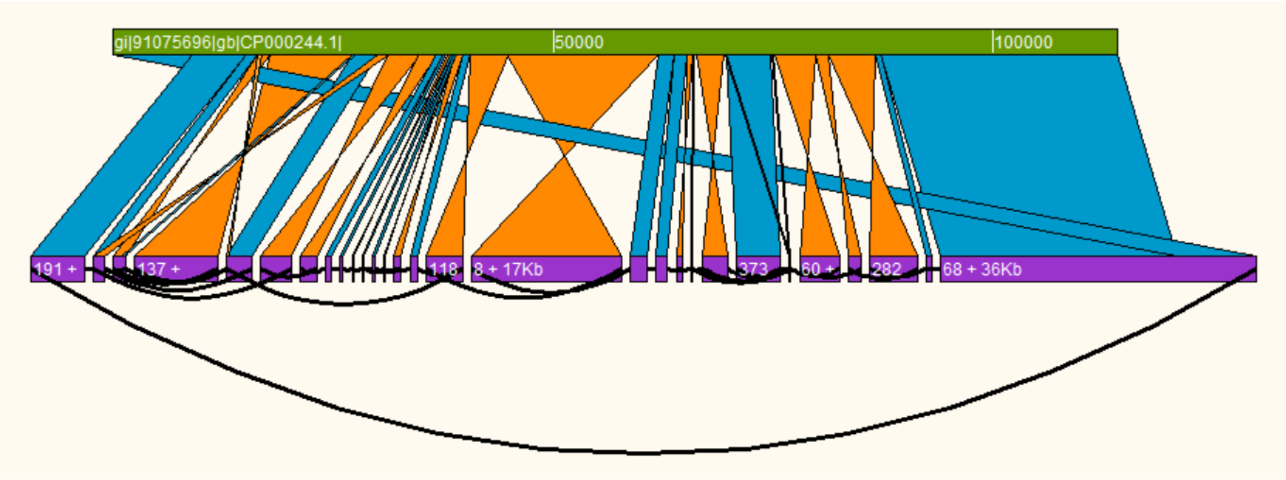


Contiguity

Construction and exploration of assembly graphs

Version 1.0

http://mjsull.github.io/Contiguity/

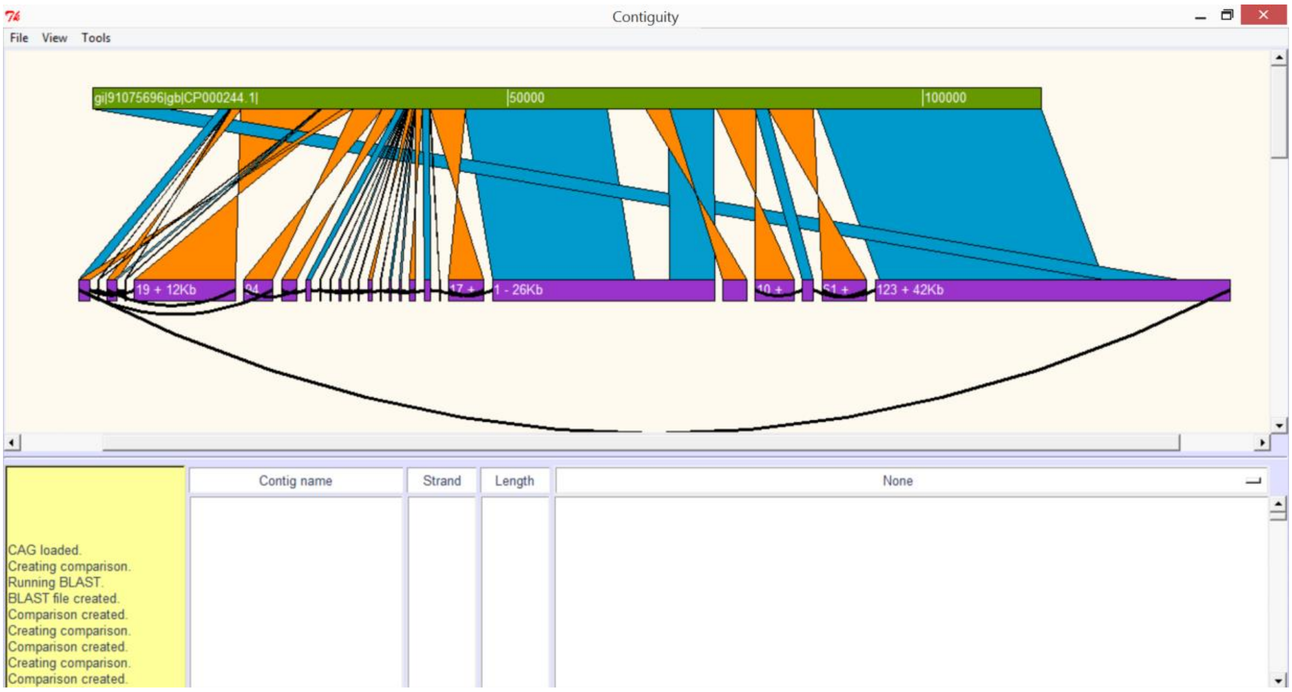
<Citation here>

mjsull@gmail.com

Contents

1. Introduction to Contiguity

Contiguity is a Python application that allows the interactive visualization and manipulation of unfinished *de novo* genome assemblies, chiefly bacterial genomes. It enables a user to create and display information about contig adjacency via a graphical user interface (GUI). Sequence comparisons between the assembled contigs and a reference sequence can be displayed simultaneously to help guide the resolution of novel sequence or structural variants in a draft assembly. Contiguity also provides the first sequencing and assembly independent approach for the creation of contig adjacency graphs. Read pair mapping, sequence overlap and De Bruijn graph exploration are combined to maximize the number of adjacencies found during graph construction. By combining adjacency information with comparative genomics, Contiguity provides an intuitive approach for exploring and improving sequence assemblies. It is also ideally suited for guiding manual closure of Pacific Biosciences SMRT sequence assemblies of bacterial genomes. Contiguity is an open source application, implemented using Python and the Tkinter GUI package that can run on any Unix, OSX and Windows operating system.



CAG loaded.
Creating comparison.
Running BLAST.
BLAST file created.
Comparison created.
Creating comparison.
Comparison created.
Creating comparison.
Comparison created.

Comparison of an Illmuina assembly of E. coli to a virulence plasmid visualised with Contiguity. The Plasmid reference is shown in green. BLAST hits and inverted BLAST hits are shown in blue and orange respectively. Contigs are displayed in purple, a shortened version of the contig name on the contig when room allows. Its orientation and length is also shown where possible. If read information suggests that these contigs may be adjacent to each other, they are joined by a curved black line. This plasmid has been misassembled by Velvet (tsk tsk).

1.1 Requirements

Contiguity comes precompiled for Windows, OSX and GNU/Linux. If you would like Contiguity to automatically generate comparison files for you, please have BLAST installed and in your path. Contiguity also accepts user generated comparison files in BLAST's tabbed output format with no headers (-outfmt 6).

Running Contiguity as a python script

Python – Contiguity has been tested using python 2.7, use earlier versions at your own risk.

Tkinter (GUI) – Included in most installations of python.

Constructing contig adjacency graphs with Contiguity

Khmer – While not required, it is strongly recommended you use Khmer to generate the CAG file unless you have a large amount of memory (>=16GB) a smallish sequencing run (<6Mbp genome with < 200x coverage) and a lot of time. Available at https://github.com/ged-lab/khmer.

NCBI-BLAST+ – BLAST is required for generating CAG files. Available at http://blast.ncbi.nlm.nih.gov/Blast.cgi?PAGE_TYPE=BlastDocs&DOC_TYPE=Download.

Bowtie 2 – Bowtie 2 is required for generating CAG files with paired-end data. Available at http://bowtie-bio.sourceforge.net/bowtie2/index.shtml/.

1.2 Installation

The source code for Contiguity is also available on git https://github.com/mjsull/Contiguity.
To run ensure the requirements have been satisified, download Contiguity.py, or clone the git repository and run using Python.

If you have both python, blastn and bowtie2 you need to (if not already present) install pip.

You can check if pip exists with:

```
$ which pip
```

If you get a "not found", please read the pip installation instructions.

If you already have pip we do suggest you upgrade it. We are using version 1.5.6 at the time of writing this document.

You can upgrade pip like this:

```
$ pip install --upgrade pip
```

pip based installation of Contiguity

If you have root/admin something like:

```
$ pip install Contiguity
```

Otherwise (not root/admin or permission denied errors running above):

```
$ pip install --user Contiguity
```

If you installed using the --user option of pip, Contiguity will typically end up in: /home/$USER/.local/bin/ You need to add this location to you ~/.bash_profile.

Add Contiguity to your path:

```
$ echo 'export PATH=$PATH:/home/$USER/.local/bin/' >> ~/.bash_profile
$ source !$
```

Testing the installation of Contiguity

Run (in the Terminal):

```
$ Contiguity
```

Upgrading Contiguity

You can upgrade like this:

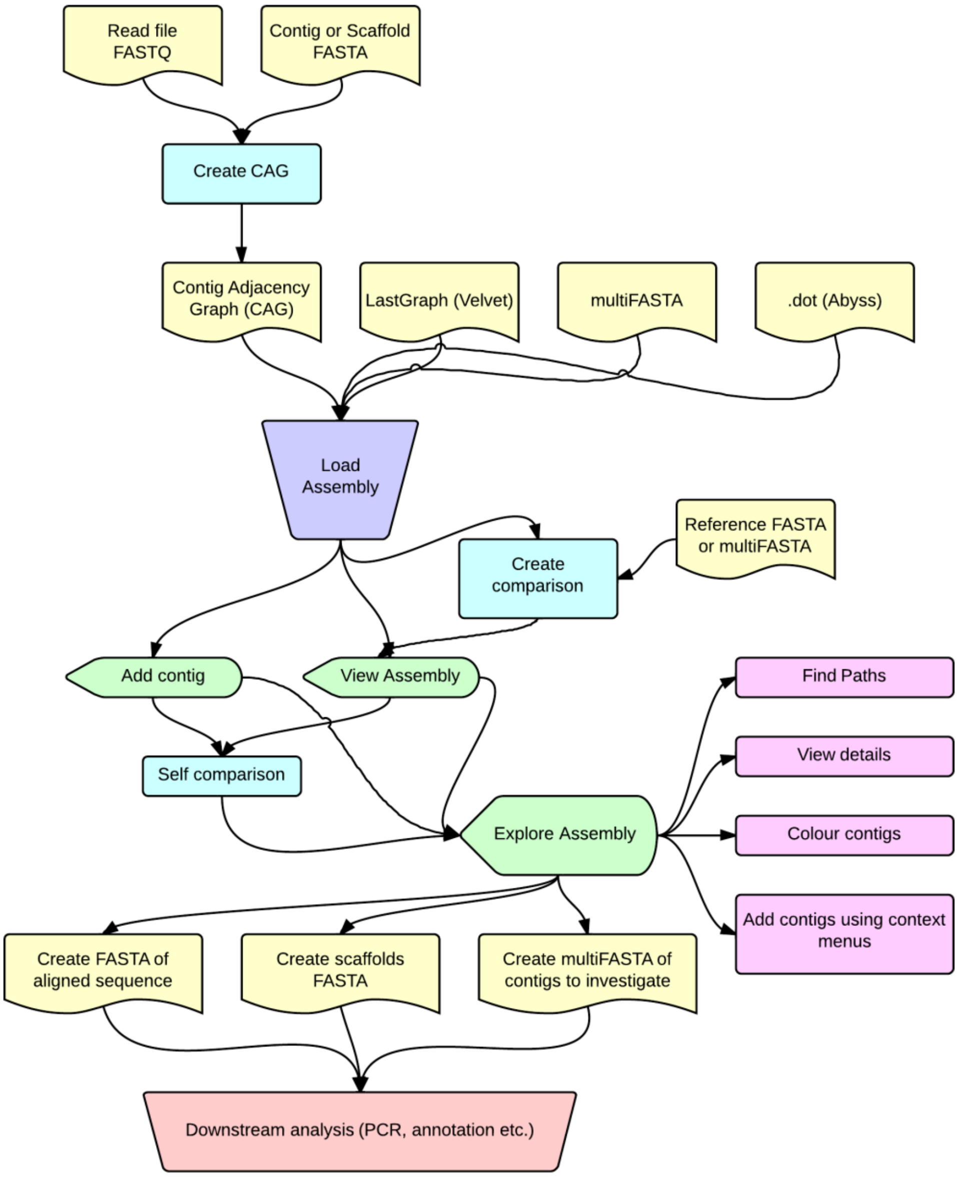
```
pip install --upgrade Contiguity
```

1.3 Citing Contiguity

<Paper will be available soon>

2. Overview

2.1 Workflow



Flowchart of possible workflows, for videos illustrating the uses of Contiguity please visit https://mjsull.github.io/Contiguity (more videos to come).

FASTG (Spades) now supported. GFA support coming soon.

2.1.1 Example workflow for finishing PacBio HGAP Assembly

1. Load a multiFASTA file of the assembly by selecting "Load Assembly" from the file menu
2. View the assembly by selecting "View Assembly" from the "View" menu, ensure "All" is selected in the dropdown box next to "Contigs to view:"
3. To identify spurious contigs perform a self-comparison between contigs using the "Self-comparison tool" from the "View" menu. Phenomena such as prophage tail fibre allele switching can cause contig breaks and spurious contig to form. Remove small contigs that align internally to larger contigs from the assembly.
4. To identify plasmids and reconstruct the chromosome perform another self-comparison. This time ensure the "show intra contig hits" and "only show edge hits" checkboxes are ticked. Show intra-contig hits will allow us to identify circularising contigs and "only show edge hits" will treat the resulting comparison as a graph so when scaffolds are made it will remove overlapping edges.
5. Identify circularising contigs and write to individual FASTA files remove contig from canvas once written.
6. Order and reorientate the remaining contigs, choose "Select All" from the tools menu.
7. Check order and orientation of contigs is correct in the Contig list and then select "Write to FASTA" from the tools menu.
8. Map reads back to output contigs using your favourite read mapper to confirm new assembly.

2.1.2 Example workflow for ordering contigs and scaffolding regions of difference in a bacterial assembly (Illumina)

1. Create Contiguity-CAG using the command-line or GUI.
2. Load CAG by selecting "Load Assembly" from the "File" menu.
3. Generate sequence alignment by selecting "Create comparison" from "File" menu and selecting the FASTA file of a reference.
4. View the graph and comparison by selecting "View assembly" from the "View" menu, select "BLAST" in contigs to view.
5. Order and orientate contigs in the Canvas, contigs can be reverse complemented and duplicated by right clicking on a contig and selecting "Reverse" or "Duplicate" from the context driven menu.
6. Identify gaps in the assembly, adjacent contigs that are not connected by an edge. These are caused by gaps in sequence coverage or insertion elements.
7. To identify potential insertion sequence double click the two contigs adjacent to the gap and select "Find paths" from the "Tools" menu, then select ok.
8. This can also be done manually by selecting "add" or "move" contigs from "To" or "From" in the context driven menu created by right clicking on the contig.
9. Repeat for entire chromosome.
10. Choose "Select all" from the "Tools" menu.
11. Choose "Write to FASTA" from the "Tools" menu.
12. Proceed to annotation.

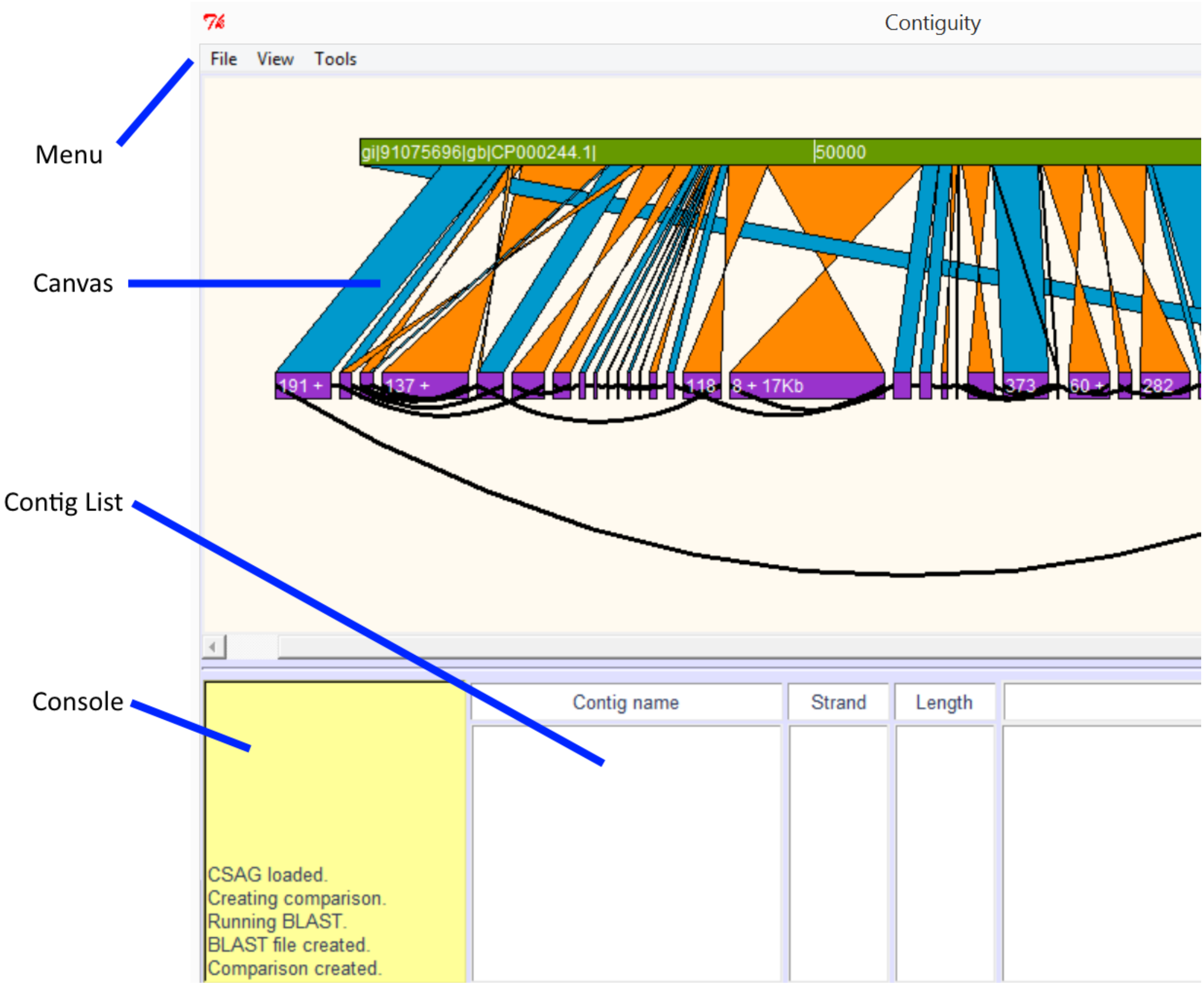
2.1.3 Example workflow for finding passenger genes of a transposon with long terminal repeats (LTRs) in a bacterial genome (Illumina)

1. Create Contiguity-CAG using the command-line or GUI.
2. Load CAG by selecting “Load Assembly” from the “File” menu.
3. If the LTR contig has been identified select “Add contig” from “View menu”.
4. Alternatively a multiFASTA of potential LTRs can be compiled and LTR contigs can be identified by generating a comparison to this file.
5. Double click on transposon and select “Find paths” from the “Tools” menu.
6. Check “Find all paths” and reduce “Max bp” to the maximum bases predicted for passenger genes.
7. Select “Ok”.
8. Arrange contigs (Duplicate LTR) and create scaffolds if needed.

2.1.4 Example workflow for finding plasmids in bacterial assembly (Illumina)

1. Create Contiguity-CAG using the command-line or GUI.
2. Load CAG by selecting “Load Assembly” from the “File” menu.
3. Generate sequence alignment by selecting “Create comparison” from “File” menu and selecting the FASTA file of a reference.
4. View the graph by selecting “View assembly” from the “View” menu, select filter from the drop down menu next to “Contigs to view” and then click “Ok”.
5. Too add contigs that may be shared by the plasmid and chromosome choose “Select all” from the tools menu and then select “Find paths” and click “Ok”.
6. Order contigs so that contigs that share an edge are adjacent, there will be multiple ways to order contigs. Duplicate contigs where necessary by right-clicking on the contigs and selecting duplicated form the context-driven menu.
7. Create scaffolds of the potential plasmid by selecting “Write to FASTA” from the “Tools” menu.
8. These scaffolds can then be used to design primers to confirm presence and arrangement of contigs of the plasmid.

2.2 General layout



```
CSAG loaded.
Creating comparison.
Running BLAST.
BLAST file created.
Comparison created.
```

Menu – Dropdown menus
Canvas – Area where Contig, mapping and graph information is shown.
Console – Progress of processes running in the background is reported here.
Contig List – List of selected contigs

2.3 Menu

2.3.1 File

Create CAG file – Construct a CAG file (for more details see "Constructing a CAG" below)

Create Comparison – Create comparison (for more details see "Creating comparisons" below)

Load Assembly – Load FASTA/FASTG/CAG/LastGraph/.dot. For user generated .dot files or .dot generated using Abyss please concatenate with a FASTA file containing the forward sequence of each contig or scaffold this can be done from the command line using a command such as cat assembly.dot assembly.fa > assembly.contiguity.dot

Save Image – Save canvas as a postscript image

Change Working Directory – Change the working directory from .contiguity_wd

Cancel Running Process – Cancel any currently running processes

Exit – Cancel any currently running processes and quit Contiguity

2.3.2 View

View Assembly – View loaded assembly (for more details see "view assembly" below)

Self-Comparison – Find and display sequence similarity between and optionally within contigs (for more details see "Self-comparisons" below)

Add Contig – Add contig (long or short name)

Find Contig – Go to specified contig on the canvas

Zoom in – Zoom in on the canvas

Zoom out – Zoom out on the canvas

Stretch – Stretch the canvas in the x-axis

Shrink – Shrink the canvas in the x-axis

2.3.3 Tools

Select All – Select all contigs currently displayed on the canvas from left to right

Clear Selected – Clear all currently selected contigs

Find Paths – Find paths between selected contigs (For more information see "Find Paths" below)

Write FASTA – Write a scaffold of currently selected contigs, if an edge is found between two sequential contigs, use the edge to fill in sequence or remove overlaps, otherwise the user defined scaffolding characters will be use.

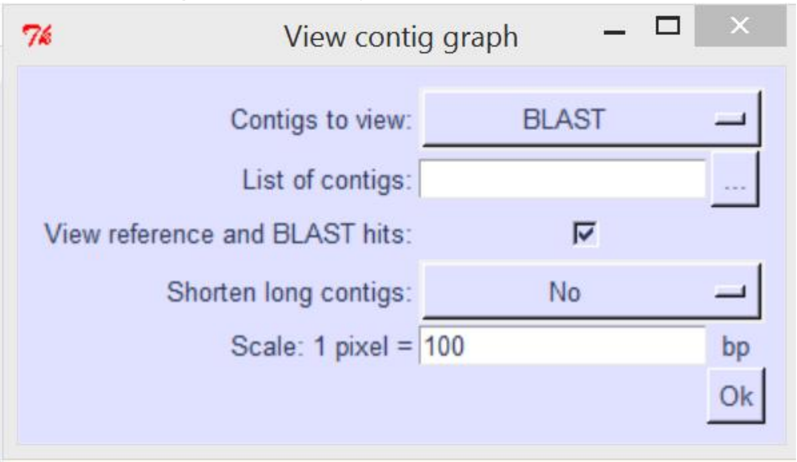
Write multiFASTA – Write currently selected contigs as a multiFASTA, each selected contig will be written as its own entry

Colour options – Change options relating to contig colours (for more information see "Contig Selection" below)

Load Special – Load custom colours from a text file (for more information see "Contig Selection" below)

2.4 Viewing the Assembly

Once the assembly has been loaded by selecting “View” and then “View assembly”. If you also wish to view a comparison between the assembly and a reference ensure it has been created before viewing the assembly.



View contig graph

Contigs to view: BLAST

List of contigs:

View reference and BLAST hits: ☑

Shorten long contigs: No

Scale: 1 pixel = 100 bp

Contigs to view

- **BLAST –** view only contigs that align to the reference
- **ALL -** view all contigs
- **List –** view a user defined set of contigs
- **Filter –** view only contigs that do not align to the reference

List of contigs – If “List” is selected from “Contigs to view:” please provide a text file with a list of contig names with each name in its own row

e.g.

```
node_8_length_10123_cov_52.121
node_9_length_32191_cov_24.121
node_10_length_1021_cov_100.111
```

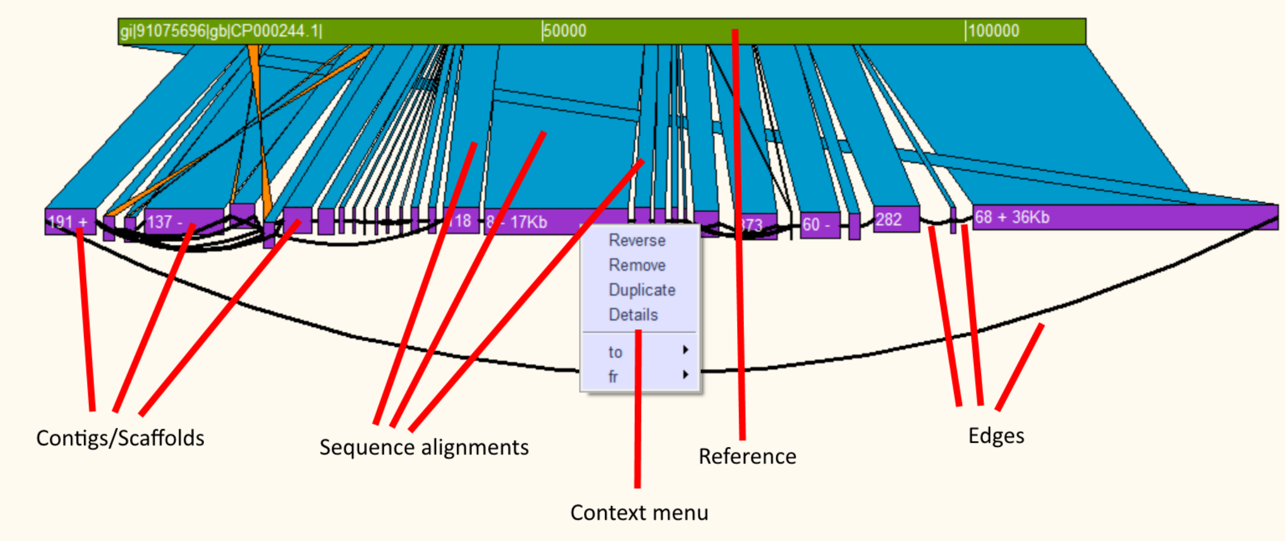
View reference and BLAST hits – When the graph is drawn on the canvas, also draw the reference and alignments

Shorten long contigs

- **No –** Contig length on canvas proportional to length in base pairs
- **Constant –** All contigs 200px wide
- **Min –** All contigs at least 200px wide
- **Log –** Contig length on canvas proportional to the log of its length

Scale: 1 pixel = … bp – Ratio for determining size of contigs in pixels, this can be adjusted once loaded by selecting “Stretch” or “Shrink” from the menu or alternatively using the “A” and “D” keys.

2.5 Canvas overview



Using the canvas

The contigs and the reference may be moved individually by dragging them with the left mouse button. The canvas can be zoomed in and out by using the mousewheel or using the “w” or “d” key. Contigs may be shrunk or stretched in the x dimension by using the “a” and “d” key. Context menus can be brought up by right clicking a contig or sequence alignment allowing the user to reverse, remove, duplicate or bring up details about that contig or alignment.

2.5.1 Context menus

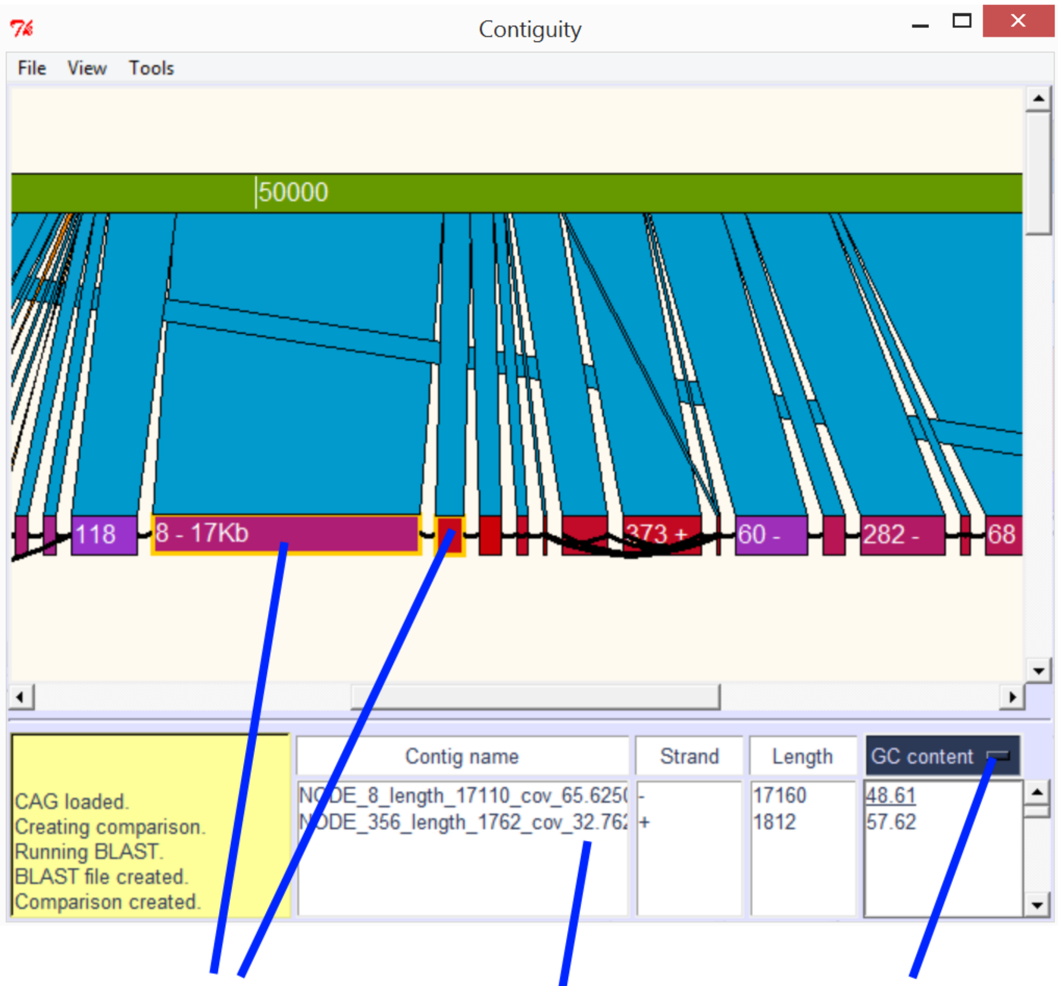
Contig

- **Reverse –** Reverse compliment the contig
- **Remove –** Remove the contig from the canvas
- **Duplicate –** Duplicate the contig
- **Details** – Bring up details about the contigs length, GC content, skew, coverage and edges
- **to/fr –** Bring up a menu that allows the user to highlight edges. Also allows the user to add, duplicate or move contigs attached to the 3’ end of the selected contig (to) or the 5’ end (fr)

Blast

- **Query/Subject**
  - **Move –** Move to query/subject so it is aligns with its match
  - **Goto –** Move to the query or subject
  - **Write –** Write the aligned query or subject sequence to a FASTA file
- **Remove –** Remove the hit
- **Details –** Show details about the hit such as identity, mismatches etc.
- **To back/front –** Move the blast hit to the back or front of the canvas
- **Highlight –** Outline the hit with a specific colour

2.6 Contig Selection



CAG loaded.
Creating comparison.
Running BLAST.
BLAST file created.
Comparison created.

Contig name	Strand	Length	GC content
NODE_8_length_17110_cov_65.625	-	17160	48.61
NODE_356_length_1762_cov_32.76	+	1812	57.62

Selected Contigs

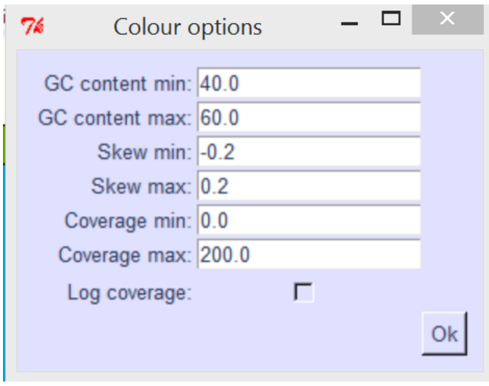
List of Selected Contigs

Colour options

ABOVE: Assembled contigs aligned to a plasmid. Contigs 8 and 356 have been selected, and are displayed along with their orientation and length in the contig list section. GC content has been selected from a drop down menu (colour options) and the GC content of both contigs is displayed. Contigs on the canvas are now also coloured according to their GC content. Contigs with 60% or greater GC content are coloured red, contigs with 40% or lower GC content are coloured purple, contigs in-between are coloured on a gradient between red and purple. Selected contigs are outlined with a yellow box.

2.6.1 Colouring contigs

Contigs can be coloured according to GC content, GC Skew, AT Skew, Coverage or user provided colours. Contigs are coloured purple if the selected value is equal to or under the minimum value or red if the colour is equal to or more than the maximum value, contigs with a value in between the minimum and maximum value are coloured on a gradient from purple to red. Minimum and maximum values can be changed from the colour options menu under tools.



GC content min: 40.0
GC content max: 60.0
Skew min: -0.2
Skew max: 0.2
Coverage min: 0.0
Coverage max: 200.0
Log coverage: □

Custom colours and values

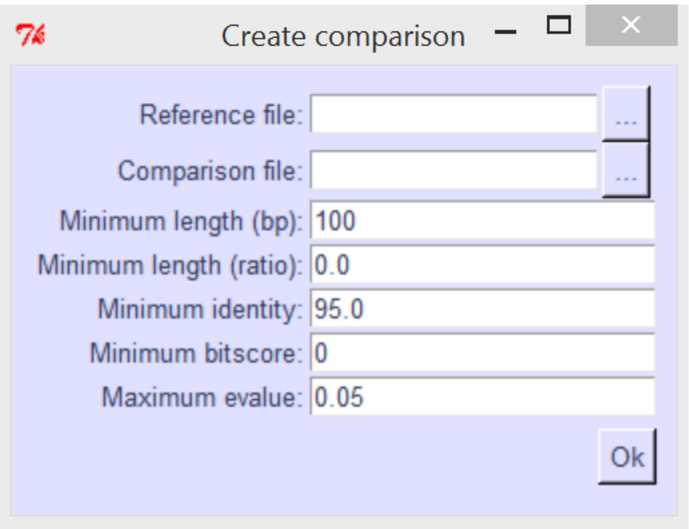
Custom colours and values may be loaded. To load a file select “Load Custom” from the “Tools” menu and select a file with the custom values. The file should be in a tab delimited format with the first column being full contig names, the second column containing the custom value for the corresponding contig and the third column containing hex colour codes.

e.g.

```
node_8_length_10123_cov_52.121     20   #FF1C28
node_9_length_32191_cov_24.121     30   #12FF28
node_10_length_1021_cov_100.111    40   #12311C
```

2.7 Creating comparisons

Comparison can be created between a loaded assembly and a reference by selecting “Create comparison” from the “File” menu. If you have NCBI-BLAST+ installed and in your path comparisons can be generated on the fly by providing Contiguity with a reference file. Alternatively, an alignment file in BLAST’s tab-delimited (no header) format may be provided. Alignment results will be displayed if they satisfy the following default parameters:



Reference file:

Comparison file:

Minimum length (bp): 100

Minimum length (ratio): 0.0

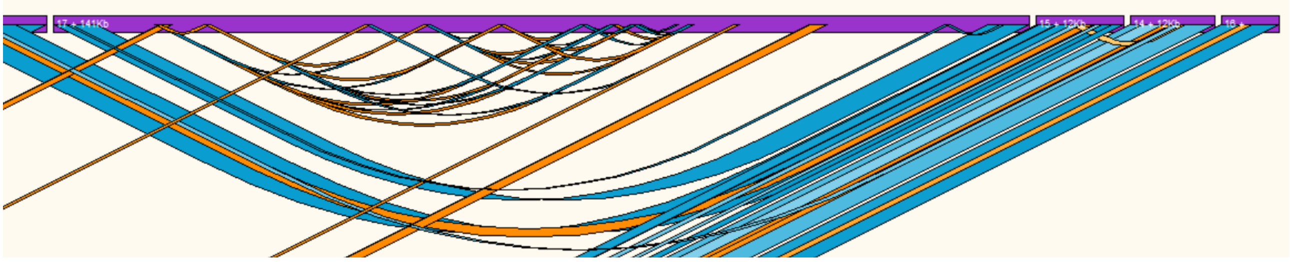
Minimum identity: 95.0

Minimum bitscore: 0

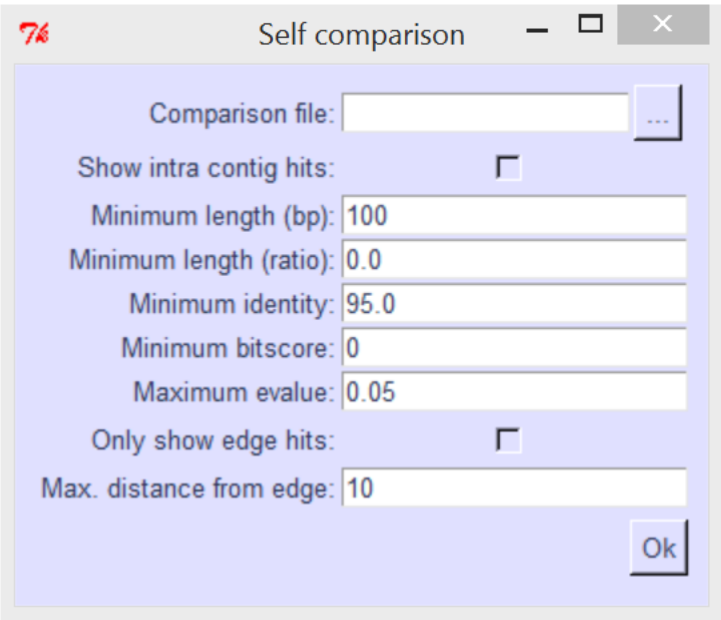
Maximum evalue: 0.05

Reference file – Reference file in FASTA or multiFASTA format
Comparison file – Alignment file in BLAST’s tab-delimited format
Minimum length – Only alignments of length (in base pairs) equal to or greater than this value will be shown
Minimum length (ratio) – Fraction of the contig sequence aligned to show alignment i.e. 0.5 means that an alignment needs to cover at least half the query to be shown
Minimum identity – Only alignments with an identity equal to or greater than this value (in percent) will be shown
Minimum bitscore – Only alignments with a bitscore equal to or greater than this value will be shown
Maximum evalue – Only alignments with an expect score lower than or equal to this value will be shown

2.8 Self comparisons



ABOVE: Self comparison of a pacbio assembly, direct repeats are shown with blue ribbons, inverted repeats are shown with orange ribbons. Hits with a low identity are lighter than hits with a high identity.



Comparison file:

Show intra contig hits: □

Minimum length (bp): 100

Minimum length (ratio): 0.0

Minimum identity: 95.0

Minimum bitscore: 0

Maximum evalue: 0.05

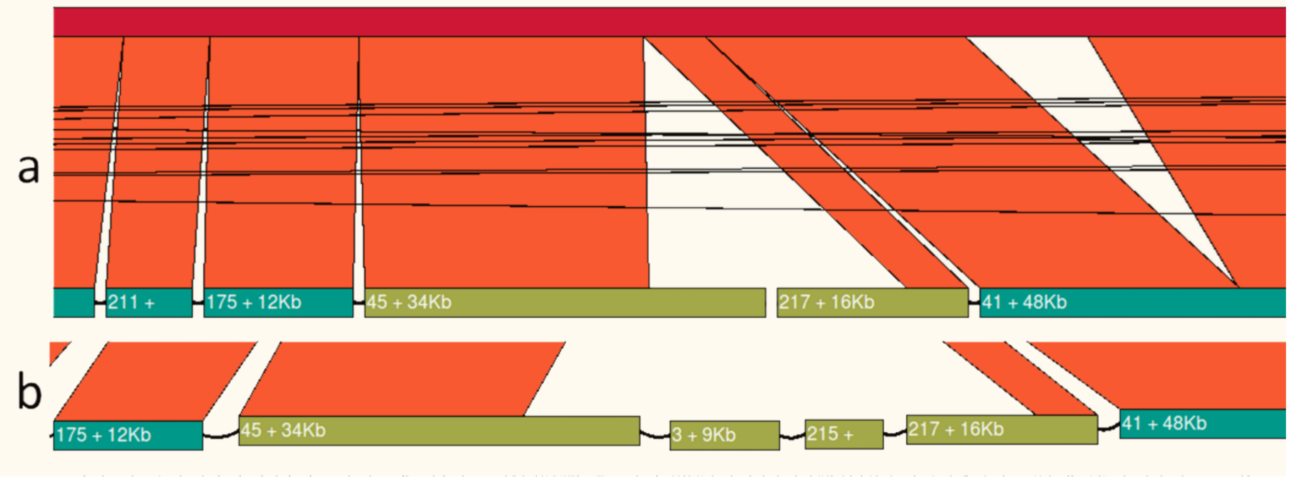
Only show edge hits: □

Max. distance from edge: 10

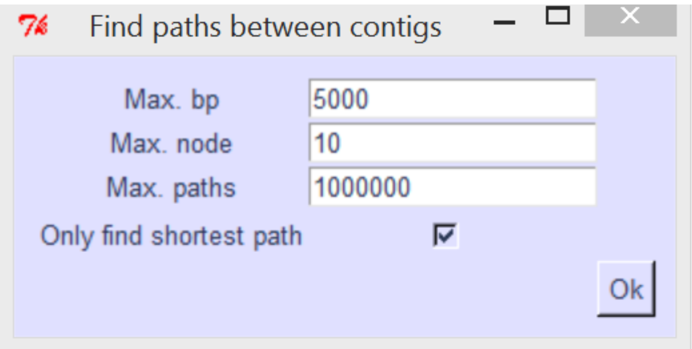
Comparison file – Alignment file in BLAST’s tab-delimited format
Show intra contig hits – Show hits mapping from and to the same contig
Minimum length – Only alignments of length (in base pairs) equal to or greater than this value will be shown
Minimum length (ratio) – Fraction of the contig sequence aligned to show alignment i.e. 0.5 means that an alignment needs to cover at least half the query to be shown
Minimum identity – Only alignments with an identity equal to or greater than this value (in percent) will be shown
Minimum bitscore – Only alignments with a bitscore equal to or greater than this value will be shown
Maximum evalue – Only alignments with an expect score lower than or equal to this value will be shown
Only show edge hits – Only show hits that map to the edge of the contig
Max. distance from edge – Distance from the edge of a to consider hit an “edge hit”

2.9 Find Paths

Find paths allows the user to find the shortest or all paths between all selected contigs. Simply select the contigs you would like to find paths between and then select “Find paths” from the “tools” menu. “Find paths” will search through the contig adjacency graph until it finds another selected contig or reaches the user defined node or base pair limit.



Illumina assembly mapped to a reference genome. There is a gap in the assembly between contigs 45 and 217 (panel a) by selecting both contigs and using the “Find paths” tool Contiguity is able to reconstruct this region of the chromosome (panel b).


Find paths between contigs

Max. bp: 5000
Max. node: 10
Max. paths: 1000000
Only find shortest path ☑

Max bp – Depth of search in base pairs
Max node – Depth of search in contigs
Max paths – If this value is reached Contiguity will lower the amount of nodes being searched
Only find shortest path – Only report the shortest path (in base pairs) found

3. Creating Scaffolds

Creating scaffolds can be done by selecting “Write FASTA” from the tools menu. Contiguity will attempt to reconstruct sequence and remove overlapping regions between contigs using information found in the graph file or found using self-comparisons. **IT IS IMPORTANT TO NOTE THAT THIS IS A “BEST GUESS” AND MAY NOT BE ACCURATE.** We suggest sanity checking any scaffolds created by mapping reads back to the newly created scaffold.

3.1 Creating scaffolds from a CAG

First ensure the contigs are orientated correctly on the canvas then select the contigs from which you want to create a scaffold from in order. Alternatively if you want to create a scaffold from all the contigs on the screen order the contigs on the canvas from left to right and select “Select All” from the “tools” menu. Then select “Write FASTA” from the tools menu, if there is an edge between the two adjacent contig ends Contiguity will insert the sequence or remove the overlap associated with that edge, if there is no edge contig will insert the user defined buffer sequence.

3.2 Creating scaffolds from a PacBio assembly

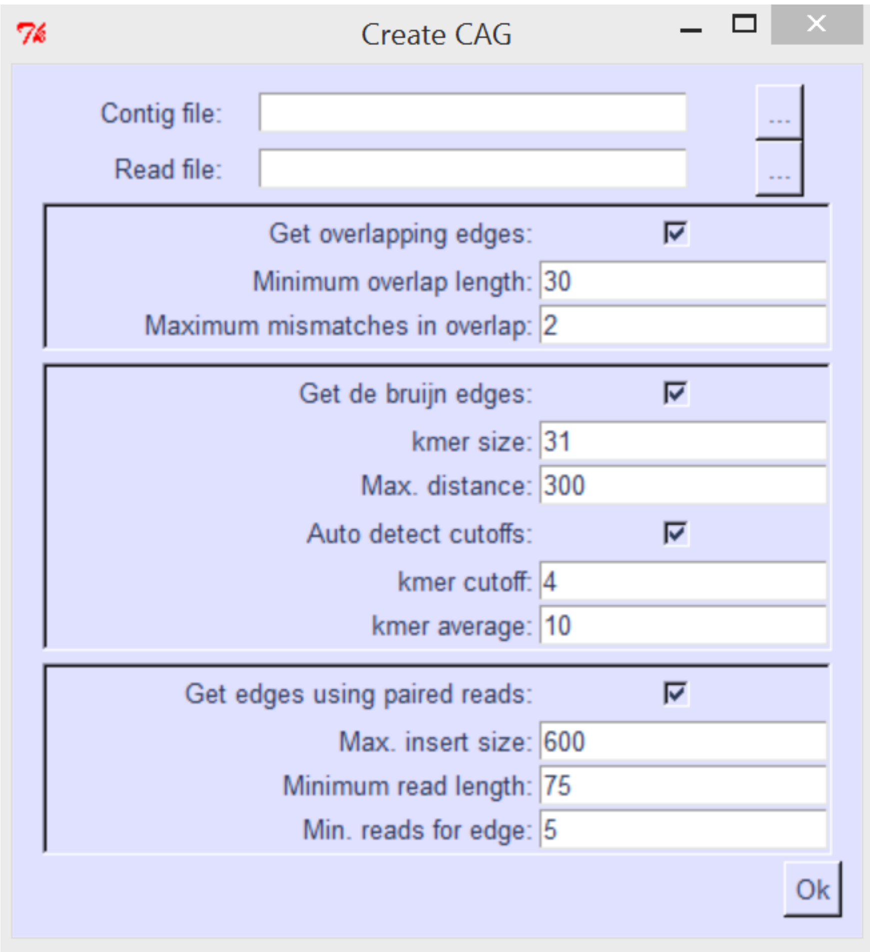
Once the assembly has been loaded and displayed on the canvas select “Self-comparison” from the “View” menu, ensure that you check the “Only show edge hits” checkbox. Order and orientate the contigs in the correct direction and then select contigs you wish to make up the scaffold. Contiguity will remove merge the contigs removing the overlapping sequence. Contiguity uses the sequence from the larger contig if there are mismatches in the overlap.

4. Constructing a Contig Adjacency Graph

A CAG is a directed graph where each contig in a de novo assembly is represented as two nodes (forward and reverse strand). A directed edge is created from node A to node B if the sequence represented by node B occurs directly after the sequence represented by node A.

4.1 GUI

Contig adjacency graphs can be created by selecting "Create CAG" from the "File" menu


Create CAG

Contig file: ...

Read file: ...

Get overlapping edges: ☑

Minimum overlap length: 30

Maximum mismatches in overlap: 2

Get de bruijn edges: ☑

kmer size: 31

Max. distance: 300

Auto detect cutoffs: ☑

kmer cutoff: 4

kmer average: 10

Get edges using paired reads: ☑

Max. insert size: 600

Minimum read length: 75

Min. reads for edge: 5

Contig file – FASTA file of contigs or scaffolds
Read file – Interleaved fastq file - read1_left, read1_right, read2_left... orientated as such --> <--
Get overlapping edges – Find edges by looking for overlapping edges (Requires BLAST)
Minimum overlap length – Minimum overlap to consider two contigs adjacent
Maximum mismatches in overlap – Maximum number of mistmatches allowed in that overlap
Get de bruijn edges – Find edges using a De Bruijn graph
Kmer size – kmer size used to construct De Bruijn graph
Max. distance: Distance (in bp) to search in the De Bruijn graph for an adjacent contig (this value + kmer size)
Auto detect cutoffs – Automatically detect coverage cutoff and median unique kmer frequency
kmer cutoff – kmers with a frequency less than this value will not be traversed
kmer average – median unique kmer frequency: all kmers with a frequency greater than this value will be traversed
Get edges using paired reads: Find edges by mapping paired-end reads (Bowtie-2 required)
Max. insert size: Only reads with an insert size smaller than this will be counted

Minimum read length: Only reads that have at least this value of bases mapping will be counted
Min. reads for edge: Minimum mapping reads to create an edge between two contigs

4.2 Command-line

Contig adjacency graphs can also be created from the command line

```
USAGE: Contiguity.py -cl -c <contig_file.fa> -fq <read_file.fq> -o
<output_folder>


contig file: FASTA file of contigs or scaffolds

read file: Interleaved fastq file - read1_left, read1_right, read2_left etc...
orientated as such --> <--

output folder: folder to put output files in, can and will overwrite files in
this folder, will create folder if folder doesn't exist


Only other option to keep in mind is -rl if the read length is not 101bp


optional arguments:
  -h, --help            show this help message and exit
  -co CONTIG_FILE, --contig_file CONTIG_FILE
                        fasta file of assembled contigs or scaffolds
  -rf READ_FILE, --read_file READ_FILE
                        read file
  -o OUTPUT_FOLDER, --output_folder OUTPUT_FOLDER
                        output folder
  -k KMER_SIZE, --kmer_size KMER_SIZE
                        k-mer size for finding adjacent contigs [31]
  -max_d MAX_DISTANCE, --max_distance MAX_DISTANCE
                        maximum distance apart in the de bruijn graph for
                        contigs to count as adjacent [300]
  -kmer_a KMER_AVERAGE, --kmer_average KMER_AVERAGE
                        All k-mers above half this value will be traversed
                        [auto]
  -kmer_c KMER_CUTOFF, --kmer_cutoff KMER_CUTOFF
                        cutoff for k-mer values [auto]
  -ov OVERLAP, --overlap OVERLAP
                        minimum overlap to create edge [kmer_size-1]
  -rl MIN_READ_LENGTH, --min_read_length MIN_READ_LENGTH
                        Minimum read length [75]
  -max_mm MAX_MISMATCH, --max_mismatch MAX_MISMATCH
                        maximum number of mismatches to count overlap [2]
  -lo LONG_OVERLAP_IDENT, --long_overlap_ident LONG_OVERLAP_IDENT
                        minimum percent identity to create an edge where there
                        is a long overlap [85]
  -mp MINIMUM_PAIRS_EDGE, --minimum_pairs_edge MINIMUM_PAIRS_EDGE
```

```
                        Minimum pairs to create edge [2]
  -is MAX_INSERT_SIZE, --max_insert_size MAX_INSERT_SIZE
                        Upper bound on insert size [600]
  -cl, --command_line   Run contiguity in command line mode
  -no, --no_overlap_edges
                        Don't get overlap edges
  -nd, --no_db_edges    Don't get De Bruijn edges
  -np, --no_paired_edges
                        Don't get paired-end edges
  -km, --khmer          Don't use khmer for De Bruijn graph contruction (not
                        recommended)

Thanks for using Contiguity
```