Graph of Sequences Viewer Users Guide

Overview

Graph of Sequences Viewer (GSV) is a tool based on cytoscape.js. GSV is dedicated to visualize graphs representing textual sequences with additional informations. It is well suited for visualizing genomic sequences, in particular the assembly graph obtained from a compacted de Bruijn graph in which the coverage and quality are stored. Each sequence is represented by a node. Each node stores explicitly a genomic sequences and implicitly its reverse complement. An overlap (of size k-1 with respect to the de Bruijn Graph) between two sequences is represented by a directed edge. Each directed edge is labeled with two letters. The first letter indicates the associated sequence of the source node: the explicit sequence (forward='F') or the implicit sequence (reverse='R'). The second letter indicates the sequence of the target node also by a 'F' or a 'R'. Thus each edge is labeled with "FF", "RR", "FR" or "RF". Note that, by construction, if an edge goes from node A to node B labeled respectively FF, RR, FR or RF, then another edge goes from node B to node A labeled respectively RR, FF, FR or RF. GSV shows only one of the two edges.

Any traversal of the graph must respect the traversed edge labels (for instance a node considered as forward cannot be left as "reverse"). With this constraint, GSV allows to check possible paths in the graph and to generate the corresponding sequence. This is useful while reconstructing locally and manually genomic parts of interest.

GSV includes a vizmapper that applies graphical styles (shape, color, size) on graph elements depending on its data properties (sequence length, average coverage, ...). Each element of the graph is clickable, allowing to see various information in a retractable panel. This panel has several functions, especially for nodes, for use the sequences displayed like concatenation, comment and highlight. It is possible to export all sequences displayed in this panel (nodes sequence and concatenated sequences) in text files.

This tool has been initially developed for visualization of the JSON output of Masembler2. Mapsembler2 input are one or more starters (references sequences) and a set of reads used to extend these starters. The result can be a JSON file that contains for each starter several graphs in which one node is a starter. The other nodes are unitigs or contigs (depending on the user choice) connected together with respect to their overlaps. GSV can also visualize simple JSON that is not an output of Mapsembler2. This JSON must contain, at least, a graph description with nodes and edges (see details of simple JSON format on part 3.1).

Compatibility table

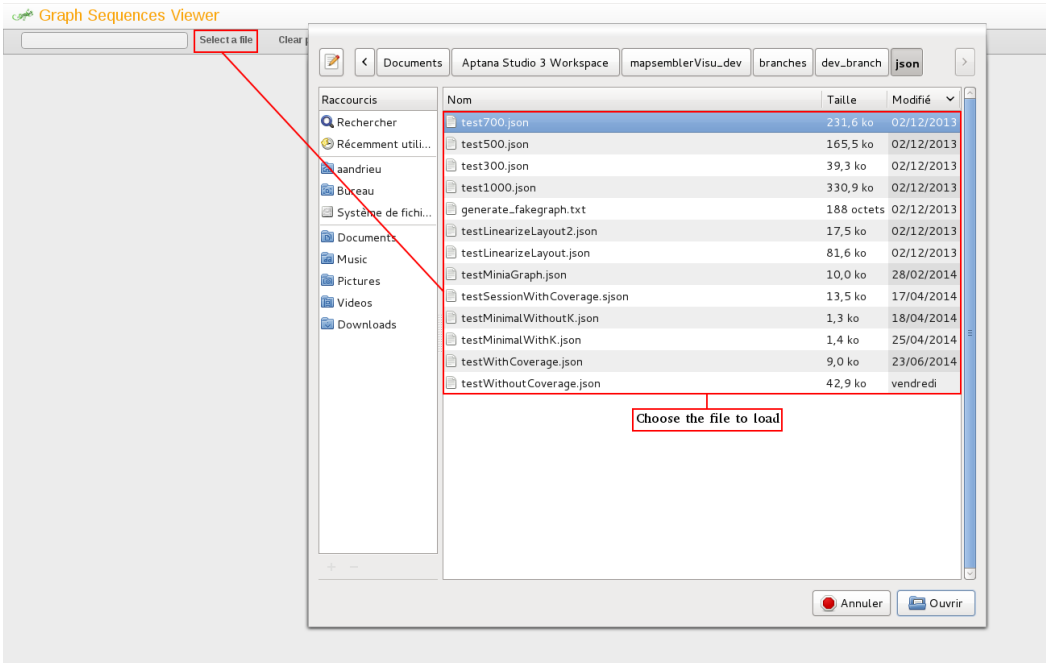
 4.0+  5.0+  5.1.7+  9.0+  11.6+

Compatibility table is valid on MacOS X, Windows and Linux.
Tests had been performed on :
- Windows 7 with IE 10, IE 11, Firefox 26, Chrome 31, Opera 12.16
- MacOS X Mountain Lion with Safari 6.0.2, FireFox 26, Firefox 27, Chrome 30, Chrome 31 and Opera 12.16
- MacOS X Maverick with Safari 7.0, FireFox 26, Firefox 27, Chrome 31
- Fedora 17 with FireFox 15, Chrome 22
For others compatibilities problems or questions, please contact authors of the tool.

1. Start Page (Using a JSON generated by Mapsembler2)

1.1. Load a File

The start page allows loading a graph file (.json) or a session file (.sjson). Any compatible file save in your hard drive can be open.

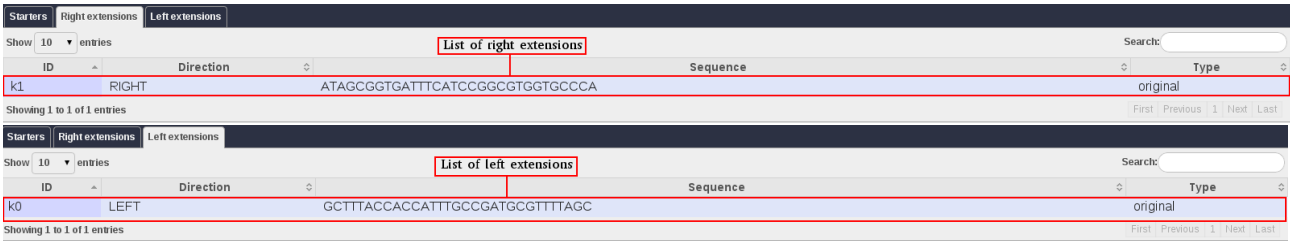


a. Starters

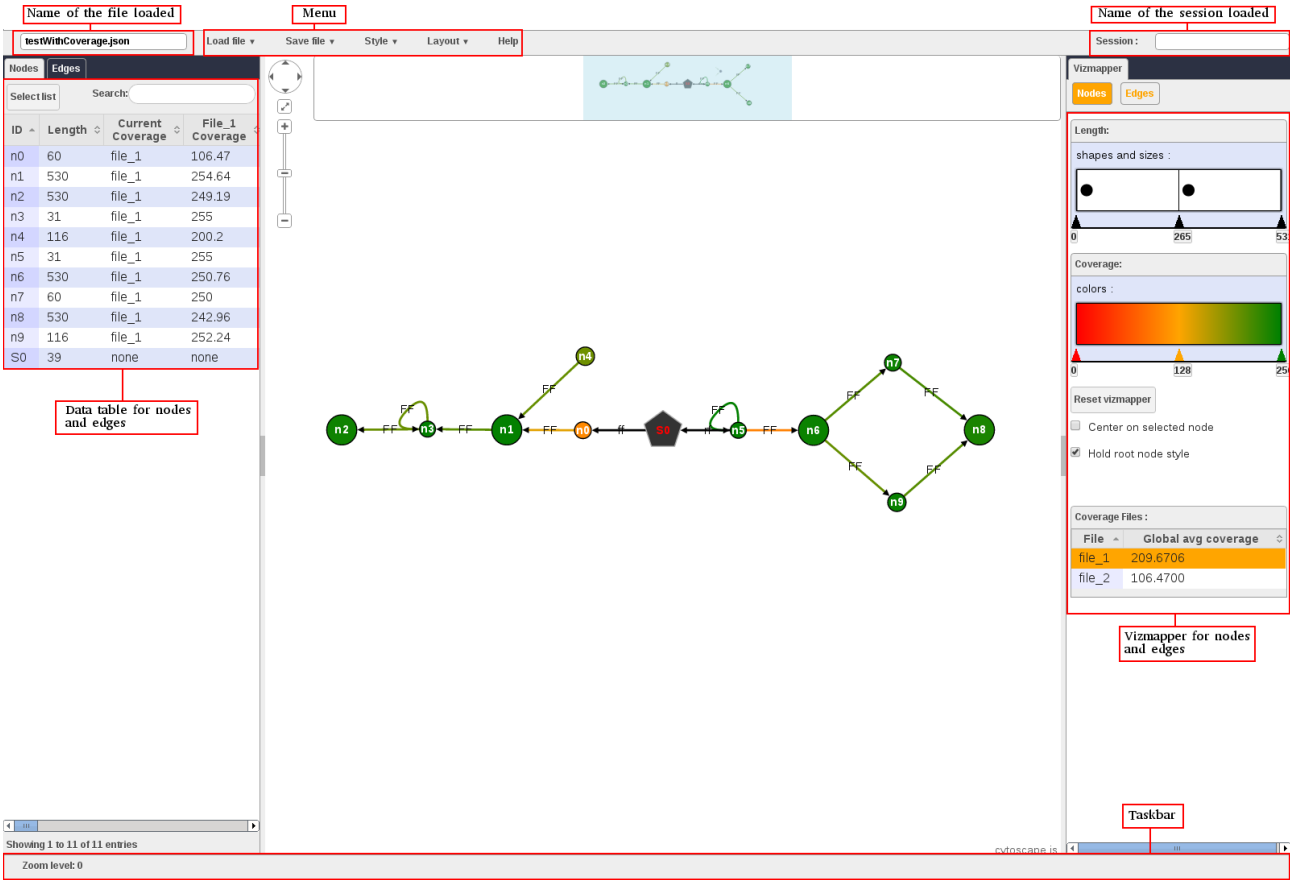
When output JSON of Mapsembler 2 is loaded, a table of starters has been displayed (several starters are possibly stored in a unique JSON file).

Click on one of them selecting and displaying extensions tables in a new internal tab.

b. Extensions



Select a right extension and a left extension displaying the graph viewer in a new internal tab.

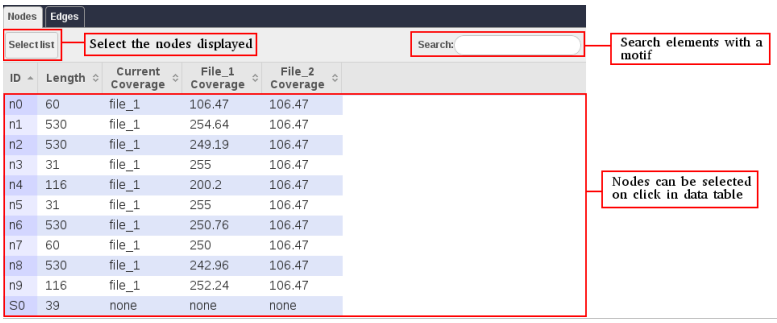


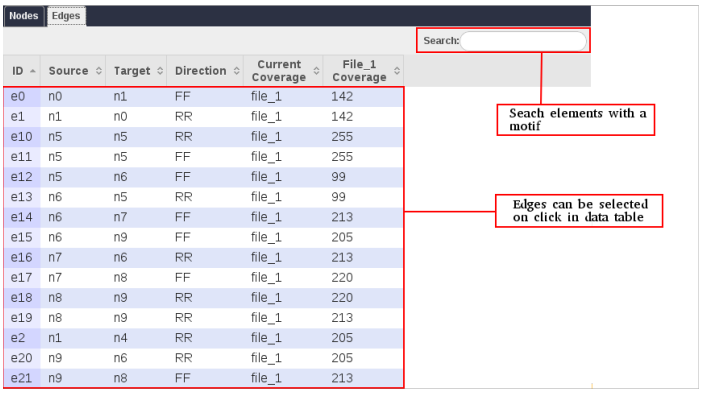
2. Graph viewer

2.1. Data tables

In the graph viewer, left panel contains nodes' data table and edges' data table in two tabs. In this area, data elements are displayed:

- For nodes : ID, length, sequence and average coverage (if present in JSON).
- For edges: ID, source, target and average coverage (if present in JSON).





Select/unselect one of this elements in table select/unselect this element in the graph. Multiselection is possible with hot key "Shift+Left Click". Select an element display a retractable bottom panel where several data of the elements are shown (like id, sequence, coverage files, current average coverage, ...). The search functionality allows to find an elements by its ID or to find nodes with a specific (sub)sequence. The button "Select list" in "Nodes" tab allows to select all nodes displayed in the data table. If the motif is a part of a sequence, this motif will be highlighted in the sequences displaying in bottom panel.

2.2. Vizmapper

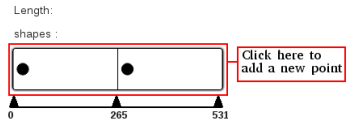
The vizmapper allows to define styles with respect to the length and average coverage properties. For nodes, it is possible to define the shape and the size with respect to length. If average

coverage is present, edges tab appears and it is possible to define color with respect to the average coverage for nodes and edges. Each cursor defines a point in the distribution of nodes length or nodes/edges average coverage. By default, the style of the root (node containing the starter) is locked.

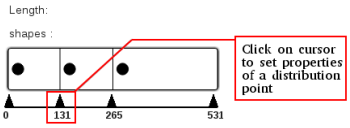
Interestingly, coverage property from several read sets can be stored in the JSON file. In this case, the user may choose (and change at any time) which read set is selected while using the vizmapper (see Table of coverage files section).

a. Length property

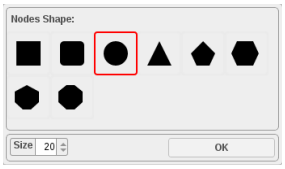
Points in distribution of length sequence define points in distribution of shapes (discrete) and in distribution of size (gradient). By default, shape is round for all nodes except the root and the distribution of sizes contains three points with the size :10px (minimum length), 40px (medium length), 70px(maximum length). For example, with the default values of size, nodes have length sequence between 0 and median values have a size values between 10px and 40px. Size values grow up gradually with values of length sequence.



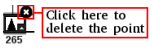
Click on preview of distribution add a new point (and a new cursor) in the distribution.



Click on a cursor display a selector to allow setting the properties (shape, size) of the distribution point.

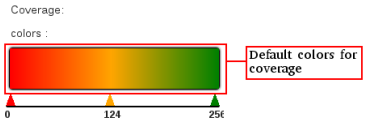


This action displays a frame with a cross around the cursor too. Click on the cross delete this distribution point (and the cursor).

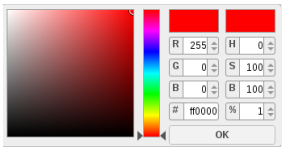


b. Coverage property

If coverage is present, points in distribution of average coverage define points in distribution of colors (gradient). By default, the distribution of colors contains three points : red (poor coverage), orange (medium coverage), green (good coverage).

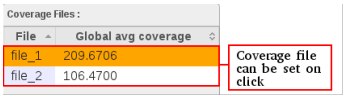


Similarly to length property, it's possible to add, remove and set the style (here the color). For example, with the default value of color, nodes have an average coverage between 0 and median values have a color values between red and orange with a gradient transition between these.



c. Table of coverage files

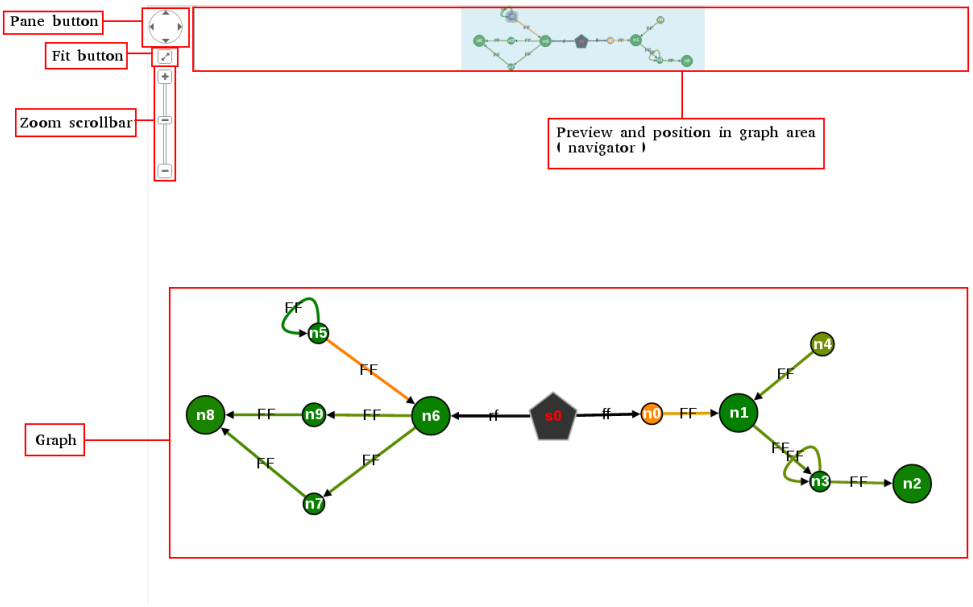
The table of coverage files allows to check the coverage files using for visualizing the graph. The orange one is the selected one. This can be changed at anytime, updating the graph visualization properties.



2.3. Graph panel

The graph viewer allows to:

- Select one element on click
- Select several elements with the hot-key "Shift+Left Click" or "Left Click pushed + Mouse move" (selection box)
- Move nodes with drag and drop
- Zoom/De-zoom with mouse wheel, click on the zoom scroll bar
- Navigate with "long Left Click pushed (1 sec) + Mouse move" or with navigation button
- Preview the global graph in navigator frame
- Fit the graph with fit button



a. Load file

Load file button allows returning to start page to set starter and extensions, and allows loading a previous session file (.sjson) or a vizmapper configuration file(.vjson) .

b. Save file

A Save file button allows saving the current session in a SJSON file or a vizmapper configuration in a VJSON file.

c. Style

The style button allows showing/hiding labels of graph elements (nodes and edges).

d. Layout

The layout button allows recalculating the current layout (reset), and set type of layout. Ten types of layout are available:

- Null : all nodes in the same positions
- Preset : fit nodes
- Random : nodes with random positions
- Grid : nodes into a grid
- Arbor : force directed layout
- Circle : nodes on a circle
- Concentric : nodes in several circles
- Cose : group by nodes similarities
- BreadthFirst : layout with breadth first algorithm
- Hierarchical : hierarchical layout (in dev)

2.4. Data viewer

Select nodes displaying a bottom panel that contains the properties of the elements selected. The panel gives:

For edges : ID, source, target, average coverage.

```
ID: e9 Source: n4 Target: n1 Coverage: 205
```

For nodes : ID, sequence length, average coverage and sequence. The panel has a menu and display the interval of selection for the current node.

The panel is close, after elements are unselected. The hold button allows keeping open the panel after elements are unselected.

a. Sequence format

It's possible to set the format of sequence(s) displayed (Set button). Four formats are available:

FASTA

```
ID:n7 Length:60bp Coverage: 250
GAACACGAACTCAGCTATGACACTTATGCATTTGGCATACTGGCCGCTTGCGTCTTACCG
```

CODATA

```
ENTRY       n7
SEQUENCE
            5         10        15        20        25        30        35        40        45        50        55
      1  G A A C A C G A A C T C A G C T A T G A C A C T T A T G C A T T T G G C A T A C T G G C C G C T T G C G T C T T A C C G

///
```

PRIDE

```
00001  GAACACGAACTCAGCTATGACACTTATGCATTTGGCATACTGGCCGCTTGCGTCTTACCG  00060
```

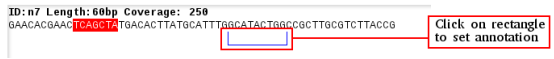
RAW

```
gaacacgaactcagctatgacacttatgcatttggcatactggccgcttgcgtcttaccg
```

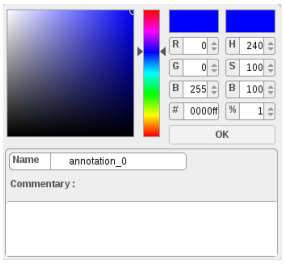
b. Annotation and highlight

It's possible to spot part(s) of sequence(s) with highlight or annotation (Add button):

- To add highlight select part of the sequence and click on "Highlight". The selected part of the sequence become highlighted with red color.
- To add an annotation select part of the sequence and click on "Annotation". Under the part of selected sequence a rectangle is displayed.



Click on the annotation rectangle displaying a color selector with a text area to define the name of the annotation and a commentary.



After the selector is hidden, name and commentary are showing on mouse over annotation rectangle.

The highlight and annotation are persistent, so if nodes are unselected these have not been loose.

To remove a highlight, just select it and click on "Highlight" in "Remove" menu and to remove an annotation click on the crux displayed on mouse over on it. It's also possible to remove all highlight or/and all annotations displayed (Remove menu).

c. Export

The export function allows generating and download a text file, contains all sequences displayed in the bottom panel, in the selected format.

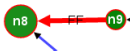
d. Concatenation

The concatenation function allows concatenating sequences of two nodes or more(available only for output JSON of Mapsembler 2 or minimal JSON with k in data nodes). This function involves some constraints to allow or disallow the concatenation of sequences. It's important to know that if two nodes are linked by an edge, there are n characters similar (overlap) in the two sequences:

- Edge direction is FF (forward-forward): The last k-1 characters of the sequences of nodes source are identical to the first k-1 characters of the sequence of the target nodes.
- Edge direction is RR (reverse-reverse): The last k-1 characters of the reverse complement of the sequences of nodes source are identical to the first k-1 characters of the reverse complement of the sequence of the target nodes.
- Edge direction is RF (reverse-forward): The last k-1 characters of the reverse complement of the sequences of nodes source are identical to the first k-1 characters of the sequence of the target nodes.
- Edge direction is FR (forward-reverse): The last k-1 characters of the sequences of nodes source are identical to the first k-1 characters of the reverse complement of the sequence of the target nodes.

To concatenate two nodes the order of click is very important because only just one direction is shown on the graph but in some cases another direction exists. (This can be checked in the edges' data table).

So for example if click on n9 nodes first, and after click on n8. The direction followed is FF.

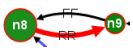


ID:n9-n8 Length:616bp Coverage:
GAACACGAACTCAGCTATGACACTTATGCAAACCACAGGCTCCATGATACCCCAGGAGCTGCCAAGCTCCGGGGGTTGGGTACAAGTTTGGCATACTGGCCGCTTGCGTCTTACCGAAAGTTATTCAGCGAGACTAACGCGGAGACCCATACTCCCTGGAAAAGAGTCATGGCAGTTTGATCTATAACAGATCC
CTCTTTGCTACCGTTTACTTCGATGATGATACCGGGCACTACCGAACGAGGTAGAGCGCACTAGCGTAACCTGATGTATTCACTGACCCCGAGTTTTACATCGTAGGATGCCTTGGCCCGAAGGTCCGTGAGTGAGAATTATCCTCCCTTATTCGTTATTTCACTTTCTGCAATGAATAAATCTGAGCAGTAGC
TCCGCACAGAGACCATGTTAAGTTTTAAAGATGTTCCCCATACCACGATCCGCCCCCGTGCCCGTGCCTCCCGCTACAATTTTGGATTGAACCCAGACATAACGTACGGGAGAGAATATGGAAATTGACAGCTTTCCGTATCTTTCTGAAAGTGAAACGGCATAGTTGTGAACCGCCAATACACTAATCCTAGA
GTACTTTTTTTTTTTTTTTTTTTTTTTTTTTTTT

ID:n9 Length:116bp Coverage: 252.24
GAACACGAACTCAGCTATGACACTTATGCAAACCACAGGCTCCATGATACCCCAGGAGCTGCCAAGCTCCGGGGGTTGGGTACAAGTTTGGCATACTGGCCGCTTGCGTCTTACCG

ID:n8 Length:530bp Coverage: 242.96
TTTGGCATACTGGCCGCTTGCGTCTTACCGAAAGTTATTCAGCGAGACTAACGCGGAGACCCATACTCCCTGGAAAAGAGTCATGGCAGTTTGATCTATAACAGATCCCTCTTTGCTACCGTTTACTTCGATGATGATACCGGGCACTACCGAACGAGGTAGAGCGCACTAGCGTAACCTGATGTATTCACTGA
CCCCGAGTTTTACATCGTAGGATGCCTTGGCCCGAAGGTCCGTGAGTGAGAATTATCCTCCCTTATTCGTTATTTCACTTTCTGCAATGAATAAATCTGAGCAGTAGCTCCGCACAGAGACCATGTTAAGTTTTAAAGATGTTCCCCATACCACGATCCGCCCCCGTGCCCGTGCCTCCCGCTACAATTTTGGA
TTGAACCCAGACATAACGTACGGGAGAGAATATGGAAATTGACAGCTTTCCGTATCTTTCTGAAAGTGAAACGGCATAGTTGTGAACCGCCAATACACTAATCCTAGAGTACTTTTTTTTTTTTTTTTTTTTTTTTTTTTT

But if click on n8 before n9, the direction follows is RR.
So the result of concatenation will be different.



ID:n8-n9n8-n9 Length:616bp Coverage:
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAGTACTCTAGGATTAGTGTATTGGCGGTTCACAACTATGCCGTTTCACTTTCAGAAAGATACGGAAAGCTGTCAATTTCCATATTCTCTCCCGTACGTTATGTCTGGGTTCAATCCAAAATTGTAGCGGGAGGCACGGGCACGGGGGCGGATCGTGGTATGGGGA
ACATCTTTAAAACTTAACATGGTCTCTGTGCGGAGCTACTGCTCAGATTTATTCATTGCAGAAAGTGAAATAACGAATAAGGGAGGATAATTCTCACTCACGGACCTTCGGGCCAAGGCATCCTACGATGTAAAACTCGGGGTCAGTGAATACATCAGGTTACGCTAGTGCGCTCTACCTCGTTCGGTAGTGCC
CGGTATCATCATCGAAGTAAACGGTAGCAAAGAGGGATCTGTTATAGATCAAACTGCCATGACTCTTTTCCAGGGAGTATGGGTCTCCGCGTTAGTCTCGCTGAATAACTTTCGGTAAGACGCAAGCGGCCAGTATGCCAAACTTGTACCCAACCCCCGGAGCTTGGCAGCTCCTGGGGTATCATGGAGCCTGT
GGTTTGCATAAGTGTCATAGCTGAGTTCGTGTTC

ID:n9 Length:116bp Coverage: 252.24
GAACACGAACTCAGCTATGACACTTATGCAAACCACAGGCTCCATGATACCCCAGGAGCTGCCAAGCTCCGGGGGTTGGGTACAAGTTTGGCATACTGGCCGCTTGCGTCTTACCG

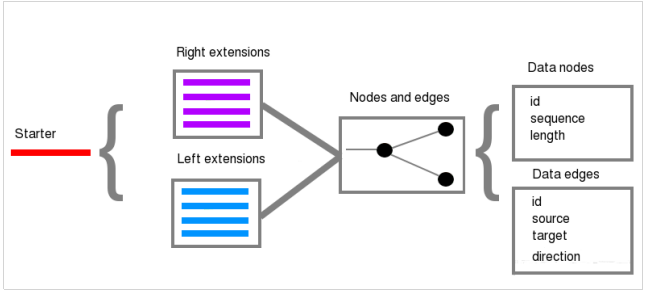
ID:n8 Length:530bp Coverage: 242.96
TTTGGCATACTGGCCGCTTGCGTCTTACCGAAAGTTATTCAGCGAGACTAACGCGGAGACCCATACTCCCTGGAAAAGAGTCATGGCAGTTTGATCTATAACAGATCCCTCTTTGCTACCGTTTACTTCGATGATGATACCGGGCACTACCGAACGAGGTAGAGCGCACTAGCGTAACCTGATGTATTCACTGA
CCCCGAGTTTTACATCGTAGGATGCCTTGGCCCGAAGGTCCGTGAGTGAGAATTATCCTCCCTTATTCGTTATTTCACTTTCTGCAATGAATAAATCTGAGCAGTAGCTCCGCACAGAGACCATGTTAAGTTTTAAAGATGTTCCCCATACCACGATCCGCCCCCGTGCCCGTGCCTCCCGCTACAATTTTGGA
TTGAACCCAGACATAACGTACGGGAGAGAATATGGAAATTGACAGCTTTCCGTATCTTTCTGAAAGTGAAACGGCATAGTTGTGAACCGCCAATACACTAATCCTAGAGTACTTTTTTTTTTTTTTTTTTTTTTTTTTTTT

Be careful on unselected nodes, concatenation disappears and all annotations or highlight are loosed (for concatenated sequence). If two (or more) nodes can't be concatenated, for example tow nodes not linked, an error message has been displayed in the corner right of the application.

3.1. Graph File

The output JSON of Mapsembler 2 has a particular structure and can include several graphs. For each starter, there are, at least, one or more right extensions and one or more left extension. Mapsembler output looks like:

Schema :



- Starter: reference sequence that must be extended.
- Right extensions: define all the possibilities of extension for the right side of a selected starter in the form of a graph.
- Left extensions: define all the possibilities of extension for the left side of a selected starter in the form of a graph.
- Nodes and edges: make up each extension.
- Data: Each node and each edge have several data.

Code :

```
{
"Starter_0":{"data":{"id":"S0", "sequence":"ATGC","length":4,
"extremGraphs":[ {"data":{"id":"k0", "sequence":"ATGC", "direction":"RIGHT", "type":"original",
"nodes": [
    {"data": {"id": "0","sequence": "ATGC"}},
    {"data": {"id": "1","sequence": "ATGC"}},
    {"data": {"id": "2","sequence": "ATGC" }},
    {"data": {"id": "3","sequence": "ATGC"}},
    {"data": {"id": "4","sequence": "ATGC"}}
],
"edges": [
    {"data": {"id": "e0","source": "0","target": "1","direction": "FF"}},
    {"data":{"id": "e1","source": "1","target": "0","direction": "RR"}},
    { "data": {"id": "e2","source": "0","target": "3","direction": "FF"}},
    { "data": {"id": "e3","source": "3","target": "0","direction": "RR"}},
    {"data": {"id": "e4","source": "1","target": "2","direction": "FF"}},
    {"data": {"id": "e5","source": "2","target": "1","direction": "RR"}},
    {"data": {"id": "e6","source": "3","target": "4","direction": "FF"}},
    {"data": {"id": "e7","source": "4","target": "3","direction": "RR"}}
]
}}
]
}
}
```

The graph file can be output of Mapsembler 2, but also a simple JSON. In the minimal structure for a compatible JSON is:

Structure :

```
nodes
    id_node
    sequence
edges
    id_edge
    source
    target
    direction
```

Code :

```
{
"nodes": [
    {"data": {"id": "0","sequence": "ATGC"}},
    {"data": {"id": "1","sequence": "ATGC"}},
    {"data": {"id": "2","sequence": "ATGC" }},
    {"data": {"id": "3","sequence": "ATGC"}},
    {"data": {"id": "4","sequence": "ATGC"}}
],
"edges": [
    {"data": {"id": "e0","source": "0","target": "1","direction": "FF"}},
    {"data":{"id": "e1","source": "1","target": "0","direction": "RR"}},
    { "data": {"id": "e2","source": "0","target": "3","direction": "FF"}},
    { "data": {"id": "e3","source": "3","target": "0","direction": "RR"}},
    {"data": {"id": "e4","source": "1","target": "2","direction": "FF"}},
    {"data": {"id": "e5","source": "2","target": "1","direction": "RR"}},
    {"data": {"id": "e6","source": "3","target": "4","direction": "FF"}},
    {"data": {"id": "e7","source": "4","target": "3","direction": "RR"}}
]
}
```

For better performances, the property "length", referred to the size of the sequence, can be added in the data nodes. In this way, the application must not calculate the size of sequences and save several times, especially with large graph.

Code :

```
    {"data": {"id": "0","length":4 "sequence": "ATGC"}},
```

So the JSON structure has become :

```
nodes
    id_node
    length
    sequence
edges
    id_edge
    source
    target
    direction
```

The data nodes and edges can contain also the property coverage. Coverage is an average come from different files. So coverage property is an array containing the identity of the files and value of average coverage:

Code :

```
{"data": {"id": "0","length":4 "sequence": "ATGC",
coverage":[
    {"id":"file_1","avg_coverage": 106.47},
    {"id":"file_2","avg_coverage": 106.47}}
]
```

With average coverage property, the structure is:

```
nodes
    id_node
    length
    sequence
    coverage
       id_file
       avg_coverage
edges
    id_edge
    source
    target
    direction
    coverage
       id_file
       avg_coverage
```

To use the concatenation function , it's necessary to know the value of k, referred to the overlap between sequences of nodes linked. With minimal JSON this value can be defined in Option >

Define k (not implemented yet).

When minimal JSON is loaded, the graph viewer is displayed automatically.

b. Session File

It's possible to load a session file (.sjson). Session files contain a graph description with all the data for nodes and edges, the positions of nodes and vizmapper properties defined previously for nodes and edges. Loaded a session file, display automatically the graph viewer.

c. Vizmapper File

It's possible to load a previous configuration of the vizmapper with a vizmapper file(.vjson). Vizmapper files contain all the configuration for nodes and edges to restore all distribution points with for each its color, size or shape. Loaded a vizmapper file apply the saved vizmapper configuration to the current graph.