

BRIG 0.95 Manual

Nabil Alikhan

June 27, 2011

<http://sourceforge.net/projects/brig/>

Contents

1	Introduction	3
2	Licence	9
3	Installation	9
3.1	Installing BLAST	9
4	Warning when using BLAST	11
4.1	Low complexity filtering	11
4.2	Expected values(e-values) and bit scores	11
5	Visualising whole genome comparisons	13
5.1	Step 1: Load in sequences	13
5.2	Step 2: Configure rings	14
5.3	Step 3: Review and submit	16
6	Working with a Multi-FASTA reference	18
6.1	Step 1: Load in sequences	18
6.2	Step 2: Configure rings, annotations and spacer value	19
6.3	Step 3: Configure image settings and submit	23
7	Visualising graphs and genome assemblies	25
7.1	Walkthrough for visualising SAM file mapping coverage.	26
7.2	Walk through for visualising ace file assembly coverage.	31
8	Walkthroughs on creating custom annotations	39
8.1	Adding custom annotations from a tab-delimited file, GenBank or EMBL file	39
8.1.1	Step 1: Load in sequences	39
8.1.2	Step 2: Configure rings	40
8.1.3	Step 3: Adding annotations	41
8.1.4	Step 4: Review and submit	43
8.2	How to create tab-delimited files for BRIG	45
9	Configuration options	46
9.1	Saving and reopening your work	46
9.2	BLAST options	46
9.3	Setting BRIG options	48
9.4	Setting Image options	50
9.5	Loading a preset image template	52

1 Introduction

The BLAST Ring Image Generator (BRIG) is a cross-platform desktop application written in Java 1.6. It uses CGView[5] for image rendering and the Basic Local Alignment Search Tool (BLAST) for genome comparisons. It has a graphical user interface programmed on the Swing framework, which takes the user step-by-step through the configuration of a circular image generation. Figure 1 is an example of an image BRIG can create.

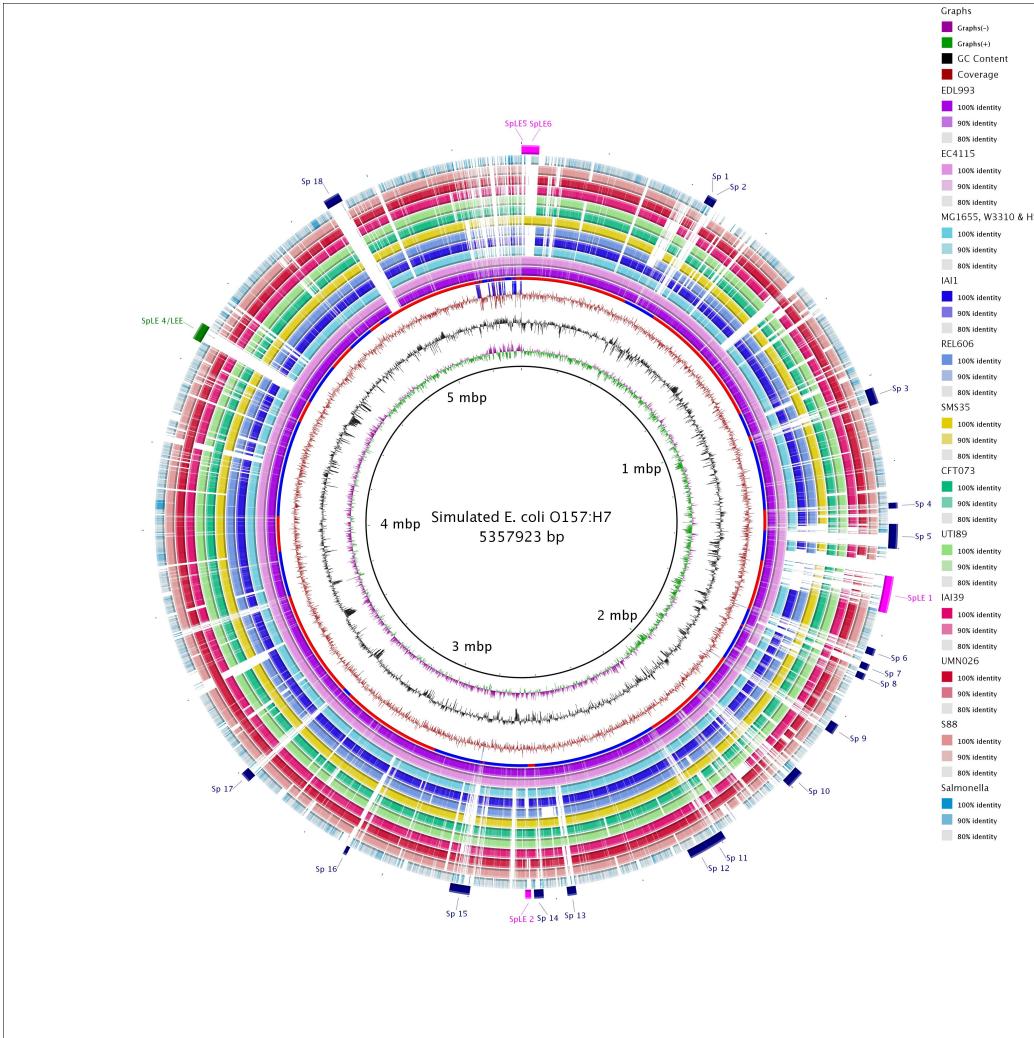


Figure 1: BRIG example output image of a simulated draft *E. coli* O157:H7 genome. The figure shows BLAST comparisons against 28 published *E. coli* and *Salmonella* genomes against the simulated draft genome.

Figure 2 shows a magnified view of the same example image showing similarity between a central reference genome in the centre against other query sequences as a set of concentric rings, where colour indicates a BLAST match of a particular percentage identity. BRIG does not represent sequences that are not present in the reference genome. The image shows:

- GC skew,
- GC content,
- Genome coverage and contig boundaries (calculated from an assembly file),
- Genome alignment results, customs annotations.

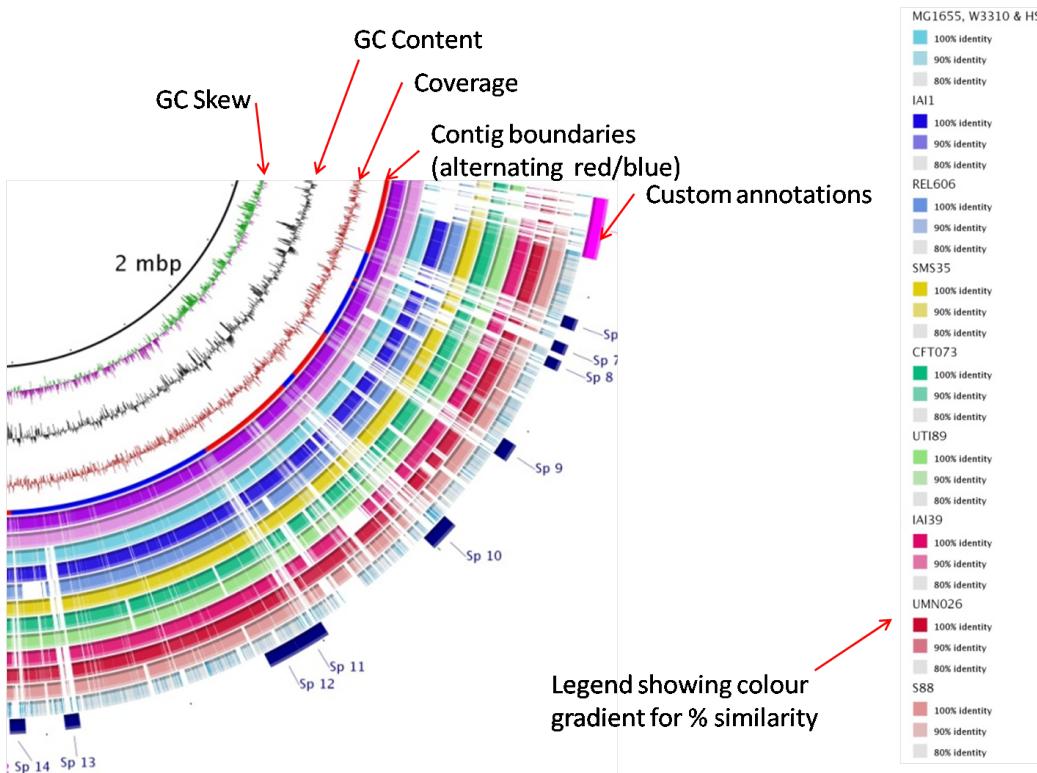


Figure 2: A magnified view of BRIG example image

How to use this manual

This manual contains a set of detailed walk throughs where readers are taken step by step through a worked example. Each walkthrough highlights different features of BRIG and users should work through each one. If you are interested in a particular aspect of BRIG, please turn to the relevant walkthrough:

- Whole genome comparisons, including how to load in coverage graphs, e.g. Figures 1 & 3, see Section 5 on page 13.
- Using a user-defined list of genes as a reference (in Multi-FASTA), e.g Figure 4, see Section 6 on page 18.
- Creating and visualising graphs generated from assembly (.ace) or read mapping coverage (.SAM), e.g Figure 5, see Section 7 on page 25.
- Labeling images with information from GenBank, Tab-delimited or Multi-FASTA files, like those seen in Figure 3, 4 & 5, see Section 8 on page 39.

The manual also has detailed instructions for how to install and configure BRIG:

- For instructions on how to install BRIG, see Section 3 on page 9.
- For instructions on how to configure BRIG and save BRIG settings, see Section 9 on 46.

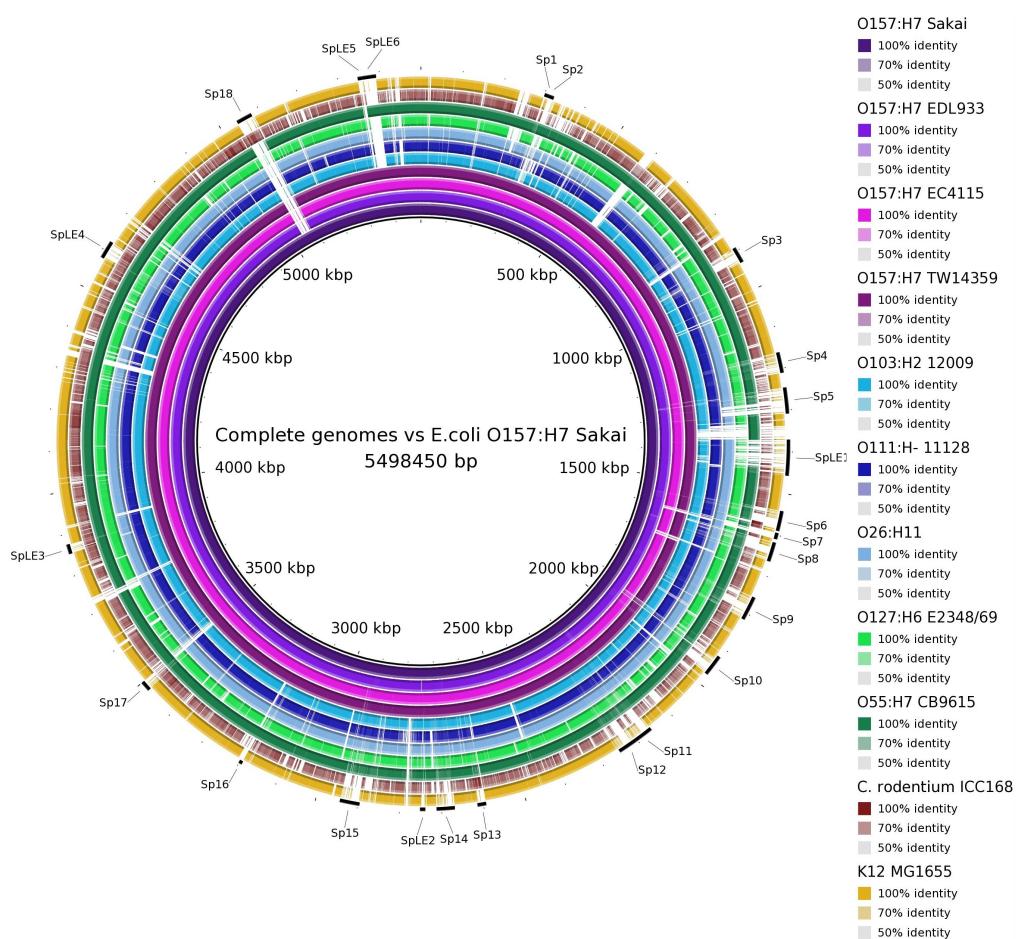


Figure 3: **Reference:** Published *E.coli* O157:H7 Sakai genome. **Query:** Complete genome sequences of related strains, listed in the key. The prophage regions from the Sakai genome are marked in alternating black & blue. To make an image like this please refer to Section 5 on page 13.

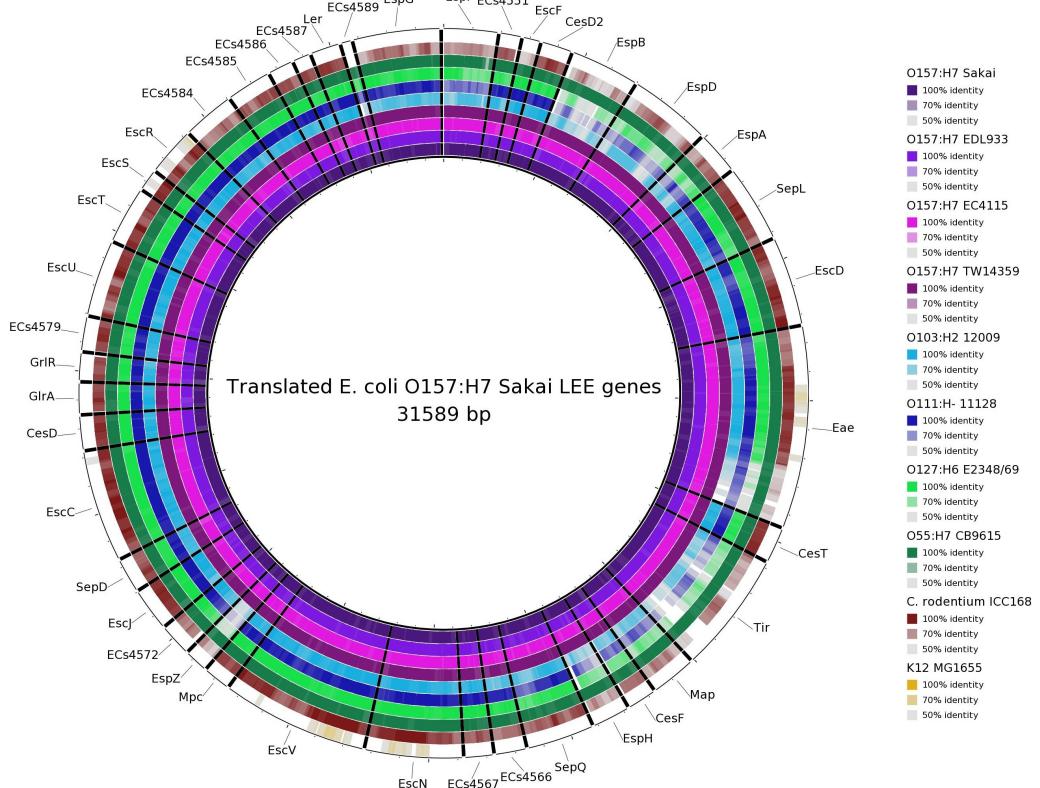


Figure 4: Reference: A list of translated genes that make up the Locus of Enterocyte Effacement (LEE), which encodes a Type III secretion system. **Query:** Raw sequencing reads simulated from several complete LEE+ published genomes (nucleotide sequence) and *E. coli* K12, (negative control; LEE-). You can clearly see gene presence/absence, and divergence (the colour represents sequence identity on a sliding scale, the greyer it gets; the lower the percentage identity). To make an image like this please refer to Section 6 on page 18.

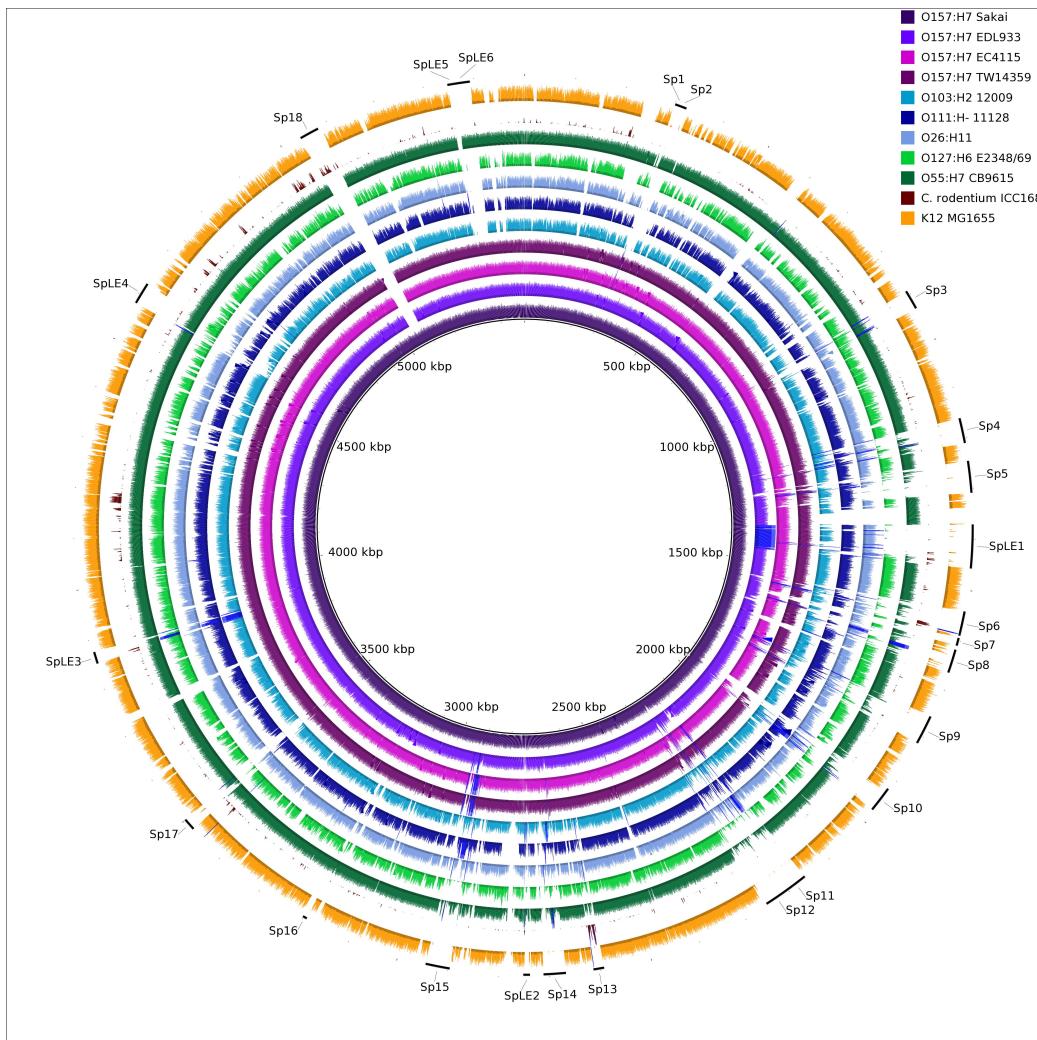


Figure 5: Reference: Published *E.coli* O157:H7 Sakai genome. **Query:** Read mapping coverage of sequencing reads simulated from complete genomes, indicated in the key. Simulated sequencing reads were mapped onto the published complete Sakai genome using BWA. The read coverage for each genome was generated from the resulting SAM files. Compare this with Figure 3, which is based on the original published genome sequences. To make an image like this please refer to Section 7 on page 25.

2 Licence

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but without any warranty; without even the implied warranty of merchantability or fitness for a particular purpose. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <<http://www.gnu.org/licenses/>>.

Please note that these restrictions do not apply to the third party libraries bundled with this software.

3 Installation

There's no real "Installation" process for BRIG itself. However, BLAST+[2] or BLAST legacy[1] must already be installed and BRIG needs to be able to locate the BLAST executables (See Section 3.1).

To run BRIG users need to:

1. Download the latest version (BRIG-x.xx-dist.zip) from
<http://sourceforge.net/projects/brig/>
2. Unzip BRIG-x.xx-dist.zip to a desired location.
3. Run BRIG.jar, by double clicking.

Users who wish to run BRIG from the command-line need to:

1. Navigate to the unpacked BRIG folder in a command-line interface (terminal, console, command prompt).
2. Run "java -Xmx1500M -jar BRIG.jar". Where -Xmx specifies the amount of memory allocated to BRIG.

3.1 Installing BLAST

The latest version of BLAST+[2] can be downloaded from:

<ftp://ftp.ncbi.nlm.nih.gov/blast/executables/blast+/LATEST/>
BLAST+ offers a number of improvements on the original BLAST implementation and comes as a bundled installer, which will walk users through the installation process. Please read the published paper on BLAST+:

Camacho, C., G. Coulouris, *et al.* (2009). "BLAST+: architecture and applications." *BMC Bioinformatics* **10**(1): 421 Available online at : <http://www.biomedcentral.com/1471-2105/10/421>

The latest version of BLAST legacy[1] can be downloaded from:

<ftp://ftp.ncbi.nlm.nih.gov/blast/executables/release/LATEST/>

BLAST legacy comes as a compressed package, which will unzip the BLAST binaries where ever the package is. We advise users to first create a BLAST directory (in either the home or applications directory), copy the downloaded BLAST package to that directory and unzip the package.

BRIG supports both BLAST+ & BLAST Legacy. Users can specify the location of their BLAST installation in the BRIG options menu which is:

Main window >Preferences >BRIG options.

The window is shown in Figure 6. If BRIG cannot find BLAST it will prompt users at runtime.

PRO TIP 1: *BRIG uses BLAST, do not use wwwblast or netblast with BRIG.*

PRO TIP 2: *If BOTH BLAST+ and legacy versions are in the same location, BRIG will prefer BLAST+.*

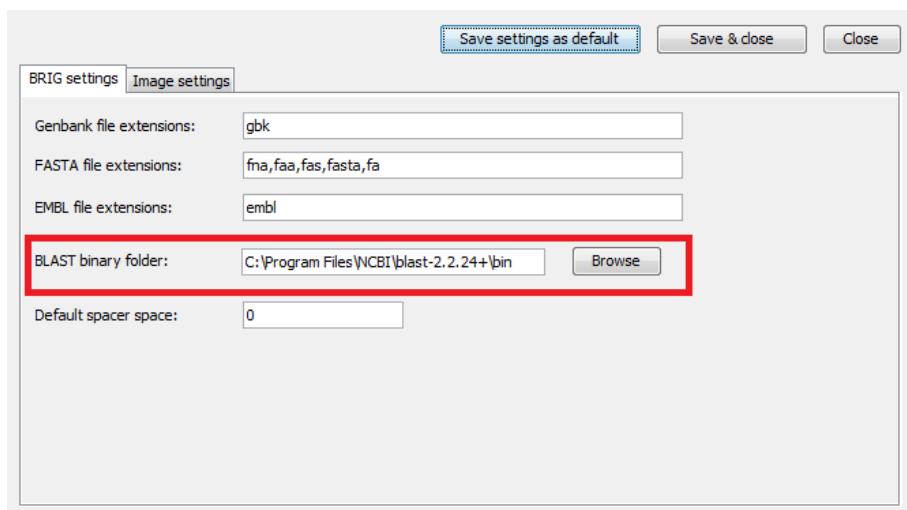


Figure 6: You can change where BRIG looks for BLAST in the BRIG options window. For more information about BRIG options see Section 9.2 on page 48.

4 Warning when using BLAST

BRIG relies on the Basic Local Alignment Search Tool (BLAST) for genome comparisons. BLAST has a number of behaviours that may seem counterintuitive and we encourage users to learn about local alignment and the BLAST algorithm to fully understand the images that BRIG produces. There are a few concepts to keep in mind when using BRIG:

4.1 Low complexity filtering

PRO TIP 3: *BLAST filters may cause gaps in alignments, which will show up as blank regions in BRIG images.*

BLAST filters (BLAST legacy -F flag or BLAST+ -dust/-seg no flag) filter the query sequence for low-complexity sequences by default. This includes sequences that are highly repetitive or contain the same nucleotide for long lengths of the sequences. Low-complexity filtering is generally a good idea, but it may break long matches into several smaller matches.

This is often shown in BRIG images as truncations or gaps in alignments, it is particularly obvious in very small reference sequences where alignments are shown on a gene-by-gene level.

To prevent this, either turn off filtering or use soft masking.

4.2 Expected values(e-values) and bit scores

PRO TIP 4: *BLAST's bitscore filtering may cause different results in BRIG if users swap the query and reference sequences, particularly if these are very different sizes.*

BLAST uses statistical thresholds to filter out “bad alignments”; alignment matches that appear random to BLAST. One of these thresholds is the e-value, which is the probability of the alignment occurring by chance, given the complexity of the match, sequence composition and the size of the database. It is more likely in a larger sequence that an alignment could occur by chance, so BLAST is more critical of these matches.

This can create different expected values if BLAST is used with the same reference sequence against databases of different sizes and may potentially filter out significant matches or include poor scoring ones.

Because of this, users might notice different results in BRIG images if they swap the order of the database and reference sequences around in the BLAST, especially if the two sequences are quite different in size. The differences are often due to a few very low-scoring hits.

Users should consider what an appropriate e-value threshold is for the comparisons that they run. Remember, that BLAST runs with an e-value of 10 by default, we recommend that users change this value. Users can set the final threshold (e-value) with the -e flag in BLAST legacy or -evaluate flag in BLAST+.

PRO TIP 5: *BLAST does not handle spaces in filenames, BRIG will prompt users if they have spaces in file locations.*

5 Visualising whole genome comparisons

In this section we will walk through the basics of generating an image. This walk through will be comparing an *E. coli* genome with five other *E. coli* genomes and mapping the read coverage from the underlying genome assembly onto the same image. For this walk through, users will need BRIG_examples.zip, which is available from the BRIG website (<http://sourceforge.net/projects/brig/files/>). This contains all the genomes and files needed to follow along with this walk through. Unzip it somewhere easily accessible, like the home directory or desktop.

About the reference genome

The reference genome used in this walk through is a simulated *E. coli* genome assembly. We took the published *E. coli* O157:H7 Sakai genome (Accession number BA000007) sequence and had assembly reads simulated by METASIM[4] and then assembled these using Newbler version 2.3. The resulting contiguous sequences were ordered using Mauve[3] against the published Sakai genome. This simulated *E. coli* is useful for illustrating some of BRIG's graphing features for assembly read coverage.

Enterohemorrhagic *E. coli* are gram-negative, enteric bacterial pathogens. They can cause diarrhea, hemorrhagic colitis, and hemolytic uremic syndrome. This particular genome we are using in this example was based on an *E. coli* O157:H7 isolated from the Sakai, Japan outbreak.

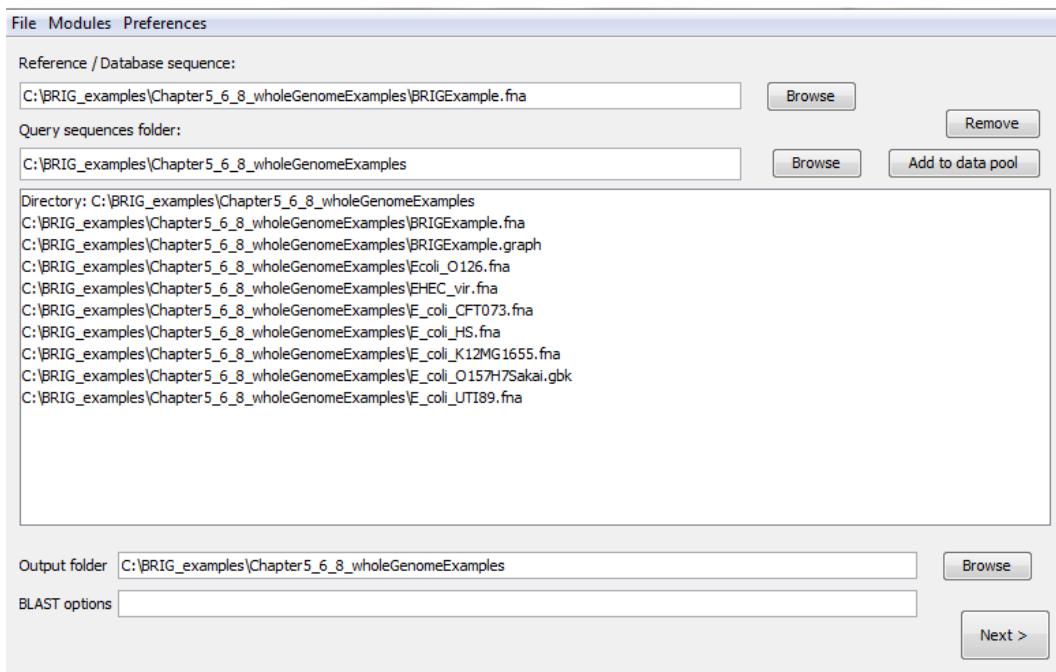
5.1 Step 1: Load in sequences

The walk through will work out of the unzipped BRIG_examples.zip in the Chapter5_6_8_wholeGenomeExamples folder. The walk through and related figures will use C:\BRIG_examples\Chapter5_6_8_wholeGenomeExamples as that location.

To keep the final image consistent with the walk through, please open "ExampleProfile.xml" from the Chapter5_6_8_wholeGenomeExamples folder. This file configures BRIG to the same image settings in the walk through.

1. First, set BRIGExample.fna as the reference sequence.
2. Set <unzipped BRIG_examples folder>\Chapter5_6_8_wholeGenomeExamples as the query sequence folder.
3. Press "add to data pool", this should load several items into the pool list, there should be nine files.

4. Set the Chapter5_6_8_wholeGenomeExamples as the output folder.
5. The BLAST options box should be left blank.



6. Click next

PRO TIP 6: *Users can add individual files to the data pool too.*

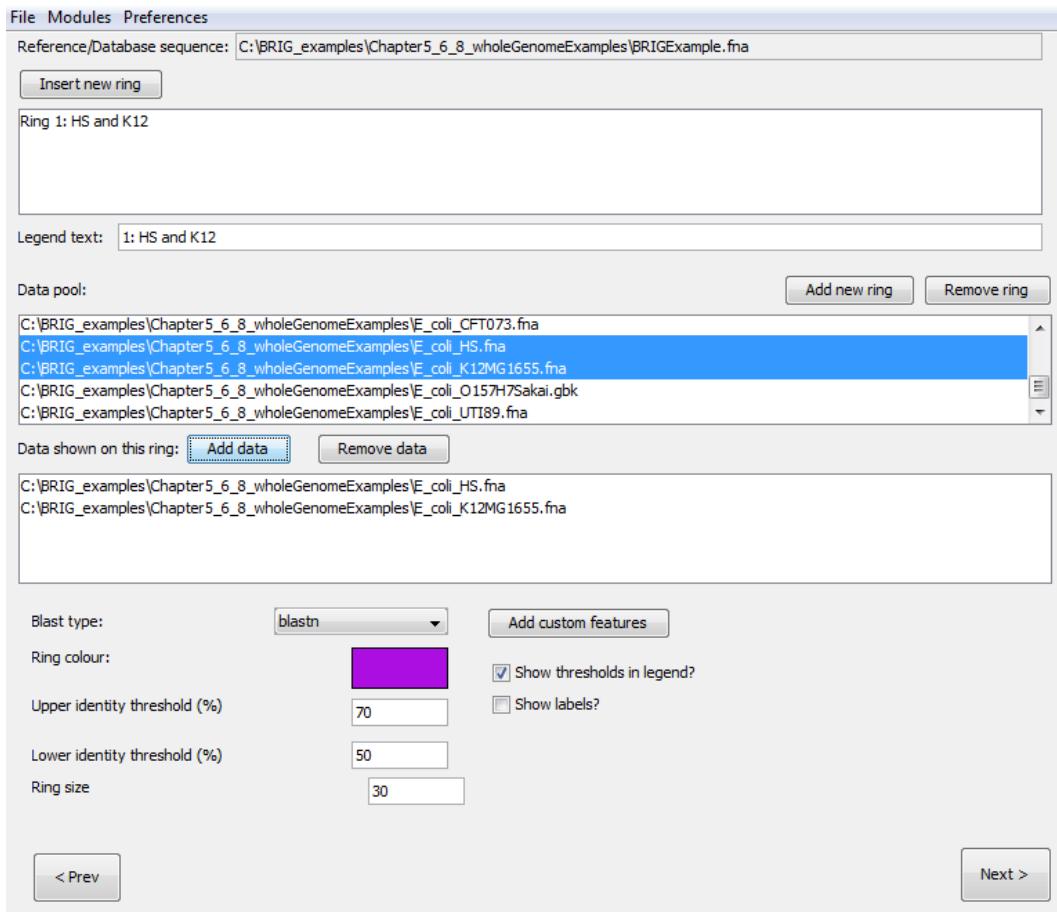
5.2 Step 2: Configure rings

The next step is to configure what information is shown on each of the concentric rings in BRIG. Create six rings, for each ring:

1. Set the legend text for each ring
2. Select the required sequences from the data pool and click on “add data” to add to the ring list.
3. Choose a colour
4. Set the upper (90) and lower (70) identity threshold.

5. Click on “add new ring” and repeat steps for each new ring required.

The values required for each ring are detailed in the table below. Notice that that



sequences can be collated into a single ring, like the example of K12 & HS. The ring will show BLAST matches from both HS and K12.

Legend text	Required sequences	Colour
GC Content	GC Content	Ignore
GC Skew	GC Skew	Ignore
Coverage	BRIGExample.graph	153,0,0
O157:H7	E.coli_O157H7Sakai.gbk	0,0,153
HS and K12	E.coli_HS.fna E.coli_K12MG1655.fna	0,153,0
CFT073 and UTI89	E.coli_CFT073.fna E.coli_UTI89.fna	153,0,153

PRO TIP 7: *Rings can be reordered by dragging them in the Ring List pane.*

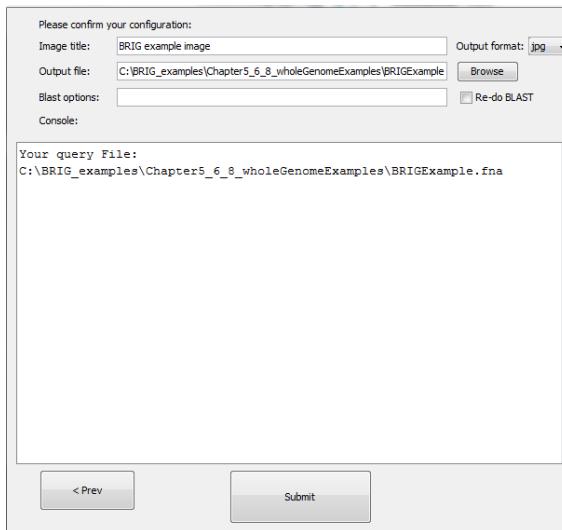
PRO TIP 8: *You can set default threshold values in “BRIG options”. See section 9.2 (page 48) for more details.*

PRO TIP 9: *When using a Genbank/EMBL file as a reference, users can choose whether to use the protein or nucleotide sequence.*

5.3 Step 3: Review and submit

The last window allows us to change the BLAST options, the location of the image file and set the image title, which will appear in the centre of the ring. For the walkthrough configure the third window as:

1. Set the image title as “BRIG example image”.
2. Hit submit.
3. The image will be created in the specified output directory and should look something like Figure 7.



BRIG will format Genbank files, run BLAST, parse the results and render the image. The final image (Figure 7) shows GC Content and Skew, the Genome coverage, contig boundaries, and the BLAST results against the other *E. coli* genomes. The results for HS and K12 have been collated into a single ring, likewise for UTI89 and CFT073.

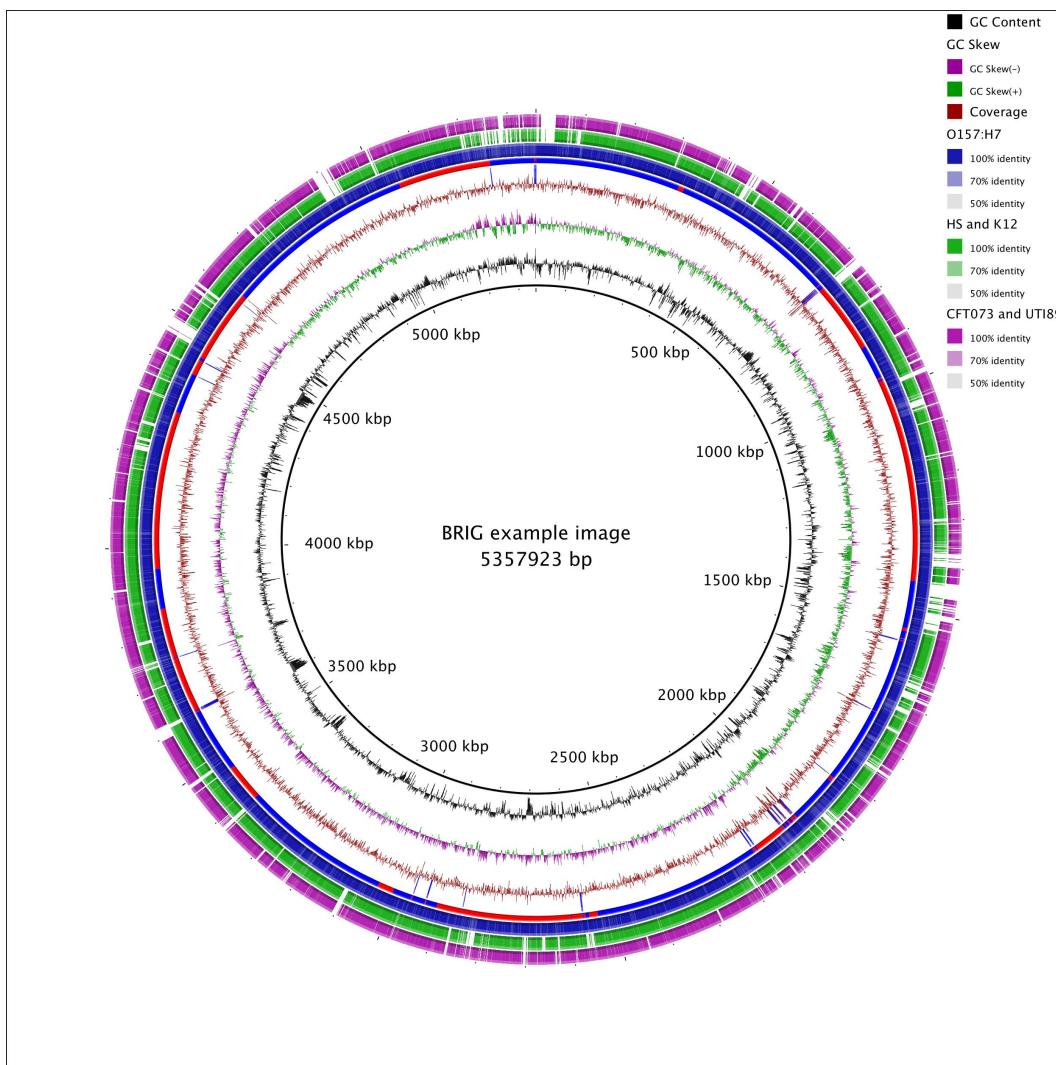


Figure 7: The final BRIG image

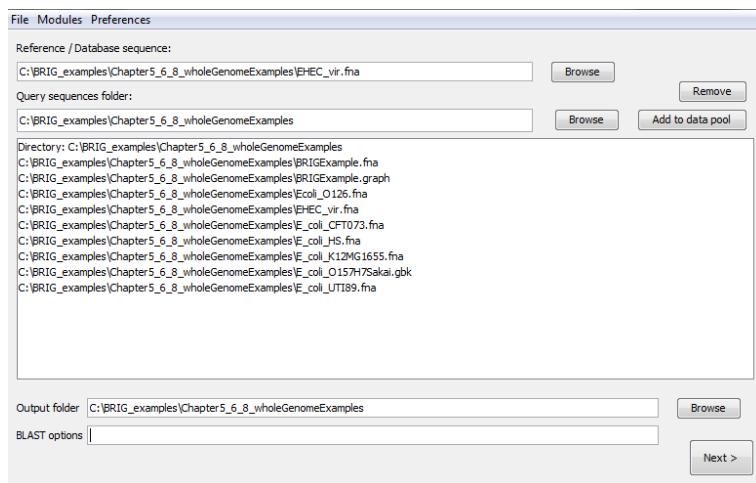
PRO TIP 10: *Image settings, like size, fonts, etc can be configured in: Main window > Preferences > Image options..*

6 Working with a Multi-FASTA reference

6.1 Step 1: Load in sequences

This section is a walk through of how to use BRIG to generate an image using a list of genes in Multi-FASTA format as a reference. The multi-FASTA file in this example is a number of virulence genes from enterohemorrhagic and uropathogenic *E. coli*, which includes EHEC polar fimbriae (*ecpA* to *ecpR*), EHEC Locus of Enteroocyte Effacement (*espF* to *espG*) and the UPEC F1C Fimbrae (*focA* to *focI*), which will be compared against the whole genome sequences of *E. coli* strains O157:H7 Sakai, K12 MG1665, O126:H7 and CFT073. Start a new session in BRIG and load in the files from the Chapter5_6_8_wholeGenomeExamples folder in the unzipped BRIG-Example folder:

1. Set the reference sequence as “Ecoli_vir.fna”. Users can use the browse button to traverse the file system.
2. Set <unzipped BRIG_examples folder>/Chapter5_6_8_wholeGenomeExamples as the query sequence folder.
3. Press “add to data pool”, this should load several items into the pool list.
4. Set the output folder as unzipped BRIG-Example folder.
5. Make sure the BLAST options box is blank.



6. Click “Next”.

6.2 Step 2: Configure rings, annotations and spacer value

The next step is to configure what information is shown on each concentric ring in BRIG. Figure 8 is an example of how one of the windows should be set up. There should be five rings. Do the follow for each ring, according to the table below:

1. Set legend text for each ring.
2. Select the required sequences from the data pool and click “add data” to add.
3. Choose a colour
4. Set upper (90) and lower (70) identity thresholds.
5. Click “add new ring” and repeat these steps for each new ring.

Legend text	Required sequences	Colour
O157:H7	E.coli_O157H7Sakai.gbk	172,14,225
O126:H7	E.coli_O126.fna	255,0,51
CFT073	E.coli_CFT073.fna	0,0,102
K12	E.coli_K12MG1655.fna	161,221,231
null	none	ignore

After each ring is configured, users need to make the following changes:

1. Set the spacer field to 50 base pairs.
2. Set the ring size of ring 5 as “2”.

PRO TIP 11: *The Spacer field determines the number of base pairs to leave between FASTA sequences.*

The next step is to add the gene annotations, which will be fetched from the Multi-FASTA headers:

1. Click Add custom features in the second BRIG window to bring up custom annotation window (Figure 9).
2. Double click “Ring 5”.
3. Set “input data” as Multi-FASTA.
4. Set “colour” as alternating red-blue
5. Click add.

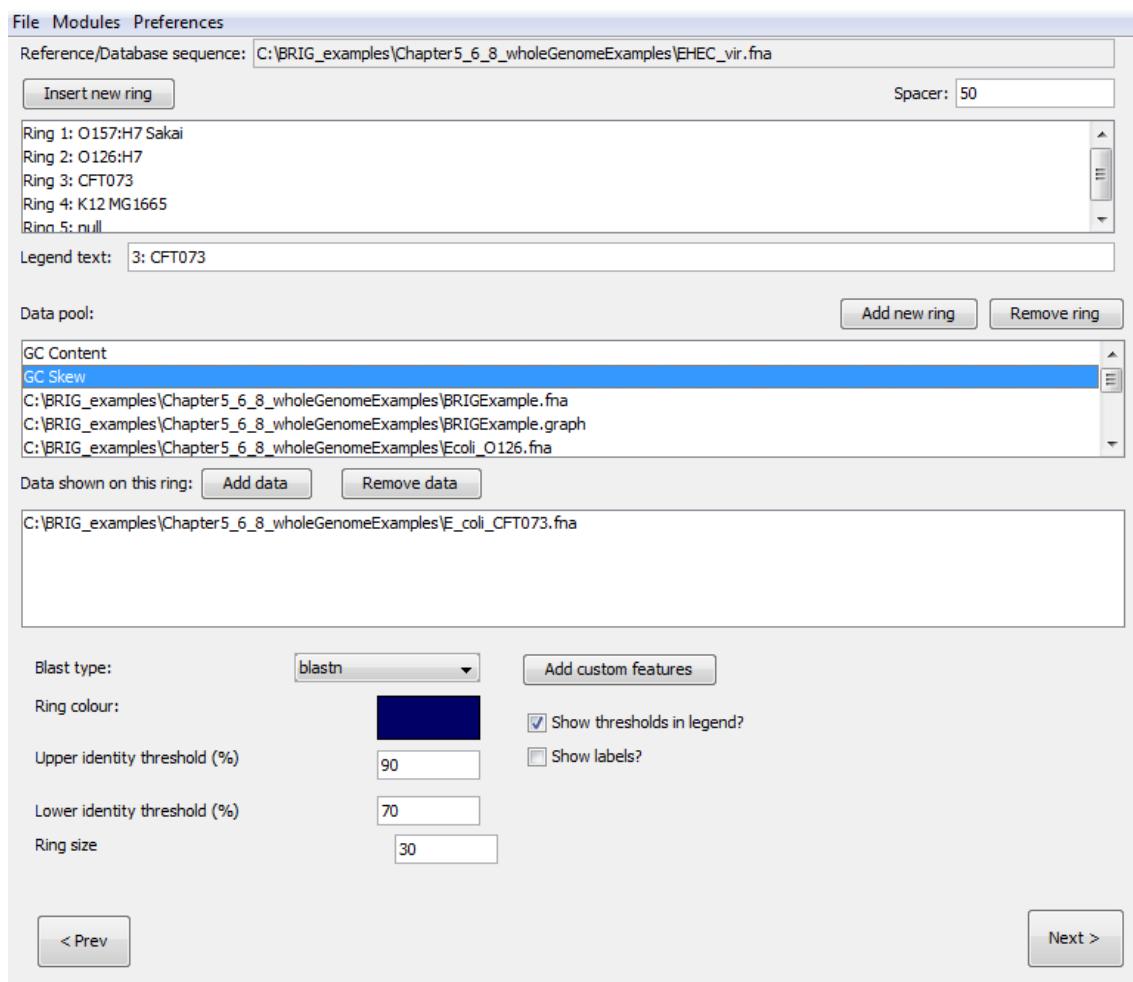


Figure 8: Ring set-up for Multi-FASTA file

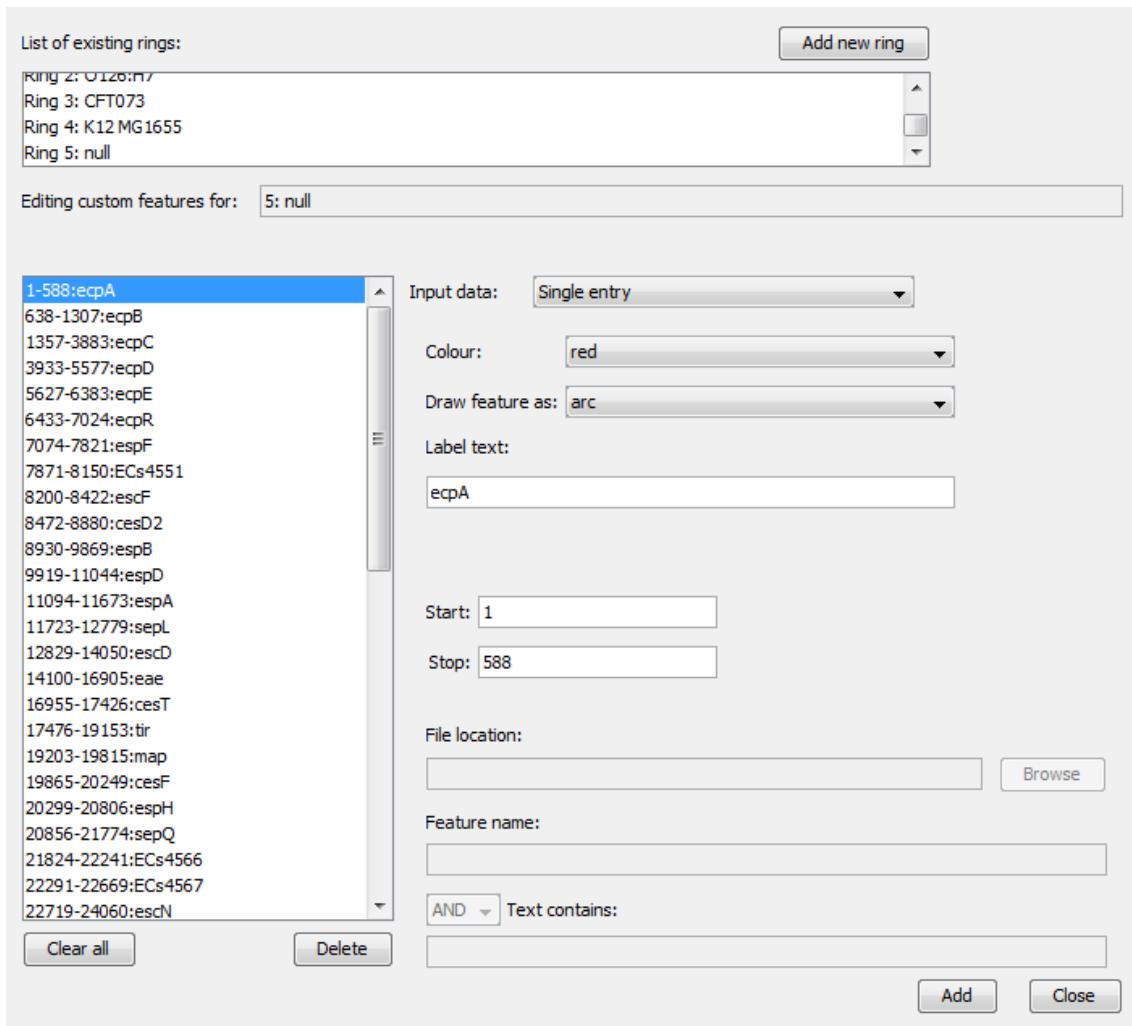


Figure 9: Custom annotation window - adding gene annotations

This step colours the gaps between FASTA entries, the gaps are calculated from the Multi-FASTA file (Figure 10). For each genome ring, do the following:

1. Set “input data” as Multi-FASTA.
2. Set “colour” as black
3. Check “load gaps only”.
4. Click add.

The results should be similar to Figure 10 in the left hand pane. Close the window when this is done.

PRO TIP 12: A spacer value can be set when using protein sequences from a Genbank/EMBL file.

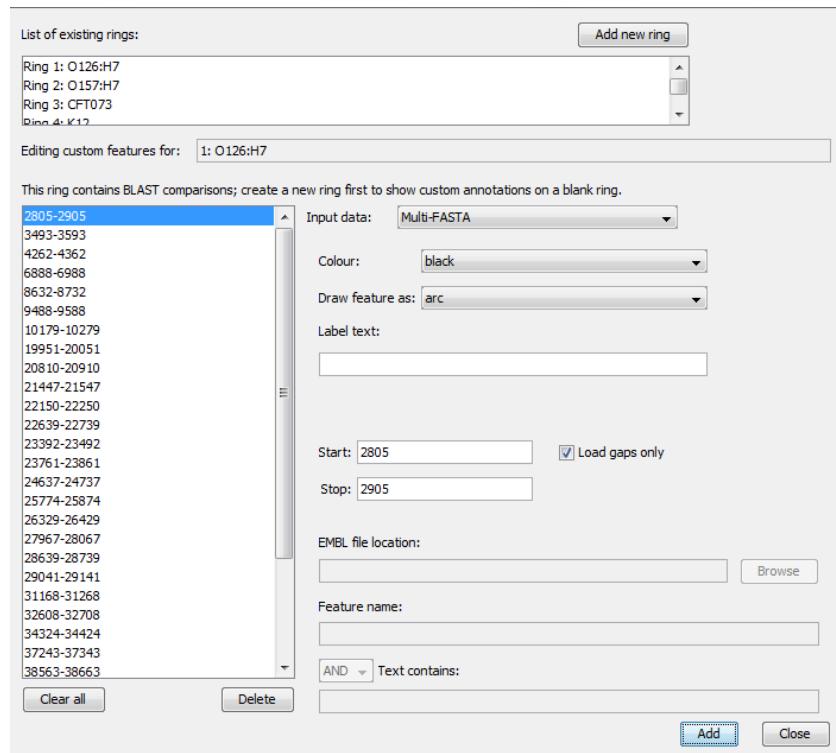


Figure 10: Custom annotation window - adding spacers

6.3 Step 3: Configure image settings and submit

There are a few more steps to complete and then the image is finished. In the customize ring window:

1. Make the following changes in **Preferences >Image options**
 - (a) Set “show shading” in “Global settings” as false.
 - (b) Set “featureSlot” spacing in “Feature settings” as x-small.
2. Return to customize ring window, click “Next” to go to the final BRIG confirmation window
3. Set the image title as “Various E. coli virulence genes” and press submit.

The output image should be something like Figure 11. The alternating red-blue option has automatically alternated the red and blue colours for the gene labels. This option is available whenever a multi-FASTA file is used as a reference sequence. This same option could be used to show contig or genome scaffold boundaries. This image shows some real biological information very clearly.

1. CFT073 (UPEC) and K12 MG1655 (Commensal) do not carry the Locus of Enterocyte Effacement. These virulence factors are specific to EHEC and EPEC.
2. All *E. coli* shown carry the common pilus (*tcpA-R*).
3. Only CFT073 carries the F1C fimbriae.

PRO TIP 13: You use protein sequences as a multi-FASTA reference and use *blastx* to improve alignment accuracy for divergent sequences.

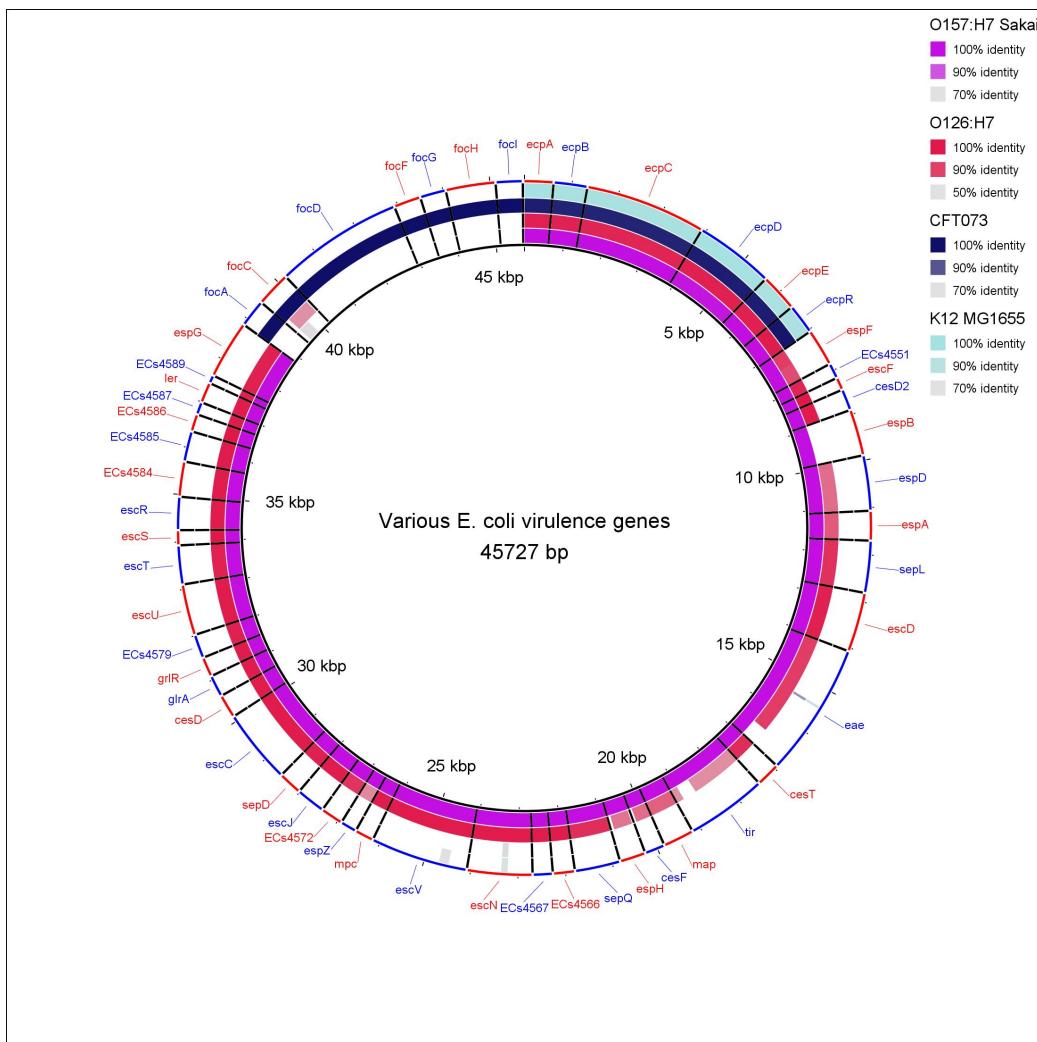


Figure 11: Output image from Multi-FASTA walkthrough. This was generated using BLAST+[2], BLAST legacy[1] will produce slightly different results.

7 Visualising graphs and genome assemblies

BRIG can produce any user-specified graph e.g coverage, read mapping, expression data etc. For example, the coverage graph in Figure 2 was produced from a tab-delimited text file, with a start, stop and value for that range.

BRIG supports .ace files (produced by Newbler, 454/Roche's propriety assembler, and used by PHRAP/Consed) or SAM files (used for read mapping and some de-novo assemblers). BRIG has a number of modules for handling assembly information. These tools are:

- **Contig mapping:** BRIG will use BLAST to try and map contigs from an .ace or Multi-FASTA file onto a reference genome and produce a .graph file that can show frequency of BLAST hits and the best BLAST hit position of contigs. It will then produce a .graph file of the frequency of BLAST hits and the best BLAST hit position of contigs and another .graph file, with the suffix "rep.graph" showing all the other BLAST hits.
- **Coverage graph:** BRIG requires an .ace or .sam file and an output location (Figure 7.1). BRIG will calculate coverage values over a user-defined window and produce a .graph file in the output folder. This will create a tab-delimited .graph file, which can be loaded into back BRIG.
- **Convert graph:** A draft genome is usually modified post assembly; adding spacers, reordering contigs, etc. These changes are often not reflected in the original .ace files BRIG uses to generate coverage graphs. BRIG can use BLAST to align the original assembly output with the newer sequence and map the coverage information to the new sequence. BRIG requires:
 - Original 454AllContigs.fna produced by Newbler.
 - Graph file created by BRIG's "Coverage graph" module, based on Newbler's ace file.
 - The modified sequence or another suitable reference genome.

BRIG will produce a new .graph file in the output folder, using the filename of the original file, with "new.graph" appended to the end.

To create work with graph files: **Main window >Modules >Create graph files**

Using graph files in BRIG images

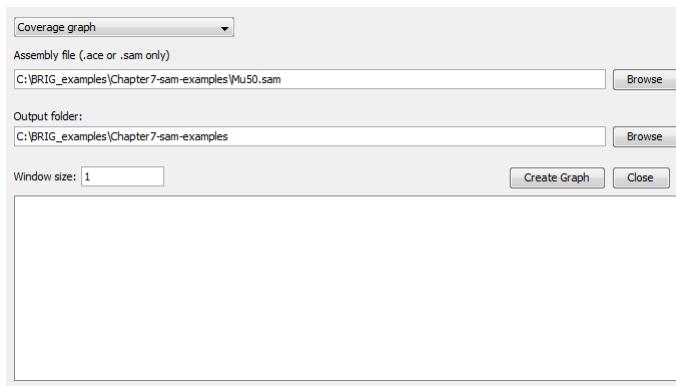
.graph files should visible when users load a directory into the query sequence pool (Figure 7.1). Graphs can be treated like any other sequence file in BRIG; the example from Figure 7.1 shows a graph file loaded into the first ring of a particular BRIG session.

PRO TIP 14: *Graph files cannot be shown on same ring as sequence files (protein or nucleotide).*

7.1 Walkthrough for visualising SAM file mapping coverage.

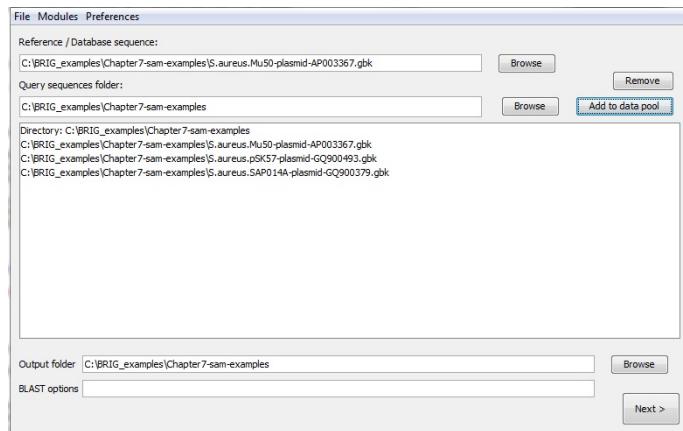
This section will give a worked example of producing a BRIG image showing mapping reads coverage from a SAM input file. The final image will look like Figure 12. This walkthrough requires BRIG_examples.zip from the BRIG website (<http://sourceforge.net/projects/brig/files/>). Unzip this somewhere convenient. The general procedure is to first generate the graph files from the SAM file, add additional files to data pool, edit the rings and annotation, then render the image.

1. Open a new BRIG session.
2. Create the graph file from the graph files modules: **Main window >Modules >Create graph files.**
 - (a) Set drop down to coverage graph, fill in fields (Figure 7.1).
 - (b) Set Assembly file as “Mu50.sam” from the BRIG_examples/Chapter7-sam-examples folder.
 - (c) Set Output folder as the location of the Chapter7-sam-examples folder.
 - (d) Window size as “1”..
 - (e) Click Create Graph. This will add the graph file to the data pool when it has finished.



Close the coverage graph window and return to the first main window.

1. Set reference file as “S.aureus.Mu50-plasmid-AP003367.gbk” from the Chapter7-sam-examples folder. Users can use the browse button to traverse the file system.
2. Set <unzipped BRIG_examples folder>/Chapter7-sam-examples as the query sequence folder.
3. Press “add to data pool”, this should load several items into the pool list.
4. Set the output folder as unzipped Chapter7-sam-examples folder.
5. Make sure the BLAST options box is blank.



Click next to move to the next window to configure the rings and add in annotations.

1. Create 4 rings, name them “Mapping Coverage”, “pSK57”, “SAP014A”, “CDS”.
2. Ring 1 Settings:
 - (a) Add “Mu50.sam.graph” from data pool to Ring 1.
 - (b) Set graph maximum value as “10”.
 - (c) Set colour as rgb(204,0,0).
 - (d) Set legend title as “Mapping coverage”.
 - (e) Check show red/blue.

3. Ring 2 Settings:

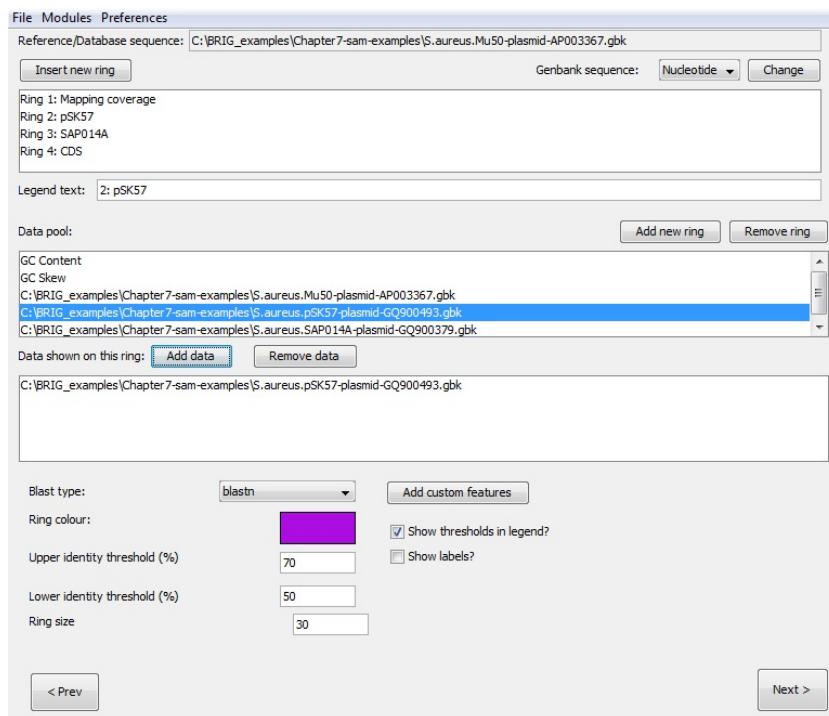
- (a) Add “S.aureus.pSK57-plasmid-GQ900493.gbk” from data pool to Ring 2.
- (b) Set colour as $\text{rgb}(0,0,102)$.
- (c) Set legend title as “pSK57”.

4. Ring 3 Settings:

- (a) Add “S.aureus.SAP014A-plasmid-GQ900379” from data pool to Ring 3.
- (b) Set colour as $\text{rgb}(102,0,102)$.
- (c) Set legend title as “SAP014A”.

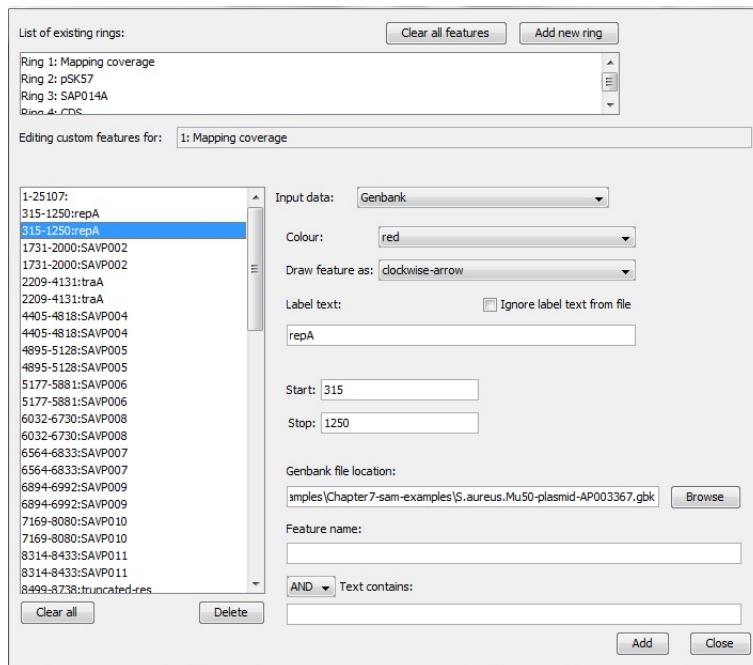
5. Ring 4 Settings:

- (a) Set colour as $\text{rgb}(0,0,0)$.
- (b) Set legend title as “CDS”.



Click “Add Custom features”.

1. Double-click Ring 4.
2. Set Input data to “Genbank”.
3. Set colour to “black”.
4. Set Draw feature as “default”.
5. Set Genbank file location to the location of “S.aureus.Mu50-plasmid-AP003367.gbk”.
6. Set Feature as “CDS”
7. Click add.



This will load all the coding sequences from the Genbank file. These annotations will be drawn as arrows, indicating orientation. Close this window and click next on the second window.

1. Set title as “S. aureus Mu50 plasmid”.
2. Click Submit.

This will generate the final image, it should look like Figure 12.

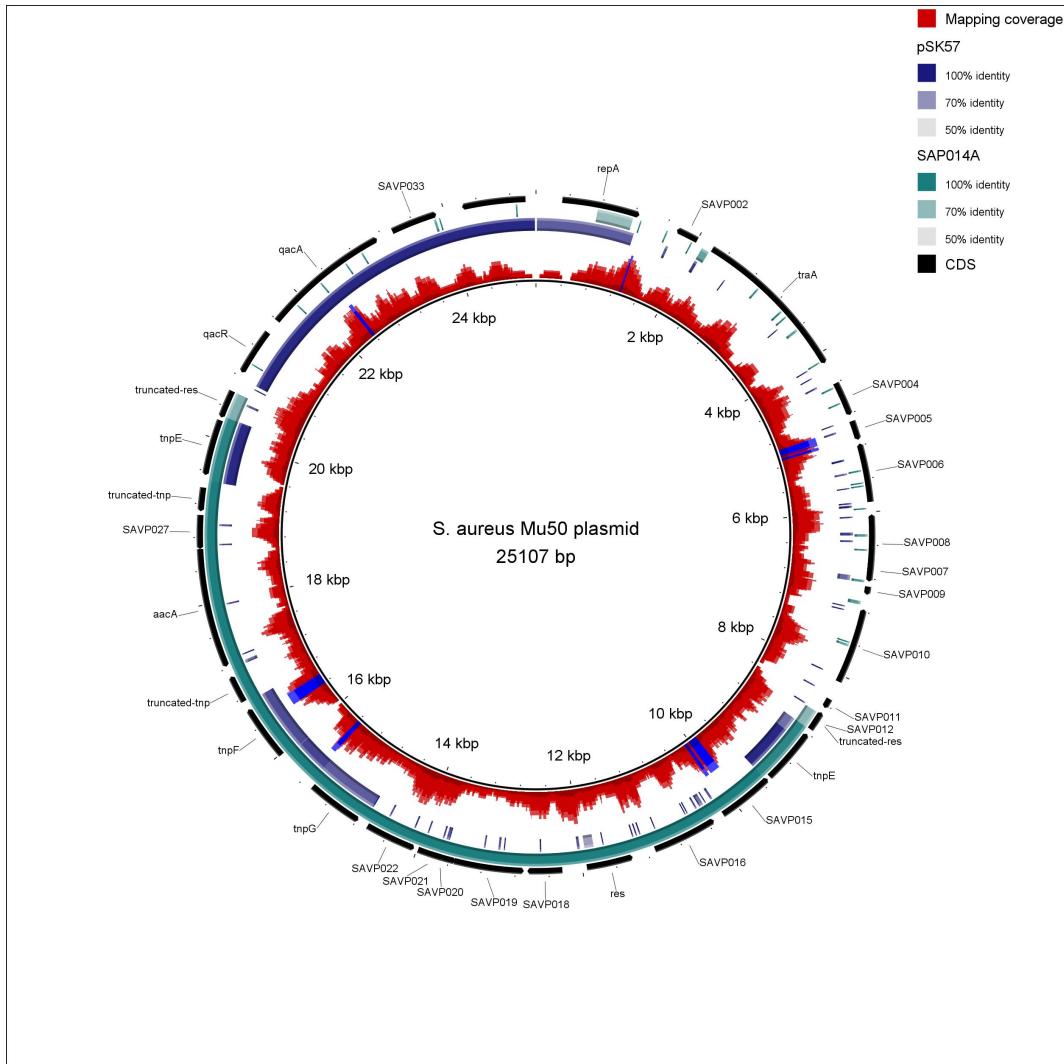


Figure 12: *S. aureus* Mu 50 plasmid, showing read mapping from simulated 454 reads, CDSs, and genome comparisons to other *S. aureus* plasmids, pSK57 & SAP014A. Alignments were performed with BLAST+

7.2 Walk through for visualising ace file assembly coverage.

This section will give a worked example of producing a BRIG image showing assembly coverage read from an ace file. The final image will look like Figure 13. This walk through requires BRIG_examples.zip from the BRIG website (<http://sourceforge.net/projects/brig/files/>). Unzip this somewhere convenient.

The general procedure is to first generate the graph files from the ace file, convert the coverage information to reference sequence if necessary, add additional files to data pool, edit rings and annotation, then render the image.

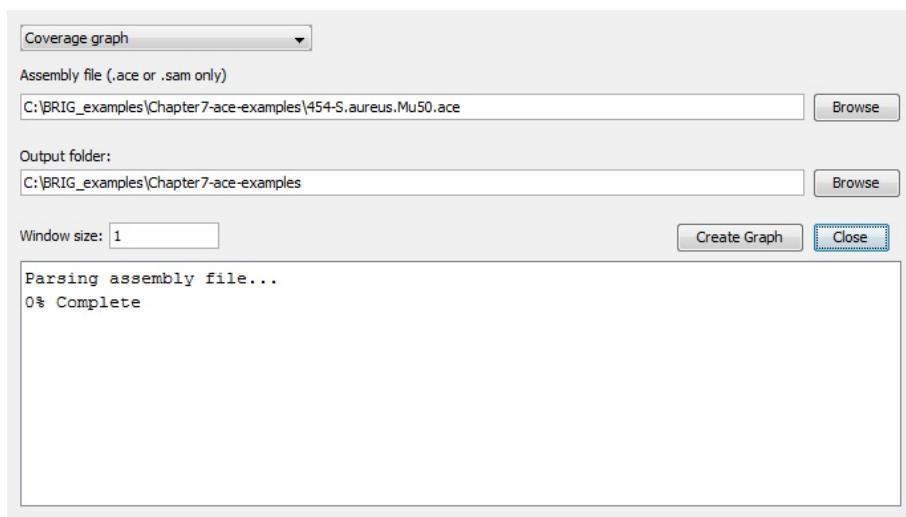
Draft genome sequences are often modified to be consistant with other information (e.g genome scaffolding, PCR sequencing of gaps) after being initially assembled. This may change the order and size of the final genome sequence compared to the original assembly.

To show the read coverage from the assembly on the final sequence correctly the “Convert graph” module within BRIG can be used to map the coverage information from the ace file onto the new sequence.

This module can also be used to map read coverage from an assembly onto a closely-related reference genome.

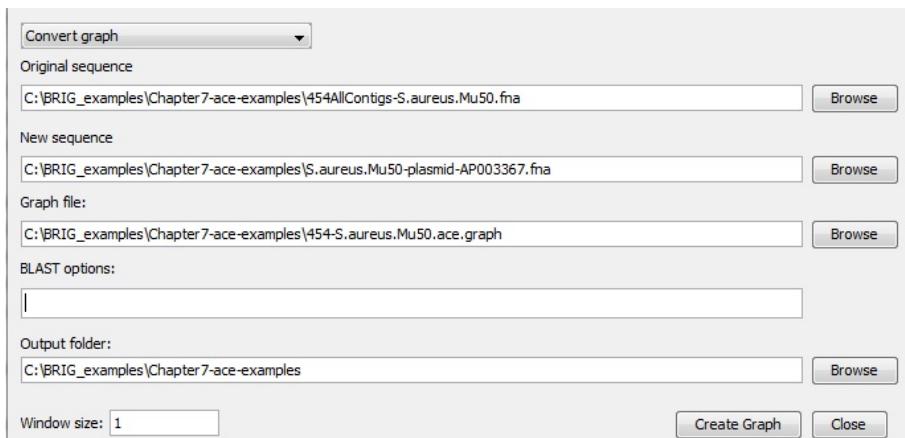
First, produce the coverage graph file based off the assembly (ace file):

1. Open a new BRIG session.
2. Create the graph file from the graph files module: **Main window >Modules >Create graph files.**
3. Set drop down to coverage graph, fill in fields (Figure 7.2).
 - (a) Set Assembly file as “454-S.aureus.Mu30.ace” from the BRIGEXAMPLE2-ace folder.
 - (b) Set Output folder as the location of the BRIGEXAMPLE2-ace folder.
 - (c) Set window size as “1”.
4. Click “Create Graph”. This will add the graph file to the data pool when it has finished.



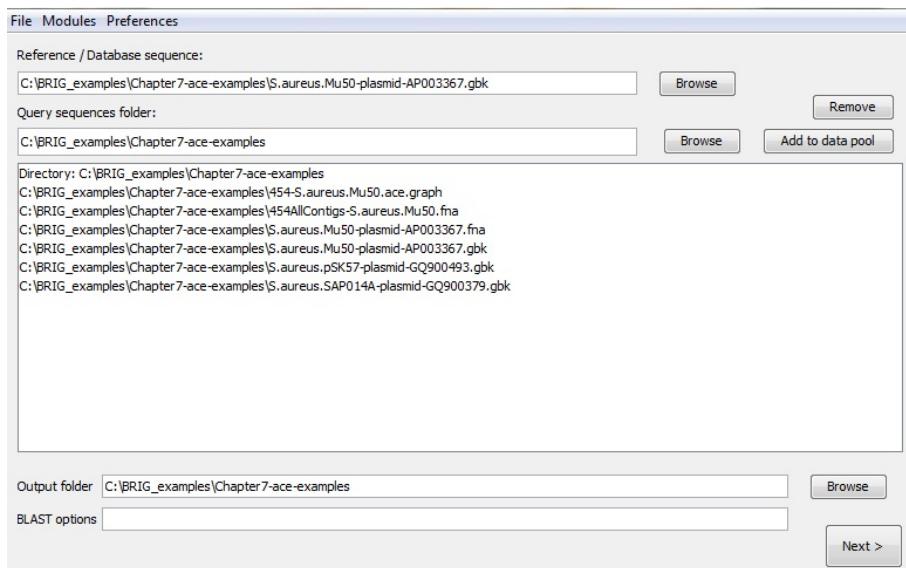
Next, map the coverage generated in the previous graph file to the modified genome sequence.

1. Remain in the “Create custom graph” window: **Main window >Modules >Create graph files.**
2. Set drop down to convert graph, fill in fields as below.
 - (a) Set Original sequence as “454AllContigs-S.aureus.Mu50.fna”.
 - (b) Set New sequence as “S.aureus.Mu50-plasmid-AP003367.fna”.
 - (c) Set graph file as “454-S.aureus.Mu50.ace.graph”.
 - (d) Set Output folder as the location of the BRIGEXAMPLE2-ace folder.
 - (e) Window size as “1”.
3. Click “Create Graph”. This will add the graph file to the data pool when it has finished.



Close the “Create custom graph” window and return to the main window.

1. Set reference file as “S.aureus.Mu50-plasmid-AP003367.gbk” from the BRIGEXAMPLE2-ace folder. Users can use the browse button to traverse the file system.
2. Set <unzipped BRIGEXAMPLE2-ace folder>/genomes as the query sequence folder.
3. Press “add to data pool”, this should load several items into the pool list.
4. Set the output folder as unzipped BRIGEXAMPLE2-ace folder.
5. Make sure the BLAST options box is blank.



Click next to move to the next window and configure the rings.

1. Create 4 rings.

2. Ring 1 Settings:

- (a) Add “454AllContigs-S.aureus.Mu50.fnanew.graph” to Ring 1.
- (b) Set colour as `rgb(204,0,0)`.
- (c) Set legend title as “Assembly coverage”.
- (d) Set graph maximum value as “700”.
- (e) Check show red/blue.

3. Ring 2 Settings:

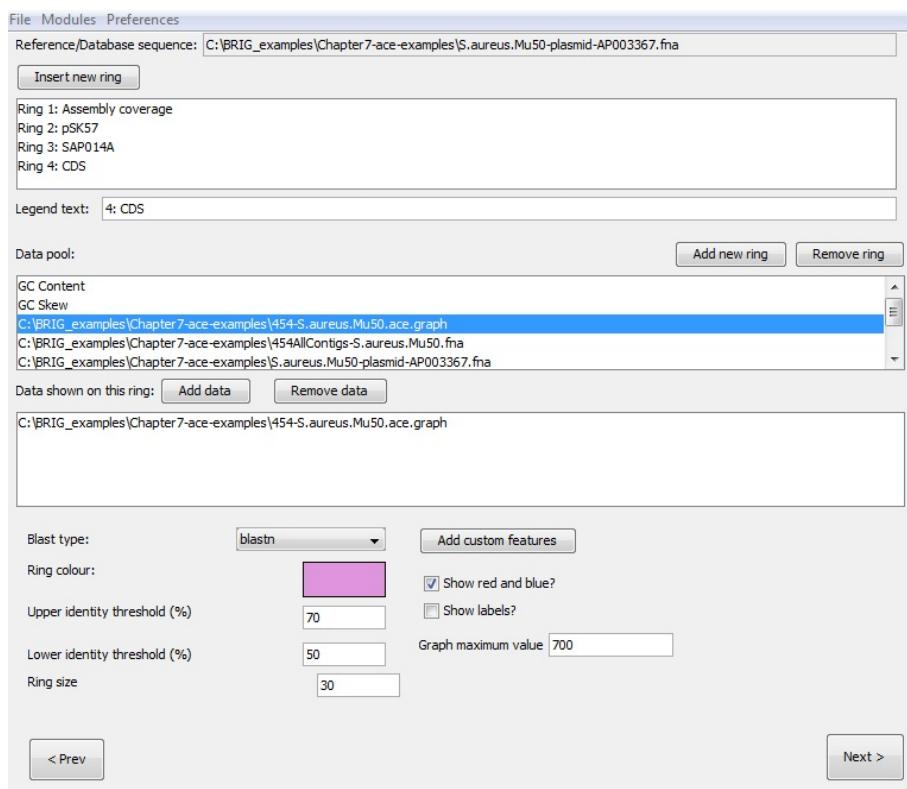
- (a) Add “S.aureus.pSK57-plasmid-GQ900493.gbk” to Ring 2.
- (b) Set colour as `rgb(0,0,102)`.
- (c) Set legend title as “pSK57”.

4. Ring 3 Settings:

- (a) Add “S.aureus.SAP014A-plasmid-GQ900379” to Ring 3.
- (b) Set colour as `rgb(102,0,102)`.
- (c) Set legend title as “SAP014A”.

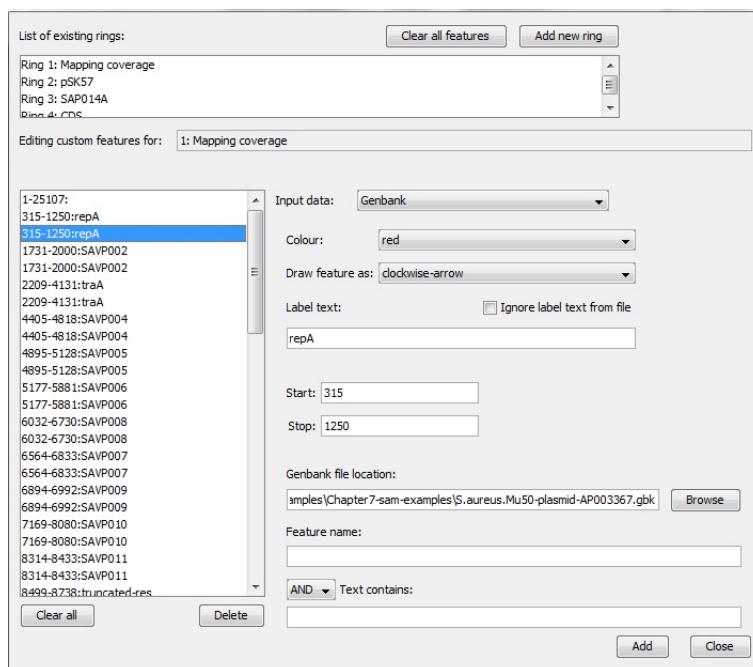
5. Ring 4 Settings:

- (a) Set colour as `rgb(0,0,0)`.
- (b) Set legend title as “CDS”.



The rings are now step up with the correct colour, data and labels. The next step is to mark the CDS on Ring 4 as “custom features”. Click “Add Custom features” to open a new window.

1. Double-click on Ring 4.
2. Set Input data to “Genbank”.
3. Set colour to “black”.
4. Set Draw feature as “default”.
5. Set GenBank file location to “S.aureus.Mu50-plasmid-AP003367.gbk” in the BRIGEXAMPLE2-ace folder.
6. Set Feature as “CDS”
7. Click add.



This will load all the coding sequences from the GenBank file. These annotations will be drawn as clockwise or counter-clockwise arrows, indicating orientation. Close this window and click next on the main window. From the third/confirmation window:

1. Set title as “S. aureus Mu50 plasmid”.
2. Click submit.

This will generate the final image, which should look like Figure 13.

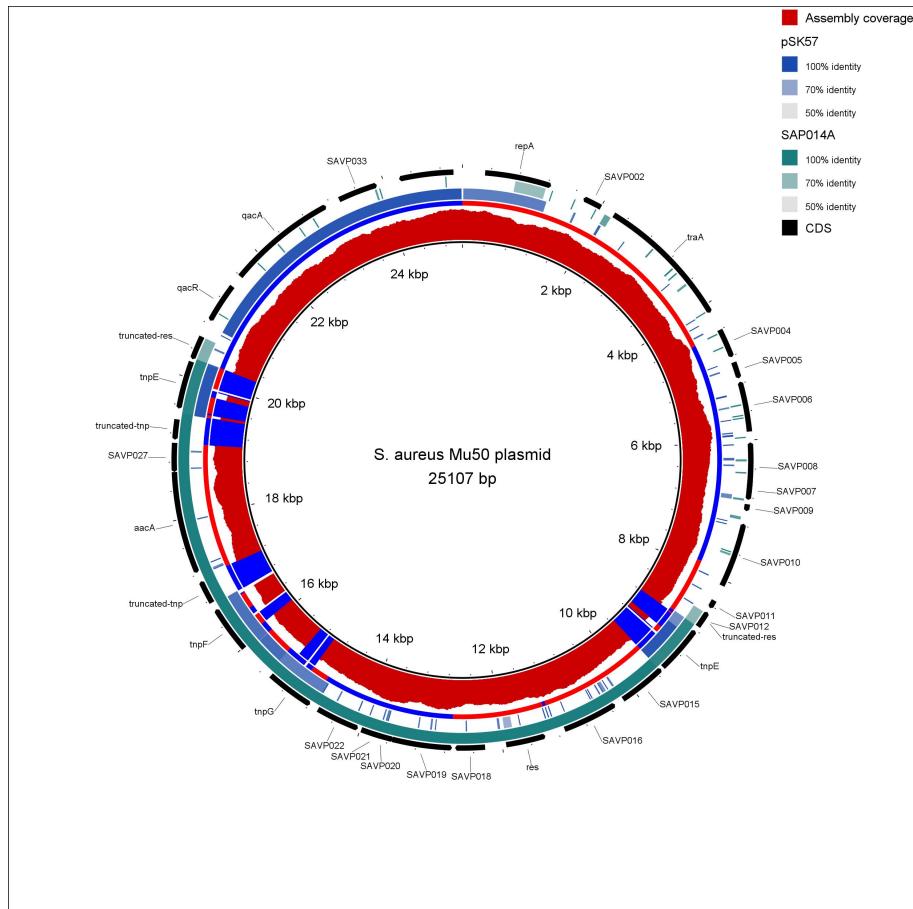


Figure 13: *S. aureus* Mu 50 plasmid published genome sequence, showing assembly coverage from a draft assembly based on simulated 454 reads. The image also shows CDS, and genome comparisons to other *S. aureus* plasmids, pSK57 & SAP014A. Alignments were performed with BLAST+.

8 Walkthroughs on creating custom annotations

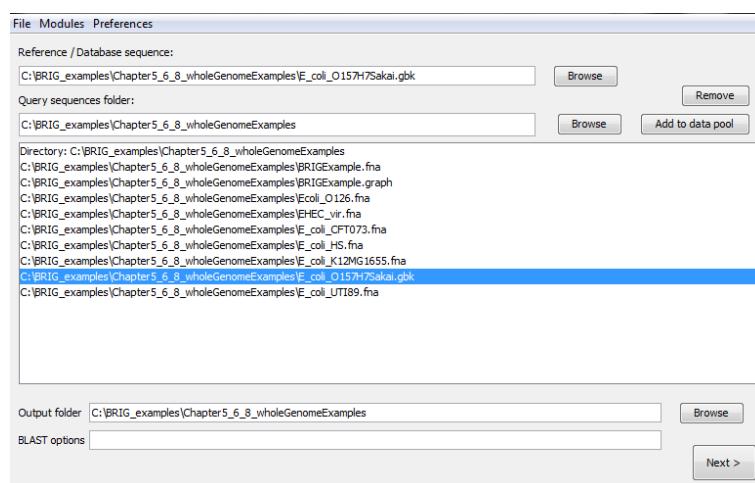
8.1 Adding custom annotations from a tab-delimited file, GenBank or EMBL file

BRIG can add features from a tab-delimited file, Multi-FASTA or GenBank/EMBL file. This section deals with GenBank/EMBL & tab-delimited files. A walkthrough for generating annotations from a Multi-FASTA file can be found in Section 6.2 on page 19.

8.1.1 Step 1: Load in sequences

The walk through will work out of the unzipped BRIG_examples.zip folder. The walk through and related figures will use C:\BRIG_examples\Chapter5_6_8_whole GenomeExamples as that location.

1. First, set “E_coli_O157H7Sakai.gbk” as the reference sequence from the BRIGEXAMPLE folder.
2. Set <unzipped BRIG_examples folder>\Chapter5_6_8_whole GenomeExamples as the query sequence folder.
3. Press “add to data pool”, this should load several items into the pool list.
4. Set the Chapter5_6_8_whole GenomeExamples folder as the output folder.
5. The BLAST options box should be left blank.



6. Click next

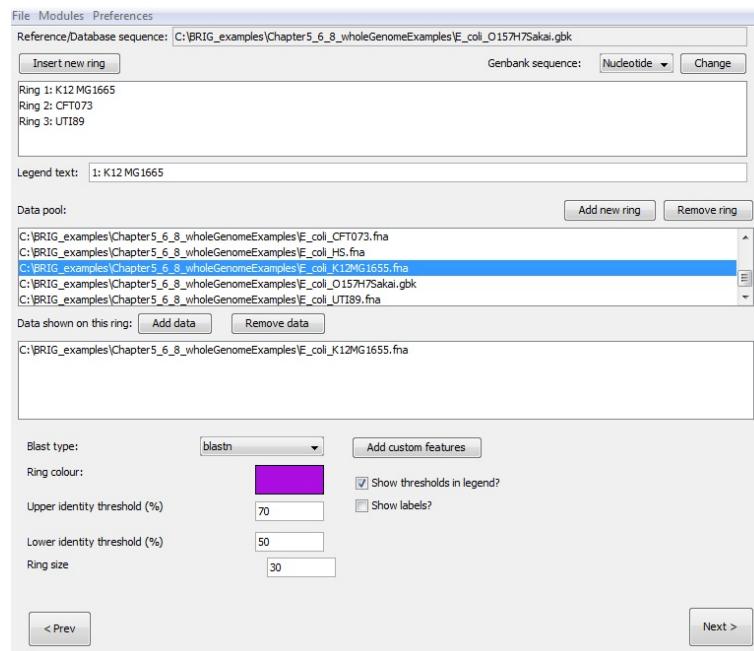
8.1.2 Step 2: Configure rings

The next step is to configure what information is shown on each of the concentric rings in BRIG. There should be three rings, for each ring:

1. Set the legend text for each ring
2. Select the required sequences from the data pool and click on “add data” to add to the ring list.
3. Choose a colour
4. Set the upper (70) and lower (50) identity threshold.
5. Click on “add new ring” and repeat steps for each new ring required.

The values required for each ring are detailed in the table below.

Legend text	Required sequences	Colour
K12 MG1665	E_coli_K12MG1655.fna	0,153,255
CFT073	E_coli_CFT073.fna	204,0,255
UTI89	E_coli_UTI89.fna	153,0,102

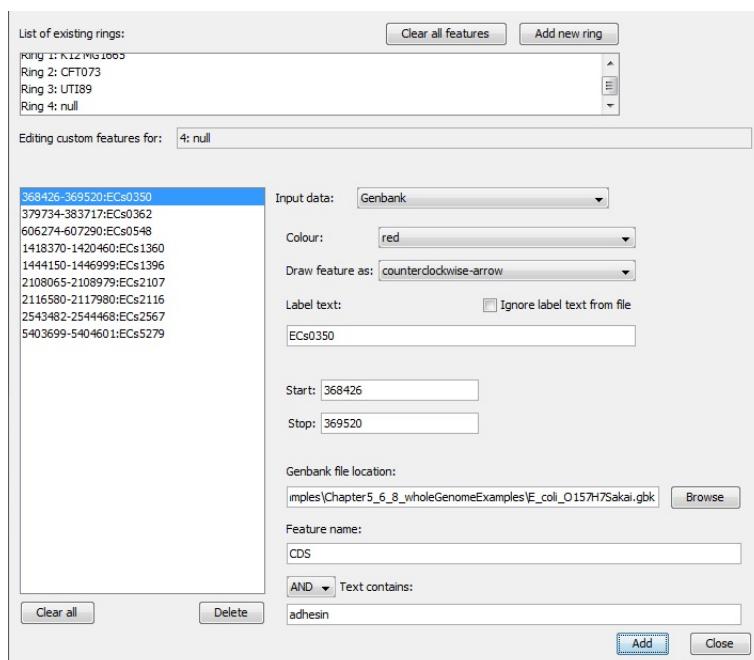


8.1.3 Step 3: Adding annotations

Adding annotations from GenBank files

This step adds annotations read from a Genbank/EMBL file. These files can be searched for a specific feature type containing a specific value. This walk through searched the *E. coli* O157:H7 Sakai genome for all CDSs that are annotated as adhesins. From the “Configure Rings” window, click “Add Custom features”. These are the steps to load the GenBank features:

1. Create a new ring (Ring 4) and double-click Ring 4 in the ring list.
2. Set “input data” to Genbank.
3. Set “colour” to red.
4. Leave “Draw feature as” default.
5. Leave label text as a blank.
6. Set Genbank file location as Chapter5_6_8_whole GenomeExamples folder\
E_coli_O157H7Sakai.gbk
7. Set “Feature name” as CDS
8. Set “And” as AND
9. Set “Text contains” as adhesin.
10. Click add.
11. Remain in the “Add Custom features” for the next step.



PRO TIP 15: *The same process applies for EMBL files.*

By default, CDS on the sense strand will be drawn as clockwise arrows and counter-clockwise arrows for anti-sense. All other features will be shown as block arcs, by default. This can be overriden by changing “Draw feature as” from “default”.

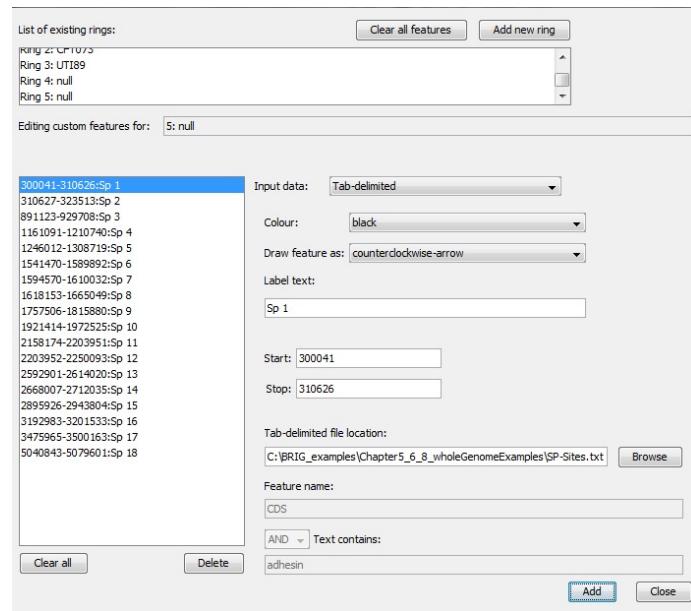
Adding annotations from Tab-delimited files

This step adds annotations of the *E. coli* O157:H7 Sakai prophage site (SP Sites) on to the image, which will be read from a tab-delimited file (SP-Sites.txt, which can be found in the Chapter5_6_8_whole GenomeExamples folder).

To add annotate the image with the SP-Sites:

1. Create a new ring (Ring 5) and double-click Ring 5 in the ring list.
2. Set “input data” as Tab-delimited.
3. Leave “label” blank.
4. Set “colour” to black.

5. Set tab-delimited file location as Chapter5_6_8_wholeGenomeExamples\Sp-Sites.txt
6. Press Add.



If there are a lot of entries, this may take a few seconds. All the values should show up in the left hand pane. Individual changes can be made by double clicking and editing an entry. Users can load any kind of results or annotations into BRIG using this approach.

8.1.4 Step 4: Review and submit

The last window allows us to change the BLAST options, the location of the image file and set the image title, which will appear in the centre of the ring. For the walkthrough we need to configure the third window as:

1. Set the image title as “E. coli O157:H7 Sakai”.
2. Press submit.
3. The image will be created in the specified output directory and should look something like Figure 14.

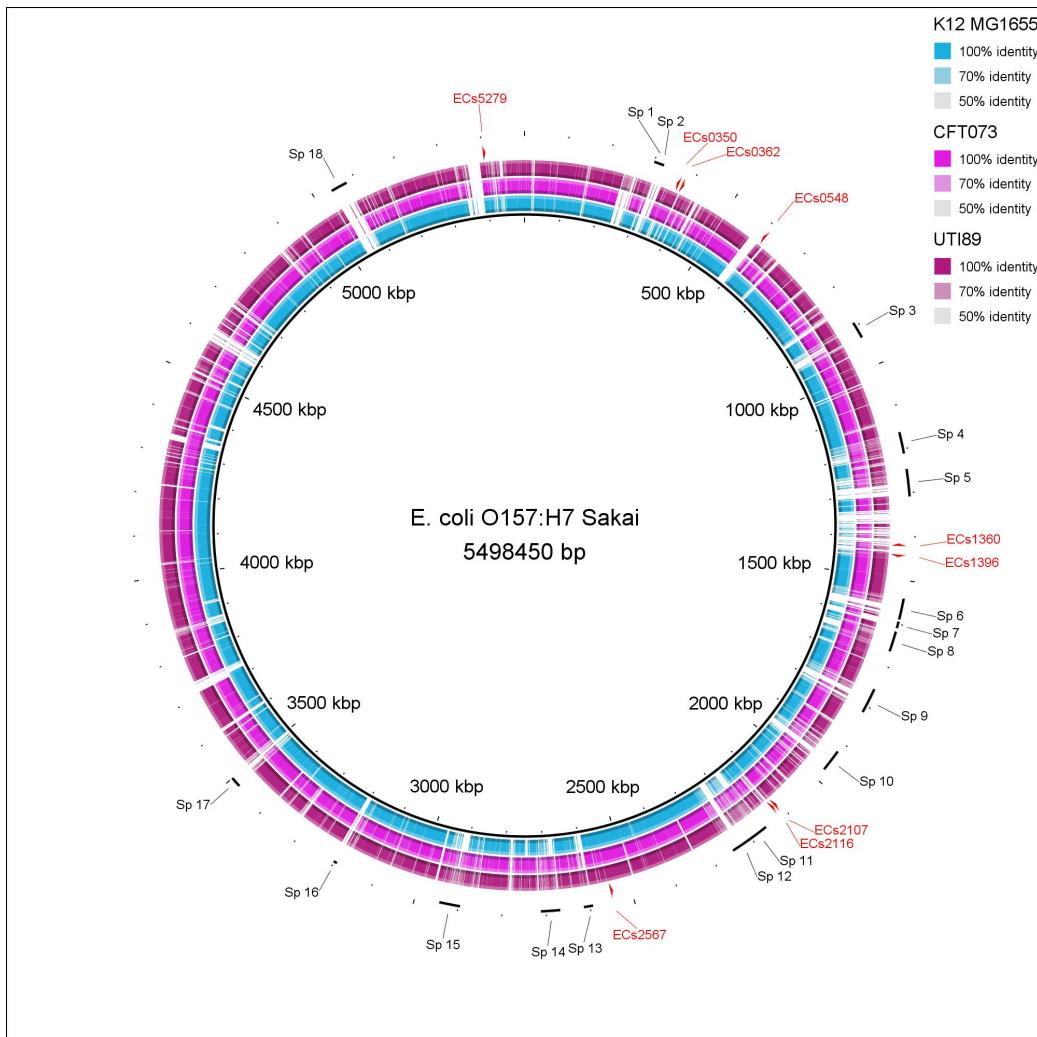


Figure 14: Output image from tab-delimited annotations. This figure shows *E. coli* O157:H7 Sakai as the central reference sequence, with genome comparisons to K12, CFT07 and UTI89 and the SP sites annotated in black.

8.2 How to create tab-delimited files for BRIG

BRIG can load custom annotations or results from a tab-delimited file. A tab delimited file; a plain text file that represents a table, with a tab between each column in the text. Rows are on each line and columns are separated by a tab character (called the delimiter). BRIG only supports tab-delimited files, not other delimiters e.g commas, space etc.

For BRIG to accept custom annotation files, it must fulfill the following requirements:

- Lines that contain column headers, or comments must start with a “#”. See the first line in Figure 8.2.
- The first two columns MUST be Start and Stop values and they must have a value or BRIG will ignore them.
- Specifying Labels, colours or decoration is not necessary but if they must be in the right column. (Label must be 3rd, Colour must be 4th, Decoration must be 5th).
- Acceptable Colour values are: default, red, aqua, black, blue, fuchsia, gray, green, lime, maroon, navy, olive, orange, purple, silver, teal, white, yellow.
- Acceptable Decoration values are default, arc, hidden, counterclockwise-arrow, clockwise-arrow.
- Files must be in plain-text tab-delimited.

The easiest way to make BRIG tab-delimited files is to set up a spreadsheet with exactly the same headers as below (Order and text is important) and then fill in the columns with data, leave the field blank to use default colours and decorations. Leave the label field blank if no label is required.

#START	STOP	Label	Colour	Decoration
2587201	2587211			
599319	599329			
2517947	2517957			
2355144	2355154			
3075341	3075351			
708320	708330			
1466400	1466410			
3693452	3693462			
1168759	1168769			
892710	892720			
3662858	3662868			
946527	946537			
3259147	3259157			
4342306	4342316			
2404991	2405001			
4318955	4318965			
4637001	4637011			

9 Configuration options

9.1 Saving and reopening your work

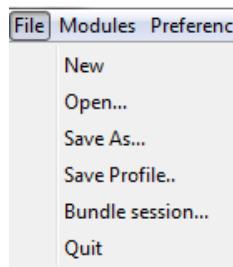


Figure 15: File menu in BRIG

Users can save their work in BRIG from the File menu (Figure 15) and can open these save files from **File menu >Open...**. There are three options for saving:

- “Save As...” will save all the image settings, BRIG settings, rings configuration, files users have imported into BRIG.
- “Save Profile...” will save just the global image settings & BRIG settings like image dimensions and font size and colours. It will not save the data pool or any ring settings. These files are designed to be used as templates for other images.
- “Bundle Session...” will copy all files used in the BRIG session and save all the current settings into a single directory. This directory can be compressed and kept as an archive or sent to someone else who can open the .xml file and work on the image with all of the original settings.

9.2 BLAST options

BRIG will run with default BLAST options if the BLAST options field is left blank. BLAST options field (Figure 16) can be used to add custom BLAST options.

PRO TIP 16: *BRIG handles all the file input and output into BLAST, so DO NOT use -o, -d, -p, -i, -m in BLAST legacy or -out, -db, -query, -outfmt in BLAST+.*

The list below highlights some BLAST parameters for BLAST+[2] and BLAST legacy[1]; more common Parameters are in bold. For more information please read the blastall or BLAST+ command-line usage messages (or <http://www.ncbi.nlm.nih.gov/books/NBK1763/>).

BLAST+ options:

- **-evalue** Expectation value (E) [Real] default = 10.0
- **-dust** or **-seg** Filter query sequence (DUST with blastn, SEG with others) [String] default = yes
- **-task** Task to execute. Permissible values: 'blastn','blastn-short','dc-megablast','megablast','vecscreen'. Default for BLAST+ is 'megablast'. BRIG overrides this by forcing blastn, unless the user specifies otherwise.
- **-num_threads** Number of processors to use [Integer] default = 1
- **-html** Produce HTML output [T/F] default = F
- **-gilist** Restrict search of database to list of GI's [String] Optional

PRO TIP 17: *BRIG runs BLAST+ with task as blastn, unless overridden by the user.*

BLAST legacy options:

- **-e** Expectation value (E) [Real] default = 10.0
- **-F** Filter query sequence (DUST with blastn, SEG with others) [String] default = T
- **-a** Number of processors to use [Integer] default = 1
- **-T** Produce HTML output [T/F] default = F
- **-l** Restrict search of database to list of GI's [String] Optional
- **-W** Word size, default if zero (blastn 11, megablast 28, all others 3) [Integer] default = 0
- **-n** MegaBlast search [T/F] default = F

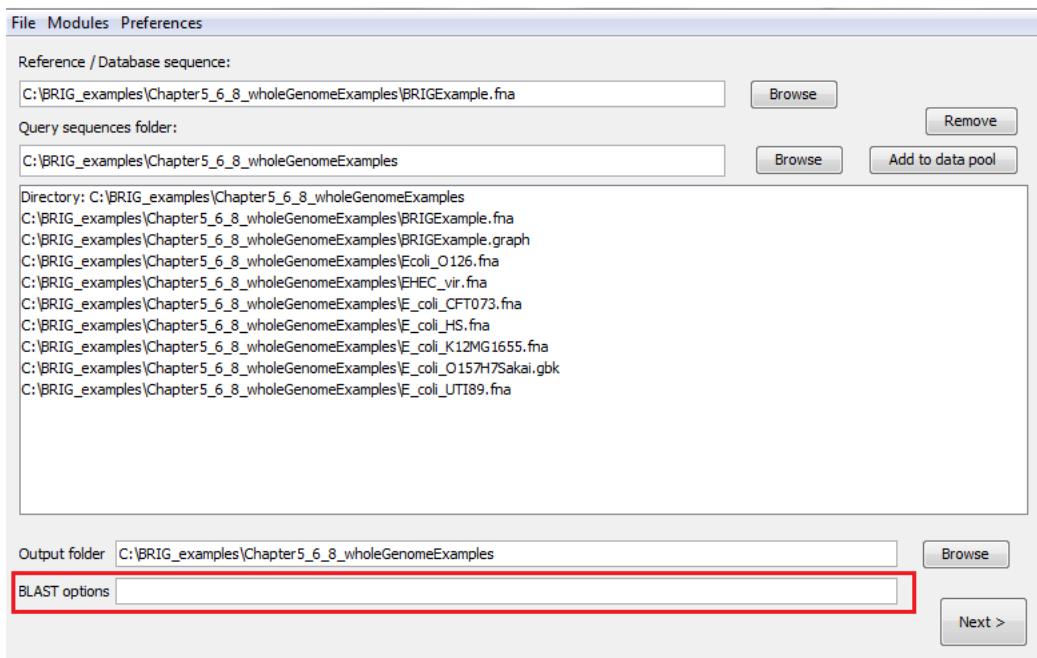


Figure 16: Set custom BLAST options on the first window (shown) or the third window

9.3 Setting BRIG options

The BRIG options window under **Preferences>BRIG options** in the first main screen allows users to set options that include valid Genbank, EMBL or FASTA file name extensions; percentage identity thresholds; and memory allocated to CGView.

Changes can be applied to just the current session through “Save & Close” or changes can be saved as the default settings for every new session in BRIG through “Save settings as default”. There are two BRIG option tabs.

Tab 1 (Figure 17) has the following options:

- **Genbank file extensions:** Users can specify the file extensions they use as GenBank files (with commas between extensions). e.g .gbk.
- **FASTA file extensions:** Users can specify the file extensions they use as FASTA files (with commas between extensions). e.g .fa,.fna,.fas.
- **EMBL file extensions:** Users can specify the file extensions they use as EMBL files (with commas between extensions). e.g .embl.

- **BLAST binary folder:** Users must specify the location of BLAST executables, leave this blank if BLAST is on their PATH.
- **Default spacer space:** Users can set a default spacer value that BRIG uses when using Multi-FASTA files.

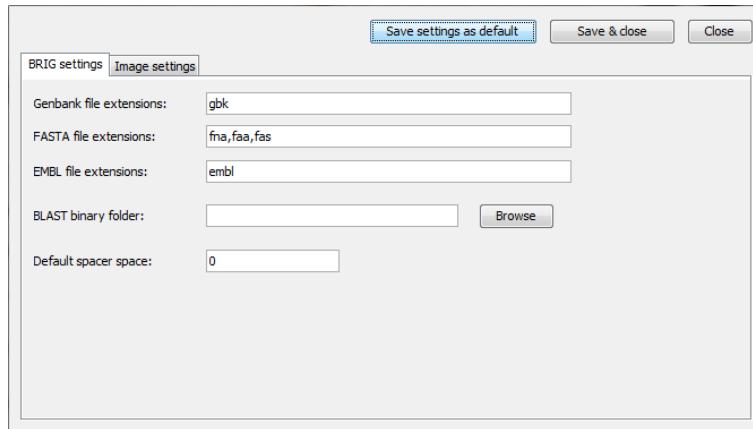


Figure 17: First BRIG options window, accessible from **Preferences >BRIG options**

Tab 2 (Figure 18) has the following options:

- **Graph label thickness:** Graph rings have another smaller ring to show labelled regions, users can set how many times smaller these rings are compared to normal rings here.
- **Graph ring thickness multiplier:** Graph rings need to be larger than other rings, users can set how many times larger Graph rings are compared to normal rings here.
- **Image render memory (MB):** Amount of memory given to CGView for image rendering, if users run out of memory when rendering their image they should increase this value.
- **Default upper & lower threshold:** The colour of BLAST matches in BRIG are shaded in a sliding scale between 100% and the lower threshold percentage identity. These values and their corresponding colour are shown in the legend. The upper threshold is used to highlight a particular percentage

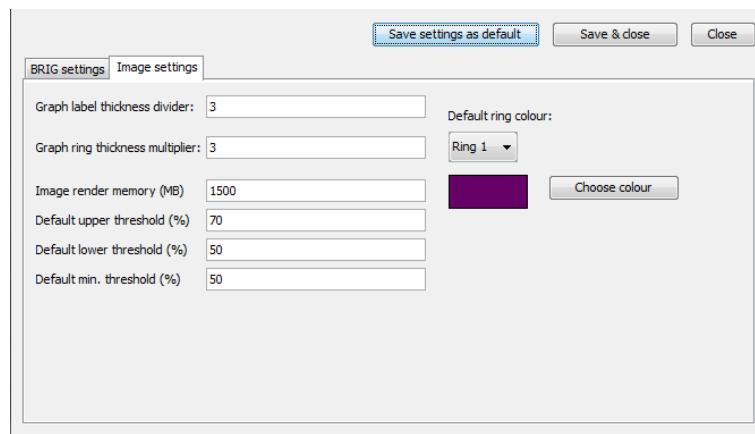


Figure 18: Second BRIG options window, accessible from **Preferences >BRIG options**

identity in the legend and has no bearing on the colour of BLAST matches.

- **Default min. threshold:** Minimum percent identity for BLAST results that BRIG will include on an image.

9.4 Setting Image options

Image rendering is done through CGView[5], these settings can be changed from **Preferences >Image options**. A detailed explanation of these options can be found at http://wishart.biology.ualberta.ca/cgview/xml_cgview.html.

Here is list of some of the more useful options:

- “*Global settings*” tab
 - **Height and Width:** These values change the height and width of image by specifying the number of pixels.
 - **Title font and color:** These values change the font type, size and colour.
 - **Show shading?:** Set to false to turn off the embossing on annotations.

- “*Feature settings*” tab
 - **FeatureSlot spacing:** Sets the amount of gaps between rings.
 - **Feature thickness:** This sets the default thickness of rings on image. This can be overridden by the thickness parameter on ring customization window.
- “*Tick settings*” tab
 - **Tick density:** This value changes how often ticks appears on the ruler, requires a decimal number between 0 and 1.
 - **Long tick colour:** This value changes the colour of long (or short ticks).
- “*Backbone and ruler settings*” tab
 - **Backbone radius:** Sets the radius size of the ring circle. Increase this to make a larger ring and decrease to make it smaller. N.B this will not automatically scale with the Image height and width.
- “*Label settings*” tab
 - **Labels on?**: Global settings for whether feature labels should be drawn on this image.
 - **Global label colour:** Default colour of labels, will be overridden with individual settings.
 - **Label font:** Font of all labels, set globally.
 - **Labels to keep:** Number of labels to try and keep on the image.
- “*Legend & Warning font settings*” tab
 - **Legend and Warning font:** Font of all legend and warning messages, set globally.

9.5 Loading a preset image template

BRIG has a set of sample templates the can be used for optimal dimensions and font sizes for a given image. This can be accessed from **Preferences >Load template**. BRIG will show a drop down of all the template in the profiles folder in the BRIG installation folder, and a preview of the template (Figure 19), from this users can choose the best template for their image and make adjustments in “Image options”.

Users can add their own templates by saving a BRIG session as a profile and copying it to the profile folder in their BRIG directory. To add a thumbnail for a profile, copy a BRIG image for that session into the profile folder as a JPEG with the same name as that profile. (i.e profilename.jpg).

PRO TIP 18: *Loading a preset template will not overwrite the ring colour settings. This is only to get the right proportions for image sizes, fonts and ring sizes.*

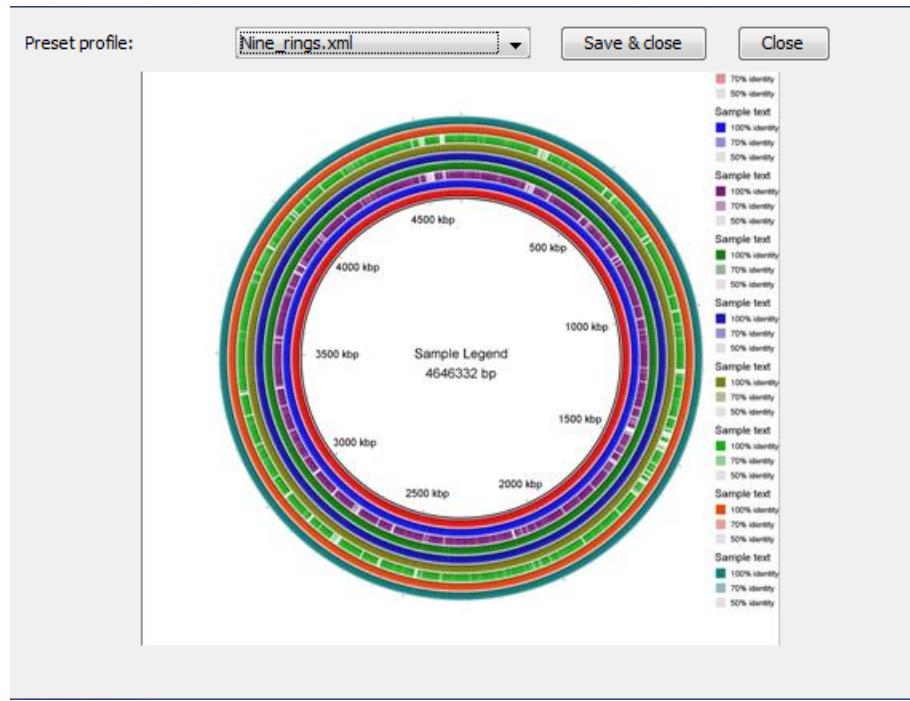


Figure 19: Loading in a preset template window

Availability and requirements

Project name: BLAST Ring Image Generator (BRIG)
Project home page: <http://sourceforge.net/projects/brig/>
Operating system(s): Platform independent
Requirements: Java 1.6 or greater
Licence: GPLv3

References

- [1] ALTCHEL, S., GISH, W., MILLER, W., MYERS, E., AND LIPMAN, D. Basic Local Alignment Search Tool. *Journal of Molecular Biology* 215, 3 (OCT 5 1990), 403–410.
- [2] CAMACHO, C., COULOURIS, G., AVAGYAN, V., MA, N., PAPADOPoulos, J., BEALER, K., AND MADDEN, T. L. BLAST plus : architecture and applications. *BMC BIOINFORMATICS* 10 (DEC 15 2009).
- [3] DARLING, A., MAU, B., BLATTNER, F., AND Perna, N. Mauve: Multiple alignment of conserved genomic sequence with rearrangements. *GENOME RESEARCH* 14, 7 (JUL 2004), 1394–1403.
- [4] RICHTER, D. C., OTT, F., AUCH, A. F., SCHMID, R., AND HUSON, D. H. MetaSim-A Sequencing Simulator for Genomics and Metagenomics. *PLOS ONE* 3, 10 (OCT 8 2008).
- [5] STOTHARD, P., AND WISHART, D. Circular genome visualization and exploration using CGView. *BIOINFORMATICS* 21, 4 (FEB 15 2005), 537–539.