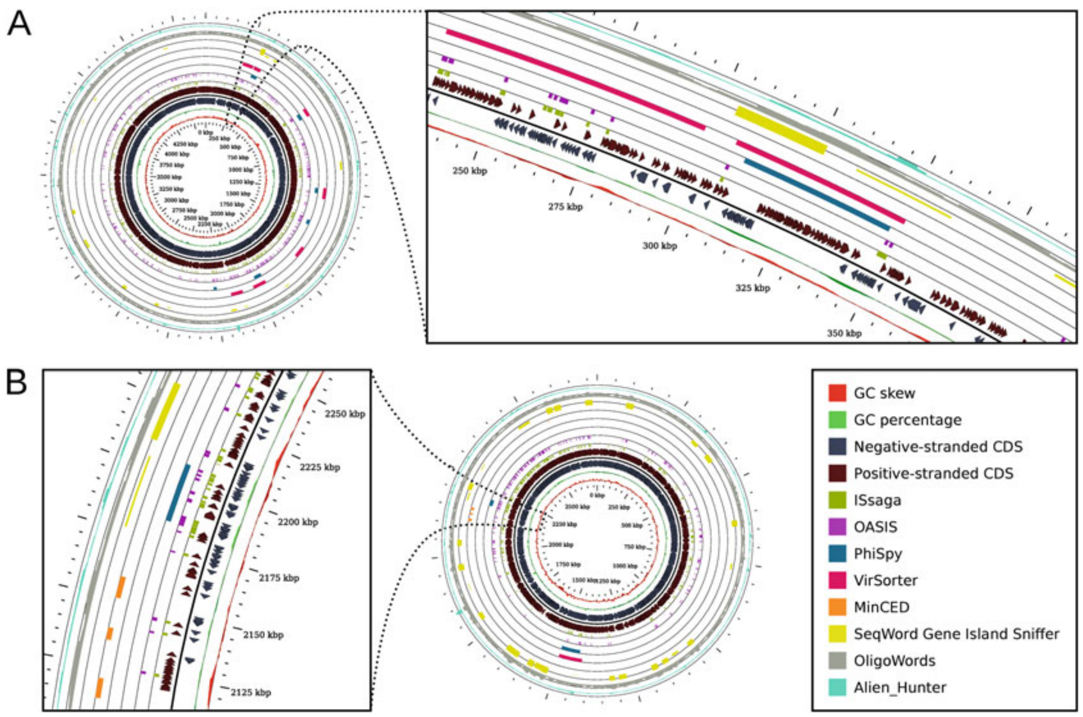

Fig. 3 Interpreting the circular maps results. Detailed Gview map (zoom-in) highlighting genomic regions with MGE predictions in bacterial and archaeal strains. (**a**) *Shigella flexneri* 2002017. (**b**) *Sulfolobus solfataricus* SULA

and flagellar-related genes, and several hypothetical ORFs were identified. Figure 2-region C illustrates a typical pathogenicity island detected by Alien_Hunter and SeqWord Sniffer, with atypical GC content and encoding ORFs related to a type III secretion system. Figure 2-region D and the other regions marked by PhiSpy and VirSorter are a typical prophage region, encoding integrases, holin, lysozyme and structural phages genes with homology to the lambda phage.

The left panel of Fig. 3a shows that *Shigella flexneri* carries a large number of ISs spread over its chromosome, together with some prophage regions and GEIs. The right panel of Fig. 3a shows a zoom-in of a specific region pointing that both ISsaga and OASIS generated similar IS predictions, indicating a consistent result. However, some differences were noted between PhiSpy and VirSorter predictions: while PhiSpy detected only one prophage (blue bar), VirSorter detected two regions (pink bar). CDS analyses of the region detected only by VirSorter indicate a typical GEI, containing integrase, phage-related and IS transposase ORFs. The region detected by both PhiSpy and VirSorter was also marked by SeqWord Sniffer (yellow bar) and OligoWords (gray lines) predictions, which indicates a prophage region encoding several phage-related genes.

The right panel of Fig. 3 shows that *Sulfolobus solfataricus* carries large numbers of ISs and GEIs, and few prophages. The low number of prophages might be due to the presence of at least three CRISPR loci detected by MinCED (orange bars on the left panel). The CDS analysis of the surrounding genes of the CRISPR loci indicate the presence of Cas1 to 7 ORFs (CRISPR-associated genes), suggesting that this archaea may have had a long history of battle against phages. Indeed, a comparative analysis with the use of the CRISPRdb indicates that this genus carries several CRISPR loci. The other regions marked on the right panel follow the pattern shown in previous figure examples.

Overall, these results emphasize that multiple predictors and databases must be used for a proper MGE analysis (*see* **Note 20**), and that manual expert annotation is a mandatory second annotation step.

3.8.1 Chimeric MGEs

As mentioned above and illustrated in Figs. 2 and 3, it is relatively frequent to find chimeric MGEs. For instance, prophages and GEIs often carry ISs which are the product of different LGT events and must be considered separate entities with a distinct evolutionary history. Similarly, prophages are often integrated into GEIs, and these can undergo further IS invasion. Indeed, in certain cases, some GEIs may be generated by a number of different combinations and recombinations between prophages and IS.

Excluding plasmids, which exist as autonomously replicating extrachromosomal entities and are generally annotated separately, elements that are normally integrated into the chromosome but are self-transmissible by conjugation are classified as ICE. Interestingly, some GEIs are transferred by conjugation and carry genes with similarities to known plasmid transfer systems, indicating a common origin, whereas others can be transferred by packaging into phages [8]. ISCRs (Insertion Sequence with a Common Region) are related to ISs of the IS*91* family, but unlike IS*91* family members are generally associated with several antibiotic resistance genes [7]. In addition, an IS can insert within other ISs, resulting in structures like "russian dolls." A similar phenomenon, generated by a number of consecutive phage invasions, is also observed with prophages [2].

Thus, certain MGEs may include a combination of ISs, prophages, GEIs, integrons, and ICEs, generating chimeric structures by illegitimate and homologous recombination. The logic to be used in their analysis is often complex, especially since certain MGEs are fast evolving and can be interdigitated in a number of subtle ways. MGE annotation best practices must commence with a bottom-up approach: first detecting the simplest and smallest elements before passing to the more complex and bigger elements (e.g., ISs → prophages → GEIs/ICEs and integrons). This maximizes annotation efficiency, and largely avoids misinterpretation of these "intertwined" MGEs and the identification of false negatives and positives. This is an important concept which must be taken into account before starting MGE analysis of a prokaryote genome. A second concept is that all predictions must be validated by manual and expert annotation (*see* **Notes 21** and **22**).

3.8.2 Non-autonomous and Partial Elements

The occurrence of non-autonomous MGEs in Prokaryotic genomes is quite common. These mainly include: (a) MITEs (Miniature Inverted-Repeat Transposable Elements), composed of the inverted repeats of an IS or Tn and some short, interstitial, non-coding DNA, capable of impacting prokaryotic genome evolution by acting as substrates for genome rearrangements and by modifying gene expression by creating new promoters; (b) partial MGEs representing relics of prophages, ISs, and GEIs, mostly generated by genome decay or other recombination events.

In spite of their abundance, the predictors and software used in this guide are not able to detect MITEs, or to classify complete and partial MGEs. This level of MGE analysis requires manual and expert annotation of the predicted results, mainly based on BLAST analysis against the MGEs databases, such as ISfinder, PHAST, ACLAME, ICEberg and INTEGRAL, and literature search (*see* **Note 23**).

4 Notes

1. Some programs can recognize files only when they have certain extensions. If you are running into file recognition errors, try changing the extension. The GenBank flatfile format (.gbff) is virtually identical to the previous GenBank format (.gb or . gbk), but most prediction software only recognizes the ".gbk" or ".gb" file extensions. In such circumstances, the ".gbff" extension should be renamed ".gbk."
2. For bacteria with multiple chromosomes, it is better to analyze each chromosome separately instead of grouping them into a multi-fasta or multi-gb file. This is due to the different ways that programs deal with such files, which might result in errors when comparing their outputs.
3. Although some software has been developed for Linux distributions and OS X, they may work on other Unix-like systems such as BSD and Solaris as well. Likewise, some software that apparently targets bacteria exclusively may also work for archaeal genomes. Try including the software in your workflow before discarding it.
4. When using the command line, press <enter> after every command, type your password when required, and type *yes* or *y* whenever required. Make sure you have memorized the password you registered when installing Ubuntu, as it will

frequently be requested after commands requiring superuser authorization (the ones preceded by *sudo*).

5. Typing errors are the major source of errors when using the command line. Getting used to using auto-completion for filenames and programs by typing their first letters and then pressing <tab> prevents several mistakes.
6. If you are not using Ubuntu or another Debian-based Linux distribution as your operating system, the apt-get package manager may not be available. Alternatively, most Python packages may be installed using the *pip* or *easy_install* managers, and Perl packages are also available from *cpan*.
7. Some programs include a compiled version of the BLAST software that may run into library issues. If you encounter this issue, remove the embedded BLAST executable and create a symbolic link to your system's executable by starting a terminal on the executable directory and typing the following command:

   ```
   rm blastn; ln -s $(which blastn) blastn
   ```
8. The general IS annotation pipelines do not consider the ISCR (Insertion Sequence with Common Regions) elements. For that level of analysis and annotation it is necessary to follow the standards proposed by Toleman and collaborators [7] and to use the ISCR Database (http://medicine.cf.ac.uk/infect-immun/research/infection/antibacterial-agents/iscr-elements/). After drawing the Gview map, investigate the IS genomic context of the gene to identify possible ISCRs.
9. Avoid using spaces in file or directory names as some commands and programs (such as Alien_Hunter) may have trouble recognizing them. Suppress spaces or replace them with underlines.
10. New ISs should be submitted to ISfinder for attribution of a standard name and inclusion in the database.
11. Since many prophages are inserted near tRNA/tmRNA genes, results from tRNA predictor software compared by BLASTn with the complexity filter disabled against the given genome should be used to improve the detection of insertion sites and to define the prophage borders. This can be done with tRNAScan-SE [106] or Aragorn [107].
12. For prophage prediction refinement, further comparisons can be made by using the PHAST (http://phast.wishartlab.com) and ACLAME (http://aclame.ulb.ac.be/) databases.
13. After compiling a program and exporting its path to the global environment, it is necessary to reload bash, the default Ubuntu terminal interpreter, so that its location can be updated. If the exported program path is no longer accessible after closing a

terminal emulator window, reload bash by typing the command *source ~/.bashrc* in a terminal or by restarting your computer.

14. If you have trouble starting Docker, first make sure that the service is still running by typing *sudo service docker start* in the terminal.
15. For GEI prediction refinement, further comparisons can be made by using the SeqWord Genome Browser (http://www.bi.up.ac.za/SeqWord/mhhapplet.php) and IslandViewer3 (http://www.pathogenomics.sfu.ca/islandviewer).
16. Artemis is not particularly well suited for visualizing analyses from programs that generate coordinates for large sections of the genome (such as Alien_Hunter and OligoWords). GView is a better tool for such cases.
17. In Artemis, it is possible to visualize concurrently the results for different feature tables. To do so, first create a directory containing the feature tables alone and concatenate them into a single file by using the following command in the terminal:

    *cat *.ft > all.ft*
18. If you are analyzing a draft genome, which is made up of a number of contigs, it is possible to visualize results for single contigs with GView if you analyze each contig separately and ignore the circular view.
19. Always compare the GView MGEs circular map results against the GIPSy predictions and the SeqWord Gene Island Sniffer circular map. If you followed the ISfinder/ISsaga gold standards for IS annotation, you can also compare these maps with the ISsaga circular map. For that go to the ISsaga webpage, and on the "Annotation" tab, click on the "ISbrowser preview."
20. Sometimes, the latest versions of some programs have not been thoroughly tested and present execution errors that the original versions described in the published papers do not. If you suspect you may be encountering such errors, look for previous versions.
21. Annotations regarding joined genes at the extremities of sequences may lead some programs to errors. Workarounds may include deleting such annotations or manually joining these genes (e.g., taking a part from the beginning of the genomic sequence to its end) and reannotating the genome.
22. Some programs rely on previous genome annotation for detecting MGE sequences. However, annotation is often not reliable for this purpose as it may overlook some elements. In addition, false positives are commonly produced by MGE analysis software. Always try generating a consensus from different software and refine the results by manual curation.

23. Last and most important: the results generated by this protocol are based on automated predictions. For high-quality results and downstream analysis, all predictions must be validated by manual, expert annotation through your preferred genome browser. Validated results can be exported to GView by editing the csv files generated by the MoGECE script. The final circular map can be exported as an svg figure for publication.

Acknowledgments

This work was supported by a project from the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES-BIGA, number 3385/2013). DOA was supported by a postdoctoral research fellowship from the São Paulo Research Foundation (FAPESP n° 2015/14600-5).

References

1. Canchaya C, Proux C, Fournous G et al (2003) Prophage genomics. Microbiol Mol Biol Rev 67:238–276
2. Canchaya C, Fournous G, Chibani-Chennoufi S et al (2003) Phage as agents of lateral gene transfer. Curr Opin Microbiol 6:417–424
3. Canchaya C, Fournous G, Brüssow H (2004) The impact of prophages on bacterial chromosomes. Mol Microbiol 53:9–18
4. Burrus V, Waldor MK (2004) Shaping bacterial genomes with integrative and conjugative elements. Res Microbiol 155:376–386
5. Frost LS, Leplae R, Summers AO et al (2005) Mobile genetic elements: the agents of open source evolution. Nature Rev Microbiol 3:722–732
6. Mazel D (2006) Integrons: agents of bacterial evolution. Nature Rev Microbiol 4:608–620
7. Toleman MA, Benett PM, Walsh TR (2006) ISCR elements: novel gene-capturing systems of the 21st century? Microbiol Mol Biol Rev 70:296–316
8. Juhas M, van der Meer JR, Gaillard M et al (2009) Genomic islands: tools of bacterial horizontal gene transfer and evolution. FEMS Microbiol Rev 33:376–393
9. Wozniak RA, Waldor MK (2010) Integrative and conjugative elements: mosaic mobile genetic elements enabling dynamic lateral gene flow. Nature Rev Microbiol 8:552–563
10. Varani AM, Monteiro-Vitorello CB, Nakaya HI et al (2013) The role of prophage in plant-pathogenic bacteria. Annu Rev Phytopathol 51:429–451
11. Fortier LC, Sekulovic O (2013) Importance of prophages to evolution and virulence of bacterial pathogens. Virulence 4:354–365
12. Siguier P, Gourbeyre E, Chandler M (2014) Bacterial insertion sequences: their genomic impact and diversity. FEMS Microbiol Rev 38:865–891
13. Siguier P, Gourbeyre E, Varani AM et al (2015) Everyman's guide to bacterial insertion sequences. In: Craig N, Chandler M, Gellert M et al (eds) Mobile DNA III. ASM Press, Washington
14. Camacho C, Coulouris G, Avagyan V et al (2009) BLAST+: architecture and applications. BMC Bioinformatics 10:421
15. Eddy SR (2011) Accelerated profile HMM searches. PLoS Comput Biol 7:e1002195
16. O'Leary NA, Wright MW, Brister JR et al (2016) Reference sequence (RefSeq) database at NCBI: current status, taxonomic expansion, and functional annotation. Nucleic Acids Res 44:D733–D745
17. Hunter S, Apweiler R, Attwood TK et al (2009) InterPro: the integrative protein signature database. Nucleic Acids Res 37: D211–D215
18. Punta M, Coggill PC, Eberhardt RY et al (2012) The Pfam protein families database. Nucleic Acids Res 40:D290–D301
19. Haft DH, Selengut JD, White O (2003) The TIGRFams database of protein families. Nucleic Acids Res 31:371–373

20. Lima T, Auchincloss AH, Coudert E et al (2009) HAMAP: a database of completely sequenced microbial proteome sets and manually curated microbial protein families in UniProtKB/Swiss-Prot. Nucleic Acids Res 37:D471–D478
21. Seemann T (2014) Prokka: rapid prokaryotic genome annotation. Bioinformatics 30:2068–2069
22. Aziz RK, Bartels D, Best AA et al (2008) The RAST server: rapid annotations using subsystems technology. BMC Genomics 9:75
23. Overbeek R, Olson R, Pusch GD et al (2014) The SEED and the rapid annotation of microbial genomes using subsystems technology (RAST). Nucleic Acids Res 42:D206–D214
24. Markowitz VM, Mavromatis K, Ivanova NN et al (2009) IMG ER: a system for microbial genome annotation expert review and curation. Bioinformatics 17:2271–2278
25. Varani AM, Siguier P, Gourbeyre E et al (2011) ISsaga is an ensemble of web-based methods for high throughput identification and semi-automatic annotation of insertion sequences in prokaryotic genomes. Genome Biol 12:R30
26. Siguier P, Perochon J, Lestrade L et al (2006) ISfinder: the reference centre for bacterial insertion sequences. Nucleic Acids Res 34: D32–D36
27. Bi D, Xu Z, Harrison EM et al (2012) ICEberg: a web-based resource for integrative and conjugative elements found in bacteria. Nucleic Acids Res 40:D621–D626
28. Moura A, Soares M, Pereira C et al (2009) INTEGRALL: a database and search engine for integrons, integrases and gene cassettes. Bioinformatics 25:1096–1098
29. Zhou Y, Liang Y, Lynch KH et al (2011) PHAST: a fast phage search tool. Nucleic Acids Res 39:W347–W352
30. Zomer A, Burghout P, Bootsma HJ et al (2012) ESSENTIALS: software for rapid analysis of high throughput transposon insertion sequencing data. PLoS One 7:e43012
31. Kichenaradja P, Siguier P, Pérochon J et al (2010) ISbrowser: an extension of ISfinder for visualizing insertion sequences in prokaryotic genomes. Nucleic Acids Res 38: D62–D68
32. Wagner A, Lewis C, Bichsel M (2007) A survey of bacterial insertion sequences using IScan. Nucleic Acids Res 35:5284–5293
33. Hawkey J, Hamidian M, Wick RR et al (2015) ISMapper: identifying transposase insertion sites in bacterial genomes from short read sequence data. BMC Genomics 16:667
34. Biswas A, Gauthier DT, Ranjan D et al (2015) ISQuest: finding insertion sequences in prokaryotic sequence fragment data. Bioinformatics 31:3406–3412
35. Robinson DG, Lee MC, Marx CJ (2012) OASIS: an automated program for global investigation of bacterial and archaeal insertion sequences. Nucleic Acids Res 40:e174
36. Chen CL, Chang YJ, Hsueh CH (2013) PRAP: an ab initio software package for automated genome-wide analysis of DNA repeats for prokaryotes. Bioinformatics 21:2683–2689
37. Bao Z, Eddy SR (2002) Automated de novo identification of repeat sequence families in sequenced genomes. Genome Res 12:1269–1276
38. Girgis HZ (2015) Red: an intelligent, rapid, accurate tool for detecting repeats de-novo on the genomic scale. BMC Bioinformatics 16:227
39. Volfovsky N, Haas BJ, Salzberg SL (2001) A clustering method for repeat analysis in DNA sequences. Genome Biol 2: research0027.1–researc0027.11
40. Smit AFA, Hubley R, Green P (2015) RepeatMasker. http://www.repeatmasker.org. Accessed 24 Mar 2016
41. Achaz G, Boyer F, Rocha EPC et al (2007) Repseek, a tool to retrieve approximate repeats from large DNA sequences. Bioinformatics 23:119–121
42. Singh V, Mishra RK (2010) RISCI–repeat induced sequence changes identifier: a comprehensive, comparative genomics-based, in silico subtractive hybridization pipeline to identify repeat induced sequence changes in closely related genomes. BMC Bioinformatics 11:609
43. Riadi G, Medina-Moenne C, Holmes DS (2012) TnpPred: a web service for the robust prediction of prokaryotic transposases. Comp Funct Genomics 2012:678761
44. Leplae R, Lima-Mendez G, Toussaint A (2010) ACLAME: a classification of mobile genetic elements, update 2010. Nucleic Acids Res 38:D57–D61
45. Fouts DE (2006) Phage_Finder: automated identification and classification of prophage regions in complete bacterial genome sequences. Nucleic Acids Res 34:5839–5851
46. Akhter S, Aziz RK, Edwards RA (2012) PhiSpy: a novel algorithm for finding prophages in bacterial genomes that combines similarity- and composition-based strategies. Nucleic Acids Res 40:e126

47. Srividhya KV, Alaguraj V, Poornima G et al (2007) Identification of prophages in bacterial genomes by dinucleotide relative abundance difference. PLoS One 11:e1193
48. Lima-Mendez G, Van Helden J, Toussaint A et al (2008) Prophinder: a computational tool for prophage prediction in prokaryotic genomes. Bioinformatics 24:863–865
49. Roux S, Enault F, Hurwitz BL et al (2015) VirSorter: mining viral signal from microbial genomic data. PeerJ 3:e985
50. Skennerton CT, Imelfort M, Tyson GW (2013) Crass: identification and reconstruction of CRISPR from unassembled metagenomic data. Nucleic Acids Res 41:e105
51. Rousseau C, Gonnet M, Le Romancer M et al (2009) CRISPI: a CRISPR interactive database. Bioinformatics 25:3317–3318
52. Grissa I, Vergnaud G, Pourcel C (2008) CRISPRcompar: a website to compare clustered regularly interspaced short palindromic repeats. Nucleic Acids Res 36:W145–W148
53. Grissa I, Vergnaud G, Pourcel C (2007) The CRISPR database and tools to display CRISPRs and to generate dictionaries of spacers and repeats. BMC Bioinformatics 8:172
54. Pinello L, Canver MC, Hoban MD et al (2015) CRISPResso: sequencing analysis toolbox for CRISPR-Cas9 genome editing. bioRxiv. https://doi.org/10.1101/031203
55. Grissa I, Vergnaud G, Pourcel C (2007) CRISPRFinder: a web tool to identify clustered regularly interspaced short palindromic repeats. Nucleic Acids Res 35:W52–W57
56. Lange SJ, Alkhnbashi OS, Rose D et al (2013) CRISPRmap: an automated classification of repeat conservation in prokaryotic adaptive immune systems. Nucleic Acids Res 41:8034–8044
57. Bland C, Ramsey TL, Sabree F et al (2007) CRISPR recognition tool (CRT): a tool for automatic detection of clustered regularly interspaced palindromic repeats. BMC Bioinformatics 8:209
58. Alkhnbashi OS, Costa F, Shah SA et al (2014) CRISPRstrand: predicting repeat orientations to determine the crRNA-encoding strand at CRISPR loci. Bioinformatics 30:i489–i496
59. Biswas A, Gagnon JN, Brouns SJJ et al (2013) Bioinformatic prediction and analysis of crRNA targets. RNA Biol 10:817–827
60. Angly F, Skennerton C (2015) MinCED. https://github.com/ctSkennerton/minced. Accessed 24 Mar 2016
61. Edgar RC (2007) PILER-CR: fast and accurate identification of CRISPR repeats. BMC Bioinformatics 8:18
62. Che D, Hasan MS, Wang H et al (2011) EGID: an ensemble algorithm for improved genomic island detection in genomic sequences. Bioinformation 7:311–314
63. Che D, Hockenbury C, Marmelstein R et al (2010) Classification of genomic islands using decision trees and their ensemble algorithms. BMC Genomics 11:S1
64. Che D, Wang H, Fazekas J et al (2014) An accurate genomic island prediction method for sequenced bacterial and archaeal genomes. J Proteomics Bioinform 7:214–221
65. Soares SC, Geyik H, Ramos RTJ et al (2015) GIPSY: genomic island prediction software. J Biotechnol 232:2–11. https://doi.org/10.1016/j.jbiotec.2015.09.008
66. Hasan MS, Liu Q, Wang H et al (2012) GIST: genomic island suite of tools for predicting genomic islands in genomic sequences. Bioinformation 8:203–205
67. Che D, Wang H (2013) GIV: a tool for genomic islands visualization. Bioinformation 9:879–882
68. Jain R, Raminemi S, Parekh N (2011) IGIP-T–integrated genomic island prediction tool. Bioinformation 7:307–310
69. Hudson CM, Lau BY, Williams KP (2015) Islander: a database of precisely mapped genomic islands in tRNA and tmRNA genes. Nucleic Acids Res 43:D48–D53
70. Baichoo S, Goodur H, Ramtohul V (2014) IslandHunter–a java-based GI detection software. PeerJ Preprints 2:e716v1
71. Hsiao W, Wan I, Jones SJ et al (2003) IslandPath: aiding detection of genomic islands in prokaryotes. Bioinformatics 19:418–420
72. Langille MGI, Hsiao WWL, Brinkman FSL (2008) Evaluation of genomic island predictors using a comparative genomics approach. BMC Bioinformatics 9:329
73. Langille MGI, Brinkman FSL (2009) IslandViewer: an integrated interface for computational indentification and visualization of genomic islands. Bioinformatics 25 (5):25664–25665
74. Ou HY, He X, Harrison EM et al (2007) MobilomeFINDER: web-based tools for *in silico* and experimental discovery of bacterial genomic islands. Nucleic Acids Res 35: W97–W104
75. Brito DM, Maracaja-Coutinho V, Farias ST et al (2016) A novel method to predict genomic islands based on mean shift clustering algorithm. PLoS One 11:e0146352

76. Reva ON, Tümmler B (2005) Differentiation of regions with atypical oligonucleotide composition in bacterial genomes. BMC Bioinformatics 6:251
77. Ganesan H, Rakitianskaia AS, Davenport CF et al (2008) The SeqWord genome browser: an online tool for the identification and visualization of atypical regions of bacterial genomes through oligonucleotide usage. BMC Bioinformatics 9:333
78. Waack S, Keller O, Asper R et al (2006) Score-based prediction of genomic islands in prokaryotic genomes using hidden Markov models. BMC Bioinformatics 7:142
79. Ou HY, Chen LL, Lonnen J et al (2006) A novel strategy for the identification of genomic islands by comparative analysis of the contents and contexts of tRNA sites in closely related bacteria. Nucleic Acids Res 34:e3
80. Vernikos GS, Parkhill J (2006) Interpolated variable order motifs for identification of horizontally acquired DNA: revisiting the *Salmonella pathogenicity* islands. Bioinformatics 22:2196–2203
81. Yoon SH, Park YK, Lee S et al (2007) Towards pathogenomics: a web-based resource for pathogenicity islands. Nucleic Acids Res 35:D395–D400
82. Soares SC, Abreu VAC, Ramos RTJ et al (2012) PIPS: pathogenicity island prediction software. PLoS One 7:e30848
83. Pundhir S, Vijayvargiya H, Kumar A (2008) PredictBias: a server for the identification of genomic and pathogenicity islands in prokaryotes. In Silico Biol 8:0019
84. Joss MJ, Koenig JE, Labbate M et al (2009) ACID: annotation of cassette and integron data. BMC Bioinformatics 10:118
85. Rajan I, Aravamuthan S, Mande SS (2007) Identification of compositionally distinct regions in genomes using the centroid method. Bioinformatics 23:2672–2677
86. Lee CC, Chen YPP, Yao TJ (2013) GI-POP: a combinational annotation and genomic island prediction pipeline for ongoing microbial genome projects. Gene 518:114–123
87. Tu Q, Ding D (2003) Detecting pathogenicity islands and anomalous gene clusters by iterative discriminant analysis. FEMS Microbiol Lett 221:269–275
88. Merkl R (2004) SIGI: score-based identification of genomic islands. BMC Bioinformatics 5:22
89. Al-Nayyef H, Guyeux C, Bahi J (2014) A pipeline for insertion sequence detection and study for bacterial genome. Lect Notes Informatics 235:85–99
90. Zhou F, Olman V, Xu Y (2008) Insertion sequences show diverse recent activities in cyanobacteria and Archaea. BMC Genomics 9:36
91. Bose M, Barber RD (2006) Prophage finder: a prophage loci prediction tool for prokaryotic genome sequences. In Silico Biol 6:0020
92. Upadhyay SK, Sharma S (2014) SSFinder: high throughput CRISPR-Cas target sites prediction tool. Biomed Res Int 2014:742482
93. Kamoun C, Payen T, Hua-Van A et al (2013) Improving prokaryotic transposable elements identification using a combination of de novo and profile HMM methods. BMC Genomics 14:700
94. Chen Y, Zhou F, Li G et al (2009) MUST: a system for identification of miniature inverted-repeat transposable elements and applications to *Anabaena variabilis* and *Haloquadratum walsbyi*. Gene 436:1–7
95. Herron PR, Hughes G, Chandra G (2004) Transposon express, a software application to report the identity of insertions obtained by comprehensive transposon mutagenesis of sequenced genomes: analysis of the preference for in vitro Tn5 transposition in to GC-rich DNA. Nucleic Acids Res 32:e113
96. Petkau A, Stuart-Edwards M, Stothard P et al (2010) Interactive microbial genome visualization with GView. Bioinformatics 26:3125–3126
97. Carver T, Harris SR, Berriman M et al (2012) Artemis: an integrated platform for visualization and analysis of high-throughput sequence-based experimental data. Bioinformatics 28:464–469
98. Bankevich A, Nurk S, Antipov D (2012) SPAdes: a new genome assembly algorithm and its applications to single-cell sequencing. J Comput Biol 19:455–477
99. Kajitani R, Toshimoto K, Noguchi H et al (2014) Efficient de novo assembly of highly heterozygous genomes from whole-genome shotgun short reads. Genome Res 24:1384–1395
100. Reddy TBK, Thomas AD, Stamatis D et al (2015) The genomes OnLine database (GOLD) v.5: a metadata management system based on a four level (meta)genome project classification. Nucleic Acids Res 43: D1099–D1106
101. Lang AS, Zhaxybayeva O, Beatty JT (2012) Gene transfer agents: phage-like elements of genetic exchange. Nat Rev Microbiol 10:472–482

102. Guy L, Nystedt B, Toft C et al (2013) A gene transfer agent and a dynamic repertoire of secretion systems hold the keys to the explosive radiation of the emerging pathogen *Bartonella*. PLoS Genet 9:e1003393

103. Horvath P, Barrangou R (2010) CRISPR/Cas, the immune system of bacteria and archaea. Science 327:167–170

104. Escudero JA, Loot C, Nivina A et al (2015) The integron: adaptation on demand. In: Craig N, Chandler M, Gellert M et al (eds) Mobile DNA III. ASM Press, Washington

105. Gillings M, Boucher Y, Labbate M et al (2008) The evolution of class 1 integrons and the rise of antibiotic resistance. J Bacteriol 190:5095–5100

106. 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

107. Laslett D, Canback B (2004) ARAGORN, a program to detect rRNA genes and tmRNA genes in nucleotide equences. Nucleic Acids Res 32:11–16