[v1.2]

SIMBA docs

LGCM

Federal University of Minas Gerais

2015

Summary

1. Introduction

SIMBA, **SImple Manager for Bacterial Assemblies**, is a Web interface for managing assembly projects of bacterial genomes. SIMBA was created to assist bioinformaticians to assemble bacterial genomes sequenced with Next-Generation Sequencing (NGS) platforms quickly, easily and effectively. SIMBA also is open source tool, *i.e.*, can be freely downloaded, shared and modified.

Figure 1. SIMBA logo. SIMBA visual identity and interface were developed to give users the better experience in genome assembly. SIMBA is awesome!

1.1 Why use SIMBA?

SIMBA allows bioinformaticians do not worry so much about techniques with repetitive activities, and focus on the most important: understand and resolve biological questions!

Assemble genomes requires processes with a high degree of complexity that adopt heuristics to attempt to obtain results closer to the biological reality. Thus, several software is required for complete assembly processes. In this context, to realize assemblies are required, in addition to knowledge of the biological part, a large knowledge of operating system and the specific operation of each software. Which leads to a slow learning curve!

In addition, many repetitive processes could be reduced with the adoption of automation scripts and tools organized in a simple pipeline, which controlled by graphical interface, can accelerate the process of data processing, assembly and curation – this is the contribution of SIMBA.

1.2 How SIMBA works?

SIMBA runs on the Web! Can be run through any browser on any operating system. SIMBA runs even cellular phones.

SIMBA uses wrapper! SIMBA integrates multiple tools into a single interface that can be accessed through a browser. SIMBA modules can provide:

- One file with raw data for each project.
- One project can have several attempts assemblies.
- One assembly has 5 steps of curation (v1.2).
- Client/Server: despite SIMBA can be accessed for any device with a browser and internet access, it needs a specific structure that must be set only once!

1.3 Where download SIMBA?

SIMBA can be downloaded at:

- http://ufmg-simba.sourceforge.net/
- http://github.com/dcbmariano/simba

2. Installing SIMBA

The basic requirements for SIMBA installation are:

(i) Linux Operational System 64 bit (we recommend Ubuntu 14.04 or CentOS 6.4);
(ii) Apache Server;
(iii) PHP 5.3 or superior with libraries Mcrypt and PHP-SQLite;
(iv) Python with library Biopython;
(v) NCBI-BLAST+.

2.1 Installing basic requirements in O. S. Ubuntu 14.04

Apache server Web server required to manage SIMBA pages which will be accessed by a browser. In the Linux terminal type the following command line:

```
sudo apt-get install apache2
```

SIMBA also requires Apache mod_rewrite capability. It can be enabled by editing the Apache list of mods-enabled.

```
sudo gedit /etc/apache2/sites-available/default
```

Edit the file based on the information provided in Table 1.

Table 1. Configuration to enable mod-rewrite in Apache Server.

Where you see:	Change to:
Options Indexes FollowSymLinks MultiViews AllowOverride None Order allow,deny Allow from all	Options Indexes FollowSymLinks MultiViews AllowOverride All Order allow,deny Allow from all

Now, run the command below and restart the Apache server.

```
sudo a2enmod rewrite

sudo service apache2 restart
```

2.1.1 PHP5 and libraries

These are required to interpret the source code of SIMBA back-end. Mcrypt is a security library used to encrypt data, and SQLite is the database management system (DBMS) used by SIMBA.

```
sudo apt-get install php5 php5-mcrypt php5-sqlite
```

2.1.2 Python and Biopython library

These are required to run sequence analysis by SIMBA. Python is installed by default in almost all versions of Linux. To install the Biopython library it is necessary to download the installation package at http://biopython.org/wiki/Download. Uncompressing the Biopython package, open the folder with Biopython in the Linux terminal and type the following on the command line to build and install:

```
python setup.py build

python setup.py test

sudo python setup.py install
```

2.1.3 NCBI-BLAST+

This is required to run local alignments between sequences using BLAST. In the Linux terminal, type the following command:

```
sudo apt-get install ncbi-blast+
```

2.2 Installing steps

To install SIMBA, first download the source code at http://ufmg-simba.sourceforge.net. Extract the file downloaded in the directory /var/www. Give permission to apache user through the command lines:

```
chown –R www-data:www-data /var/www/simba

chmod –R 755 /var/www/simba
```

2.2.1 Configuring SIMBA

SIMBA requires two simple configurations:

(i) the URL of your application;
(ii) a security random key of 32 bit.

Open the file “simba/app/config/app.php” with a text editor. Edit the lines 29 (if you don’t have a personal URL use http://localhost/simba/public for only local access) and 68 (type 32 random characters).

2.2.2 Accessing SIMBA by a browser

Now, SIMBA can be accessed by a browser, as Google Chrome, Firefox, Opera or Safari, using the address configured in the “app.php” file. We point out that SIMBA can run on Internet Explorer, however some pages can present layout problems. We recommend the Google Chrome browser.

3. SIMBA interface

3.1 Understanding genome assembly process

Before explaining SIMBA process it is necessary understanding the problem of genome assembly.

Current, most part DNA sequencing platforms are able to read only small DNA fragments. Thus, after the sequencing process of several DNA molecules of one same organism is important reconstruct the original genome sorting the fragment reads. This process is known as genome assembly (Figure 2).

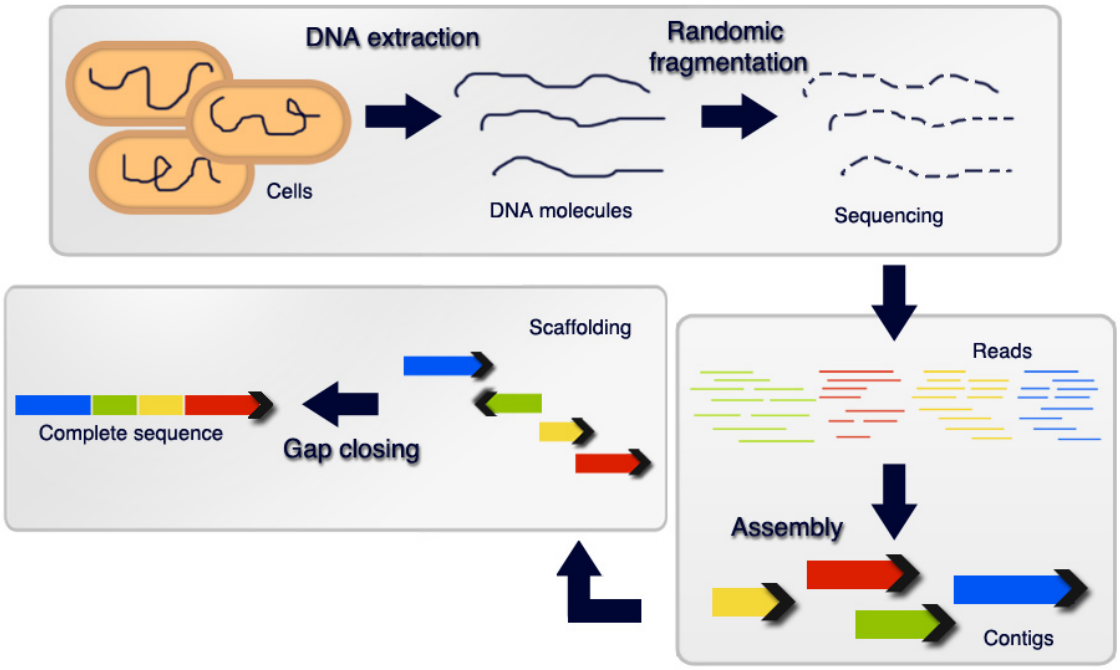


Figure 2. Sequence process. *Source*: (adapted) HUSEMANN, P. Bioinformatic Approaches for Genome Finishing, 2011.

It is possible sort the reads using an organism phylogenetically closer as reference. However, in most part of times do not have a reference to help in the assembly. Thus, it is necessary join the reads by the overlap among its. Sort reads without a reference is called *de novo* assembly or *ab initio* assembly.

Due to repetitive regions and the low coverage regions, the assembly software are not able to reconstruct the all genome. This produce gaps among the maximum continuous sequences (contigs).

After assembly, it is necessary sort contigs (scaffolding contigs). For scaffolding we can use a genome of an organism that is phylogenetically closer as reference, paired reads or physical maps. In the end, it is necessary close the

gaps among scaffolds (contigs sorted). We define as curation, the process of sort contigs and close gaps among them.

3.2 SIMBA Workflow

SIMBA was divided in three modules: (i) projects: allow projects management and data format conversions; (ii) assemblies: allow *de novo* assemblies with several software; (iii) curation: provide five steps of curation.

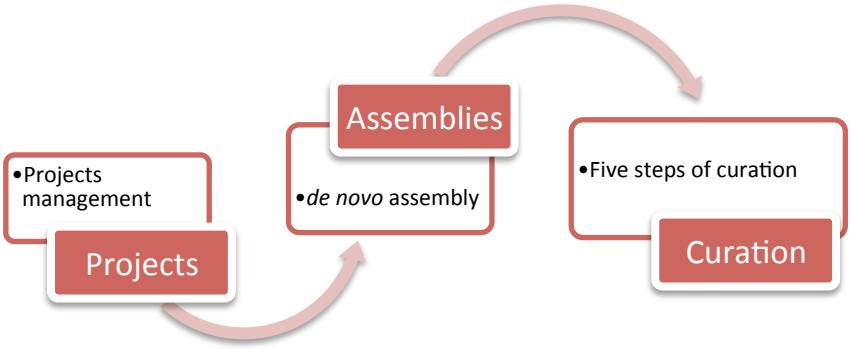


Figure 3. Modules of SIMBA.

SIMBA allows by default assemblies with Mira version 3.9.18 (default) and Mira version 4.0.2[1]. SIMBA also provide support to assemblers: (i) Newbler[2]; (ii) Minia[3]; and (iii) SPAdes[4].

The five steps of curation: (i) SIMBA allows scaffolding of contigs by reference using a modified version of CONTIGuator software (requires a genome phylogenetically closer) and optical mapping reports generated by MapSolver software (requires data from restriction maps that can be obtained using Whole Genome Mapping[5]); (ii) in circular genomes, SIMBA allows the correction of the beginning of the strand using the gene dnaA; (iii) detection of overlaps among extremities of contigs using BLAST; (iv) closing of repetitive regions using maprepeat; (v) analysis of gaps remaining.

A complete workflow of SIMBA can be seen in the Figure 4.

[1] http://mira-assembler.sourceforge.net/docs/DefinitiveGuideToMIRA.html
[2] http://www.454.com/products/analysis-software/
[3] http://minia.genouest.org/
[4] http://bioinf.spbau.ru/spades
[5] For more information visit: <http://opgen.com/>.

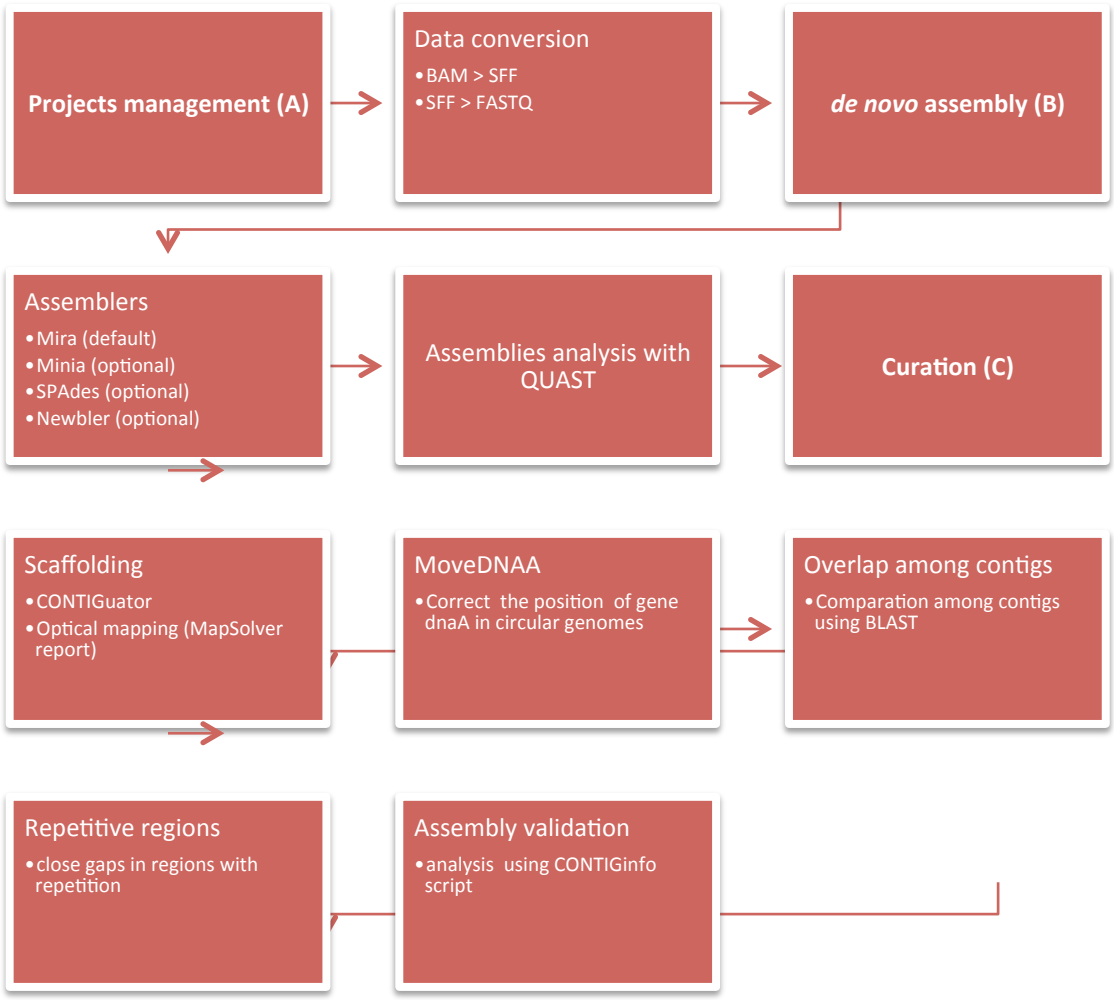


Figure 4. SIMBA workflow.

3.3 General vision of SIMBA interface

SIMBA interface is composed by toolbar, main area and footer (Figure 5). The toolbar provide access to the home page (projects page), the documentation, external tools that can be executed by SIMBA (such as CONTIGuator), and the window that shows the version of SIMBA. It also shows the user when logged and provide access to the control panel.

In the “main area” will be load the modules of SIMBA.

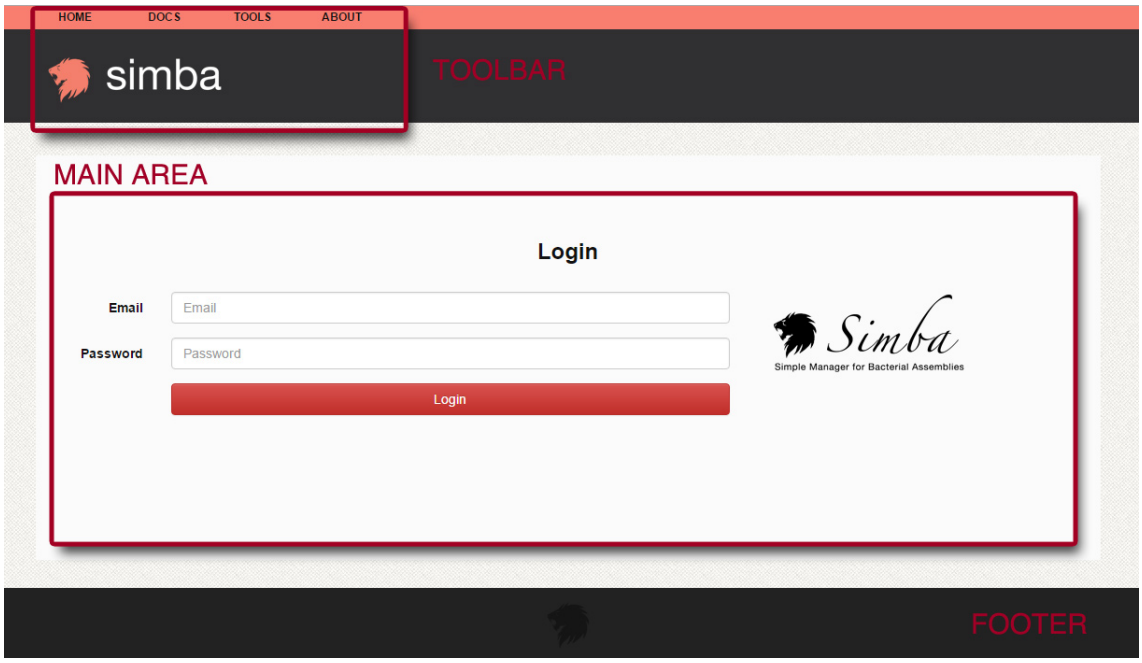


Figure 5. SIMBA interface.

SIMBA by default uses the user and password:

User: admin **Password**: admin

You can change these values in the user control panel.

3.3.1 Creating users

Click on user name on the toolbar to open the control panel. Just the user admin can create new users. Click on the pencil symbol to change the password of the admin. Click on new user to create new users (Figure 6).

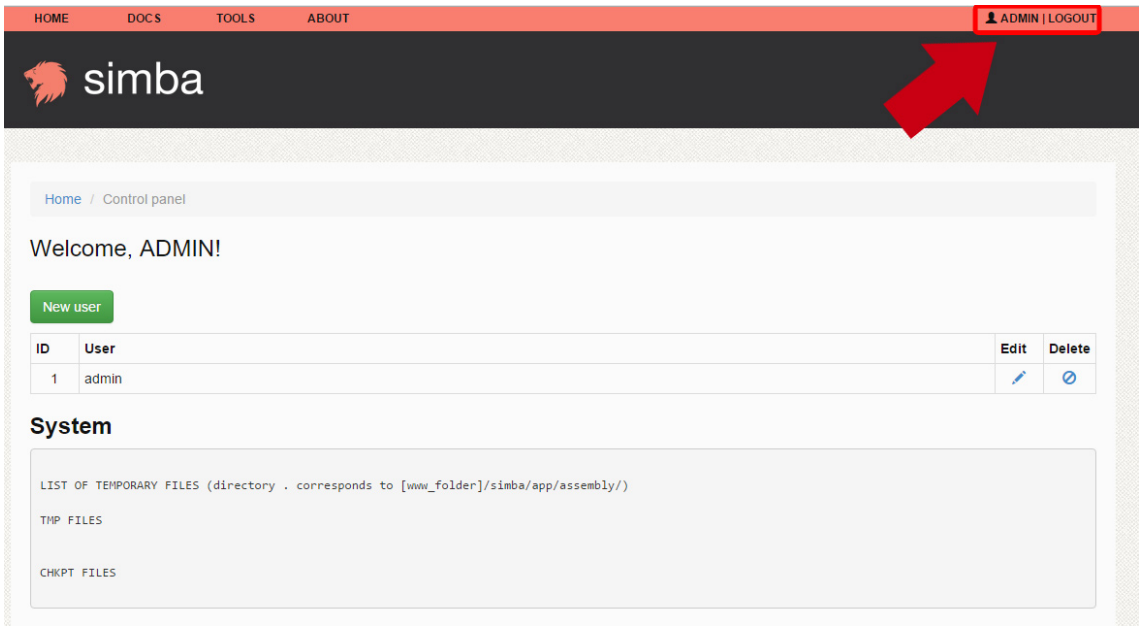


Figure 6. Creating new users.

3.4 Module projects

The module projects provide methods to managing sequencing projects (Figure 7).

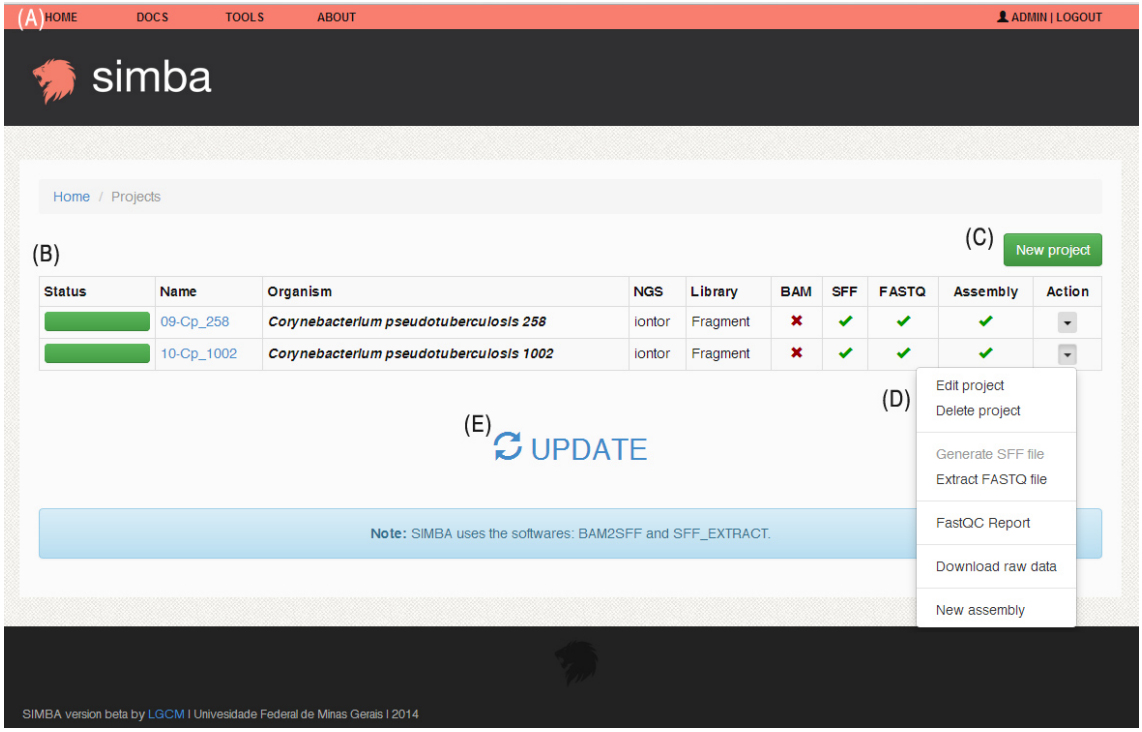


Figure 7. module projects. (A) toolbar. (B) table with genomes projects. Shows: status of the project, project name, organism name, NGS, library, format of raw data, and assemblies realized (C) Provide the creation of new projects. (D) Provide the execution of actions, such as

run new assemblies, generate data conversions or sequence analysis. (E) Update info. *Source*: Mariano (2015) – Master thesis “SIMBA: uma ferramenta Web para gerenciamento de montagens de genomas bacterianos”.

3.4.1 Creating new projects

To create new projects we recommend the creation of a folder in the directory “app/assembly” of SIMBA folder and click in “update”. SIMBA automatically will detect a new project and create a new item in the table (click in action and edit to alter the information in the row). However, you can click in the button “new project” and send the raw data to SIMBA interface (Figure 8).

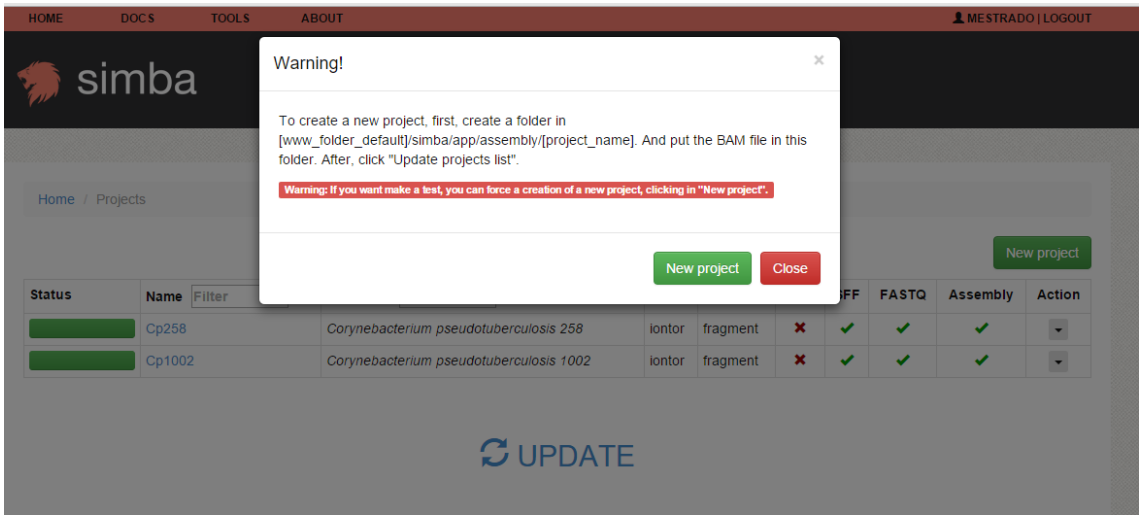


Figure 8. Creating new projects by the SIMBA interface.

Now, insert the project name, organism name, the NGS used in sequencing (SIMBA present support for Ion Torrent, however you can test SIMBA with other sequencing platforms), the library (optional) and input the raw data file (format SFF, FASTQ or BAM) (Figure 9).

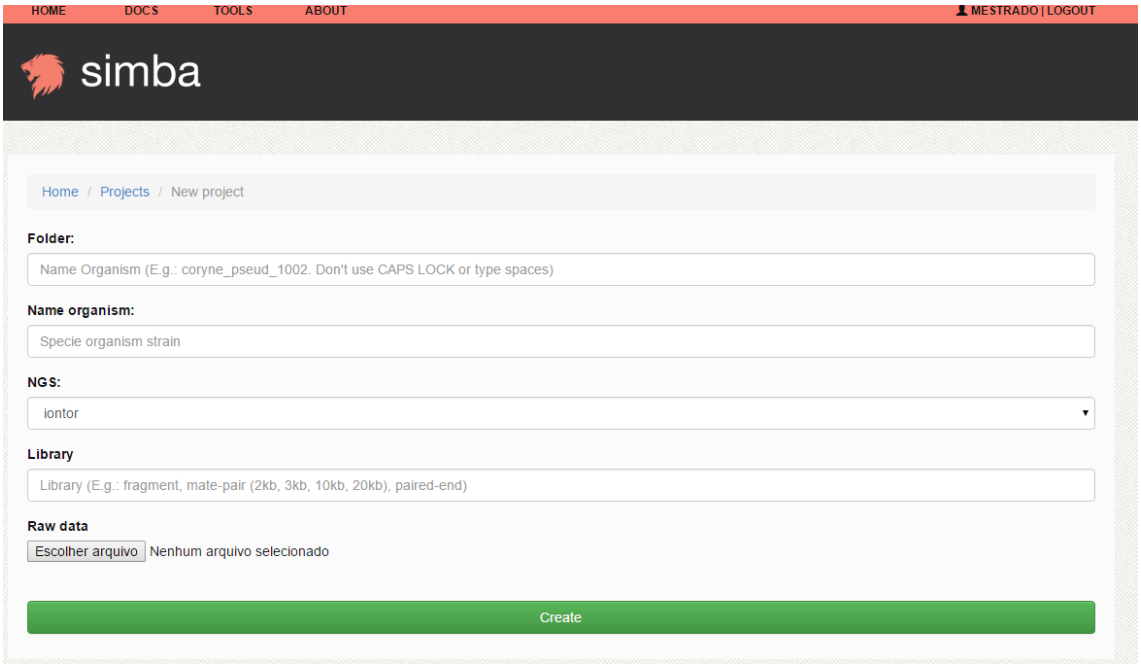


Figure 9. Creating a new project.

The new project will be listed on the projects page.

SIMBA uses FastQC[6] to generate reports of quality reads. For this, click on "action > FastQC report". SIMBA also provide format conversions, such as BAM > SFF or SFF > FASTQ (Figure 7D).

3.5 Module assemblies

Click in the link at the project name or click in "action > new assembly" to open the page of module assemblies.

The module assemblies shows all attempts of genome assembly. The module shows the version of the trial, the number of contigs obtained in the assembly, the predicted length of the genome, the length of the smaller and bigger contigs, N50 value, assembly info, parameters used in the assembly and an action button the allows download of raw data and open the curation module for a specific assembly (Figure 10). The button "update" allows the updating of assembly information in the table.

[6] http://www.bioinformatics.babraham.ac.uk/projects/fastqc/

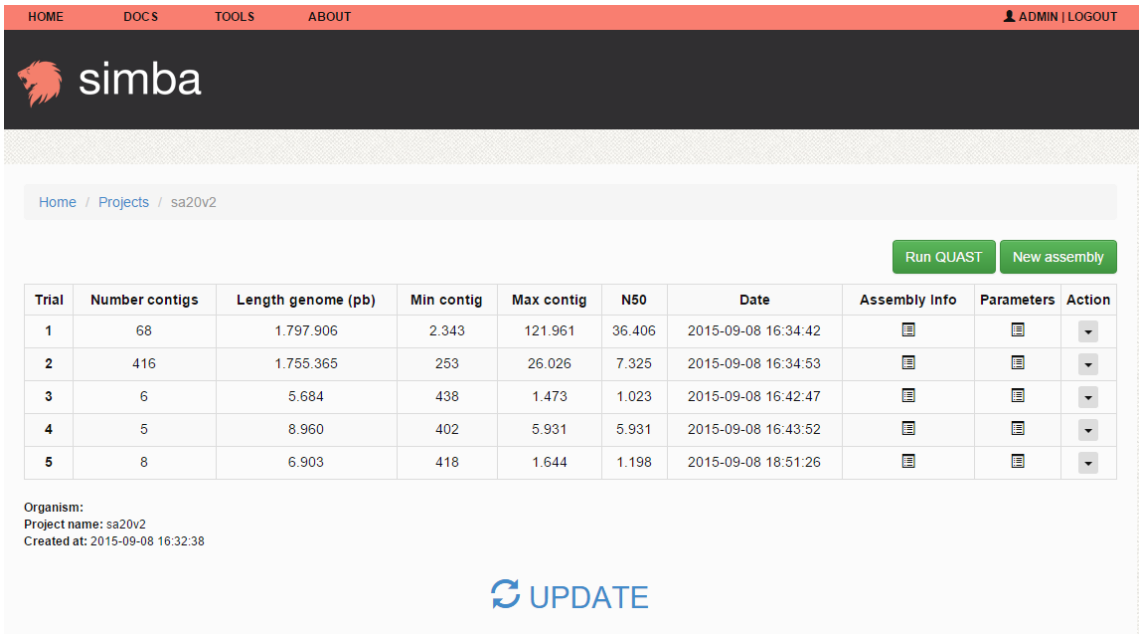


Figure 10. Module assemblies.

3.5.1 Running a new assembly

Click on the button “new assembly” to run a new assembly.

SIMBA provide default parameters for assembly (Figure 11). By default SIMBA uses Mira 3.9.

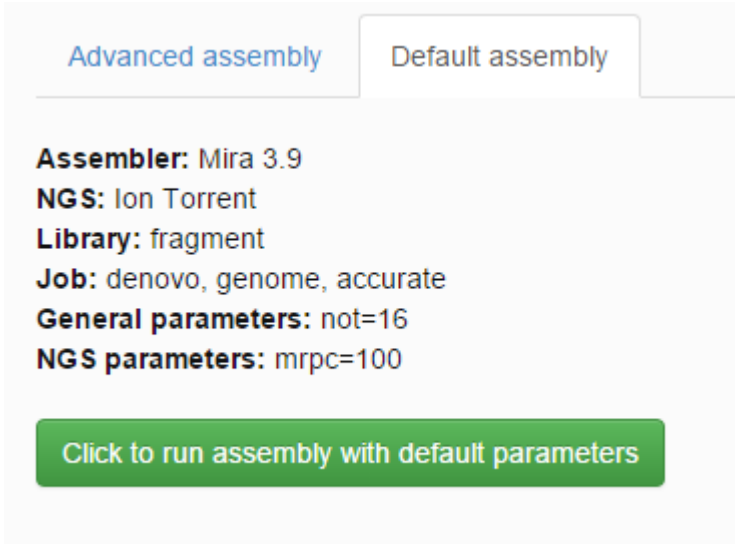


Figure 11. Default assembly. Click on the green button to run a new assembly with default parameters.

SIMBA also provide the use of other assemblers in the “advanced assembly” mode (Figure 12).

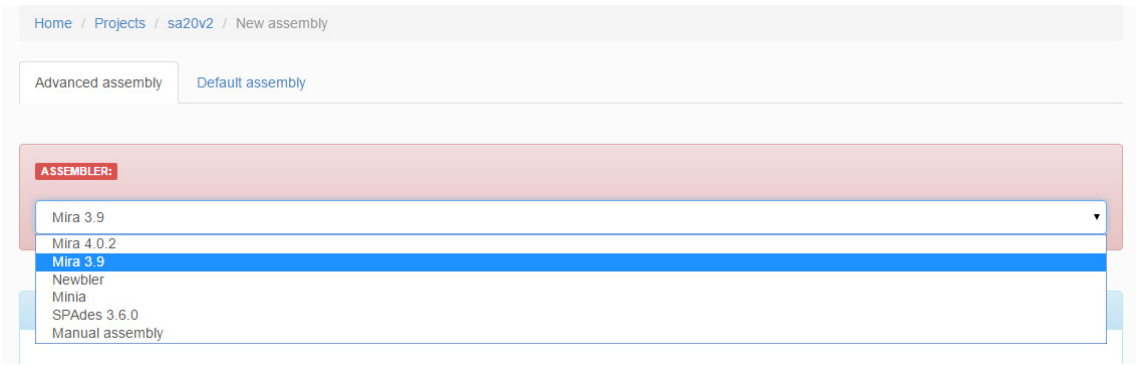


Figure 12. Advanced assembly.

WARNING: SIMBA was tested for Ion Torrent data. SIMBA allows experimentally the use of other type of sequencing raw data. However, we cannot confirm the SIMBA efficacy for different NGS data.

You can download the contigs obtained in an assembly clicking on “action > download contigs” (Figure 13).

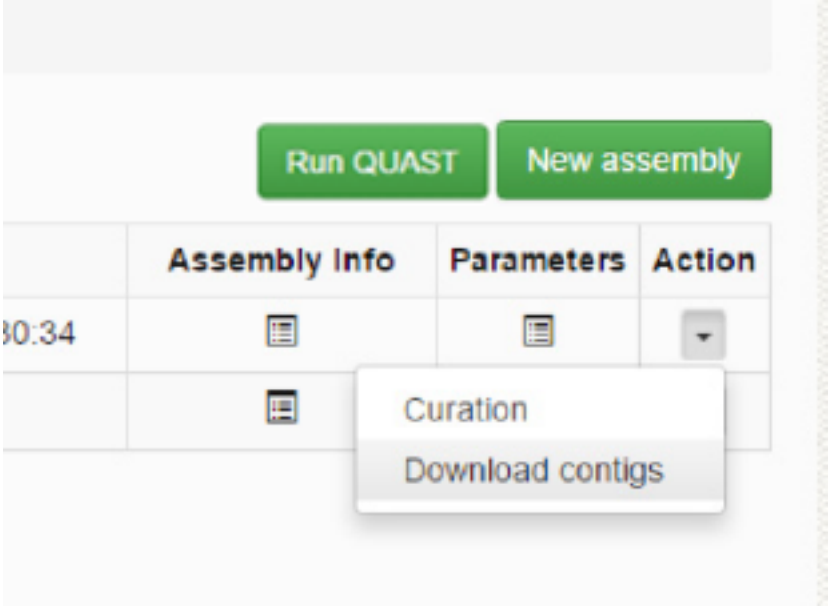


Figure 13. Module assembly: downloading contigs.

3.5.2 Validating assemblies

SIMBA uses QUAST[7] to validate assemblies (inserted in the version 1.2). To run QUAST, click on the button “Run QUAST”. The button just appears when executed at minimum one assembly attempt. QUAST will generate a quality report (Figure 14).

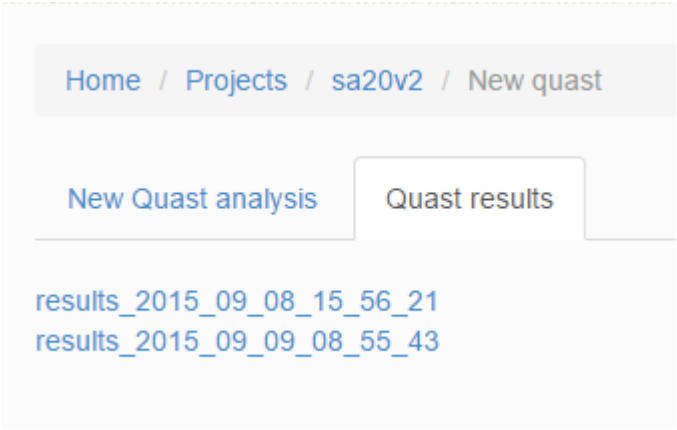


Figure 14. List of QUAST reports.

SIMBA can perform several QUAST analyses, and each can be analyzed individually (Figure 15).

QUAST report

09 September 2015, Wednesday, 08:56:05

All statistics are based on contigs of size >= 500 bp, unless otherwise noted (e.g., "# contigs (>= 0 bp)" and "Total length (>= 0 bp)" include all contigs.)

Reference size: 1 841 952 bp, G+C content: 35.48 %
1872 genes

Worst Median Best ☑ Show heatmap

Statistics without reference	t4_out.unpadded	t1_out.unpadded	t5_out.unpadded	t2_out.unpadded	t3_out.unpadded
# contigs	2	68	4	397	5
Largest contig	5931	121 961	1644	26 026	1473
Total length	7608	1 797 906	5175	1 747 027	5246
N50	5931	36 406	1481	7343	1023
Misassemblies					
# misassemblies	2	0	1	2	1
Misassembled contigs length	7608	0	1644	4414	1473
Mismatches					
# mismatches per 100 kbp	0	0.33	58	26.43	57.3
# indels per 100 kbp	0	0.33	0	5.46	0
# N's per 100 kbp	0	0	0	0	0
Genome statistics					
Genome fraction (%)	0.413	97.546	0.281	94.496	0.284
Duplication ratio	1	1.001	1.001	1.001	1.002
# genes	4 + 4 part	1751 + 68 part	1 + 3 part	1451 + 367 part	1 + 4 part
NGA50	-	36 205	-	6731	-

Extended report

Figure 15. QUAST report.

[7] http://bioinf.spbau.ru/quast

3.6 Module curation

The module curation provides strategies to try finish the genome assembly (Figure 16).

Module Curation requirement: each step need to be executed to run the next.

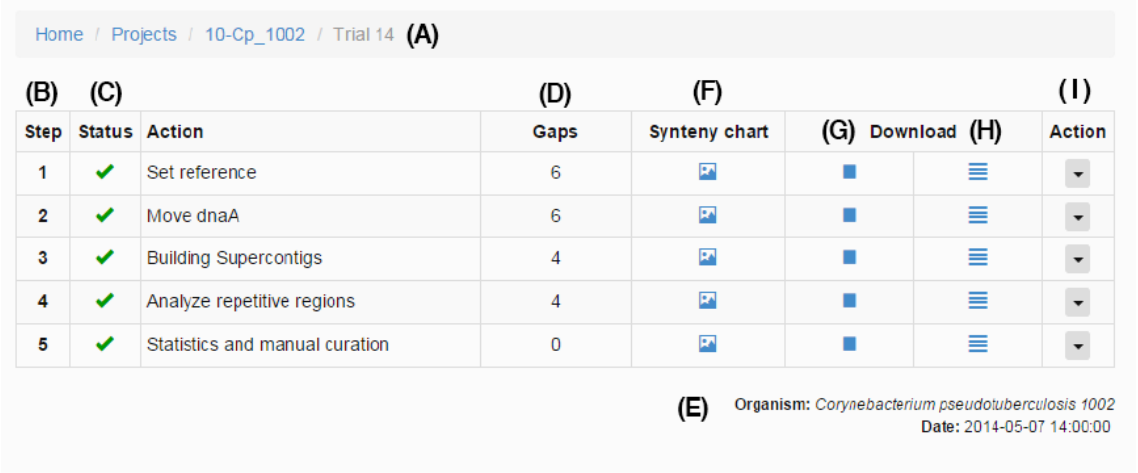
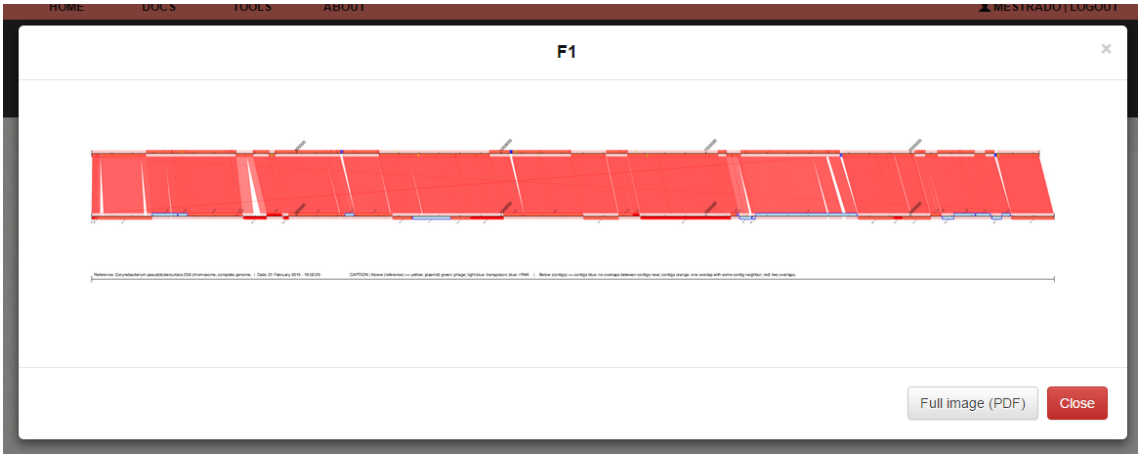


Figure 16. Module curation. (A) Inner toolbar. (B) Curation step. (C) Status of the step. (D) Number of gaps remaining after the step. (E) Organism information. (F) Synteny chart generated by CONTIGuator. (G) Download scaffolds (separated by "Ns"). (H) Download contigs. (I) Action button. *Source*: Mariano (2015) – Master thesis "SIMBA: uma ferramenta Web para gerenciamento de montagens de genomas bacterianos".

After each step, SIMBA performs a comparison using a modified version of the software CONTIGuator[8]. CONTIGuator generate a synteny graph (Figure 17), that helps users to detect assembly errors.



[8] http://contiguator.sourceforge.net/

Figure 17. Synteny graph using CONTIGuator.

In the step 1, SIMBA performs the process of scaffolding. The scaffolding by reference (Figure 18), SIMBA requires links for two files: a GenBank file (gbk) and a Genome Complete file (fna). These links can be obtained at: < ftp.ncbi.nih.gov/genomes/>.

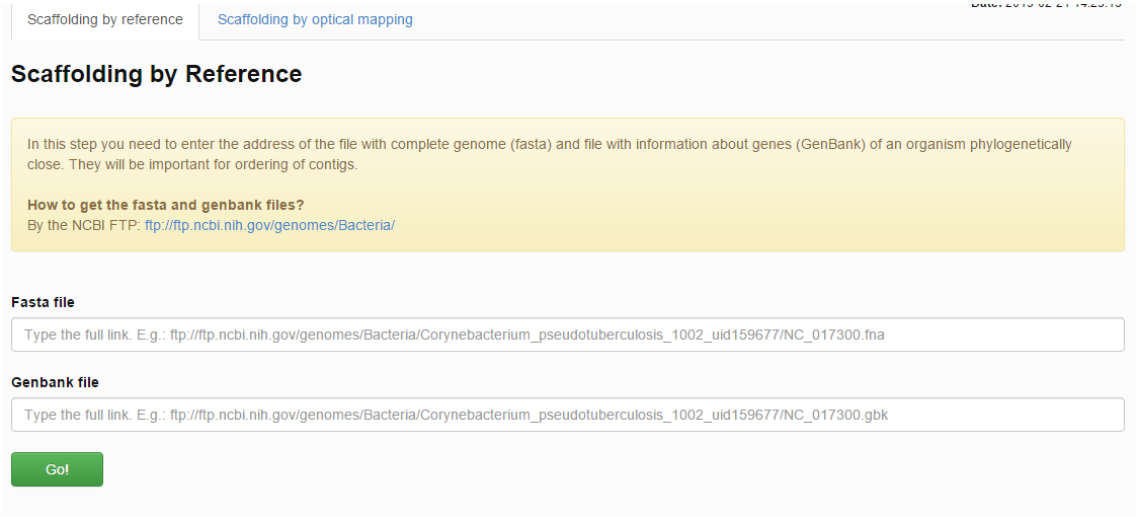


Figure 18. Step 1 - scaffolding using reference.

SIMBA also provide methods to scaffolding using Whole Genome Mapping (optical mapping). Click on the link “scaffolding by optical mapping” (Figure 19) to run this type of scaffolding. In the text area, insert the report generate by the MapSolver software.

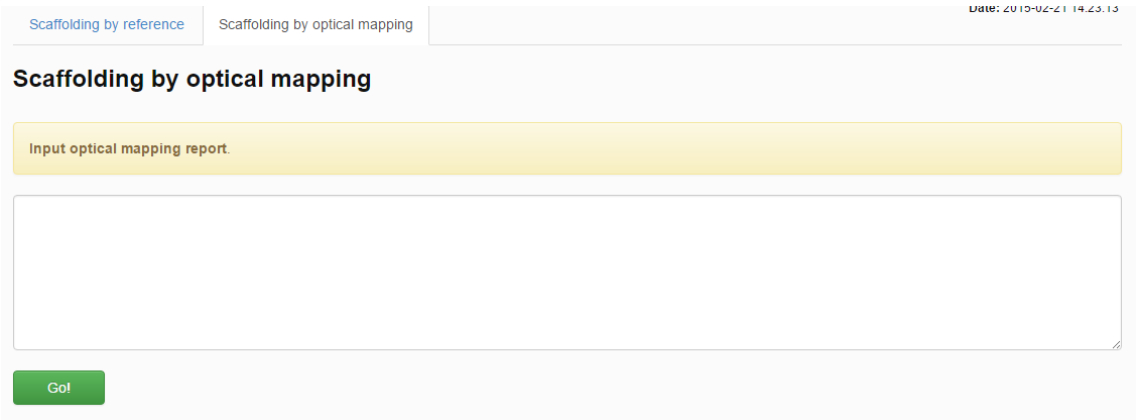


Figure 19. Step 1 - scaffolding using optical mapping.

Whole genome mapping (or optical mapping) depends of experimental data. MapSolver is proprietary software. Contact <http://opgen.com> for more information.

The step 2, can be executed clicking on “Action > RUN STEP 2”. In the next page, SIMBA shows two options: (i) run moveDNAA.py and CONTIGuator – if you have a reference; or (ii) SKIP – for optical mapping scaffolding.

Figure 20. Move dnaA.

In the step 3, SIMBA performs BLAST comparison of 3.000 bp in the extremities of contigs and allows that users merge contigs homologues extremities (Figure 21). When two contigs are joined, it receives the name of Supercontig.

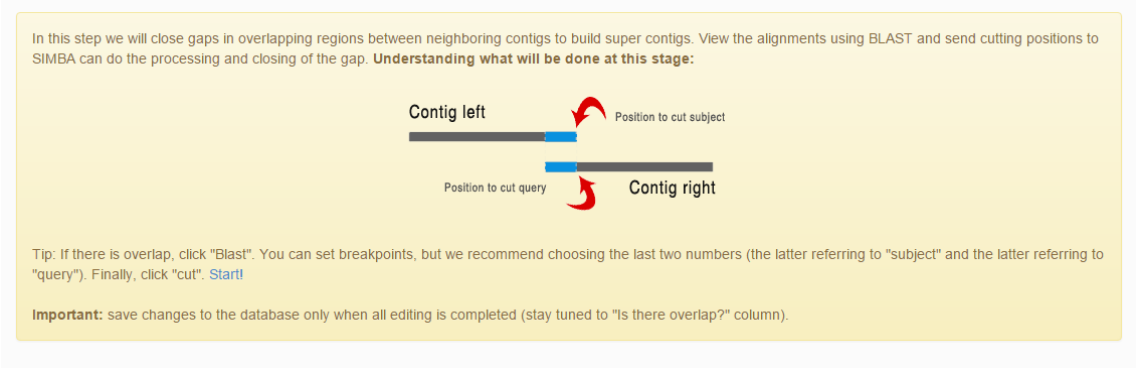
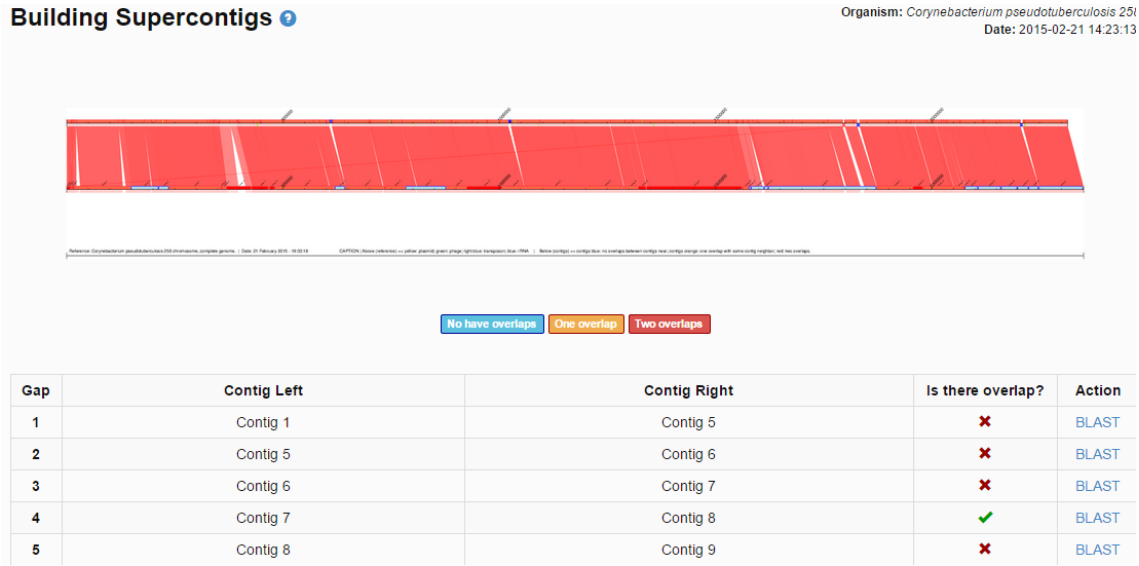


Figure 21. Move dnaA.

SIMBA also shows a synteny graph and a comparative table (Figure 22). You can skip this step clicking on the button “skip”.

Warning: only click on “save updates in database” after close all gaps

possible.



Gap	Contig Left	Contig Right	Is there overlap?	Action
1	Contig 1	Contig 5	✖	BLAST
2	Contig 5	Contig 6	✖	BLAST
3	Contig 6	Contig 7	✖	BLAST
4	Contig 7	Contig 8	✔	BLAST
5	Contig 8	Contig 9	✖	BLAST

Figure 22. Construct Supercontigs.

Clicking in the button “BLAST”, SIMBA shows the results of the BLAST comparison among two extremities of contigs (Figure 23).

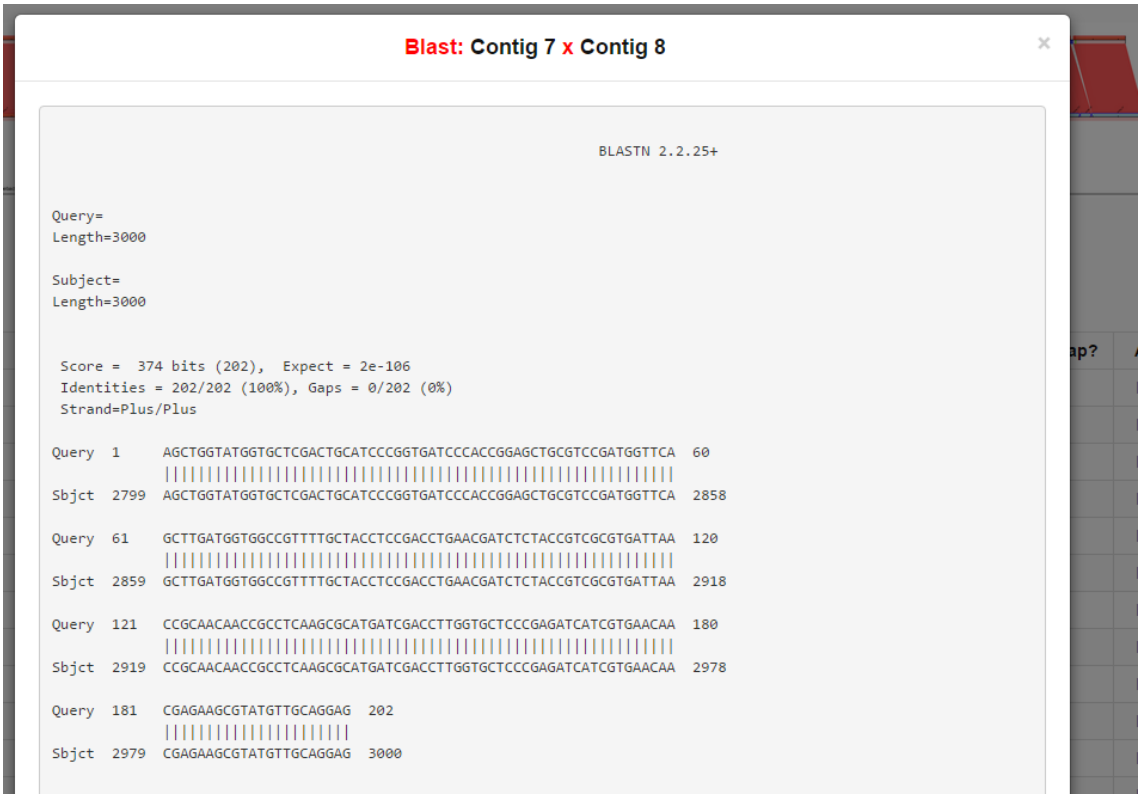


Figure 23. Construct Supercontigs: BLAST results.

If it was detect a match in the BLAST result, the user can merge the contigs sending positions for SIMBA to cut one of the homologues regions (Figure 24). We point out that the analysis of similarities among contigs must to be done carefully by the user. SIMBA gives to the user total control to do alterations in the contigs.

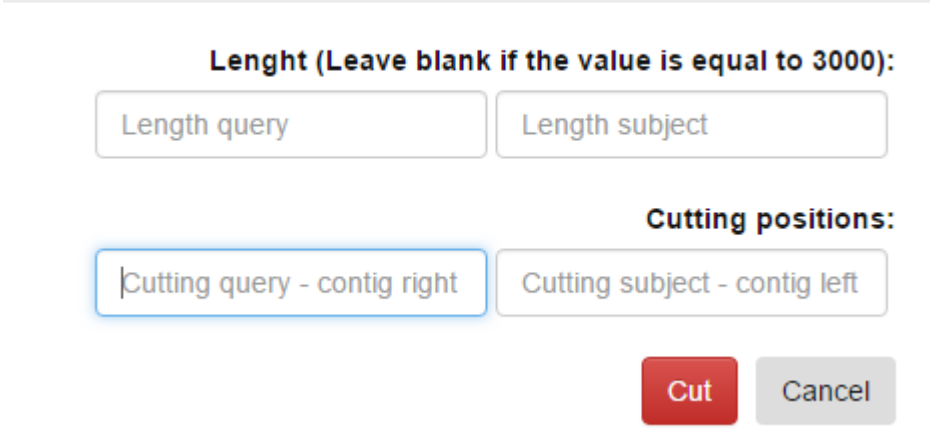


Figure 24. Construct Supercontigs: cut positions.

In the step 4, SIMBA allows close repetitive gaps using the software MapRepeat[9] (Figure 25).

[9] https://github.com/dcbmariano/maprepeat

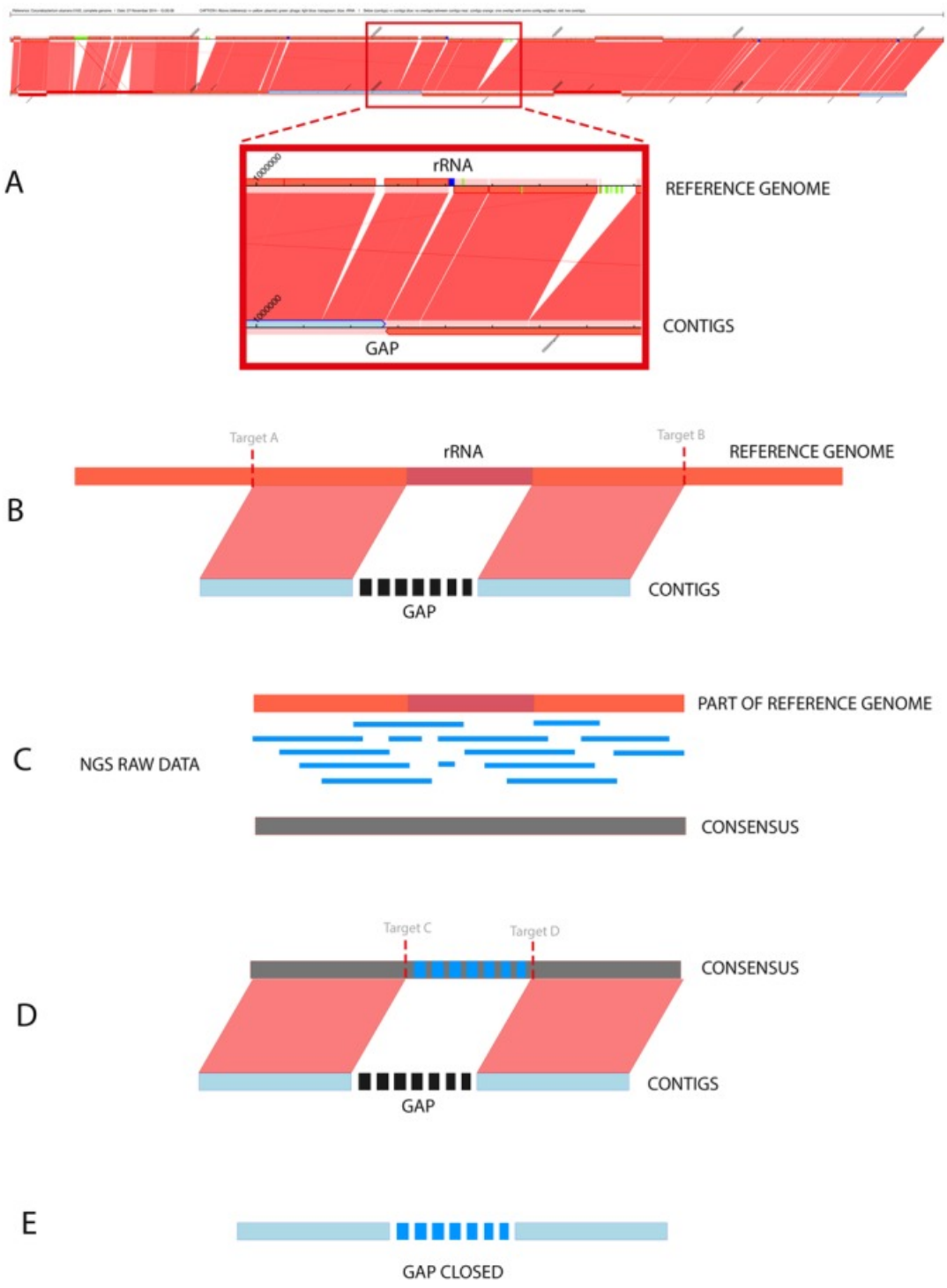


Figure 25. MapRepeat workflow. (A) detect region corresponding in reference genome. (B) Insert targets A and B. (C) Map raw data using Mira. (D) Insert targets C and D to detect gap region on reference. (E) Close gap. *Source*: Mariano *et al*. MapRepeat: an approach for effective assembly of repetitive regions in prokaryotic genomes. Bioinformation. 2015; 11(6): 276–279. Published online 2015 Jun 30. doi: 10.6026/97320630011276

In the SIMBA interface MapRepeat just require click on the button “map” to run (Figure 26).

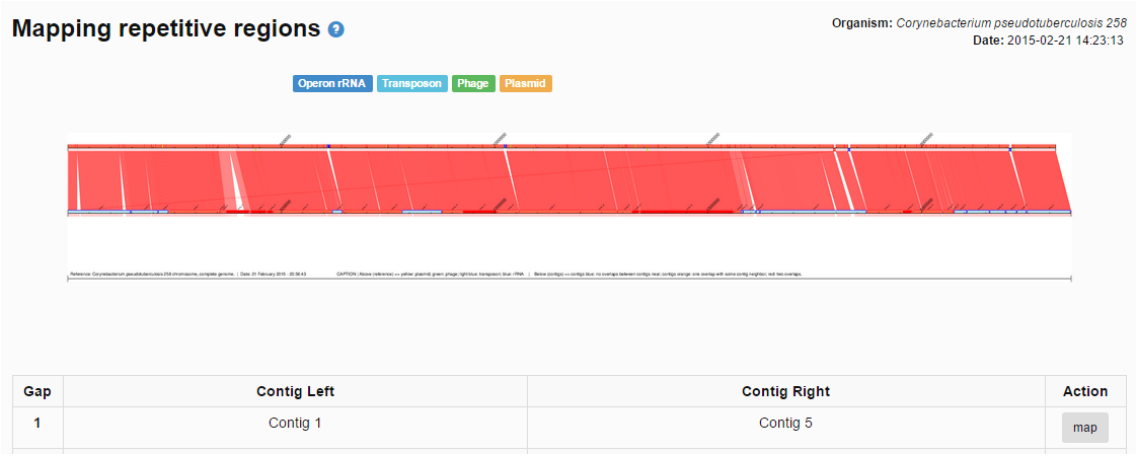


Figure 25. Step 4: running MapRepeat.

Warning: we recommend “skip” this step if: (i) you have sequencing with deep coverage more than 50-fold; (ii) you don’t have a reference phylogenetically closer; (iii) you will use experimental strategies to close repetitive gaps.

In the step 5, SIMBA shows undefined nucleotides information about the genome in the steps 4 and 5. SIMBA also shows information about the genome and allows the download of contigs excluded, files with contigs/scaffolds of step 4 and 5 (Figure 26).

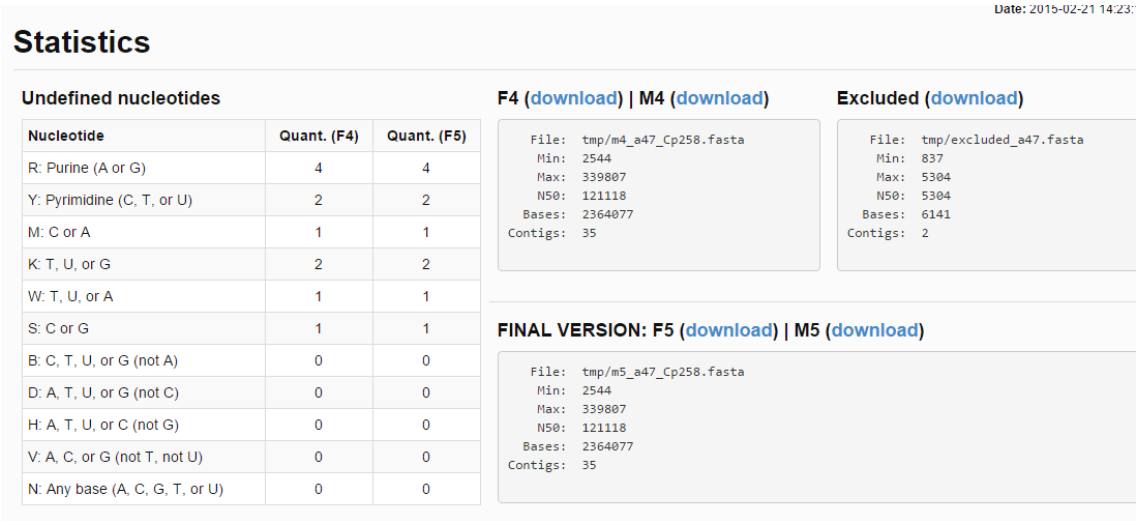


Figure 26. Step 5 – manual curation.

You can download the data of step 4 and curate with other software. After, SIMBA allows that the final file will be stored in SIMBA interface (Figure 27). The final genome can be downloaded in the field “FINAL VERSION: F5” (Figure 26).

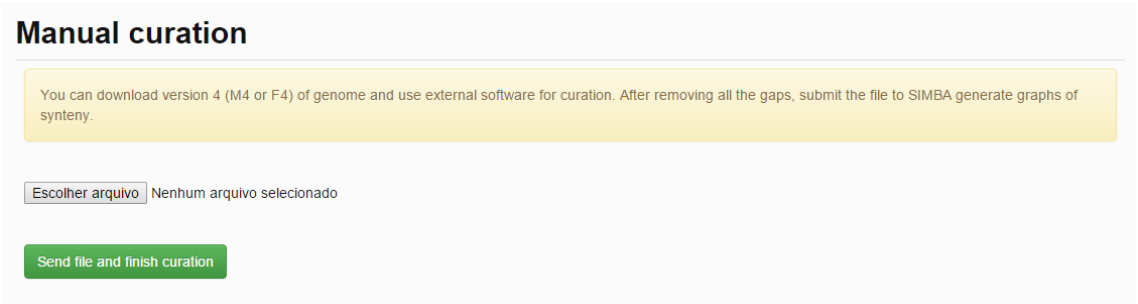


Figure 27. Step 5: sending final genome after manual curation to SIMBA.

4. SIMBA for developers

If you are a developer, you can make modifications in the source code of SIMBA.

SIMBA was developed using PHP[10], the framework Laravel[11], and a SQLite[12] databank. SIMBA also uses several scripts developed in Python[13] and the library Biopython[14] for sequence analysis.

SIMBA source code is available at:

- http://github.com/dcbmariano/simba
- http://sourceforge.net/projects/ufmg-simba/

4.1 SIMBA directories

In the main folder of SIMBA, there are four directories: (i) app; (ii) vendor; (iii) public; and (iv) bootstrap. Vendor and bootstrap are folders used by Laravel. Public store all data that can be access by the browser. The main codes are stored at “app” directory.

Laravel uses the methodology MVC (model, view, controller):

- “Model” provides access to the database (SQLite). SIMBA just store in the SQLite database statistical information, such as assembly attempts, number of contigs of assembly, curation steps performed, etc. The large files are stored in the original format. Genomes sequencing raw data are stored in the folder “app/assembly”, while contigs files and synteny graphs are stored in the folder “public”.
- “View” provides the HTML pages that SIMBA shows. All view files are stored in the folder “app/views”. The main layout file (master.blade.php) is stored at “app/views/layout”. It is responsible to load the layout of the interface (note that scripts and style sheets are stored on the public directory).
- “Controller” provides access to everything. The files in the folder “app/controllers” contain codes to load views according to the URL called by the browser (for personalize URLs see the file “app/routes.php”), provide access to the SQLite database, run python scripts and execute wrappers tools, such as BLAST, CONTIGuator, Mira, MapRepeat, etc.

[10] http://www.php.net/
[11] http://laravel.com/
[12] https://www.sqlite.org/
[13] https://www.python.org/
[14] biopython.org/

The most important file is “ProjectsController.php”. It controls the execution of assembly software and the project manager. Another important file is “ActionController.php”: responsible by parser assembly results and update tables.

4.1 Using SIMBA with TORQUE

SIMBA can be used in parallel with the TORQUE Resource Manager[15]. TORQUE provides control over batch jobs and distributed computing resources. To use SIMBA with TORQUE, first create a job queue called “assembly”. After, you will need alter the SIMBA source code.

Edit the file “app/controllers/ ProjectsController.php”. Search by the public function “run_new_assembly( )”. SIMBA was configured to run the assembly software in background without interruptions if the session end using the command structure “nohup” + command + “&” (Figure 28).

```
213                 /* Execucao com nohup */
214                 $query = "cd $folder && nohup ../../bin/mira $name.manifest > $name.log.txt &";
215
216                 /* Grava exec_mira - uso de gerenciadores de fila
217                 $exec = $this->raiz.$name_project.'/mira.sh';
218                 $exec_content = "#PBS -o mira.out\n#PBS -e mira.err\n\ncd \$PBS_O_WORKDIR\n../../bin/mira $name.manifest";
219                 $pt = fopen($exec,'w');
220                 fwrite($pt,$exec_content);
221                 fclose($pt);
222
223                 $query = "cd $folder && qsub -q assembly ./mira.sh";*/
224                 break;
```

Figure 28. Run assembly software in background – default.

To use TORQUE, comment the line with the variable “$query” and remove the comments of the lines below (Figure 29). Repeat this process for all assemblies (declared by the lines started with “case”).

```
switch($assembler){
    case 'mira':
        /* Execucao com nohup */
        # $query = "cd $folder && nohup ../../bin/mira $name.manifest > $name.log.txt &";

        /* Grava exec_mira - uso de gerenciadores de fila */
        $exec = $this->raiz.$name_project.'/mira.sh';
        $exec_content = "#PBS -o mira.out\n#PBS -e mira.err\n\ncd \$PBS_O_WORKDIR\n../../bin/mira $name.manifest";
        $pt = fopen($exec,'w');
        fwrite($pt,$exec_content);
        fclose($pt);

        $query = "cd $folder && qsub -q assembly ./mira.sh";
        break;
```

Figure 29. Run assembly software with TORQUE.

[15] http://www.adaptivecomputing.com/products/open-source/torque/

4.2 Adding new assembler software

SIMBA provide by default assemblies only with Mira. However, SIMBA supports Minia, Newbler and SPAdes. To install these software, first download the software and put the binary file in the folder “app/bin”.

Except Newbler that need to be in the folder “app/bin/454”. SIMBA execute the Newbler through the binary “454/apps/mapper/bin/runAssembly”. Check if the file is in this correct directory. You also can change the address in the file “app/controllers/ ProjectsController.php”. Look by the code block started with “case 'newbler':”.

SIMBA only provide methods to run and parsers to analyze the results of these software. Consult the license of each one.

You can also insert new assemblers. For this:

(i) insert the command line in the function “run_new_assembly( )” of “app/controllers/ProjectsController.php” file;
(ii) insert a parser for the result the software in the public function “update_assemblies_info( )” – use the parser of Newbler as reference (look for “/* Newbler parser */”);
(iii) put the binary file in the directory “app/bin”;
(iv) insert the options for use the assembly in the file “app/views/assembly_create.php” (Figure 30).

```
<br/><br/>
<div class="alert alert-danger">
    <label class="label label-danger">ASSEMBLER:</label><br/><br/>
    <select name="assembler" class="form-control">
        <option value="mira">Mira 4.0.2</option>
        <option value="mira39" selected="selected">Mira 3.9</option>
        <option value="newbler">Newbler</option>
        <option value="minia">Minia</option>
        <option value="text">Manual assembly</option>
    </select>
</div>

<br/>
```

Figure 30. Add a new assembler software in the view.