

[v1.2]

# SIMBA docs



S I M B A

LGCM

Federal University of Minas Gerais

2015

## Summary

1. Introduction .....	3
1.1 Why use SIMBA? .....	3
1.2 How does SIMBA work? .....	4
1.3 How to download SIMBA? .....	4
2. Installing SIMBA .....	5
2.1 Installing basic requirements in O. S. Ubuntu 14.04 .....	5
2.1.1 PHP5 and libraries .....	6
2.1.2 Python and Biopython library .....	6
2.1.3 NCBI-BLAST+ .....	6
2.2 Installation steps .....	7
2.2.1 Configuring SIMBA .....	7
2.2.2 Accessing SIMBA by a browser .....	7
3. SIMBA interface.....	8
3.1 Understanding genome assembly process .....	8
3.2 SIMBA Workflow .....	9
3.3 General vision of SIMBA interface .....	10
3.3.1 Creating users .....	11
3.4 Module projects.....	12
3.4.1 Creating new projects.....	13
3.5 Module assemblies .....	14
3.5.1 Running a new assembly .....	15
3.5.2 Validating assemblies.....	16
3.6 Module curation.....	18
4. SIMBA for developers.....	26
4.1 SIMBA directories .....	26
4.1 Using SIMBA with TORQUE .....	27
4.2 Adding new assembler software .....	28

## 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 is also an 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 a better experience in genome assembly.

### 1.1 Why use SIMBA?

---

**SIMBA allows bioinformaticians to not worry so much about techniques with repetitive activities, and focus on the more important activity of understanding and resolving biological questions!**

---

Genome assembly requires the integration of different processes with a high degree of complexity that often involve various heuristics that combine to obtain results closer to the biological reality. Thus, several software is required for the complete assembly process. In addition to the knowledge required for the biological components, assembly projects also require a good understanding of the underlying operating system as well as the specific operations of each software; this often leads to a slow learning curve!

Moreover, many repetitive processes could be reduced with the adoption of automation scripts and tools organized in a simple pipeline, which controlled by

a graphical interface, can accelerate the process of data processing, assembly and curation – here lies the major contribution of SIMBA.

## 1.2 How does SIMBA work?

---

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

---

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

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

## 1.3 How to download SIMBA?

SIMBA can be downloaded from:

- <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:

```
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	Options	Indexes	FollowSymLinks
MultiViews			MultiViews		
AllowOverride	None		AllowOverride	All	
Order	allow,deny		Order	allow,deny	
Allow from	all		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 the 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 Installation 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 line:

```
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

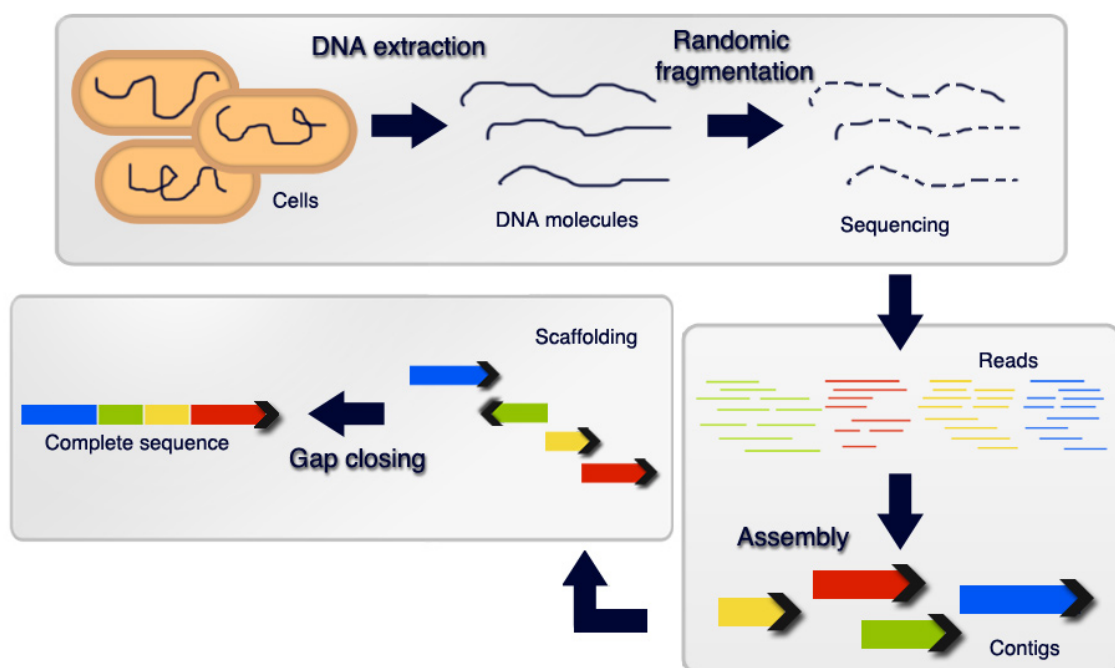
Now, SIMBA can be accessed by a browser, as Google Chrome, Firefox, Opera or Safari, using the address configured in the “app.php” file. Note that SIMBA can also run on Internet Explorer, however some pages can present layout problems. We recommend the Google Chrome browser.

### 3. SIMBA interface

#### 3.1 Understanding genome assembly process

Before explaining the SIMBA pipeline, it is necessary to first understand the problem of genome assembly.

Currently, most DNA sequencing platforms can read only small DNA fragments. Thus, after the sequencing process of several DNA molecules of the same organism is completed, it becomes important to reconstruct the original genome sorting the individual fragment reads. This process is known as genome assembly (Figure 2).



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

It is possible to sort the reads using a phylogenetically closer organism as reference. However, in most cases we may not have a reference to help in the assembly. Thus, it is necessary to join the reads based on the overlap among its sequences. Sort reads without a reference is called *de novo* assembly or *ab initio* assembly.

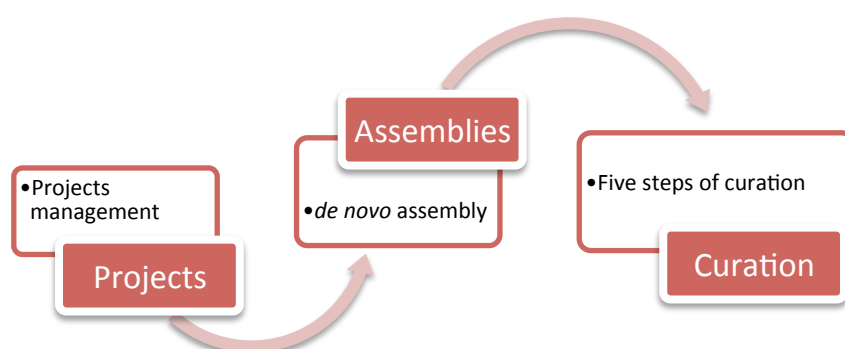
Due to the repetitive regions as well as the low coverage regions, the assembly software may not be able to reconstruct the whole genome. This produces gaps among the maximally continuous sequences (contigs).



After assembly, it is necessary to sort the 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 important to close the gaps among scaffolds (contigs sorted). We designate as curation, the process of sorting contigs and closing gaps among them.

### 3.2 SIMBA Workflow

SIMBA is divided into 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<sup>1</sup>. SIMBA also provides support to assemblers: (i) Newbler<sup>2</sup>; (ii) Minia<sup>3</sup>; and (iii) SPAdes<sup>4</sup>.

The five steps of curation are handled by SIMBA as follows: (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<sup>5</sup>); (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 is shown in Figure 4.

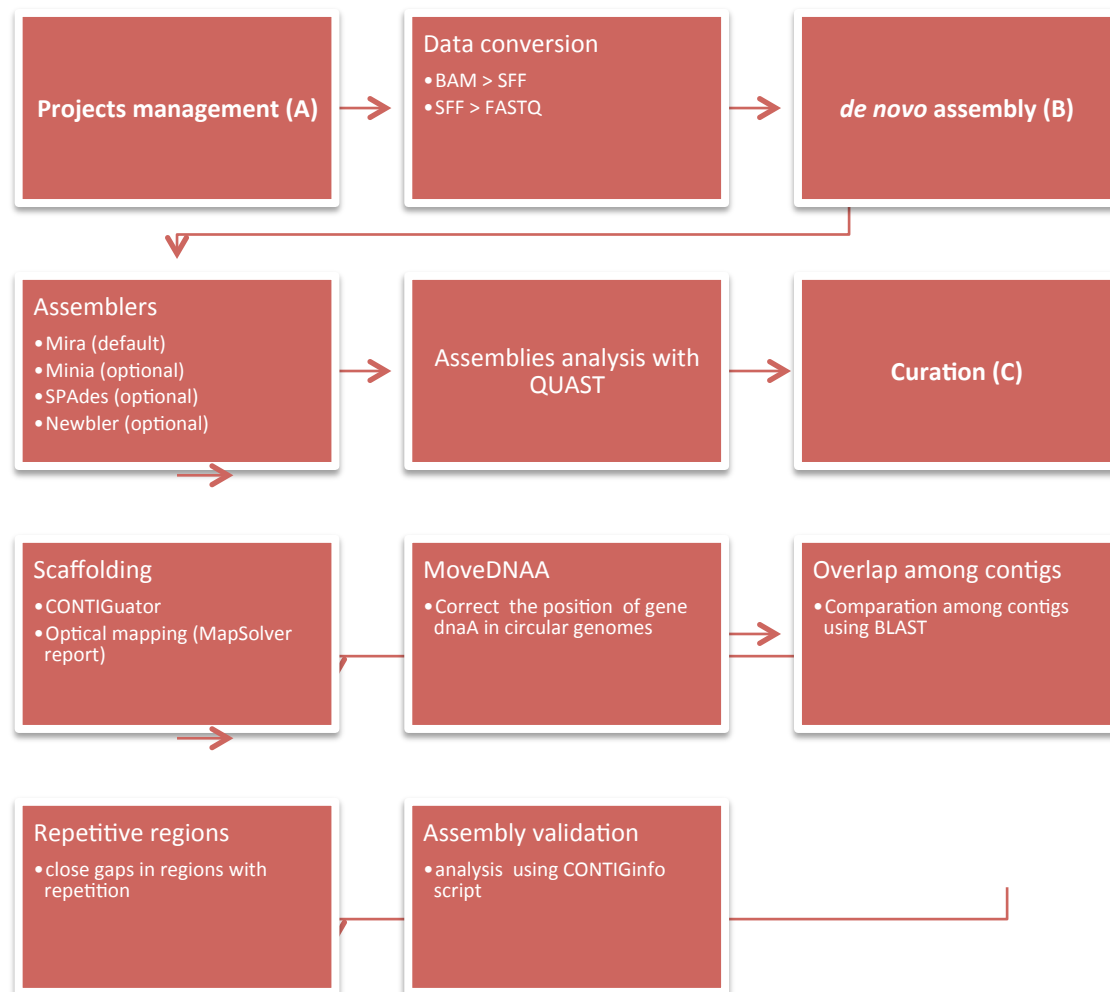
<sup>1</sup> <http://mira-assembler.sourceforge.net/docs/DefinitiveGuideToMIRA.html>

<sup>2</sup> <http://www.454.com/products/analysis-software/>

<sup>3</sup> <http://minia.genouest.org/>

<sup>4</sup> <http://bioinf.spbau.ru/spades>

<sup>5</sup> For more information visit: <<http://opgen.com/>>.



**Figure 4.** SIMBA workflow.

### 3.3 General vision of SIMBA interface

SIMBA interface is composed of a toolbar, main area and footer (Figure 5). The toolbar provides 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 when the user logged and provides access to the control panel.

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

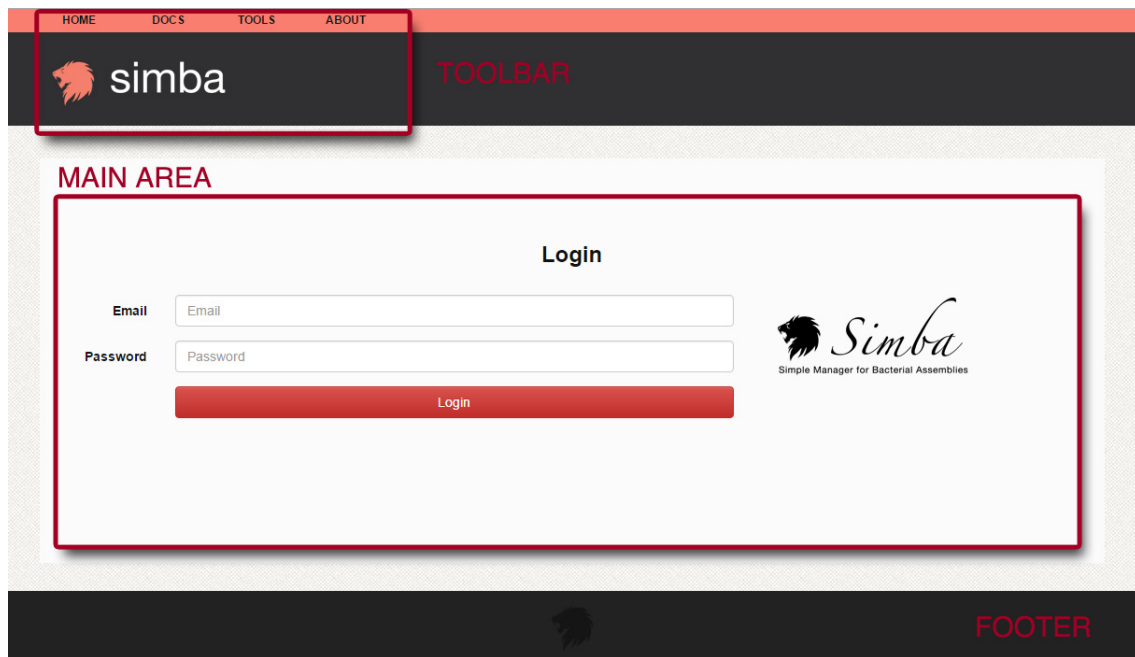


Figure 5. SIMBA interface.

SIMBA by default uses the user and password:

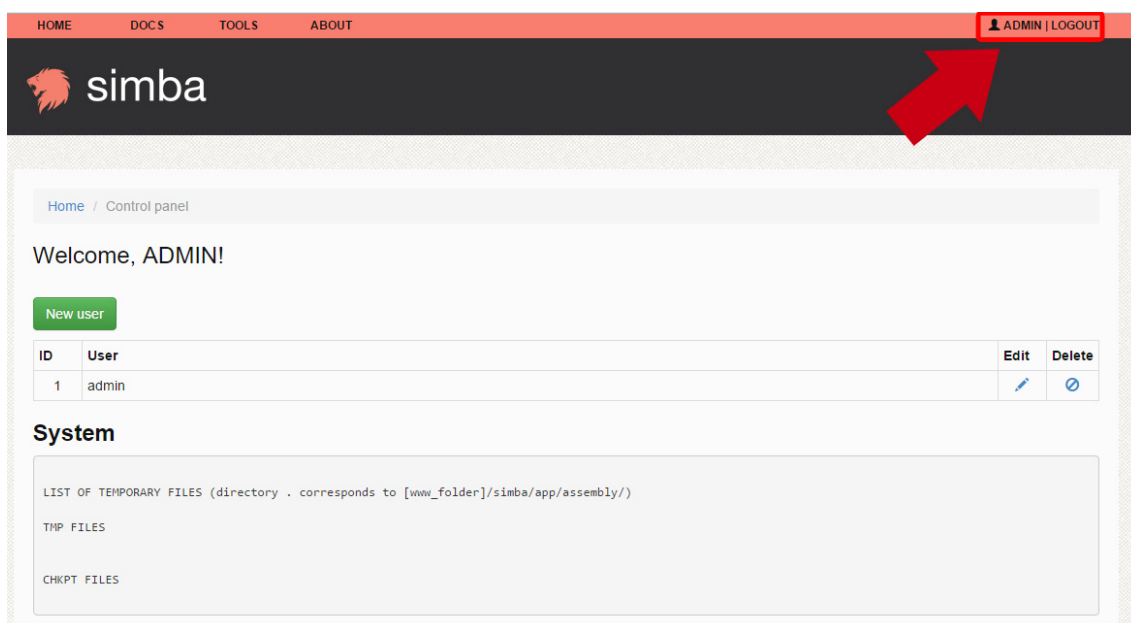
**User:** admin

**Password:** admin

You can change these values in the user control panel.

### 3.3.1 Creating users

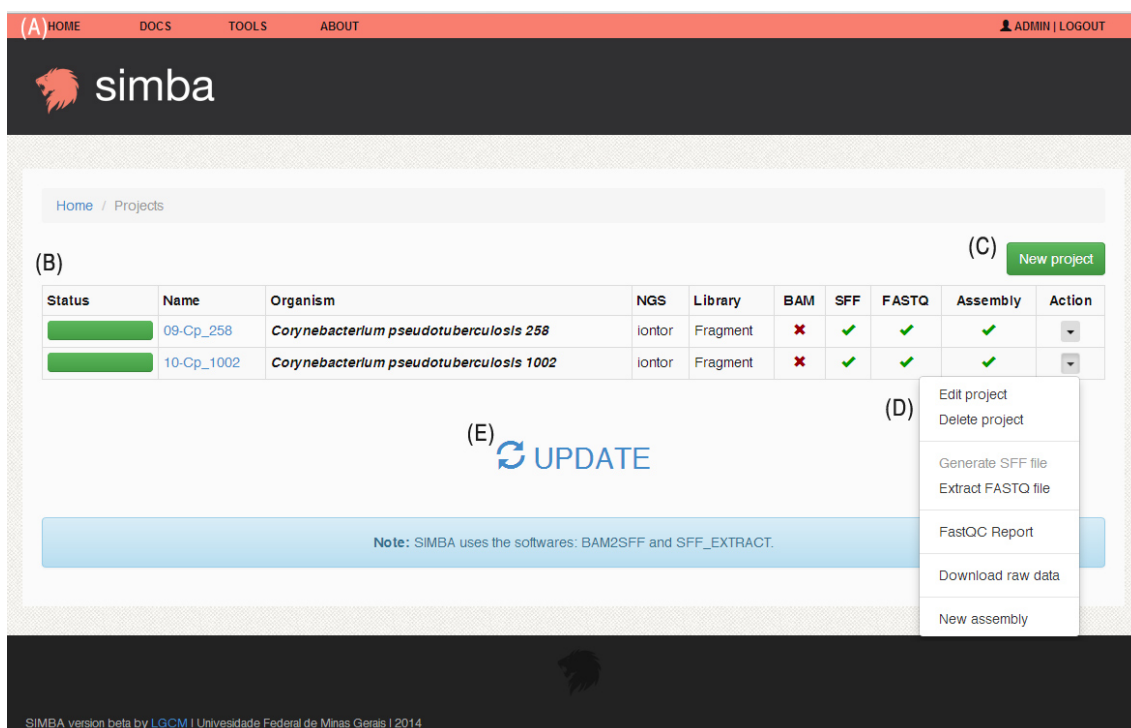
Click on user name on the toolbar to open the control panel. Only the user admin can create new users. Click on the pencil symbol to change the password of the admin. Click on new user to create new users (Figure 6).



**Figure 6.** Creating new users.

### 3.4 Module projects

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

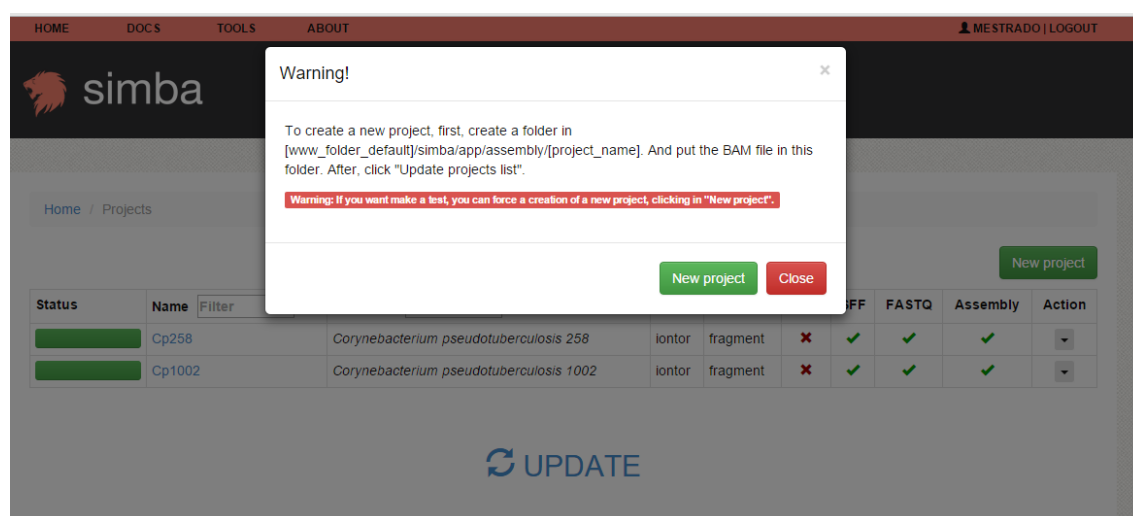


**Figure 7.** Module projects. (A) toolbar. (B) table with genomes projects. Shows: status of the project, project name, organism name, NGS, library, format of raw data, and assemblies realized (C) Allow the creation of new projects. (D) Allow the execution of actions, such as run

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 the SIMBA folder and click on “update”. SIMBA will automatically detect a new project and create a new item in the table (click on action and edit to alter the information in the row). However, you can click on 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 presents 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).

The screenshot shows the SIMBA web interface. At the top, there is a navigation bar with links: HOME, DOCS, TOOLS, ABOUT, and a user profile section for MESTRADO | LOGOUT. Below the navigation bar is a dark header with the SIMBA logo. The main content area has a breadcrumb trail: Home / Projects / New project. The form for creating a new project includes several sections:
 

- Folder:** A text input field with the placeholder text "Name Organism (E.g.: coryne\_pseud\_1002. Don't use CAPS LOCK or type spaces)".
- Name organism:** A text input field with the placeholder text "Specie organism strain".
- NGS:** A dropdown menu currently showing "iontor".
- Library:** A text input field with the placeholder text "Library (E.g.: fragment, mate-pair (2kb, 3kb, 10kb, 20kb), paired-end)".
- Raw data:** A section with a button labeled "Escolher arquivo" and the text "Nenhum arquivo selecionado".

 At the bottom of the form is a large green button labeled "Create".

**Figure 9.** Creating a new project.

The new project will be listed on the projects page.

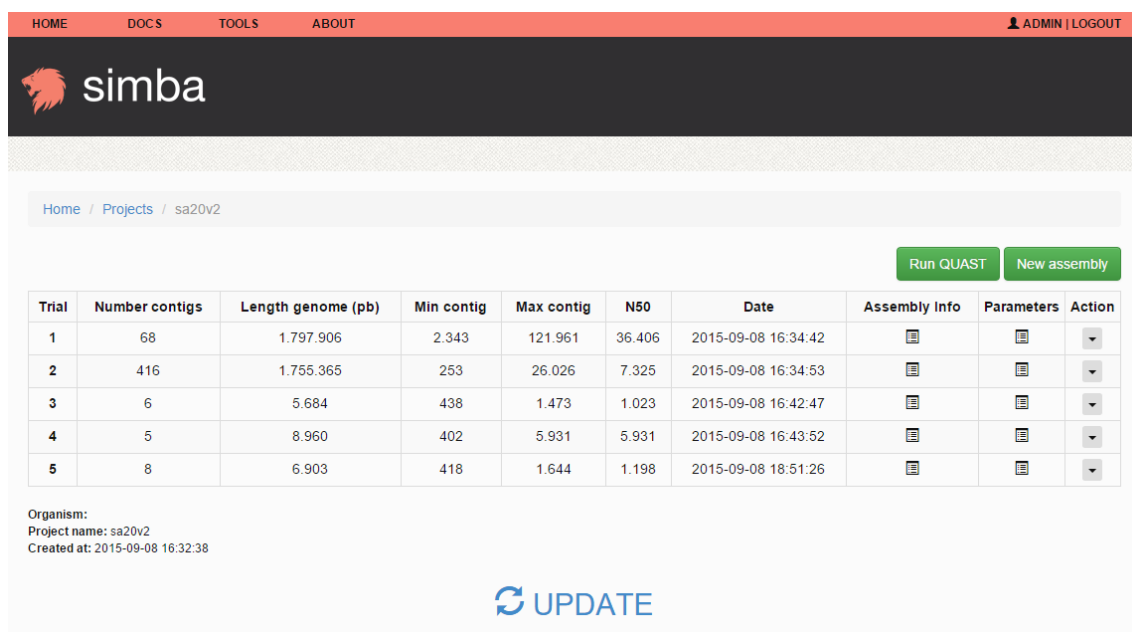
SIMBA uses FastQC<sup>6</sup> to generate reports of quality reads. For this, click on “action > FastQC report”. SIMBA also provides format conversions, such as BAM > SFF or SFF > FASTQ (Figure 7D).

### 3.5 Module assemblies

Click on the link at the project name or click on “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 that allows the 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.

<sup>6</sup> <http://www.bioinformatics.babraham.ac.uk/projects/fastqc/>



HOME DOCS TOOLS ABOUT ADMIN | LOGOUT

simba

Home / Projects / sa20v2

Run QUAST New assembly

Trial	Number contigs	Length genome (pb)	Min contig	Max contig	N50	Date	Assembly Info	Parameters	Action
1	68	1.797.906	2.343	121.961	36.406	2015-09-08 16:34:42			
2	416	1.755.365	253	26.026	7.325	2015-09-08 16:34:53			
3	6	5.684	438	1.473	1.023	2015-09-08 16:42:47			
4	5	8.960	402	5.931	5.931	2015-09-08 16:43:52			
5	8	6.903	418	1.644	1.198	2015-09-08 18:51:26			

Organism:  
Project name: sa20v2  
Created at: 2015-09-08 16:32:38

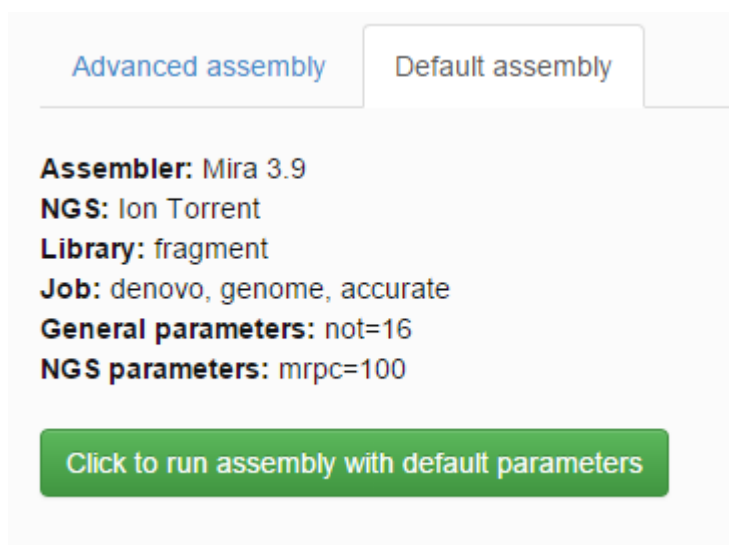
UPDATE

**Figure 10.** Module assemblies.

### 3.5.1 Running a new assembly

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

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



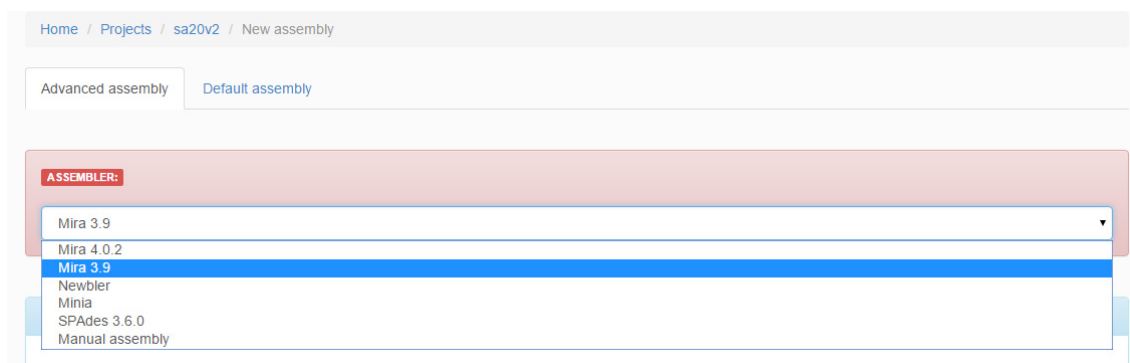
Advanced assembly Default assembly

**Assembler:** Mira 3.9  
**NGS:** Ion Torrent  
**Library:** fragment  
**Job:** denovo, genome, accurate  
**General parameters:** not=16  
**NGS parameters:** mrpc=100

Click to run assembly with default parameters

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

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



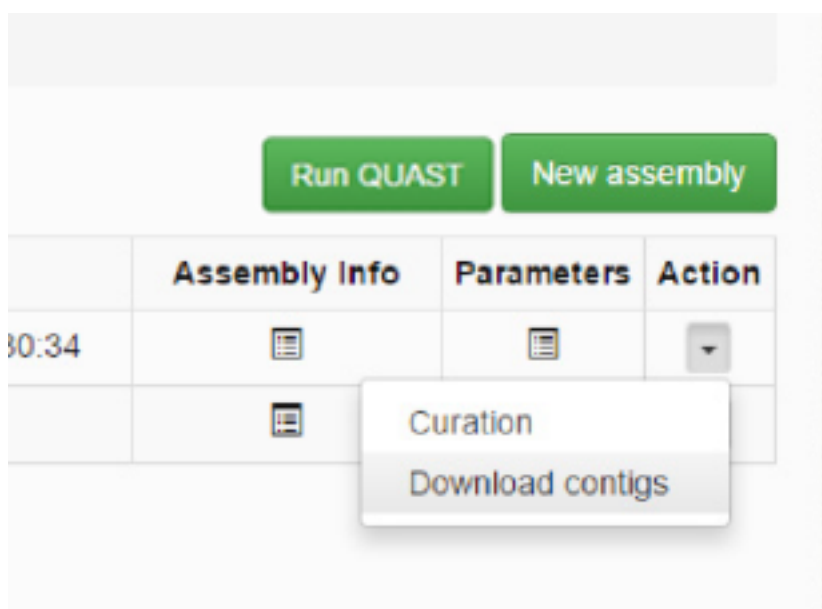
**Figure 12.** Advanced assembly.

---

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

---

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

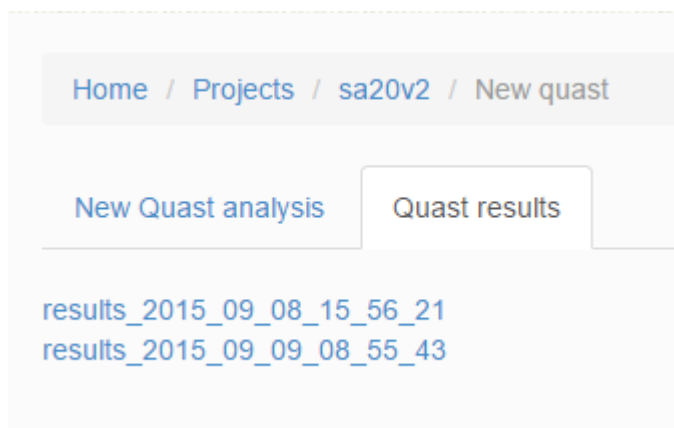


**Figure 13.** Module assembly: downloading contigs.

### 3.5.2 Validating assemblies



SIMBA uses QUAST<sup>7</sup> to validate assemblies (inserted version 1.2). To run QUAST, click on the button “Run QUAST”. The button just appears when executed at a minimum of 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  $\geq 500$  bp, unless otherwise noted (e.g., “# contigs ( $\geq 0$  bp)” and “Total length ( $\geq 0$  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
<b>Misassemblies</b>					
# misassemblies	2	0	1	2	1
Misassembled contigs length	7608	0	1644	4414	1473
<b>Mismatches</b>					
# 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
<b>Genome statistics</b>					
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.





















<sup>7</sup> <http://bioinf.spbau.ru/quast>

### 3.6 Module curation

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

**Module Curation requirement:** each step needs to be executed before triggering the next.

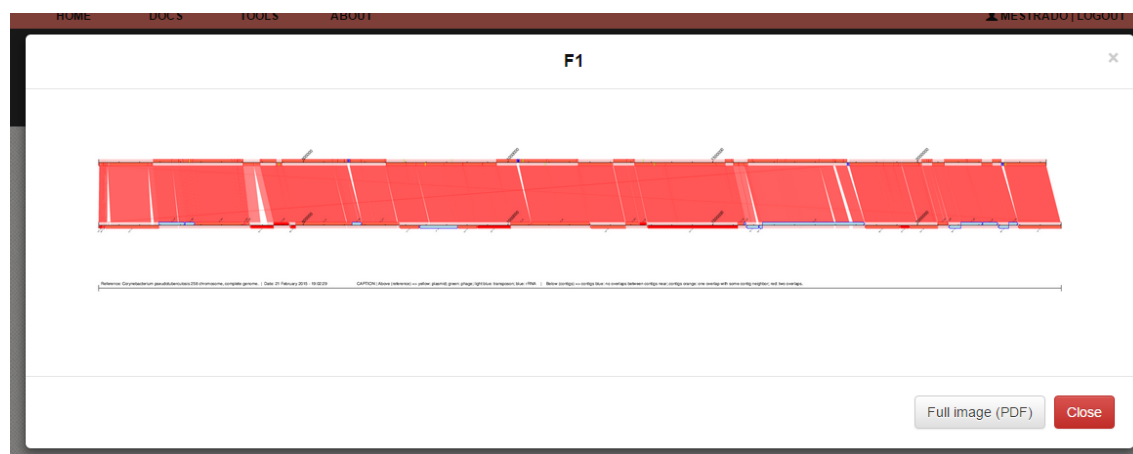
Home / Projects / 10-Cp\_1002 / Trial 14 (A)

(B) Step	(C) Status	Action	(D) Gaps	(F) Synteny chart	(G) Download	(H)	(I) Action
1	✓	Set reference	6				
2	✓	Move dnaA	6				
3	✓	Building Supercontigs	4				
4	✓	Analyze repetitive regions	4				
5	✓	Statistics and manual curation	0				

(E) Organism: *Corynebacterium pseudotuberculosis* 1002  
Date: 2014-05-07 14:00:00

**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”.

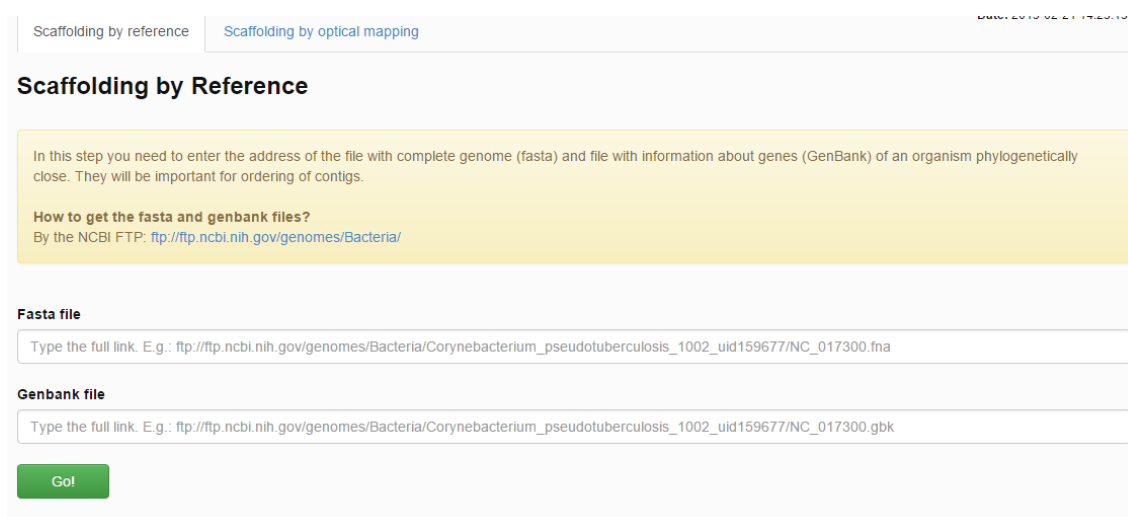
After each step, SIMBA performs a comparison using a modified version of the software CONTIGuator<sup>8</sup>. CONTIGuator generates a synteny graph (Figure 17), that helps users to detect assembly errors.



<sup>8</sup> <http://contiguator.sourceforge.net/>

**Figure 17.** Synteny graph using CONTIGuator.

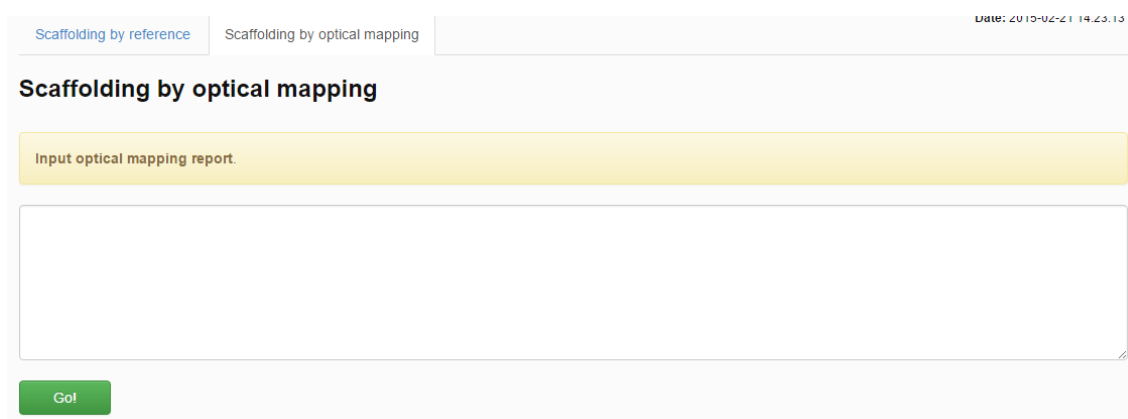
In the step 1, SIMBA performs the process of scaffolding. For scaffolding by reference (Figure 18), SIMBA requires links for two files: a GenBank file (gbk) and a Genome Complete file (fna). These links can be obtained at: < <ftp.ncbi.nih.gov/genomes/>>.



The screenshot shows the 'Scaffolding by reference' tab selected. It includes a yellow instruction box stating that users need to provide a complete genome (fasta) and a GenBank file (gbk) for scaffolding. Below this, there are two text input fields: one for the 'Fasta file' and one for the 'Genbank file'. Both fields contain a sample URL: 'ftp://ftp.ncbi.nih.gov/genomes/Bacteria/Corynebacterium\_pseudotuberculosis\_1002\_uid159677/NC\_017300.fna' and 'ftp://ftp.ncbi.nih.gov/genomes/Bacteria/Corynebacterium\_pseudotuberculosis\_1002\_uid159677/NC\_017300.gbk' respectively. A green 'Go!' button is located at the bottom left of the form.

**Figure 18.** Step 1 - scaffolding using reference.

SIMBA also provides 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.



The screenshot shows the 'Scaffolding by optical mapping' tab selected. It features a yellow instruction box that says 'Input optical mapping report.' Below this is a large, empty text area for pasting the report. A green 'Go!' button is positioned at the bottom left of the interface.

**Figure 19.** Step 1 - scaffolding using optical mapping.

---

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

---

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

**Correcting the beginning of the file by dnaA gene**
Organism: *Corynebacterium pseudotuberculosis* 258  
Date: 2015-02-21 14:23:13

At this stage we fix the sequence so that it begins by dnaA gene. A cut will be made in the "sequence" in the start position of the first gene. The new contig formed will be moved to the beginning of the genome.

Run movednaA.py and CONTIGuator

or

SKIP

**Figure 20.** Move dnaA.

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

In this step we will close gaps in overlapping regions between neighboring contigs to build super contigs. View the alignments using BLAST and send cutting positions to SIMBA can do the processing and closing of the gap. **Understanding what will be done at this stage:**

Contig left

Position to cut subject

Position to cut query

Contig right

Tip: If there is overlap, click "Blast". You can set breakpoints, but we recommend choosing the last two numbers (the latter referring to "subject" and the latter referring to "query"). Finally, click "cut". [Start!](#)

**Important:** save changes to the database only when all editing is completed (stay tuned to "is there overlap?" column).

**Figure 21.** Move dnaA.

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

---

**Warning:** only click on “save updates in database” after the closing of all gaps

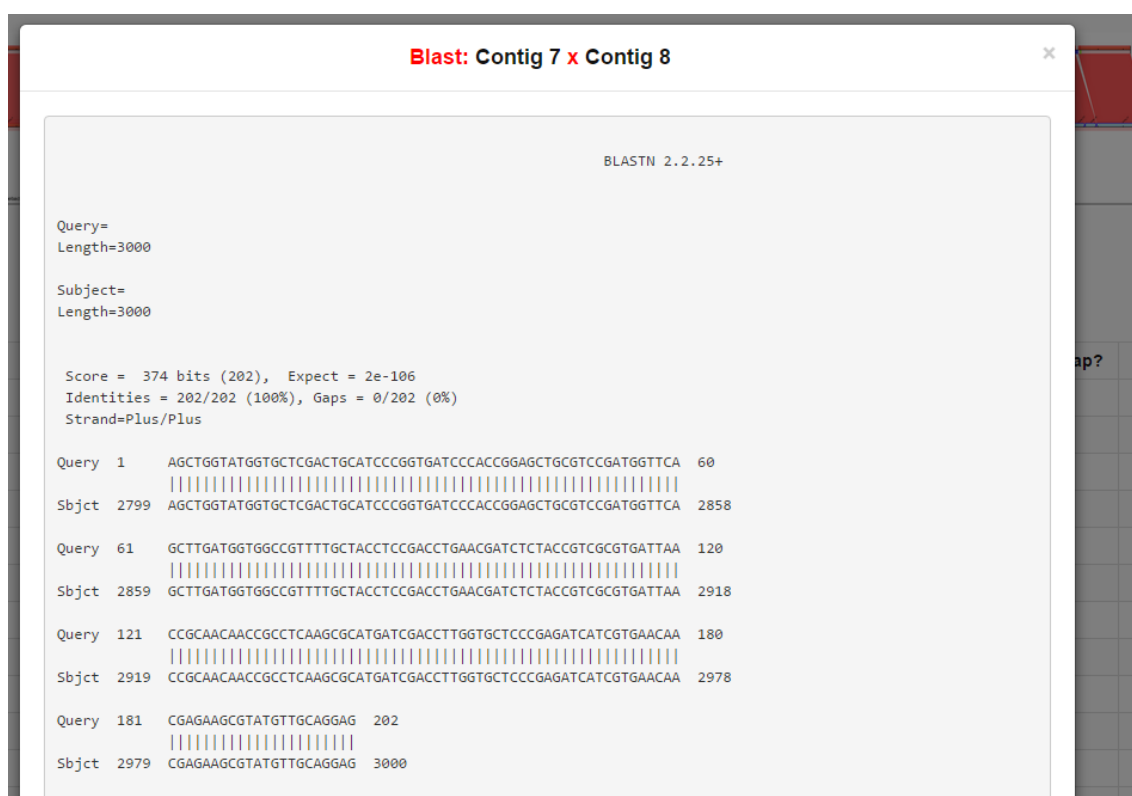
---

was possible.



**Figure 22.** Construct Supercontigs.

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



**Figure 23.** Construct Supercontigs: BLAST results.

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

---

**Lenght (Leave blank if the value is equal to 3000):**

Length query	Length subject
--------------	----------------

**Cutting positions:**

Cutting query - contig right	Cutting subject - contig left
------------------------------	-------------------------------

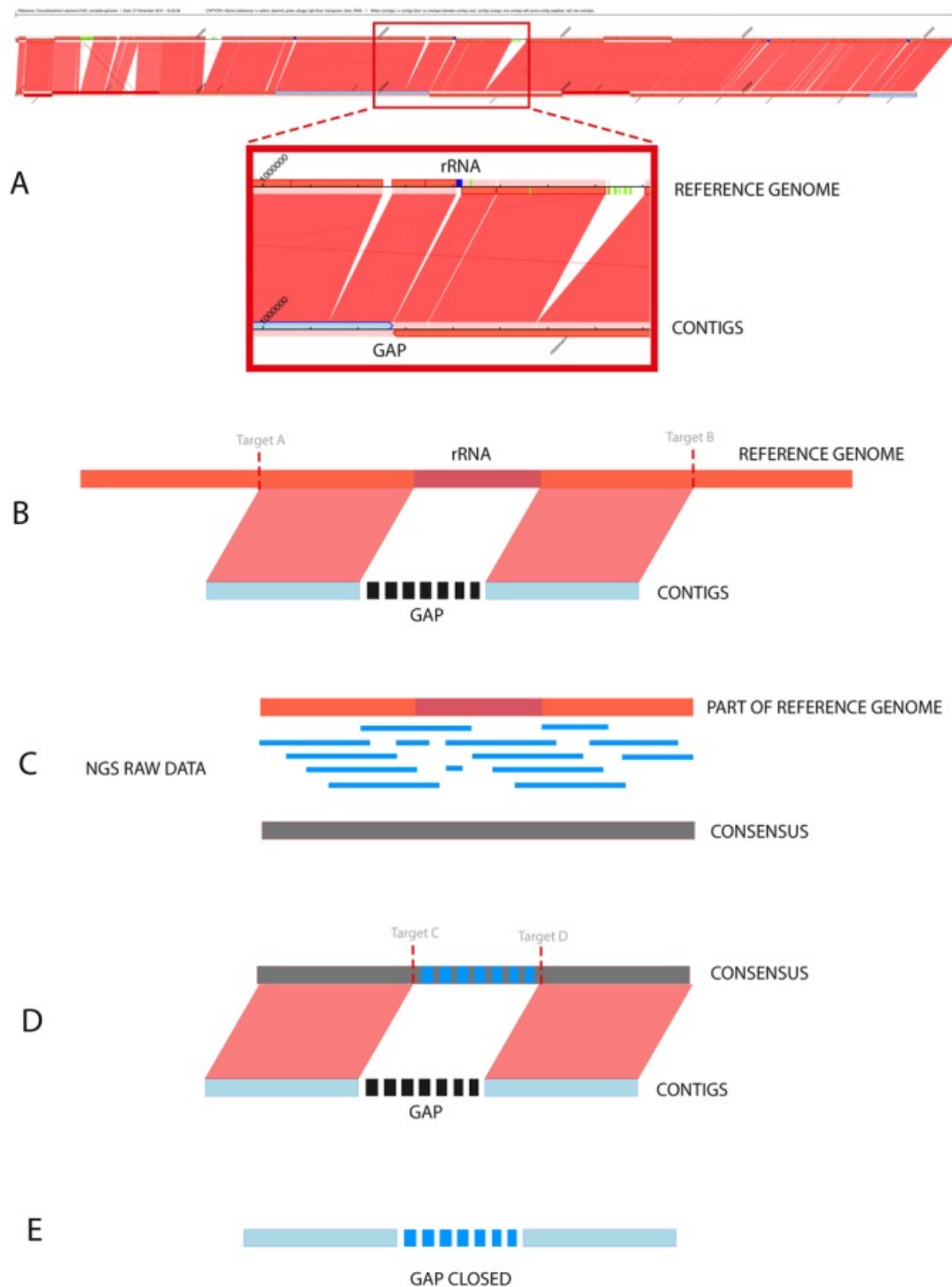
**Cut** **Cancel**

**Figure 24.** Construct Supercontigs: cut positions.

In the step 4, SIMBA helps in closing repetitive gaps using the software MapRepeat<sup>9</sup> (Figure 25).

---

<sup>9</sup> <https://github.com/dcbmariano/maprepeat>



**Figure 25.** MapRepeat workflow. (A) detect region corresponding to 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. *Bioinformatics*. 2015; 11(6): 276–279. Published online 2015 Jun 30. doi: 10.6026/97320630011276

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

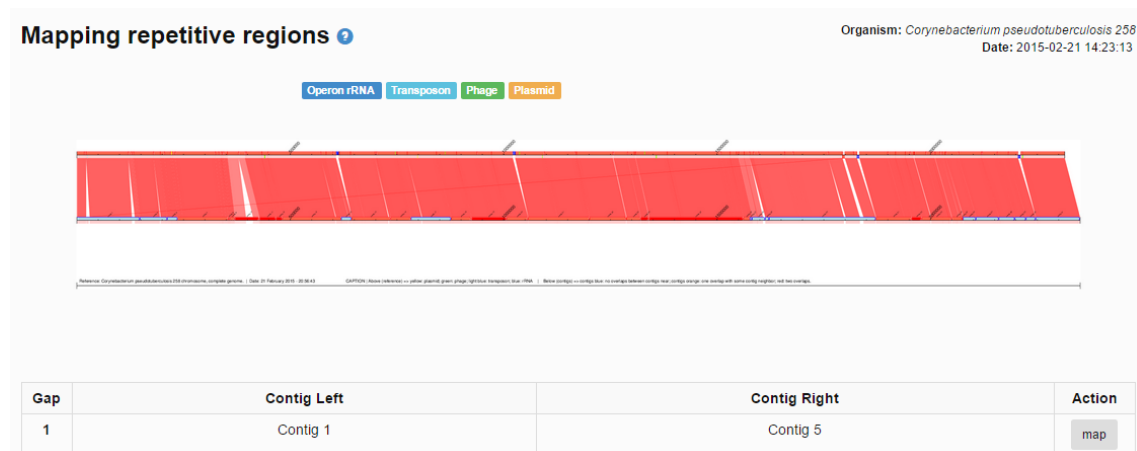


Figure 25. Step 4: running MapRepeat.

**Warning:** we recommend to “skip” this step if: (i) you have sequencing with deep coverage more than 50-fold; (ii) you don’t have a phylogenetically closer reference; (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 by excluding files with contigs/scaffolds of step 4 and 5 (Figure 26).

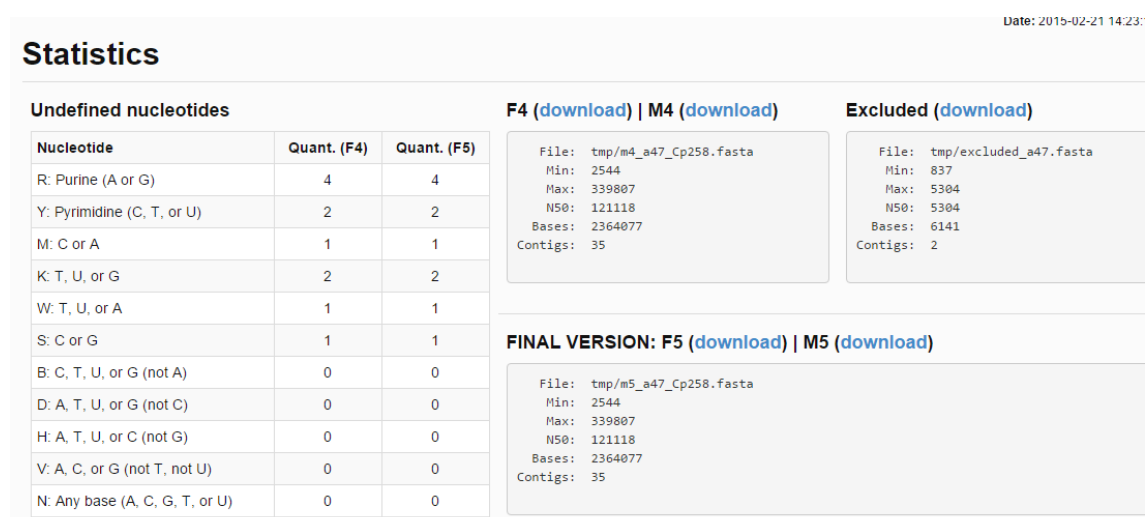


Figure 26. Step 5 – manual curation.



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

### Manual curation

You can download version 4 (M4 or F4) of genome and use external software for curation. After removing all the gaps, submit the file to SIMBA generate graphs of synteny.

Nenhum arquivo selecionado

**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 to the source code of SIMBA.

SIMBA was developed using PHP<sup>10</sup>, the framework Laravel<sup>11</sup>, and a SQLite<sup>12</sup> databank. SIMBA also uses several scripts developed in Python<sup>13</sup> and the library Biopython<sup>14</sup> 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 stores all data that can be accessed by the browser. The main codes are stored in the “app” directory.

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

- “Model” provides access to the database (SQLite). SIMBA just stores in the SQLite database different 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 personalized 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.

---

<sup>10</sup> <http://www.php.net/>

<sup>11</sup> <http://laravel.com/>

<sup>12</sup> <https://www.sqlite.org/>

<sup>13</sup> <https://www.python.org/>

<sup>14</sup> [biopython.org/](http://biopython.org/)

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

#### 4.1 Using SIMBA with TORQUE

SIMBA can be used in parallel with the TORQUE Resource Manager<sup>15</sup>. TORQUE provides control over batch jobs and distributed computing resources. To use SIMBA with TORQUE, first create a job queue called “assembly”. After that you will need to 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 a background without interruptions if the session was ended 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\nncd \${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\nncd \${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.

<sup>15</sup> <http://www.adaptivecomputing.com/products/open-source/torque/>

## 4.2 Adding new assembler software

By default, SIMBA provides 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”.

---

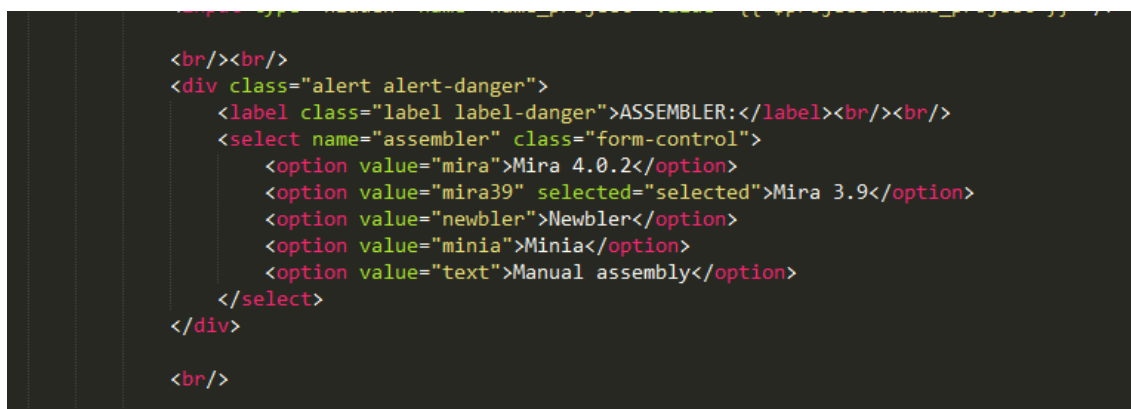
Except Newbler that needs to be in the folder “app/bin/454”. SIMBA executes the Newbler through the binary “454/apps/mapper/bin/runAssembly”. Check if the file is in this correct directory. You can also change the address in the file “app/controllers/ ProjectsController.php”. Look by the code block starting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

---

SIMBA only provides 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 of 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 in 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.