Bioinformatics, YYYY, 0–0
doi:10.1093/bioinformatics/bty1013
Advance Access Publication Date: DD Month YYYY
Available online 7 December 2018

Bioinformatics, Volume 35, Issue 15,
1 August 2019, pp. 2671–2673

Sequence Analysis

MapOptics: A light-weight, cross-platform visualisation tool for optical mapping alignment

Josephine Burgin[1], Corentin Molitor[1], Fady Mohareb[1*]

[1] The Bioinformatics Group, School of Water, Energy and Environment, Cranfield University, College Road, Bedford, MK43 0AL, UK.

*To whom correspondence should be addressed.

Associate Editor: XXXXXXX

Received on XXXXX; revised on XXXXX; accepted on XXXXX

Abstract

Summary: Bionano optical mapping is a technology that can assist in the final stages of genome assembly by lengthening and ordering scaffolds in a draft assembly by aligning the assembly to a genomic map. However, currently, tools for visualisation are limited to use on a Windows operating system or are developed initially for visualising large-scale structural variation. MapOptics is a lightweight cross-platform tool that enables the user to visualise and interact with the alignment of Bionano optical mapping data and can be used for in depth exploration of hybrid scaffolding alignments. It provides a fast, simple alternative to the large optical mapping analysis programs currently available for this area of research.

Availability and implementation: MapOptics is implemented in Java 1.8 and released under an MIT licence. MapOptics can be downloaded from https://github.com/FadyMohareb/mapoptics and run on any standard desktop computer equipped with a Java Virtual Machine (JVM).

Contact: f.mohareb@cranfield.ac.uk.

Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

Optical mapping is a technology that gives insight into the basic structure of large DNA fragments. It can be used as an extra step to a genome assembly to improve ordering, orientation and length of scaffolds. Traditionally, optical mapping suffered from a high error rate, making it unreliable for accurate genome assemblies. However, as with the recent advances of other sequencing platforms, optical mapping has largely overcome its limitations and is now used in many genomic studies (Phillippy 2017; Yuan et al. 2017). Now available are optical maps derived from high throughput nano-channels performed by *Bionano* systems (Dai et al. 2018; Pendleton et al. 2015). With modern techniques, restriction enzyme digestion of long DNA molecules (~ 500kb) is performed by one or more 'nickase' enzymes, modified to only digest single strands at a particular site. These sites are then repaired with fluorescent nucleotides which can be imaged to form a unique barcode on the molecule (Udall and Dawe 2018). Even newer technologies such as *Bionano* Direct Label and Stain Technology (DLS) offer non-destructive methods of creating optical maps that are even longer (> 2Mbp) with enzymes that label the sites without digestion. These long DNA molecules are assembled to form a genome consensus map. Genome assembly scaffolds can then be digested *in silico* to form barcoded molecules that can be aligned to this consensus map. During this alignment process, errors in the assembly can be recognised and corrected, therefore improving the sequence completeness and accuracy further (Jiao and Schneeberger 2017; Seo et al. 2016). Visualisation post-alignment can give insight into the quality of the alignment and recognition of mis-assemblies. This is key to understand the success in the technique and in assisting manual conflict resolution to improve results further.

There are five tools currently available for visualisation of optical mapping alignment: BioNumerics v7 (Applied-Maths-NV 2018), JBrowse (Skinner et al. 2009), OMView (Leung et al. 2017), *Bionano's* IrysView (https://bionanogenomics.com/support-page/irysview) and *Bionano's* Access (https://bionanogenomics.com/support-page/bionano-acces). However, only two of these tools provide views useful to a genome assembly context. The tools BioNumerics v7, JBrowse and OMView provide visualisation of map alignment at a one-to-one scale with the aim to detect large regions of structural variation between two sequences.

Published by Oxford University Press. This is the Author Accepted Manuscript issued with: Creative Commons Attribution Non-Commercial License (CC:BY:NC 4.0).
The final published version (version of record) is available online at DOI:10.1093/bioinformatics/bty1013. Please refer to any applicable publisher terms of use.

Whereas, IrysView and Access can also provide visualisation on a smaller scale, showing multiple scaffolds from a genome assembly aligned to the optical map. This view is useful to visualise mis-assemblies and mis-orientations of scaffolds within the dataset. However, IrysView and Access are primarily alignment tools; in other words, they perform the computationally heavy alignment stages, making their performance sometimes slow. There is no current lightweight and user-friendly alternative tool for alignment visualisation. In addition, IrysView is limited to use on a Windows platform and Access is only available on an online server; meaning there is no locally run Linux-based tool for this purpose.

2 MapOptics

MapOptics is developed in Java 8 to ensure cross-platform compatibility; the tool uses Java Swing components and the Graphics2D library to keep visualisation simple and fast to load (See the Supplementary Material S1 for the software overall architecture). The user is only required to input two CMAP files (a reference and query) and an XMAP file in *Bionano* format and the tool will display all the alignments between the two map files for the user to navigate through. In particular, it provides a tool for visualisation of hybrid scaffolding results generated based on optical mapping and contig/scaffold assembly. Unique to this tool is a highly interactive view of all scaffolds aligned to the optical map, which allows the user to assess the quality of these alignments and the quantity of mis-assemblies and scaffold overlaps. It also allows the user to explore the alignment of scaffolds to hybrid-scaffolds to visualise this stage of assembly.

3 Assembly visualisation

MapOptics provides tabs of 3 different views for visualisation, to allow exploration of the data at different zoom levels (See Fig.1). The Summary View tab (Fig1.A) provides an overall summary of the maps loaded and a simple alignment view panel. It allows the user to explore basic statistics of the reference dataset, such as contig lengths and label densities, the latter being useful to confirm that the correct nicking enzyme was used. Using MapOptics, the user can highlight the distribution of label densities against the recommended values(BionanoGenomics 2018b) Moreover, the user can select a reference contig of interest to have the specific values and alignment associated with it. The Reference View tab (Fig1.B) provides a more interactive view of a chosen reference contig. It visualises all scaffolds aligned to the chosen contig which can be dragged, re-oriented or deleted by the user. These changes can be saved to ease navigation in Summary View. Here, the user can also load a conflicts_cut_status.txt file generated from hybrid scaffolding and visualise contigs where the conflicts were resolved. This allows the user to navigate and assess this automated process, make their own decisions and export the file for manual conflict resolution. The Query View tab (Fig2.C) provides a close up of just one alignment for more detail on quality of individual labels. The user can upload a FASTA file and explore gaps in the sequence as well as zoom into a region of choice (See Supplementary Material S2 for additional functionalities). These Views interact with each other, updating to match the reference or query contig selected whenever it is changed, allowing ease of navigation and images from the displays can be exported so any key alignments of interest can be shared and stored for later use.

Conclusions

MapOptics provides a lightweight and user-friendly way to visualise scaffold assembly and optical mapping alignment. The data can be explored at different levels of depth in an interactive way so the user can customise how best to display their data.

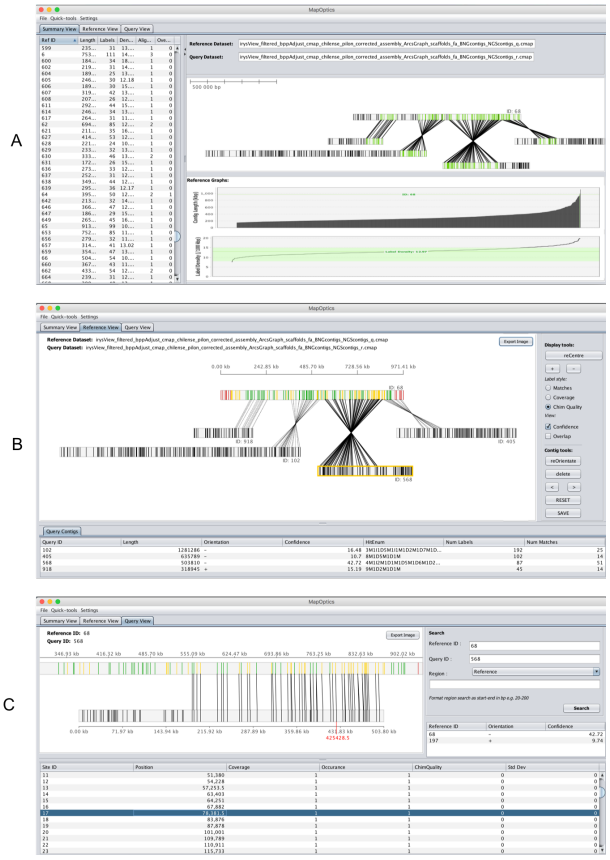


Fig. 1 ***A. (top left)*** *Summary View - provides an overall summary on the maps loaded and a simple alignment view.* ***B. (top right)*** *Reference View- provides a more interactive view of the reference map/scaffold chosen and visualises all aligned scaffolds relative to that which can be manipulated by the user.* ***C. (bottom middle)*** *Query View – provides a close up of just one alignment for more detail on quality of individual labels and the ability to zoom into a region of choice.*

Funding

MapOptics was developed as part of the BBSRC-funded project "Genomics-assisted selection of *Solanum chilense* introgression lines for enhancing drought resistance in tomatoes" - BB/L011611/1.

References

Applied-Maths-NV (2018), 'BioNumerics Seven : a unique software platform ', (Applied Maths).

Dai, Yi, et al. (2018), 'Single-molecule optical mapping enables accurate molecular diagnosis of facioscapulohumeral muscular dystrophy (FSHD)', *bioRxiv*.

Jiao, W. B. and Schneeberger, K. (2017), 'The impact of third generation genomic technologies on plant genome assembly', *Curr Opin Plant Biol,* 36, 64-70.

Leung, A. K., et al. (2017), 'OMTools: a software package for visualizing and processing optical mapping data', *Bioinformatics,* 33 (18), 2933-35.

Pendleton, M., et al. (2015), 'Assembly and diploid architecture of an individual human genome via single-molecule technologies', *Nat Methods,* 12 (8), 780-6.

Phillippy, A. M. (2017), 'New advances in sequence assembly', *Genome Res,* 27 (5), xi-xiii.

Seo, J. S., et al. (2016), 'De novo assembly and phasing of a Korean human genome', *Nature,* 538 (7624), 243-47.

Skinner, M. E., et al. (2009), 'JBrowse: a next-generation genome browser', *Genome Res,* 19 (9), 1630-8.

Udall, J. A. and Dawe, R. K. (2018), 'Is It Ordered Correctly? Validating Genome Assemblies by Optical Mapping', *Plant Cell,* 30 (1), 7-14.

Yuan, Y., et al. (2017), 'Improvements in Genomic Technologies: Application to Crop Genomics', *Trends Biotechnol,* 35 (6), 547-58.

MapOptics: Technical Documentation

MapOptics is a lightweight stand-alone cross-platform visualisation tool for optical mapping hybrid scaffolding alignment. It is developed in Java 8 using the Java SWING GUI components and the JFreeChart external library.

External libraries or scripts used:

1. JFreeChart was used for drawing graphs
   http://www.jfree.org/jfreechart/

2. A script for an editable header renderer was used for the EditableHeaderRender.java class
   https://stackoverflow.com/questions/7137786/how-can-i-put-a-control-in-the-jtableheader-of-a-jtable/29963916#29963916

1. Structure

In *MapOptics*, the user must load a file and click Run to load the data. The program is designed to parse and store all the data from the file inputs in each of the Datasets when the Run button is clicked. This means once the data is loaded, it is very fast to navigate through.

1.1. Class layout

The program is laid out in 5 main packages:

1. **Algorithms** – contains all classes that run analysis that can apply to multiple datasets such as calculating overlaps or sorting the overlap of the layout.
2. **DataTypes** – contains the different objects that the data is stored into.
3. **Datasets** – contains different classes for each Dataset. These contain static methods and data HashMaps that are accessed by different areas of the program or to store saved changes.
   a. **Default** – contains classes for datasets used to store the default data read from the file at the beginning of loading the application.
   b. **UserEdited** – contains classes for datasets customised by the user or to store any saved changes from the user.
4. **FileHandling** – contains classes used to read or write any compatible file types.
5. **UserInterface** – contains classes related to the user interface (panels for different visualisation types and main JFrame).

a. **ModelsAndRenderers** – contains classes of all the custom models and renderers set up for the JComponents of the GUI.

1.2. DataTypes

There are 5 different objects (data types) used to store data. 3 data types are used for storing raw file information during file parsing (**AlignmentInfo**, **LabelInfo** and **ContigInfo**). 2 data types are used for storing data for each contig type (**RefContig** and **QryContig**).

AlignmentInfo – all alignment information for one reference-query alignment extracted from the XMAP file.

- String qryAlignStart – start position of alignment region on query contig
- String qryAlignEnd – end position of alignment region on query contig
- String refAlignStart – start position of alignment region on reference contig
- String refAlignEnd – end position of alignment region on reference contig
- String orientation – orientation of query contig against reference contig
- String confidence – confidence value of alignment
- String hitEnum – CIGAR of structural variation of alignment
- String labelChannel – label colour channel for digestion
- String alignment – all the alignments formatted in a string as formatted in the XMAP file to be processed later

LabelInfo – all information for each label extracted from each row of the CMAP file.

- String labelChannel – label colour channel for digestion
- String labelPos – position of label along contig
- String stdDev – standard deviation of interval between current label site and following label site
- String coverage – number of Bionano molecules that overlap the label
- String occurrence – number of molecules with labels aligning to that label
- String chimQuality – percentage of molecules that align to both sides of the label out of all the molecules that align near the label

ContigInfo – all information unique to each contig extracted from each CMAP file.

- double contigLen – length of the contig (in bp)
- LabelInfo[ ] labelInfo – an array of all the labels on the contig and their properties

RefContig – all information relative to reference contig which can be changed by the user when interacting with the data or that is different depending on the View required.

- Rectangle2D rectangle – the rectangle of the reference contig displayed (relative to the size of the panel)
- Rectangle2D[ ] labels – an array of all the rectangle of the reference labels displayed (relative to the size of the panel)
- String[ ] connections – an array of all the query contig IDs that align to this query
- String sequence – FASTA sequence information (added if FASTA is loaded for reference dataset)
- String name – FASTA sequence name (added if FASTA is loaded for reference dataset)
- RefContig copy() – function that copies the current RefContig object and return an exact copy

QryContig – all information relative to query contig which can be changed by the user when interacting with the data or that is different depending on the View required.

- Rectangle2D rectangle – the rectangle of the query contig displayed (relative to the size of the panel)
- Rectangle2D[ ] labels – an array of all the rectangle of the query labels displayed (relative to the size of the panel)
- double qryAlignStart – start position of alignment region on this contig
- double qryAlignEnd – end position of alignment region on this contig
- double refAlignStart – start position of alignment region on reference contig
- double refAlignEnd – end position of alignment region on reference contig
- String orientation – orientation of this contig against reference contig
- boolean reOrientated – whether or not the contig has been oriented from it's original orientation
- String sequence – FASTA sequence information (added if FASTA is loaded for reference dataset)
- String name – FASTA sequence name (added if FASTA is loaded for reference dataset)
- RefContig copy() – function that copies the current RefContig object and return an exact copy

1.3. Datasets

All file information is stored in the Datasets. Each dataset stores the data in HashMaps of contigs or contig information accessed using the contig IDs.

Most Datasets access their **references** HashMap using the reference contig ID which contains **RefContig** objects for all the references.

All Datasets access their **queries** HashMap using the reference ID and query ID formatted as a string “refID-qryID” which contains **QryContig** objects for all the possible alignments. This is because a query contig can be aligned to more than one different reference so each alignment is saved as a different “refID-qryID”.

The only Datasets that also access their **references** hahsmap this way are the query view related datasets, this is as during query view, there are different views relative to the two contigs chosen so this again must be accessed with “refID-qryID”.

Default

1. **RawFileData** – contains HashMaps of the **ContigInfo** and **AlignmentInfo** extracted from each file. Also contains the **references** and **queries** HashMaps where the data is not scaled to match any panel so the rectangles are scaled to the absolute length of the contigs. This is the base of all the datasets.
2. **QueryViewData** – contains **references** and **queries** HashMaps where rectangles are scaled relative to the query view panel. This can’t be altered by the user and is used to reset the data when requested.
3. **RefViewData** – contains **references** and **queries** HashMaps where rectangles are scaled relative to the reference view panel. This can’t be altered by the user and is used to reset the data when requested.
4. **SummaryViewData** – contains **references** and **queries** HashMaps where rectangles are scaled relative to the summary view panel. This can’t be altered by the user.

UserEdited

1. **UserQryData** – contains **references** and **queries** HashMaps where rectangles are scaled relative to the query view panel. This can be altered by the user and is used to reset the data when requested.

2. **UserRefData** – contains **references** and **queries** HashMaps where rectangles are scaled relative to the reference view panel. This can't be altered by the user and is used to reset the data when requested.
3. **SavedQryData** – contains **references** and **queries** HashMaps where rectangles are scaled relative to the query view panel. This is used to store any saved changes.
4. **SavedRefData** – contains **references** and **queries** HashMaps where rectangles are scaled relative to the reference view panel. This is used to store any saved changes.
5. **SearchRegionData** – contains one **RefContig** object and one **QryContig** object which is updated when the user searched a region. This is scaled and positioned so that only the region searched will display on the query view panel.

The interaction between Datasets

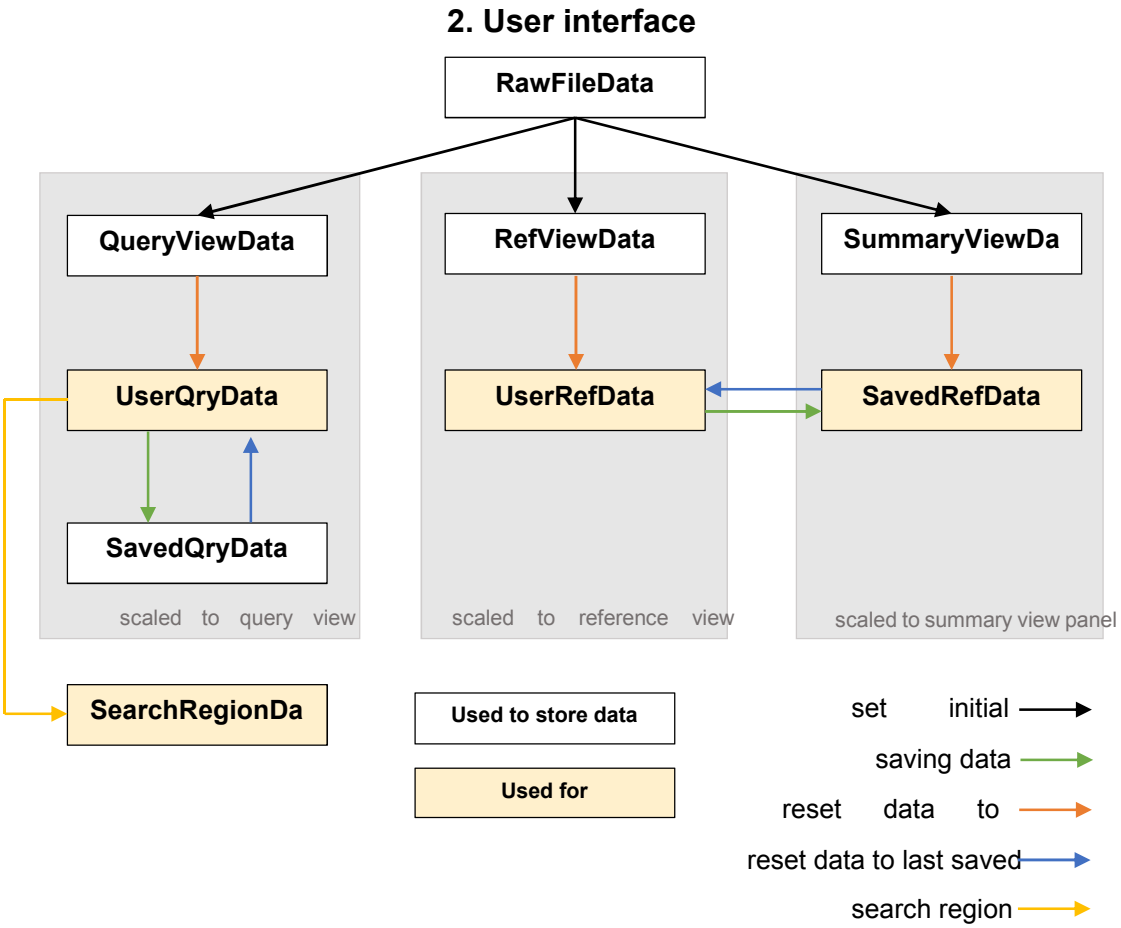


Fig. S1. Diagram showing how the datasets within MapOptics interact. RawFileData processes the raw files and populates the default datasets: QueryViewData, RefViewData and SummaryViewData which are each scaled to display the raw alignment data for each relevant panel. These are not altered and are used to reset data to default when requested. As Reference View and Query View display panels are interactive, UserQryData and UserRefData are used for visualisation and store user-made changes to the relevant view. Any saved changes are stored in SavedQryData, which is used to reset the data to last saved, and SavedRefData which is used to display the changes made in Reference View on the Summary View panel. Finally, SearchRegionData holds data about the contig display if the user searches a region. This is scaled unique to the region zoomed in to and is displayed on the Query View panel, temporarily replacing UserQryData as the class used for visualisation of this panel.

Interactions within the user interface are contained within the **MainInterface** class which extends the JFrame object. The three classes which visualise the alignments are **QueryView**, **ReferenceView**, and **SummaryView** which extend the JPanel object.

2.1 Data visualisation

The main visualisation panels are generated by overriding the paintComponent method. The resulting visualisation is controlled by a series of booleans set up to notify if certain view-types were selected. For example, when **ConfidenceView** is set to **true**, an additional method is called the paint the alignment lines relative to the confidence values of that alignment.

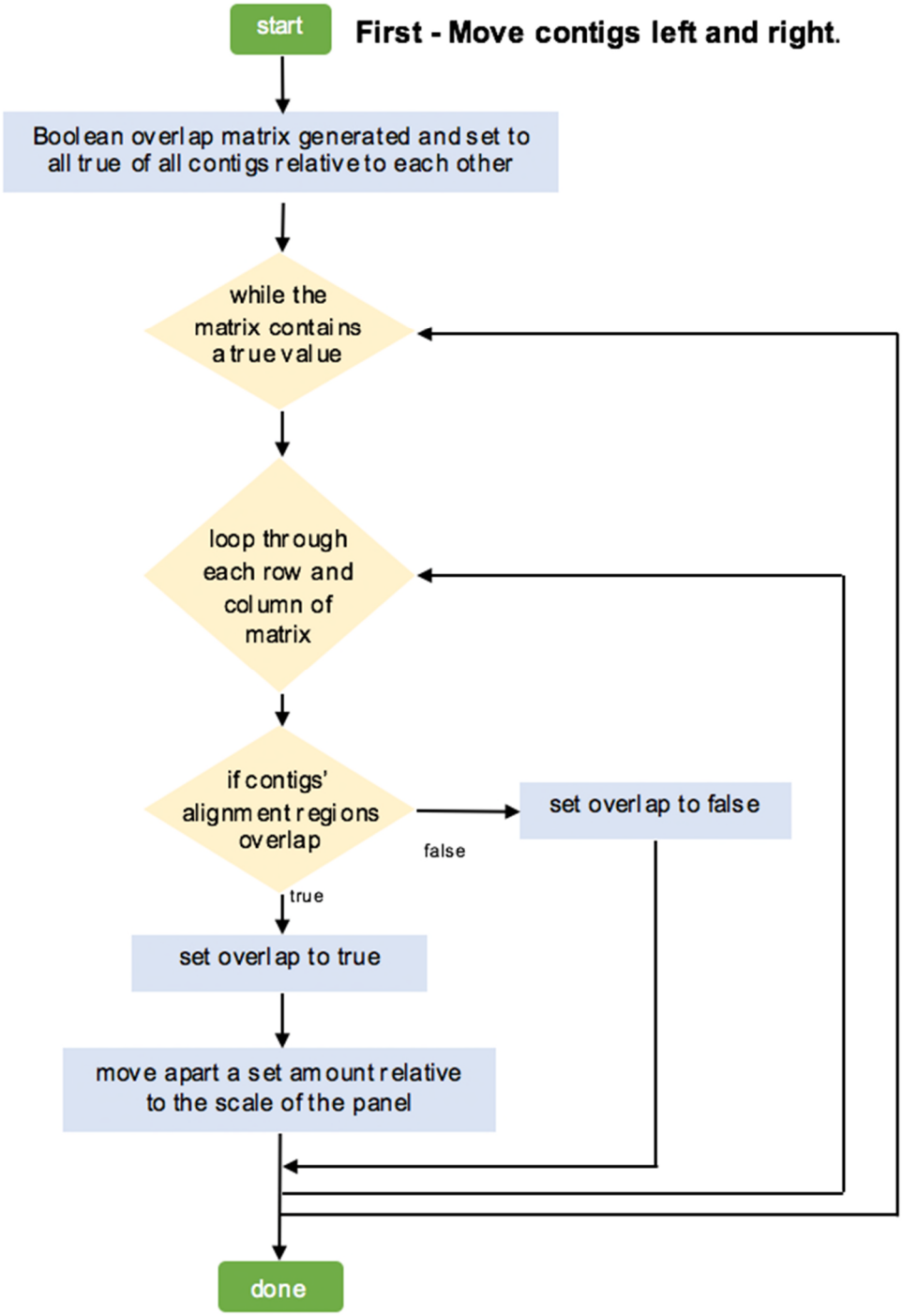
The graphs in the Summary View tab are generated and customised using the JFreeChart library.

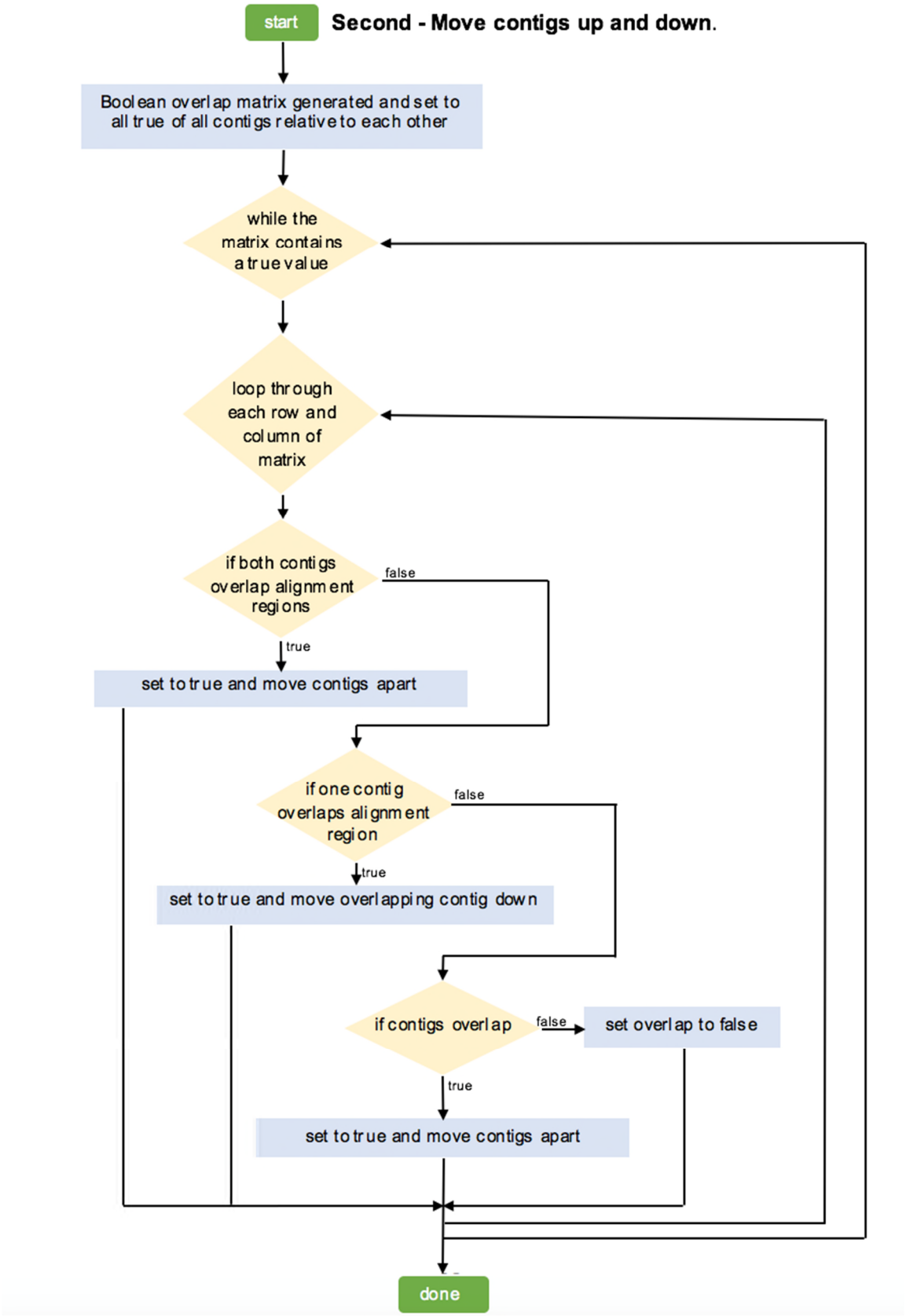
If the user interacts with the panel, such as zoom or drag a contig, the contig that is being interacted with will be translated as required. This means the actual dataset is altered and not just the view. This makes repainting the panel faster.

2.2 Overlap algorithm

The overlap algorithm is the algorithm set up for when the data is first loaded to ensure the contigs are not overlapping in the initial view.

It follows the following workflow stored in the class **Algorithms.SortOverlap**:


start
First - Move contigs left and right.
Boolean overlap matrix generated and set to all true of all contigs relative to each other
while the matrix contains a true value
loop through each row and column of matrix
if contigs' alignment regions overlap
false
set overlap to false
true
set overlap to true
move apart a set amount relative to the scale of the panel
done


start
Second - Move contigs up and down.
Boolean overlap matrix generated and set to all true of all contigs relative to each other
while the matrix contains a true value
loop through each row and column of matrix
if both contigs overlap alignment regions
false
true
set to true and move contigs apart
if one contig overlaps alignment region
false
true
set to true and move overlapping contig down
if contigs overlap
false
set overlap to false
true
set to true and move contigs apart
done

3. File handling

MAPS

1XMAP file and 2 CMAP files are parsed to initially run the program. These are parsed individually and stored in **AlignmentInfo**, **LabelInfo** and **ContigInfo** objects. After this, they are sorted into **RefContig** and **QryContig** objects within the **RawFileData** class.

XMAP reader assumes the *Bionano* format laid out in:

https://bionanogenomics.com/wp-content/uploads/2017/03/30040-XMAP-File-Format-Specification-Sheet.pdf

CMAP reader assumes the *Bionano* format laid out in:

https://bionanogenomics.com/wp-content/uploads/2017/03/30039-CMAP-File-Format-Specification-Sheet.pdf

FASTA and key

A FASTA file can also be parsed and displayed to show the gaps in the initial draft assembly. This requires a key file which is parsed first to get the names of the sequences and what ID the alignment software gave them.

The FASTA file is parsed like a text file with a BufferedReader. If the sequence name in the FASTA does not match a name in the key file, that sequence is not stored. The whole sequence data is first stored with its name as a key in a HashMap. Once the file is parsed, the sequences are added to the relevant **RefContig** or **QryContig** objects.

Only when visualised are the sequences assessed for gaps and the gaps calculated, scaled to view and visualised. This gives future scope for using FASTA sequence data to add more functionalities to the tool.

conflicts_cut_status.txt file

the conflicts_cut_status.txt file can be loaded to help assist in setting up manual conflict resolution. The file is added as rows on a JTable in Reference View as it is parsed. Each line of the file is split into two – one for the reference and one for the query – and is added as two rows on the table. This makes the file clearer for the user to navigate.

The JTable is set so the user can edit the fields and then export the table back into a file. The table is read line by line and written to a text file.

MapOptics: User Guide

MapOptics is a lightweight stand-alone cross-platform visualisation tool for optical mapping alignment. Users can visualise, interact with and export images of the resulting alignment of hybrid scaffolding as well as use this tool to make decisions about manual conflict resolution. It provides a fast, simple alternative to the large in-built optical mapping analysis programs currently available for this area of research.

1 Installation 2

2 Getting Started 2

2.1 Loading maps 2

2.2 Summary View 3

2.3 Reference View 4

1.1. Query View 5

3 Other Functionalities 6

3.1 Available quick-tools 6

3.2 Loading a FASTA file 6

3.3 Manual conflict resolution 8

4 Quick tips and tricks for easy use 9

1 Installation

The easiest way to get started is to download the stable release here. Uncompress the zipped file and run the JAR executable directly.

Alternatively, you can clone the git repository:

```
git clone https://github.com/FadyMohareb/mapoptics
cd mapoptics/dist/
java -jar MapOptics.jar
```

2 Getting Started

MapOptics was designed to be a user-friendly alternative to current optical mapping tools. It is laid out in a simple tabbed fashion with 3 main views: *Summary View, Reference View* and *Query View*. Only three MAP file inputs are required for basic visualisation.

The basic tools and settings are available at the main toolbar. The main interface is navigated by changing the reference and query contig to display their alignments. If you change the reference or query contig in one *View*, it will update for all the *Views*, keeping the navigation easy to follow between the different *Views*.

2.1 Loading maps

Maps can be loaded under *File* > *Load Maps.* Here, the program asks for an “XMAP file”, “Reference CMAP file” and “Query CMAP file”. These files are outputted at various stages of hybrid scaffolding to store information about alignment. As long as the three maps are within the same dataset, they should be visualised with ease. This could be for the display of alignment between your assembly to optical maps, or your assembly to hybrid scaffolds for example.

Where to find maps

If you are using *Bionano’s HybridScaffold* pipeline from the *IrysSolve* package to generate super-scaffolds, the output directory **hybrid_scaffolds/** stores all the maps that can be visualised. Here are some examples of alignments you can display:

To visualise	**Use MAP files ending in**
Bionano maps against draft assembly	*BNGcontigs_NGScontigs*
Draft assembly against super-scaffolds	*NGScontigs_HYBRID_SCAFFOLD*
Bionano maps against super-scaffolds	*BNGcontigs_HYBRID_SCAFFOLD*

XMAP file: this file contains the alignment information and finishes with an ".xmap" file extension.

Reference CMAP file: this contains all the contigs for the "reference" dataset, (those which the queries have been aligned to). This is often noted by the "_r.cmap" file extension.

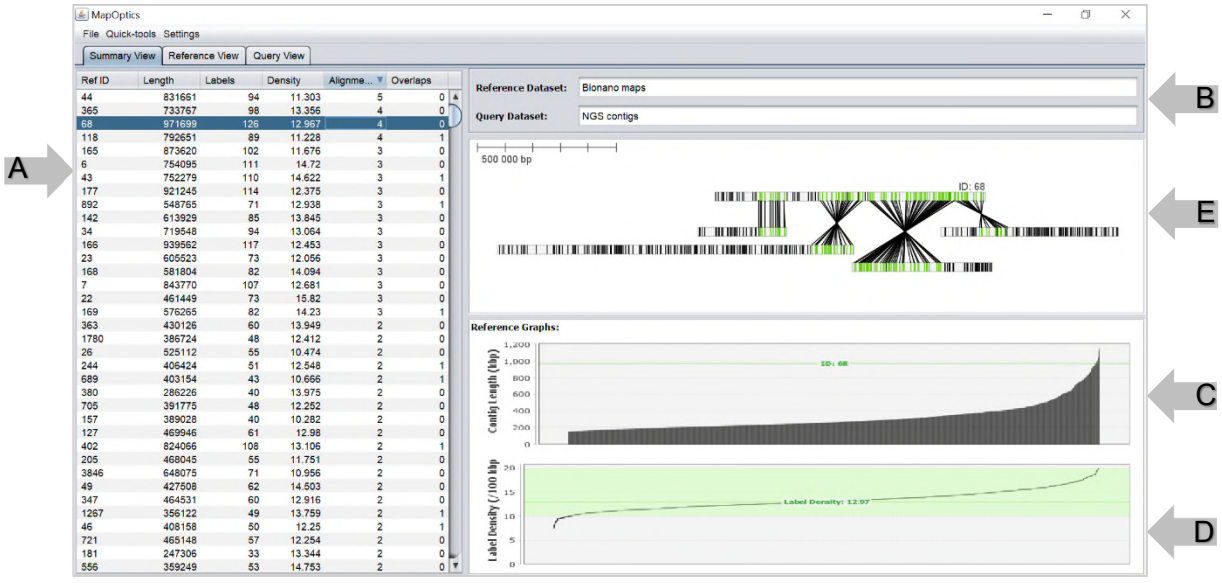
Query CMAP file: this contains all the contigs for the "query" dataset, (those which are aligned to the reference). This is often noted by the "_q.cmap" file extension.

Note: this program is aimed to visualise the alignment during the hybrid scaffolding stage of analysis, not the *de novo* map assembly – loading maps from the *de novo* step of alignment may lead to a crowded and unclear display as these alignments often contains a large number of query contigs aligned to one reference.

Once maps have been loaded, click *Run* and the software should populate the table of reference contigs and draw the reference graphs in *Summary View* so the data can begin to be explored.

2.2 Summary View

Summary View provides a good overview for the user to navigate through the contigs of the reference dataset using the table on the left, to view the alignments and quality of the contigs on the right.



(A) The table of reference contigs should be populated when the maps are loaded. This includes information on the reference contig length, number of labels, label density (/100kb), number of query contigs aligned to the reference and number of query contigs which overlap in regions of their alignment. The table can be sorted in order of any of these fields. When a row of the table is selected, the information of that reference is displayed.

(B) The names of the loaded datasets can be altered in the “Reference Dataset” and “Query Dataset” fields. As there are many different combinations of maps that can be loaded, this allows the user to keep track of what datasets they are displaying.

(C) This graph shows the distribution of reference contig lengths, highlighting the chosen contig in the distribution.

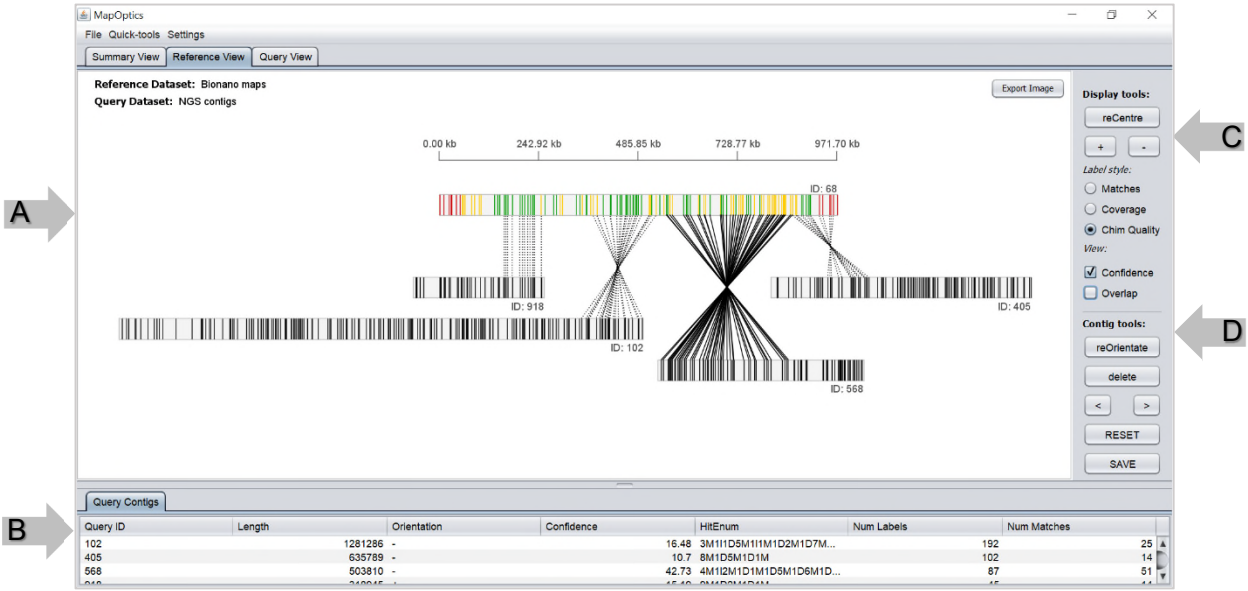
(D) This graph shows the label density distribution of the reference contigs, highlighting the chosen contig in the distribution with its label density value. Ideally label density should fall between 10 and 20 labels per 100kb.

(E) This panel displays a simple diagram of all the query contigs that are aligned to the reference contig, with the orientation of their alignment. This view can be changed and updated from Reference View.

If a user wishes to explore the alignments of a reference contig in more detail, they can do so in Reference View.

2.3 Reference View

Reference View gives the user more of an interactive view of the contig alignment with more information on aspects of their quality.



(A) The display generated for Reference View is similar to that of Summary View but is more interactive. Here, the user can drag the contigs and screen so as to centre the view as required.

(B) The query contigs that align to this reference should populate the table below the display. This shows information on their ID, length, orientation, confidence in the alignment, the CIGAR of the structural variation in the alignment, the number of labels and the number of labels that match within the alignment. Here, contigs can be selected and highlighted in the view or explored in more detail in *Query View*.

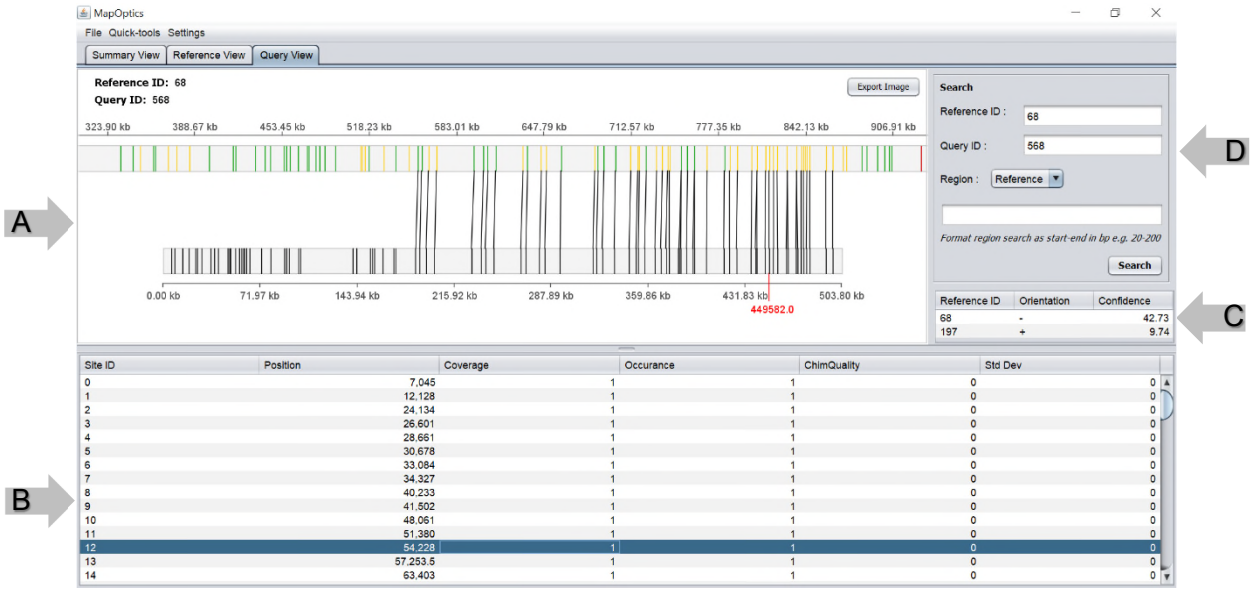
(C) This is the “Display tools” portion of the tool bar. This is for customising the display. The user can zoom in on the view and *reCentre* to re-size the display to show all contigs. The Label style can be recoloured to display the labels that match, the coverage values of the labels or the chimeric quality of the labels. *Confidence* view can be selected, this changes the alignment lines to match the confidence in the alignment (solid line = high confidence, dashed line = medium confidence, dotted line = low confidence). *Overlap* view can be selected to display the regions in which query contigs overlap in their alignment, highlighted in yellow.

(D) This is the “Contig tools” portion of the tool bar. This is for manipulating the contigs in the display and saving their positioning. They can be reoriented in the display, position the contig to match the left or rightmost part of the alignment, can be deleted from the display and the positioning of the contigs can be saved to the Summary View for ease of navigation. The changes can be reset to default or last saved if the user is unhappy with changes made.

If the user wishes to explore a single alignment between one reference and one query contig, a query contig can be selected (from the display or table) and then explored in Query View.

1.1. Query View

Query View shows the most in detail view of one alignment. Here the user can see a simplified view of one alignment and navigate through all the labels on the query contig.



(A) The display generated for Query View is not as interactive as Reference View. By default, the query contig is oriented positively and positioned to match the leftmost alignment. The display settings match those set in Reference View (e.g. label style).

(B) This table allows the user to navigate and highlight the labels on the query contig. This table contains information of the label’s position, coverage, occurrence, chimeric quality and standard deviation.

(C) This additional table shows if the query contig has been aligned to other references in addition to the one displayed. These can be navigated through and all the equivalent views will be updated to match this reference (including Summary View and Reference View).

(D) In this view, the user can search for any reference or query to be displayed (a message will be displayed of there is no alignment found between the two). There is also the option to zoom into a region of choice in either the reference or query contig. This can be of use when using a small screen, working with very long contigs or when FASTA file information is loaded and you would like to explore the position of the gaps in more detail.

3 Other Functionalities

There are also some additional functionalities to the application to further enhance the functionality of *MapOptics* within your hybrid scaffolding research.

3.1 Available quick-tools

Some quick tools are available under *Tools* in the top toolbar.

Tools > Orientate all contigs	Orientates the display of all contigs to be positive.
Tools > Save view of all contigs	Saves any changes performed in *Reference View* to *Summary View.*
Tools > Swap query and reference	Swaps the two datasets so the query dataset becomes the reference dataset and the reference dataset becomes the query dataset.

3.2 Loading a FASTA file

To explore the alignment of the maps in more detail, a FASTA file can be loaded to display the gaps in the sequence in Query View- this can help you recognise if what appear to be misalignments are actually caused by regions of unknown sequence.

/home/josie/irysSolveResults/recommended/fa2cmap/

FASTA and Key files can be loaded under *File > Load FASTA and Key files.* Here you must upload two files:

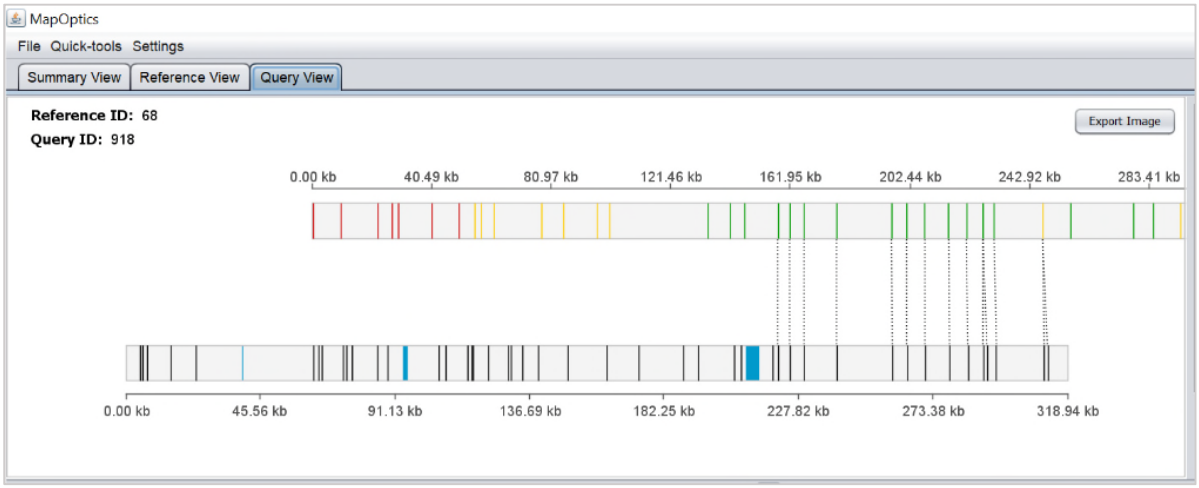
(1) The FASTA file of the original assembly that you're hybrid scaffolding.
(2) The key file (with the extension _key.txt). This contains all the names of the FASTA contigs and the IDs that were given to them during hybrid scaffolding.

Where to find the key file

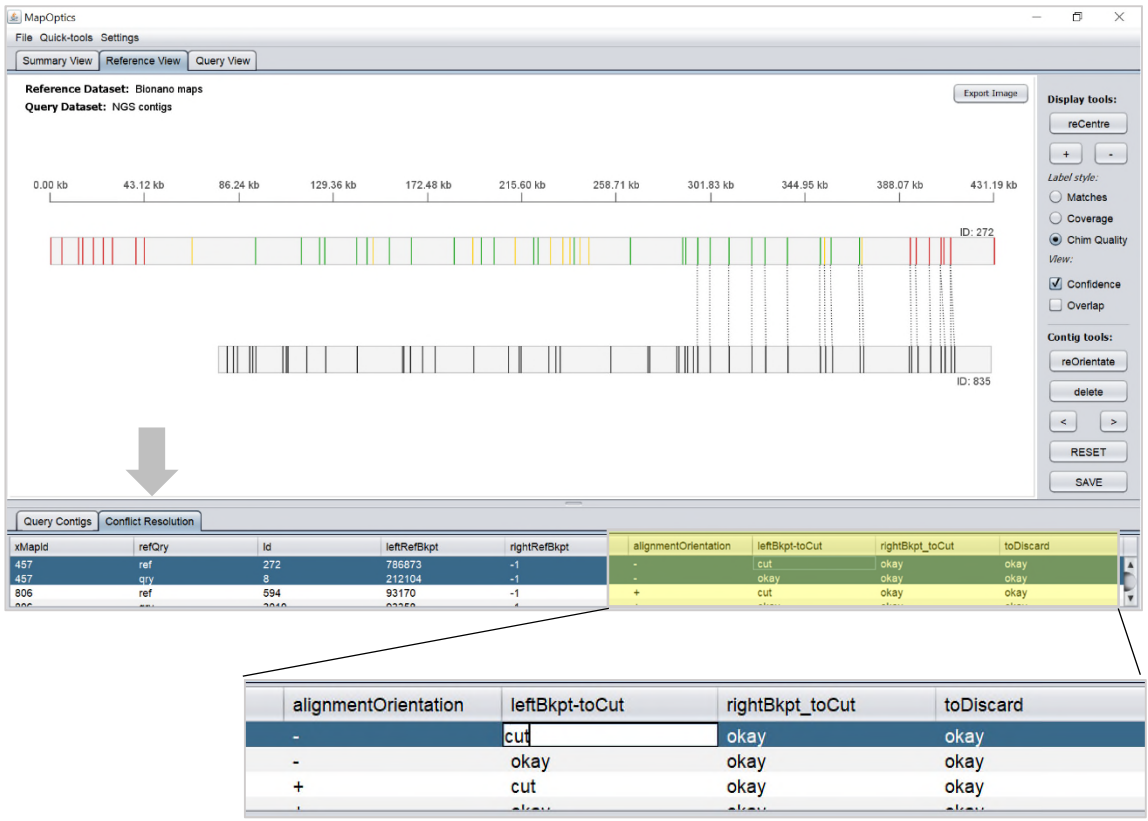
If you are using *Bionano's HybridScaffold* pipeline from the *IrysSolve* package to generate super-scaffolds, the output directory **fa2cmap/** stores all the files outputted when the FASTA is converted into a CMAP format. The key file can be found within this directory.

Both must be loaded for any gap information to be displayed.

Finally, the user must select which dataset the FASTA file is in regards to, the “reference” or “query” to ensure it is visualised properly. If this is incorrectly selected, the program won’t necessarily give an error, but the gaps will be visualised incorrectly leading to incorrect conclusions in an analysis.



The resulting gaps are displayed in blue. If you wish to explore these in more detail, you can search for the region where there are gaps.



TIP: If you are not sure which is your NGS data and which are your BNG maps, click *Coverage* as your Label Style, only the BNG contigs should show coverage data as the NGS maps are not assembled with bionano molecules and instead are digested *in silico*.

3.3 Manual conflict resolution

To easily navigate through the contigs and display their alignments to produce a conflicts_cut_status.txt file for manual conflict resolution, first you must load the conflicts_cut_status.txt file from the default run of hybrid scaffolding.

Where to find the conflicts_cut_status.txt file

If you are using *Bionano's HybridScaffold* pipeline from the *IrysSolve* package to generate super-scaffolds, the output directory **assignAlignType/cut_conflicts/** stores all the files outputted when the FASTA is converted into a CMAP format. The conflicts_cut_status.txt file can be found within this directory.

This is done under *File* > *Manual Conflict Resolution* > *Load conflicts_cut_status file.* On loading of this file, an additional tab will appear next to the query contig table in Reference View.

Here, you can navigate through the file, updating the reference and queries viewed to match those in the conflicts table. You can alter any fields in the file as necessary for your manual conflict resolution and this can be exported as a new conflicts file under *File* > *Manual Conflict Resolution* > *Save manual conflict resolution file.*

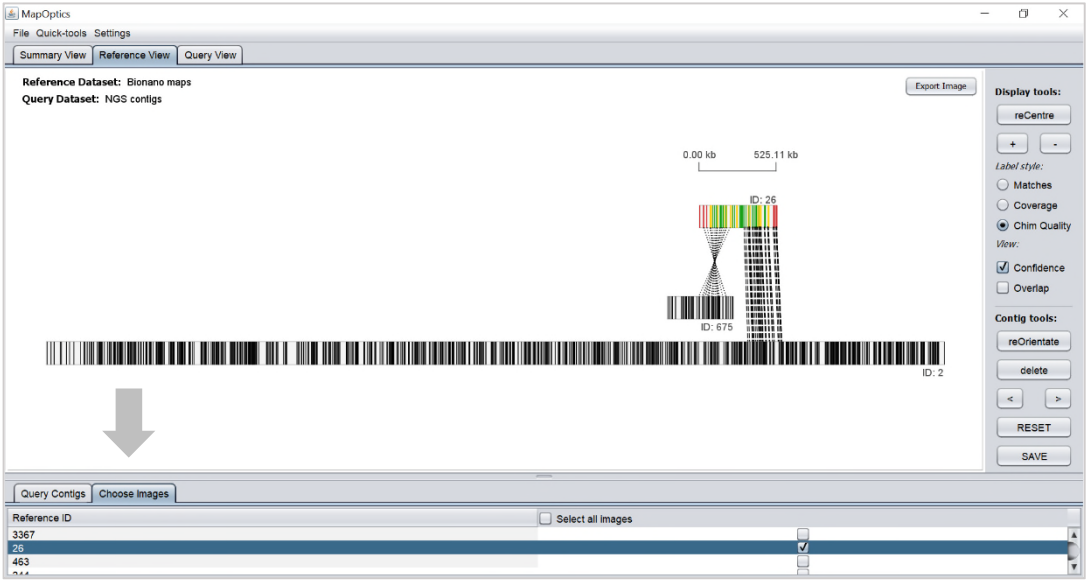
TIP: For guidance on how to alter the fields for manual conflict resolution, consult the theory of operation documentation for hybrid scaffolding provided by *Bionano*: **Bionano Solve Theory of Operation: Hybrid Scaffold.**

3.4. Exporting images

To export a single image of a panel, click the *Export Image* button in the corner of the panel. This functionality is available for the display on Reference View and Query View.

To export more than one alignment image from Reference View, open the Choose Images tab accessed from *File* > *Export Images* > *Choose images to export.* In Reference View, the tab should now be accessible next to the table of query contigs.

This shows a simple table of the reference contig IDs and a tick box that selects the view as one to be exported. You can change the view and explore that contig's alignment by selecting a contig much like in Summary View.



To then save the chosen contigs go to *File* > *Export Images* > *Export chosen images.* Here you can create a directory for all the images to be saved in.

4 Quick tips and tricks for easy use

(1) If you are not bothered about contig orientation in your analyses, it can make the view more complex than necessary. Use the quick-tool *Orientate all contigs* followed by *Save view of all contigs.* This will reorient all inverse contigs and save them to the easy-to-navigate Summary View. If you still want to know the initial orientation of the contigs, this is available in the table in Reference View.

(2) If you load a dataset and the reference and query contigs are the opposite way around than you would like (for example, the visualisation is the BNG maps aligned to the NGS contigs when you would like to see the NGS contigs aligned to the map), you can use the quick-tool *swap reference and query*. It is advised to do this at the beginning of the visualisation as changes made won't be saved.

(3) If you want to export images of all the alignments, open the *Choose Images* tab in Reference View by accessing *File* > *Export images* > *Choose images to export.* Here, click the *select all* checkbox and export all the images by accessing *File* > *Export images* > *Export chosen images.* All the images will be exported, but be patient as the program may become unresponsive while this is processing.

(4) If you wish to compare two different alignments, open the application twice and load the different datasets in each application. If you put each window side by side, you can navigate the two applications simultaneously.

https://dspace.org/

MapOptics: A light-weight, cross-platform visualisation tool for optical mapping alignment

Burgin, Josephine; Molitor, Corentin; Mohareb, Fady R.

2018-12-07
Oxford University Press
https://doi.org/10.1093/bioinformatics/bty1013
Josephine Burgin, Corentin Molitor and Fady Mohareb. MapOptics: A light-weight, cross-platform visualisation tool for optical mapping alignment. Bioinformatics, Volume 35, Issue 15, 1 August 2019, pp. 2671–2673

Downloaded from a DSpace repository. https://dspace.org/