Gene calling and bacterial genome annotation with BG7

Raquel Tobes Pablo Pareja-Tobes Marina Manrique
Eduardo Pareja-Tobes Evdokim Kovach Alexey Alekhin
Eduardo Pareja

January 4, 2014

ohnosequences!

era7 bioinformatics

Abstract

New massive sequencing technologies are providing many bacterial genome sequences from diverse taxa but, a refined annotation of these genomes is crucial for obtaining scientific findings and new knowledge. Thus, bacterial genome annotation has emerged as a key point to investigate in bacteria. Any efficient tool designed specifically to annotate bacterial genomes sequenced with massively parallel technologies has to consider the specific features of bacterial genomes (absence of introns and scarcity of non protein-coding sequence) and of next generation sequencing (NGS) technologies (presence of errors and not perfectly assembled genomes).These features make it convenient to focus on coding regions and, hence, on protein sequences that are the elements directly related with biological functions.

In this chapter we describe how to annotate bacterial genomes with BG7, an open source tool based on a protein-centered gene calling / annotation paradigm. BG7 is specifically designed for the annotation of bacterial genomes sequenced with NGS. This tool is sequence error tolerant maintaining their capabilities for the annotation of highly fragmented genomes or for annotating mixed sequences coming from several genomes (as those obtained through metagenomics samples). BG7 uses cloud computing (Amazon Web Services) as its computing infrastructure and has been designed with scalability as a requirement. It is a perfect fit for big annotation projects involving hundreds and thousands of bacterial genomes.

Contents

1 Gene calling and bacterial genome annotation with BG7

1.1 i. Summary

New massive sequencing technologies are providing many bacterial genome sequences from diverse taxa but, a refined annotation of these genomes is crucial for obtaining scientific findings and new knowledge. Thus, bacterial genome annotation has emerged as a key point to investigate in bacteria. Any efficient tool designed specifically to annotate bacterial genomes sequenced with massively parallel technologies has to consider the specific features of bacterial genomes (absence of introns and scarcity of non protein-coding sequence) and of next generation sequencing (NGS) technologies (presence of errors and not perfectly assembled genomes).These features make it convenient to focus on coding regions and, hence, on protein sequences that are the elements directly related with biological functions.

In this chapter we describe how to annotate bacterial genomes with BG7, an open source tool based on a protein-centered gene calling / annotation paradigm. BG7 is specifically designed for the annotation of bacterial genomes sequenced with NGS. This tool is sequence error tolerant maintaining their capabilities for the annotation of highly fragmented

genomes or for annotating mixed sequences coming from several genomes (as those obtained through metagenomics samples). BG7 uses cloud computing (Amazon Web Services) as its computing infrastructure and has been designed with scalability as a requirement. It is a perfect fit for big annotation projects involving hundreds and thousands of bacterial genomes.

ii. Key Words bacterial genomics, genome annotation, gene calling, Next Generation Sequencing, cloud computing, Amazon Web Services.

1.2 1. Introduction

With the availability of new massive sequencing technologies, genome annotation becomes a crucial need in order to reach new findings within the amazing world of bacteria. Annotation is the basic central step in any sequence analysis pipeline, linking raw data and biological knowledge; it is the foundation on which further analysis builds upon.

Any efficient tool designed specifically to annotate bacterial genomes sequenced with massively parallel technologies has to consider the specific features, first, of bacterial genomes and, second, of next generation sequencing (NGS) technologies. The absence of introns and the scarcity of non protein-coding space in bacterial genomes are critical differences with eukaryotic ones. These features make it convenient to focus on coding regions and, hence, on protein sequences that are the elements directly related with biological functions. The extreme (even intra-species) diversity of bacteria makes also preferable to study bacterial genomes not using the "model organism" paradigm, but more flexible approaches that fit better with the plasticity, evolutionary changes and gene flux that bacterial genomes possess.

The quickly evolving massive sequencing technologies also demand flexible algorithms, able to work with different technology-dependent error patterns and highly preliminary, heavily fragmented draft genome assemblies. Another challenge to face is the annotation of contigs coming from metagenomics samples of different non-culturable bacteria.

Classical gene prediction algorithms are based on statistical features of gene and non-gene sequences and in some specific signals and patterns present in the sequences flanking genic regions [10, 2, 5]. Many of them need a previous training phase with known annotated genes, that are not always available as in the really interesting case of genomes very distant from known ones.

In contrast with methods that separate ORF prediction from their annotation [1, 8, 3, 11, 6, 12, 4, 13, 7], BG7 [9] predicts genes and infers their function mainly based on protein similarity, integrating ORF prediction and functional annotation in a single process. If the gene function is assigned based on protein sequence similarity, why not to predict genes based on this very same sequence similarity? Among other advantages, this provides more

directly traceable gene annotations, since the similarity with a specific known gene product is responsible for both the gene prediction and the function assignment. Thus, the system is more flexible, tunable and traceable and the sequence errors can be easily managed even if the pattern of errors changes with the introduction of a new sequencing technology.

In this protein-centered gene calling / annotation paradigm the set of reference proteins is the most determinant element when setting the annotation process; having an appropriate set of reference proteins is perfectly affordable for a biologist working in bacteria. BG7 predicts genes not only based on similarity but also analyzing sequence signals as stop and start codons and joining fragments of similarity that probably correspond to the same gene. Similar reference sequences compete for annotating a region of the bacterial genome and finally the best predicted and annotated genes are selected. The problem of small contigs, frequent in NGS assemblies, is also solved with BG7 that is able to detect fragmented genes or genes only partially sequenced. Given that BG7 carries out the functional annotation in the same step as the gene prediction, the system is tolerant to the lack or gain of start / stop signals and able to annotate fragments of genes.

In this chapter we describe how to annotate a bacterial genome with BG7. It is especially designed for genomes sequenced with NGS since is tolerant to sequence error maintaining their capabilities for the annotation of highly fragmented genomes or for the annotation of mixed sequences coming from several genomes (as those obtained through metagenomics samples). BG7 has been tested with data from most of the NGS technologies currently available (454, Illumina, IonTorrent and PacBio), assembled with different assembly tools, yielding to high quality annotations in all these cases. Due to how it is designed, BG7 is tolerant to the most frequent NGS errors like errors in homopolymeric regions or any other type of insertions or deletions, substitutions, poor-quality assemblies or highly fragmented genomes.

BG7 uses cloud computing (Amazon Web Services) as its computing infrastructure, and has been designed with scalability as a requirement. It is a perfect fit for big annotation projects involving hundreds and thousands of genomes: BG7 makes possible to obtain their annotations in a time independent of the number of genomes, by adjusting the number of provisioned resources accordingly.

1.2.1 1.2 BG7 algorithm

BG7 is designed from the ground up so as to deal with the special characteristics of both NGS data and bacterial genomes.

Selection of UniProt reference proteins and reference RNAs This selection has to be objective-driven. Two common objectives are the prediction of all the genes and the assignment of their function as accurately as possible; however, in many cases the annotation is essentially focused on specific types of functionalities, as could be antibiotic resistance,

enzymatic activities, metabolic pathways or plasmidic genes. This can be accomplished through the selection of reference protein sets matching those needs.

Search of similarities between contigs and reference proteins to predict and annotate the coding regions This is carried out doing a `tblastn` search of the reference proteins against the contig sequences. As a result of this BLAST search we will have lots of BLAST hits of the proteins with the contigs, some of them with possibly lots of aligned fragments (HSPs: High-scoring Segment Pairs) of the reference proteins with the contigs.

Gene prediction First we need to define a single similarity region between the protein and the contig, by merging all the coherent HSPs from a hit. Then we look upstream and downstream for start and stop signals, and define preliminary genes accordingly. These just defined genes could suffer from a series of deficiencies: non-canonical start / stop codons, intragenic stop codons and / or frameshifts. We check for all these possibilities, and mark the corresponding non-canonical genes with their deficiencies. This is one of the main reasons why this system is so robust to NGS sequencing errors since is able to tolerate all the types of indels and substitutions covering the local errors common in each sequencing technology. Non canonical stop or start codon, intragenic stops and frameshifts are indicated in the annotation of each gene.

Selecting the best gene for each contig region At this point we have lots of preliminary genes predicted for each contig region; we thus need to select the best gene for each of them, solving overlapping conflicts between predicted genes. Each gene is predicted by similarity to one protein and logically the best gene for each genome region is that with higher similarity value in the alignment of the protein and the contig region. The rest of predicted genes are marked as dismissed genes.

RNA prediction Once we have a set of well-defined protein coding genes we search for RNA genes. This is done in a very similar way, using `blastn` to face the reference RNA sequences against the contig sequences. At the final integration step, predicted RNA genes are always preferential over protein coding genes.

1.3 2. Materials

Here we describe the inputs that you need for running BG7. In the Methods section we explain in detail how you could obtain them.

1.3.1 2.1 genome sequences to be annotated

A FASTA file (see Note 1 and Note 2 for tips on how this file should be) containing a set of contigs comprising the (pan)genome you want to annotate.

1.3.2 2.2 reference proteins

A text file (see Note 1) containing a list of UniProt identifiers, one per line, (see Note 3) corresponding to the set of proteins that will be used as reference proteins for gene prediction and annotation.

Step 3 in Methods is dedicated to how you should choose your reference proteins, and how to obtain the corresponding file in this format.

1.3.3 2.3 reference RNAs

A FASTA file (see note 2) containing the sequence of RNAs that will be used as a reference RNAs. See step 4 in Methods for details about how to obtain them.

1.3.4 2.4 metadata for generating GenBank and EMBL format files

Metadata of the genome you want to annotate like the species name, the complete taxonomic lineage or a brief description of the sampling and sequenced genome/s. See step 5 in Methods.

1.3.5 2.5 AWS keys

A text file (see note 1) containing access keys (see note 5) for an **AWS** (**A**mazon **W**eb **S**ervices) account (see note 4 and note 6).

The user will need to create a new AWS account (if he doesn't have one); instructions for this are in the Methods section, step 1.

1.4 3. Methods

1.4.1 Step 0: Set up the environment

Before running the first annotation the user has to set up the environment. This should be done only once in each machine the user wants to use to run the annotations. The only

requirements for running BG7 are a Java Virtual Machine, the Scala simple build tool (sbt) and the BG7 command line interface; up-to-date instructions for their installation can be found at the BG7 website: http://bg7.ohnosequences.com

1.4.2 Step 1: create AWS credentials (if needed)

If you don't have an AWS account (see Note ?) you need to register there first; go to aws.amazon.com, click on "sign-up" and follow the instructions.

BG7 will create and manage all AWS resources automatically, but for that a set of valid keys with the right permissions are needed. The easiest and safest way to obtain them is by creating an IAM user (see note 8) through the Amazon Web Services web console (see note 7). You are given the opportunity to download the aforementioned credentials only once, just after creating the user (see note).

1.4.3 Step 2: get the sequences to be annotated

BG7 works with genome assemblies, even still in draft status, or with any type of DNA sequences in fasta format with a minimal length (over around 500 bp). In a typical bacterial genome project the user must assemble the genome before the annotation. There are many methods for obtaining genome assemblies from sequencing data (see chapter **??** of this book), but a thorough description of them would be, however, out of the scope of this chapter.

1.4.4 Step 3: select the reference proteins dataset

The user must provide a list of UniProt accession numbers (see note 3) of the proteins that wants to use as reference proteins. The set of proteins can be composed using the UniProt search tools at the UniProt website (http://www.uniprot.org). The list of UniProt IDs can be obtained from the UniProt website in an easy way: once the user has the set of proteins that wants to use as reference just click on the "Download" button on the right and then click on the "List" option to obtain a text file with the UniProt accession numbers of the reference proteins in the required text format; see note 9 for some guidelines on how you could choose this set of proteins.

It is possible to focus the annotation on a particular biological process of interest, see note 10.

Internally, BG7 will use Bio4j (see note 11) to actually retrieve the UniProt protein sequences and their associated functional information, see note 12. It is also possible to obtain reference proteins directly from Bio4j, see note 13.

1.4.5 Step 4: select the reference RNAs dataset

The reference RNAs must be provided in a FASTA file with the headers format as NCBI provides them through `.frn` files. See note 14 for a possible selection strategy.

1.4.6 Step 5: Create the annotation project

The next step is to create the annotation project. This is done very easily using the BG7 command line interface tool, just typing the following command

```
BG7 create
```

At this point the user will be asked some questions like a project name and an e-mail address for notifications. This command will create locally a folder called like the project name given by the user.

1.4.7 Step 6: Fill metadata for your annotation and set the parameters in the configuration file

The next step is to fill the metadata for your annotation and set the parameters in the configuration file.

Genome metadata is provided in the configuration file called `configuration.scala` (see note 15). Before running the annotation the user must edit the file and change the default values for these fields:

- Locus tag prefix. See note 16.
- Organism
- Complete taxonomic lineage. See note 17.
- Genome definition

Some BG7 parameters can be set in the configuration file `configuration.scala` (see note 15, like:

- The maximum distance to search for start and stop codons at the ends of the preliminary gene regions predicted by one HSP or several merged HSPs.
- The maximum length allowed for gene overlapping
- The maximum **dif-span** value allowed for merging two HPSs of the same BLAST hit. **dif-span** is the difference between the distance between two HSPs in the reference

protein and the distance of the corresponding aligned fragments in the contigs. **diff-span** is evaluated for joining different HSPs to construct a gene with coherent fragments that probably belongs to the same gene.

Setting these parameters is optional. All of them are provided with default values that have been proved to be appropriate for most scenarios.

1.4.8 Step 7: check your input data

This step is not mandatory either, but we recommend the user to check the input data. The user should check he has:

- the file with the **AWS keys** as in step 1
- the genome sequences to be annotated as detailed in step 2.
- the text file with the list of UniProt accession numbers for the **reference proteins**, from step 3
- the FASTA file with the **reference RNAs** obtained in step 4
- the `configuration.scala` file with the **metadata** and the correct values for the **parameters**: step 6

1.4.9 Step 8: launch the annotation

For launching an annotation the user has just to follow these 2 steps:

1. Publishing the annotation project
2. Running the annotation

For publishing the annotation project. See note 18.

```
bg7 publish
```

and for running the annotation

```
bg7 run
```

The user receives notifications and updates about the progress via e-mail. About the running time, see note 19. BG7 execution costs depend on the time and type of AWS resources used, see Note 20.

1.4.10 Step 9: download the output

Once the annotation is finished the user can download the output files in two different ways

- using the Amazon console (note 7): the output files are stored in a S3 bucket
- clicking on the link provided in the notification mail that is sent once the annotation is finished. See note 21

1.5 4. Notes

1.5.1 Note 1

Incorrect text file encoding can result in erroneous results and unexpected BG7 behavior. Make sure that all your text input files are in UTF-8 without BOM. If you are using Windows as your operating system, you can check this (and correct it if needed) with a good text editor such as Notepad++.

1.5.2 Note 2

FASTA files are just text files representing a set of sequences in a specific format described in http://www.ncbi.nlm.nih.gov/BLAST/blastcgihelp.shtml. This is an example of how the fasta format looks like:

```
> sequence id1231
TACGAGGTAGATGCGAGTGCGAGAGGGGGCTGAGCGAGTGCGAGTGAGC
TCGACCCGATCCCGTGAGGATGGGCGAGGAAAGTGAGAAAGCGTGTGTT
TAAACTTACGCAGAAAATTTAA
> sequence id2167
TACGAGGTAGATGCAAGAGTGCGTTAGAGGGTTCATCCTGCGAGTGAGCC
TCGACCTGCGAGAGGGGAGGATGGGCGAGGAAAGTGAGCATCCCTGTGTT
TCCGGC
```

1.5.3 Note 3

The format of the file containing the protein IDs is as follows:

```
P62552
P62554
P04737
P03012
P14565
P10026
P08716
```

It is important to note that for the reference proteins BG7 works with the so-called Uniprot primary accession number. The user should refer to the Uniprot User manual site for more information about the accession number

1.5.4 Note 4

AWS, standing for **A**mazon **W**eb **S**ervices, is the biggest de-facto standard cloud computing provider. BG7 uses the following services

- EC2 for providing the compute infrastructure
- S3 as a storage service for input and output data
- SQS for scheduling computations and in general for communication between components
- DynamoDB for managing the state of the different components

1.5.5 Note 5

Access keys are a pair of strings, the **access key ID** and **secret access key**, which are used to sign programmatic requests made to AWS. BG7 will use these keys to create a set of resources on your behalf, needed for executing the annotation process. The input file with the keys to be provided to BG7 looks like this:

```
accessKey = DKIZI23W4SKMA4C7FL4A
secretKey = Iq2F5xHV8aqTnEgS8bVcOzZSW3ZDcc3Wd1RzvlG
```

1.5.6 Note 6

It is important not to confuse `amazon.com` accounts with AWS accounts. They are different entities; if in doubt follow the instructions in step 1 and create a new AWS account.

1.5.7 Note 7

You can manage AWS services and resources through a graphical interface, the “Amazon Web Services Web Console”, available at https://aws.amazon.com/console.

1.5.8 Note 8

IAM, part of the AWS offer, is a service providing user and access control facilities to the rest of AWS services. You can access it through the web console (see note 7). When creating the user through the web console, you can grant him full administrative access if you don't want to deal with the complexity of fine-grained permissions. You can copy the user AWS credentials or download them only once, just after creating it; however, you can regenerate them as many times as needed.

1.5.9 Note 9

For a 5Mb bacterial genome we recommend using around 200,000 proteins as reference proteins. We recommend including all the proteins from close species as well as additional proteins from more distant taxa involved in processes of interest for the user, i.e. proteins involved in host interactions, in a particular metabolic pathway or plasmidic proteins.

A good strategy in some cases is to select representative proteins from UniRef100 or UniRef90. It allows covering a higher diversity of proteins and taxa maintaining a manageable number of reference proteins. The selection of UniRef100 representative proteins in the case of species with many available genomes causes a reduction in the protein number needed to cover one species of one order of magnitude, maintaining the same number of different sequences (all the proteins included in a UniRef100 cluster shared a sequence 100% identical to the representative ones). This is the case i.e. for Escherichia coli genomes. Using UniRef90 cluster representative proteins you can cover more taxa with the same number of proteins since each cluster group all the proteins with 90% of identity to the representative ones. If you want to select UniRef protein IDs the only modification that you have to do is to remove the prefix UniRef100 or UniRef90 to compose the definitive list of UniProt reference protein IDs for BG7 input.

1.5.10 Note 10

It is possible to focus the annotation on a particular biological process, pathway or any specific aspect of interest selecting the reference proteins in a proper way.

For example, if the user is especially interested in the proteins involved in antibiotic resistance but he also wants to annotate the rest of proteins of the genome, should simply **add** a set of specifically selected antibiotic resistance UniProt proteins to the set of reference proteins. Another possibility is that the user want to annotate **only** the proteins related to antibiotic resistance. In that case he should include **only** resistance related proteins in the set of reference proteins.

1.5.11 Note 11

Bio4j is a high-performance biological data platform integrating most data available in UniProt KB (SwissProt + TrEMBL), Gene Ontology (GO), UniRef (50,90,100), RefSeq, NCBI taxonomy, and Expasy Enzyme DB (Manuscript in preparation). The graph data model is directly deployable to AWS. BG7 uses Bio4j to access all data linked with proteins such as their sequence, and functional data (Gene Ontology annotations, keywords, enzymatic activity, etc).

1.5.12 Note 12

Internally BG7 uses Bio4j for accessing the proteins defined by the UniProt identifiers provided as part of the input. Those input identifiers that correspond to proteins that are not included in Bio4j will be discarded. Given that Bio4j includes all the UniProt proteins and that is updated very frequently if you obtain the list of UniProt IDs for the reference protein set from the UniProt website probably no one protein will be dismissed.

1.5.13 Note 13

It is possible to select reference proteins directly through Bio4j in a programmatic way; this involves coding, but it can be a great option when the reference sets need to be extracted using complex consults to Bio4j database. Graph databases offer new capabilities for complex querying and consulting.

1.5.14 Note 14

We recommend retrieving the FASTA files of the reference RNAs from the NCBI FTP site. The FASTA files containing the RNAs information are those with the extension `.frn`. This is the format required for the headers of the reference RNAs:

```
>ref|NC_009925|:29248-29320|Arg tRNA| [locus_tag=AM1_0026
```

It is possible to use any RNA sequences as reference if the file is in FASTA format and the header format is compatible with this NCBI format.

Normally the RNAs from one close genome are enough for a proper annotation of the main RNAs of a genome.

1.5.15 Note 15

This file is Scala code, a hybrid functional-object-oriented programming language with Java interoperability. Writing the equivalent of configuration files and parameters as code can look a bit strange at first, but it has a key set of advantages

- your configuration is thoroughly checked before launching anything, drastically reducing the amount of run-time errors. This is particularly important here, where BG7 will be creating tens of machines and millions of tasks in the course of the annotation process.
- it makes much easier to run annotations programmatically, as the configuration you need to provide can be expressed directly as code.

1.5.16 Note 16

The locus tag prefix should be a combination of letters and numbers no longer than 4 characters to be used as unique prefix to identify the contigs of the genome/s under analysis. **EC1** could be an example of a proper locus tag prefix. Each unique contig ID will be composed by this prefix and by a number. **EC1000001** would be an example of locus tag ID for a contig from a genome with a locus tag prefix **EC1**.

1.5.17 Note 17

The complete taxonomic lineage for a given organism can be obtained pretty easily at the NCBI Taxonomy website just searching for the organism in the text search field and then clicking on the corresponding entry in the results.

For example the complete taxonomic lineage of the organism *Escherichia coli* O17 str. K12a would be `cellular organisms; Bacteria; Proteobacteria; Gammaproteobacteria; Enterobacteriales; Enterobacteriaceae; Escherichia; Escherichia coli; Escherichia coli O17` obtained from its entry at the NCBI Taxonomy site.

1.5.18 Note 18

Publishing the annotation project **does not** mean that the project data is public. It just means that the code and the files are accessible to all AWS resources that would perform the annotation but it does not mean that these files are public in any way.

1.5.19 Note 19

The real BG7 running time depends on many factors but mainly on the number and type of machine/s launched; TBLASTN of the reference proteins against the contigs is usually the most time consuming process. This BLAST computational time is directly dependent on the reference protein number and on the similarity of the proteins with the genome sequences. The total size of the genome sequences to be annotated also contributes to the computational cost, but, at a minor level since the total size of reference sequences usually is much bigger than the total contig size. It is thus impossible to give a precise estimate for the running time of one BG7 annotation; experience shows though that a normal project is finished in less than one hour. You can find some estimates of running time in specific conditions in the BG7 website BG7

1.5.20 Note 20

Each BG7 execution incurs in some costs due to the use of AWS resources. Before launching your first BG7 annotation you need to consult the prices of each type of machine at the AWS site to design your project. Some figures about BG7 execution costs for specific genome annotation examples will be available through the BG7 website BG7.

1.5.21 Note 21

Once the annotation is finished the user receives a notification by e-mail with a temporary link to download the output files. It is important to note that this is a temporary link that will be accessible for a short period of time.

1.5.22 5.1 Tables

```
era7p/bg7-redux/TABLES_input_and_output_files_for_BG7.docx
```

References

[1] Aziz, R. K., Bartels, D., Best, A. A., DeJongh, M., Disz, T., Edwards, R. A., Formsma, K., Gerdes, S., Glass, E. M., Kubal, M., Meyer, F., Olsen, G. J., Olson, R., Osterman, A. L., Overbeek, R. A., McNeil, L. K., Paarmann, D., Paczian, T., Parrello, B., Pusch, G. D., Reich, C., Stevens, R., Vassieva, O., Vonstein, V., Wilke, A. and Zagnitko, O. [2008], 'The RAST Server: rapid annotations using subsystems technology.', *BMC genomics* **9**(1), 75.

URL: http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=2265698&tool=pmcentrez&rendertype=abstract

[2] Besemer, J., Lomsadze, A. and Borodovsky, M. [2001], 'GeneMarkS: a self-training method for prediction of gene starts in microbial genomes. Implications for finding sequence motifs in regulatory regions.', *Nucleic acids research* **29**(12), 2607–18.
URL: http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=55746&tool=pmcentrez&rendertype=abstract

[3] Borodovsky, M., Mills, R., Besemer, J. and Lomsadze, A. [2003], 'Prokaryotic gene prediction using GeneMark and GeneMark.hmm.', *Current protocols in bioinformatics / editoral board, Andreas D. Baxevanis ... [et al.]* **Chapter 4**(7), Unit4.5.
URL: http://www.ncbi.nlm.nih.gov/pubmed/18428700

[4] Hemmerich, C., Buechlein, A., Podicheti, R., Revanna, K. V. and Dong, Q. [2010], 'An Ergatis-based prokaryotic genome annotation web server.', *Bioinformatics (Oxford, England)* **26**(8), 1122–4.
URL: http://www.ncbi.nlm.nih.gov/pubmed/20194626

[5] Hyatt, D., Chen, G.-L., Locascio, P. F., Land, M. L., Larimer, F. W. and Hauser, L. J. [2010], 'Prodigal: prokaryotic gene recognition and translation initiation site identification.', *BMC bioinformatics* **11**, 119.
URL: http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=2848648&tool=pmcentrez&rendertype=abstract

[6] Kumar, K., Desai, V., Cheng, L., Khitrov, M., Grover, D., Satya, R. V., Yu, C., Zavaljevski, N. and Reifman, J. [2011], 'AGeS: a software system for microbial genome sequence annotation.', *PloS one* **6**(3), e17469.
URL: http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=3049762&tool=pmcentrez&rendertype=abstract

[7] Lee, D., Seo, H., Park, C. and Park, K. [2009], 'WeGAS: a web-based microbial genome annotation system.', *Bioscience, biotechnology, and biochemistry* **73**(1), 213–6.
URL: http://www.ncbi.nlm.nih.gov/pubmed/19129632

[8] Mavromatis, K., Ivanova, N. N., Chen, I.-M. A., Szeto, E., Markowitz, V. M. and Kyrpides, N. C. [2009], 'The DOE-JGI Standard Operating Procedure for the Annotations of Microbial Genomes.', *Standards in genomic sciences* **1**(1), 63–7.
URL: http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=3035208&tool=pmcentrez&rendertype=abstract

[9] Pareja-Tobes, P., Manrique, M., Pareja-Tobes, E., Pareja, E. and Tobes, R. [2012], 'BG7: A New Approach for Bacterial Genome Annotation Designed for Next Generation Sequencing Data', *PLoS ONE* **7**(11), e49239.
URL: http://dx.plos.org/10.1371/journal.pone.0049239

[10] Salzberg, S. L., Delcher, A. L., Kasif, S. and White, O. [1998], 'Microbial gene identifica-

tion using interpolated Markov models.', *Nucleic acids research* **26**(2), 544–8.
URL: http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=147303&tool=pmcentrez&rendertype=abstract

[11] Stewart, A. C., Osborne, B. and Read, T. D. [2009], 'DIYA: a bacterial annotation pipeline for any genomics lab.', *Bioinformatics (Oxford, England)* **25**(7), 962–3.
URL: http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=2660880&tool=pmcentrez&rendertype=abstract

[12] Tanenbaum, D. M., Goll, J., Murphy, S., Kumar, P., Zafar, N., Thiagarajan, M., Madupu, R., Davidsen, T., Kagan, L., Kravitz, S., Rusch, D. B. and Yooseph, S. [2010], 'The JCVI standard operating procedure for annotating prokaryotic metagenomic shotgun sequencing data.', *Standards in genomic sciences* **2**(2), 229–37.
URL: http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=3035284&tool=pmcentrez&rendertype=abstract

[13] Van Domselaar, G. H., Stothard, P., Shrivastava, S., Cruz, J. A., Guo, A., Dong, X., Lu, P., Szafron, D., Greiner, R. and Wishart, D. S. [2005], 'BASys: a web server for automated bacterial genome annotation.', *Nucleic acids research* **33**(Web Server issue), W455–9.
URL: http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=1160269&tool=pmcentrez&rendertype=abstract