

ComponentGrapher User Guide

Mars 2017 / version 1.0

Table of contents

1. Pre-requisites	5
2. Quickstart	5
2.1 Input files.....	6
2.2 Command-line and options.....	7
2.3 Output file.....	7
2.3.1 Network files (_complete.txt, _1.txt, _2.txt, _3.txt, _4.txt)	9
2.3.2 Node identification (_id.txt)	9
2.3.3 Statistic file (_stat.txt)	9
3. Graphical version	11
3.1. Starting the graphical version of ComponentGrapher	11
3.2. Loading datasets into ComponentGrapher	11
3.3 Create and modify matrices.....	12
3.3.1 Create a new matrix.....	12
3.3.2 Modify an existing matrix	13
3.3.3 change the name of a taxon	13
3.3.4 Edit the name of a character	14
3.3.5 Matrix orientation.....	15
3.4 Saving matrix files.....	15
3.5. Running an analysis.....	16
3.5.2 Stopping an analysis	17
3.6. Saving current analysis.....	17
3.7. Importing previous analysis	18
3.8. Exporting networks	18
3.9. Exporting network statistics	19
3.10. Exporting triplets.....	19
3.11. Polymorphic characters	20
4. Results in GUI mode	21
4.1 Summary tab.....	21
4.2 Permutation statistics tab.....	22
4.2.1 Network statistics	23
4.2.2 Node statistic's	24
4.3 Complexes	25

5. Miscellaneous.....	26
5.1 External node annotation file.....	26
5.2 External node annotation file in GUI mode.....	27
5.3 Copy matrix from / to Excel.....	28
5.4 Information about the current analysis	29
5.4 Compiling the source code	29
5.5 Citation	30
6. References	31
7. Abbreviations	32

ComponentGrapher is a multi-platform java program which eases the creation of character-states networks from character matrices in NEXUS or PHYLIP format. This command-line tool also allows the characterization of the resulting networks by searching for global and local articulation points, the numbers of complexes (connect components), betweenness, and other network related statistics. The software and its source code are provided at <https://github.com/etiennelord/ComponentGrapher> using a GPL3 license. This work was developed as a collaboration between the Université de Montréal (UdeM), Université du Québec à Montréal (UQAM) and Sorbonne Université, Université Pierre et Marie Curie, UPMC Paris 6.

1. Pre-requisites

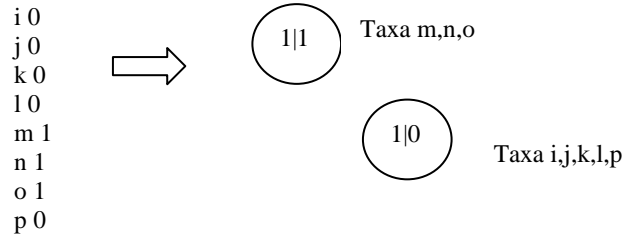
The software ComponentGrapher is implemented as a multi-platform java program. It requires Java SE5 (JDK 5) or later versions of Java. For large matrices (>1000 characters), a fair amount of RAM is required, and should be specified before launching the tool.

Example for large matrices with allocation of 6 gb of memory [-Xmx6g]:

```
java -Xmx6g -jar COMPONENT-GRAPHER.jar
```

2. Quickstart

The ComponentGrapher software can either be run as a **command-line utility** (section 2) or as a **graphical application** (section 3). Given an input matrix (section 2.1), the software ComponentGrapher will generate character state networks *i.e.* for each state (e.g. 0,1) of a character, a new node is created (see also table 1 - section 2.3). The created edges will then reflect some characteristics between the character-state nodes (see also table 2 - section 2.3). For example, given a matrix with one character (*Char.1*) and two states (0,1), this will result in 2 nodes (character state):



Note: refer to the published paper for more information.

The edges represent the link between pair of nodes. For example, four type of edges can exist if we add another character to this matrix.

	Type I Identical		Type II Inclusion		Type III Overlap		Type IV Disjoint	
	A2	B2	C2	D2	E2	F2	A2	F2
i	-	-	-	-	-	+	-	+
j	-	-	-	-	-	+	-	+
k	-	-	-	+	+	+	-	+
l	-	-	-	+	+	+	-	+
m	+	+	+	+	+	-	+	-
n	+	+	+	+	+	-	+	-
o	+	+	+	+	+	-	+	-
p	-	-	-	+	+	-	-	-

Note: the complete network is then composed of the link of types 1, 2 and 3.

2.1 Input files

The software requires as input either a character matrix either in Phylip format [1] or a character matrix in NEXUS format [2] with only one morphological MATRIX tag.

Phylip format (Sample.phy):

```
5 6
A    110110
B    110000
C    100110
D    001001
E    001110
```

Nexus format (Sample.nex):

```
#NEXUS
[Note: this is a sample matrix]
begin data;
  dimensions ntax=5;
  dimensions nchar=6;
  format datatype=standard gap=- missing=? Interleave=no;
  MATRIX
    A    110110
    B    110000
    C    100100
    D    001001
    E    001110
  ;
end;
```

*Note that multiple states at one position in the matrix are supported (e.g. {0,1}, (0 1), (0,1,2,3)) as well as interleaves for the NEXUS format. Also note that **undefined states symbol** (e.g. ?, -, *) **are not processed in this version**. For the NEXUS format, column and specific states information can be specified using the CHARSTATELABELS, STATELABELS or CHARLABELS annotation specific to NEXUS format [2].

Nexus format (Sample matrix with multiple states and annotations):

```
#NEXUS
[Note: this is a sample matrix]
begin data;
  dimensions ntax=5;
  dimensions nchar=6;
  format datatype=standard gap=- missing=? Interleave=no;
  CHARSTATELABELS
    1 Mouth / present, absent, 2 'eyes' / two, eight, 3 hands / two,
    'more than 2', 4 hairs / present, absent, 5 'eye color' / blue,
    brown, 6 nose / red, normal
  ;
  MATRIX
    A    110110
    B    1{0,1}0000
    C    100100
    D    00100{0,1}
    E    001110
  ;
end;
```

2.2 Command-line and options

To process matrices with ten of thousand of character states, we recommend to execute the software in non-GUI mode. To analyze your data in a GUI environment, please go to section 3.1.

Base command-line

```
java -jar COMPONENT-GRAPHER.jar matrixfile
```

Command-line options

```
-perm=100      : Specify the number permutation to perform.
-maxpool=10    : Specify the number of concurrent threads.
-undefined     : Remove column containing undefined states (e.g. ?, -, *).
-multiple      : Remove column containing polymorphic sites (e.g. {1,2,3}).
-bipartite     : Output bipartite network files.
-graphml       : Output graphml files (Gephi, Cytoscape compatibles).
-nodeid=file   : Provide a node identification file.
-output=dir    : Specify output directory.
-variation=X   : Specify the variation string to use.
-triplets      : Output triplets file (triplets.txt).
```

2.2.1 Base analysis (command-line version)

```
java -jar COMPONENT-GRAPHER.jar matrixfile
```

2.2.2 Analysis without column containing multiple states and with specified output file [-multiple, -output]

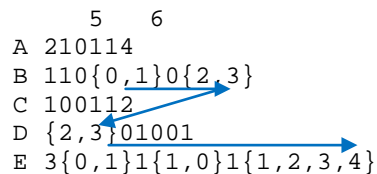
```
java -jar COMPONENT-GRAPHER.jar matrixfile -multiple -output=result
```

2.2.3 Use a specific variation [-variation]

```
java -jar COMPONENT-GRAPHER.jar Sample_n.phy -variation=032014
```

Note that the different position in the matrix are visited from left to right, from first taxon to last taxon.

```
      5   6
A 210114
B 110{0,1}0{2,3}
C 100112
D {2,3}01001
E 3{0,1}1{1,0}1{1,2,3,4}
```



2.3 Output file

With a matrix as input, the ComponentGrapher software will generate five network files (Table 3 - below) for each evaluated matrix in either an *edge list* or *graphml* (Gephi [3] compatible) format, a statistic file (.stat.txt) and a node identification file (.id.txt). In this analysis, each node correspond to a character-state meaning that each state of a particular character, column in the matrix, will be present in the complete network.

For example, processing the sample.phy will result in the following nodes (Table 1):

Sample.phy

```
5 6
A 110110
B 110000
C 100110
D 001001
E 001110
```

Table 1. Generated nodes from the Sample.phy matrix*

Node ID	Character-state	Character (column)	Associated taxa
0	char. 1 1	char. 1	A,B,C
1	char. 1 0	char. 1	D,E
2	char. 2 1	char. 2	A,B
3	char. 2 0	char. 2	C,D,E
4	char. 3 0	char. 3	A,B,C
5	char. 3 1	char. 3	D,E
6	char. 4 1	char. 4	A,C,E
7	char. 4 0	char. 4	B,D
8	char. 5 1	char. 5	A,C,E
9	char. 5 0	char. 5	B,D
10	char. 6 0	char. 6	A,B,C,E
11	char. 6 1	char. 6	D

* In bold, the nodes created for the first state column.

The different files generated correspond to the types of edges (1, 2, 3, or 4) found between each nodes (Table 2).

Table 2. Type of links (see main publication for more information)

Edge type	Directed /Undirected	Link types	Meaning
1	Undirected	Co-existence / identical	Same taxa found in both nodes
2	Directed	Inclusion	One group of taxa is included in the other group of taxa
3	Undirected	Overlap	Overlap between the group of taxa in each node
4*	Undirected	Disjoint	No overlap between the group of taxa for both nodes

*Note: the disjoint edges (type 4) are not included in the complete network.

The generated files (Table 3) for a matrix file **without multiple states** (e.g. more than one possible state per position in the matrix) are as follow:

Table 3. Generated networks, node identification and statistics files for a matrix without multiple states

Generated files	Content
Filename_complete.txt	Edge types: 1,2,3
Filename_1.txt	Edge types: 1
Filename_2.txt	Edge types: 2
Filename_3.txt	Edge types: 3
Filename_4.txt	Edge types: 4
Filename_id.txt	Node identification
Filename_stat.txt	Summary statistic of the generated networks and command-line options

2.3.1 Network files (_complete.txt, _1.txt, _2.txt, _3.txt, _4.txt)

The result edge list files are zero-based tab-separated value (tsv) files. Each line corresponds to one edge with a source id (src_id), destination id (dest_id), the edge type and the number of common taxa between each node:

Source node ID	Destination node ID	Type of edge	Number of common taxa	Percent of common taxa
----------------	---------------------	--------------	-----------------------	------------------------

```
#src_id(tab\t)dest_id(tab\t)edge_type(tab\t) number_common_taxa(tab\t)percent
2          0          2          2          1.0
0          3          3          1          0.5
0          4          1          3          1.0
0          6          3          2          0.75
...
```

*Note: that there is no fourth and fifth columns (Number of common taxa, percent of common taxa) for the type 4 network.

2.3.2 Node identification (_id.txt)

The result node identification file (tab-separated values) contains information about each node in the networks.

Node ID	Name	Char. Label	State label	Encoded State	Edge count	In edge count	Out edge count	Number of taxa	Taxa
0	char.1 1	char.1	1	1	8	1	7	3	A,B,C
1	char.1 0	char.1	0	0	8	1	7	2	D,E
...									

2.3.3 Statistic file (_stat.txt)

This file contains the command-line arguments for this particular run, as well as some results such as the number of edges in each network and the number of total nodes.

Example for the file Sample_n.nex_100.stat.txt

```
===== PARAMETERS =====
Command line options      : data/sample_n.phy -random=100
Input                    : data/sample_n.phy
N taxa                   : 5 (rows)
N characters              : 6 (columns)
Total number of multistate characters: 6      (Positions in the matrix with multi-
states)
Total number of possible variations : 128      (Number of possible matrices)
Total variations tested      : 100      (Total variation of matrices tested)
Iteration mode              : random      (Either in order or random)
Total undefined char        : 0      (Number of undefined characters.)
Total multiple char         : 6      (Number of multistate characters)
Total undefined column(s)   : 0      (Number of column with undefined char.)
Total multiple column(s)    : 4      (Number of column with multistate char.)
Remove multiple state columns : false      (Remove column with multistate char.)
Remove undefined columns    : false      (Remove column with undefined char.)
Total treated column        : 6
Total possible nodes        : 15
=====
Current iteration : 100/100      (Current variation number.)
States variations : 023004      (Current variation string)
(saving to: data/sample_n.phy_100_complete.txt)
=====
```

Results:

```
=====
Edges (total)          : 6051 (Total edges in complete network)
Edges type 1 (perfect) : 281 (Total edges in type 1 network)
Edges type 2 (inclusion) : 2367 (Total edges in type 2 network)
Edges type 3 (overlap) : 3403 (Total edges in type 3 network)
Edges type 4 (disjoint) : 3149 (Total edges in type 4 network)
```

```
Total nodes evaluated : 15 (Total generated character-state nodes)
Total nodes            : 14 (Total nodes included in the networks)
Node (unassigned)      : 1 (Total nodes that are not in any network)
Node type 1 (perfect)  : 4 (Nodes in type 1 network)
Node type 2 (inclusion) : 14 (Nodes in type 2 network)
Node type 3 (overlap)  : 12 (Nodes in type 3 network)
Node type 4 (disjoint) : 15 (Nodes in type 4 network)
```

...

```
=====
States variations : 023004 (Current variation string, see beginning of section)
Taxon->Character(column)|Value (Corresponding matrix position)
```

```
-----
B->4|0
B->6|2
D->1|3
E->2|0
E->4|0
E->6|4
=====
```

3. Graphical version

3.1. Starting the graphical version of ComponentGrapher

A. Uncompress the distributed release (github.com/etiennelord/ComponentGrapher/releases)

B. Either:

- 1) Double-click on the main jar file : COMPONENT-GRAPHER.Jar
- 2) In a terminal prompt, execute the following command-line:

```
java -jar COMPONENT-GRAPHER.jar
```

C. You should now see an interface similar to Figure 1 below.

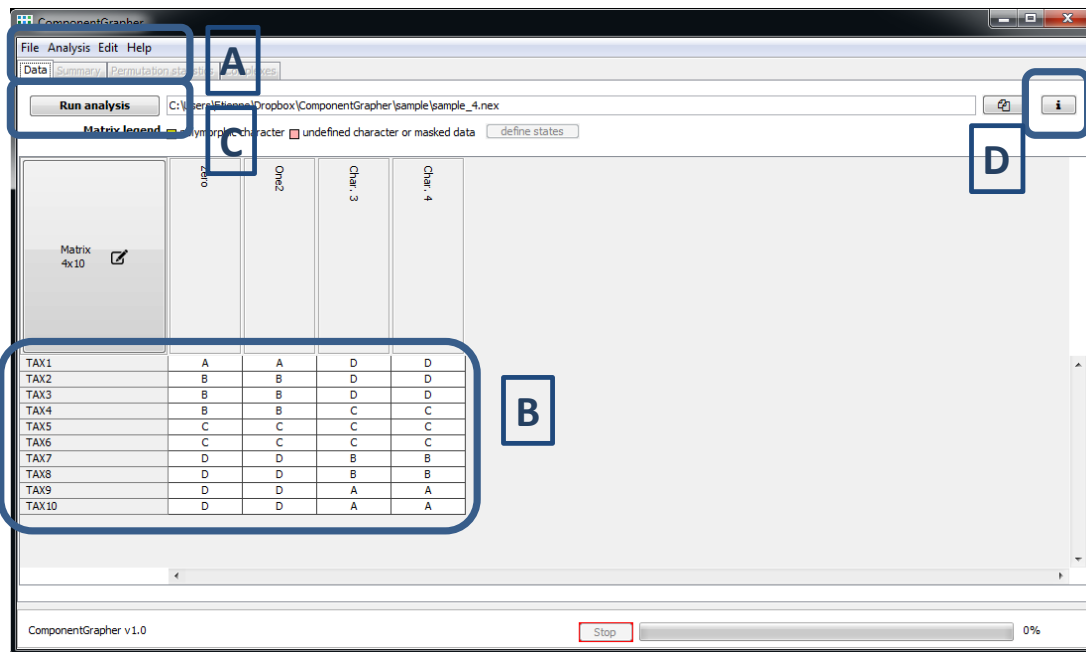
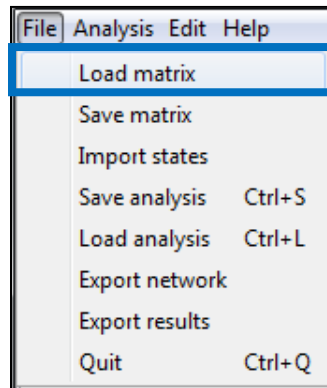


Figure 1. Main ComponentGrapher interface. In a typical analysis, you should go in the File Menu (Box A - see section 3.2) to load a matrix file. Then, modify the matrix file (Box B- see section 3.3) and perform an analysis by clicking on the run analysis button (Box C - see section 3.4). Further information about this analysis or the matrix file is available by clicking on the information button (Box D - see section 5.4) in the top right corner of the application.

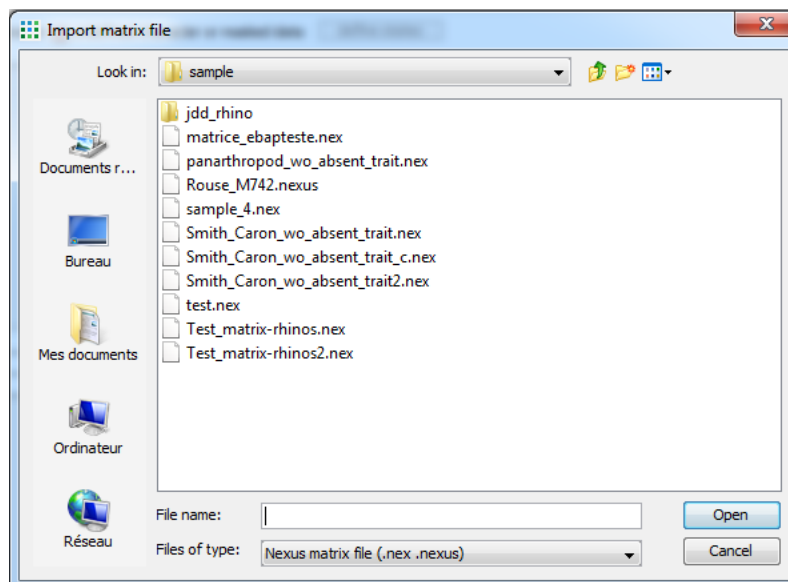
3.2. Loading datasets into ComponentGrapher

ComponentGrapher is able to import morphogenetic matrices in *Nexus*, *Phylip* and *plain matrix* formats (refer to section 2.1)

To load the matrix files, go into the **File → Load matrix** submenu. This will open a file selection box.



In this file selection box, **select the desired file format** from the *File of type* dropdown menu.

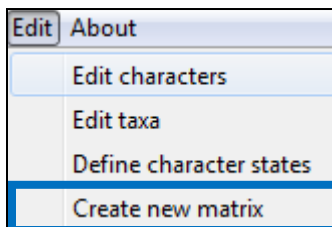


Once the selected file format is chosen. Click the **Open** button.

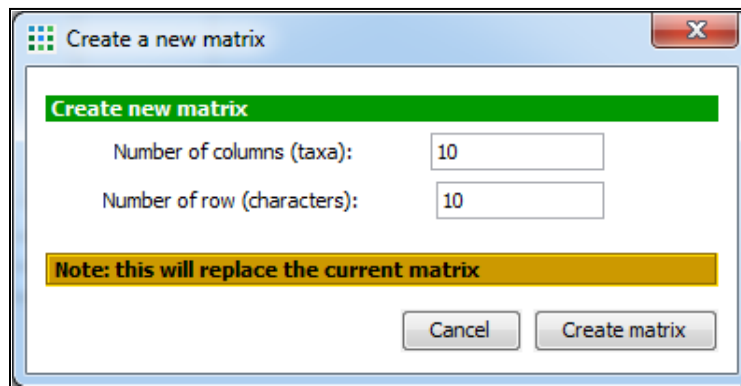
3.3 Create and modify matrices

3.3.1 Create a new matrix

Go into the **Edit → Create new matrix** submenu. A new popup menu will be displayed.




In this new menu, select the number of desired columns and rows. Then, click the **Create matrix** button.



Note: this will delete the currently loaded matrix.

3.3.2 Modify an existing matrix

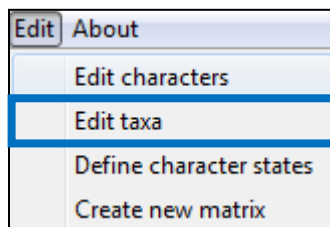
In the main matrix view, you can modify each site in the matrix by first clicking with the mouse on the desired matrix cell (it will change color), and then use the keyboard to enter the desired state.

	Char. 1	Char. 2	Char. 3	Char. 4
Matrix 3x4 				
TAX0	1	1	012	0
TAX1	0	1	1	0
TAX2	0	0	1	1

If more than one states is possible (polymorphic state), you can enter more than one value. Note that valid values are the following: numbers 0 to 9, and letters A to Y. Undefined characters : ?, *, and - are also valid, but will not be taken into account in the processing of the matrix.

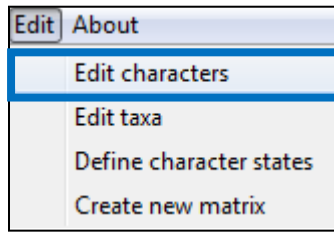
3.3.3 change the name of a taxon

To change the name of taxon, either double-click on it with the mouse left button, or go into the **Edit → Edit taxa** submenu.



3.3.4 Edit the name of a character

To edit the name character, go into the **Edit → Edit characters** submenu.



A new popup menu will be displayed (Figure 2- below). In this menu, click with the mouse on the desired characters (Figure 2 - column A) to edit them.

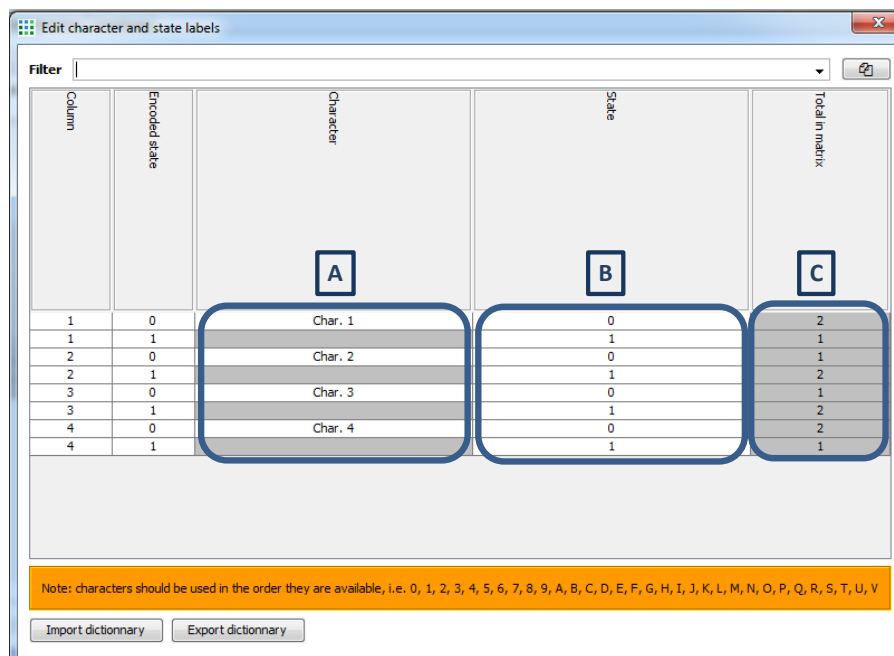


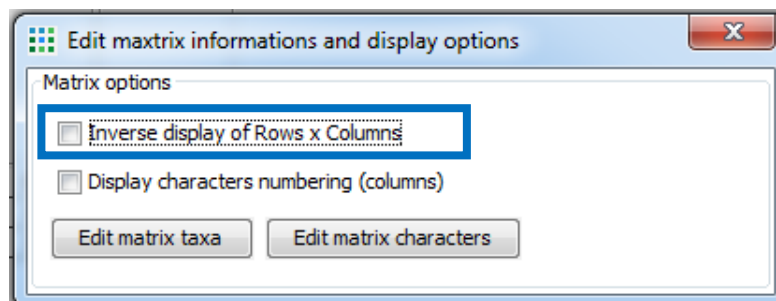
Figure 2. Main interface to edit character state. Character annotation (*name*) can be changed in column A. In column B, the title of the associated state can be modified. In column C, the total number of this character state tandem in the current matrix is indicated.

3.3.5 Matrix orientation

To change the orientation of the matrix, click the matrix button, in the top left corner of the matrix. Then, select the inverse display of Rows x Columns option.

	Char. 1	Char. 2	Char. 3	Char. 4
TAX0	1	1	012	0
TAX1	0	1	1	0
TAX2	0	0	1	1

A new popup menu will be displayed. In this new menu, click the inverse display box.



3.4 Saving matrix files

To save matrix files, go into the **File→Save** matrix submenu. Once inside, select the filename and location where you want to save the matrix file (Figure 3 - A). Also, select the desired matrix file format (Figure 3 - B).

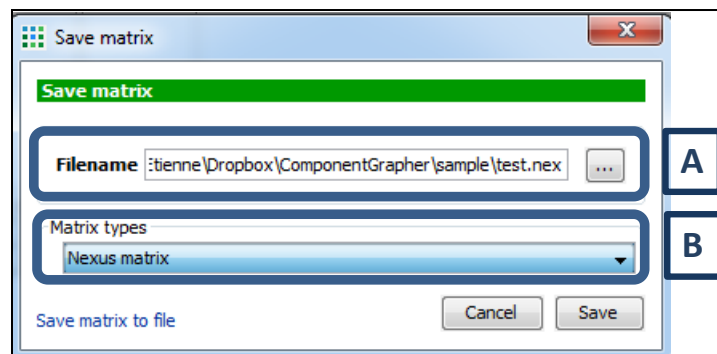


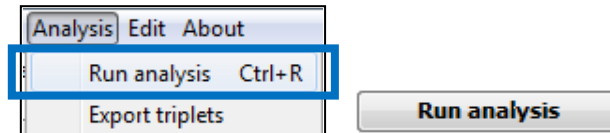
Figure 3. Main interface to save a matrix.

Note: Only the **Nexus** format support the annotation of each character states.

3.5. Running an analysis

This step will first generate the networks. Then, it will performed a number of permutation steps and re-analysis of network in order to estimate the likeliness of each networks parameters.

To run an analysis, either go in the **Analysis** → **Run analysis** submenu or click on the Run analysis button in the main GUI window.



A new popup windows will be displayed (Figure 4). In this menu, select the number of permutation of the original matrix and the output directory.

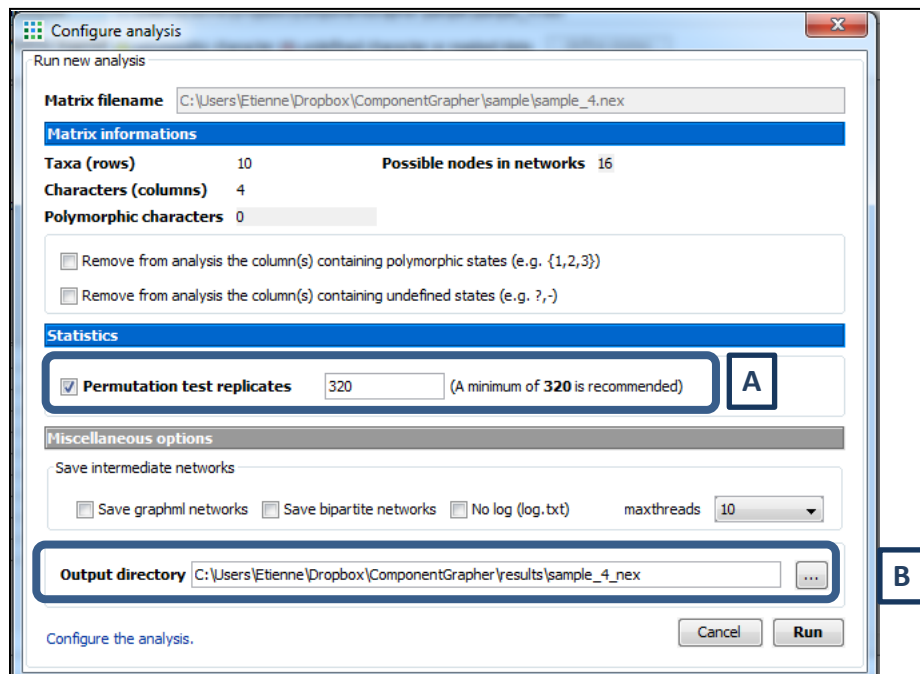


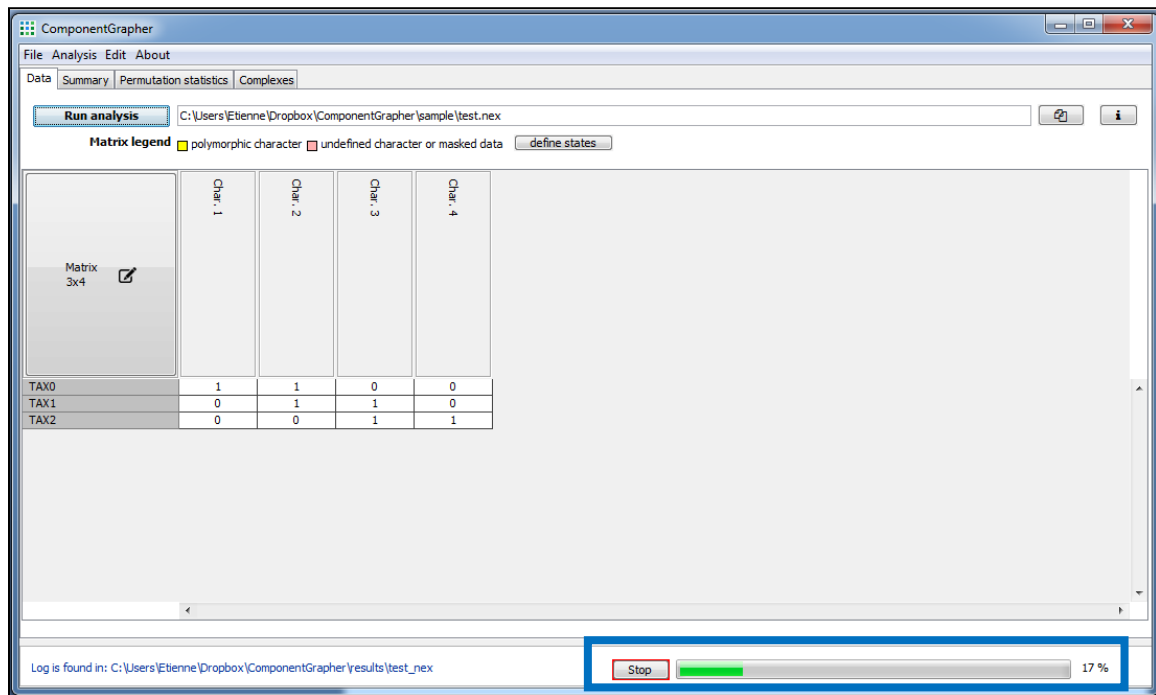
Figure 4. Main interface to run an analysis. In A, select the number of permutation of the original matrix to perform. In B) select the output directory. Other possible options are to save the resulting network as graphml files, to export bipartite networks, and to select the number of concurrent analysis.

After the different options are selected, click the **Run button** to start an analysis.

Note: all analysis output will be saved as JSON-formatted files in the selected output directory.

3.5.2 Stopping an analysis

To stop an ongoing analysis, click the **Stop button**, located in the bottom right corner of the main interface windows.

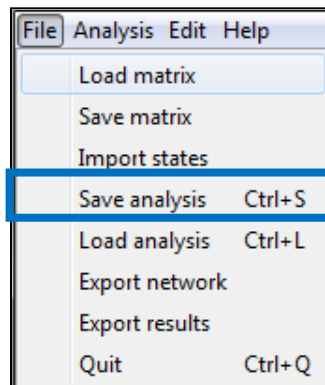


3.6. Saving current analysis

Once an analysis is completed, it is possible to save the current results.

The results will be saved in a JSON formatted file. **Please note that individual replicates are not saved. They are however found in the specify output directory.**

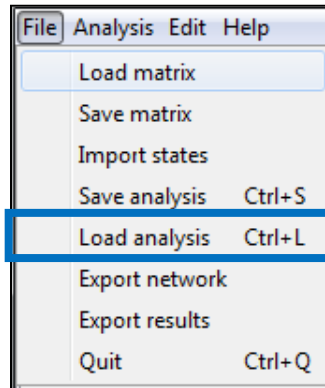
In order to save the current analysis, go to the **File → Save analysis** submenu.



You will be asked to name your analysis. Select a name, then click the **Open button**.

3.7. Importing previous analysis

In order to import an analysis, go into the **File → Load analysis** submenu. Then select the previously saved analysis (JSON file).



3.8. Exporting networks

In order to export the resulting network, go into the **File → Export network** submenu. This will open a new popup windows (Figure 5).

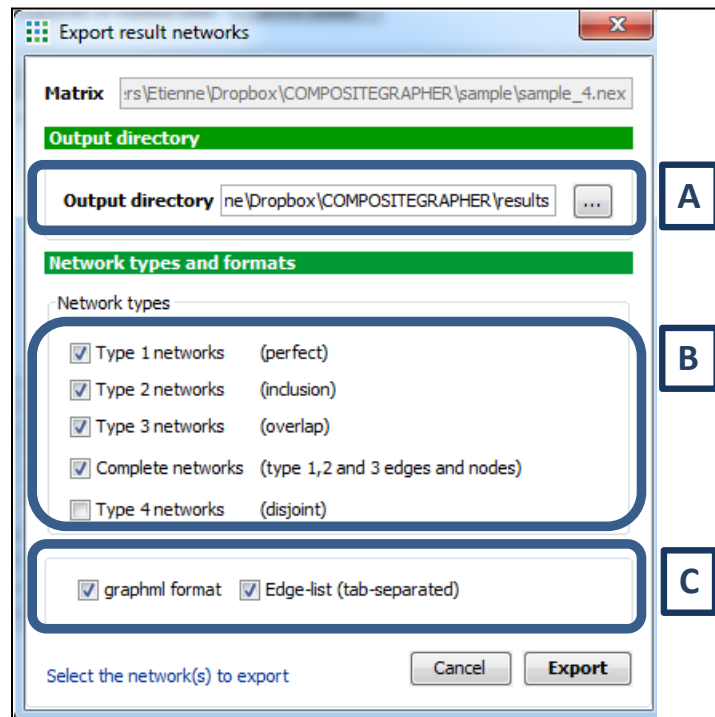
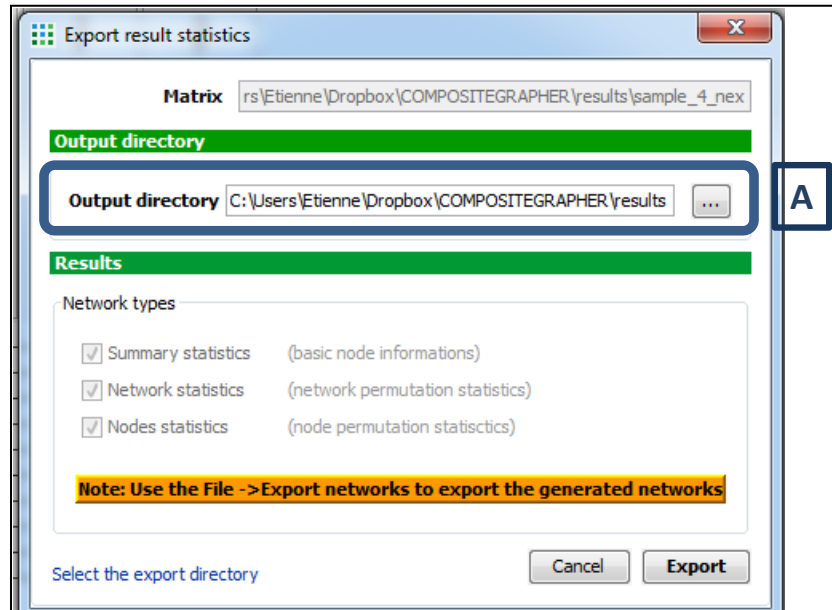


Figure 5. Main interface to export networks. In A, select the desired output directory. In B, select the network types and in C, select the desired output formats.

In this menu, first select the output directory (Figure 5 - Box A). Then, select the needed networks (Box B) and required formats (Box C). Then click the **Export button**.

3.9. Exporting network statistics

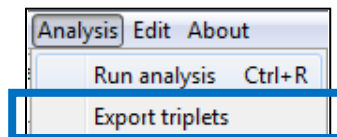
In order to export the resulting statistics, go into the **File** → **Export results** submenu. This will open a new popup windows.



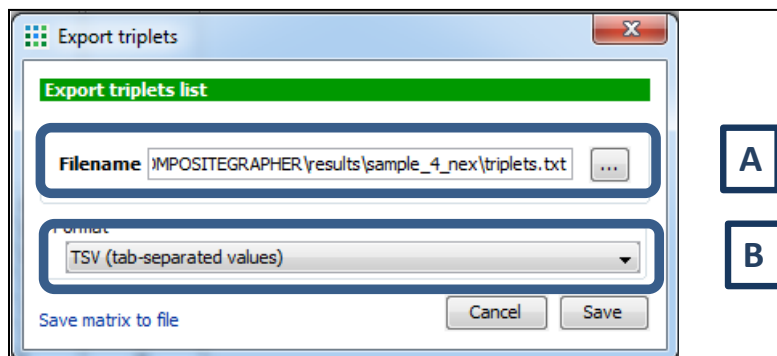
To export the network statistics, select the output directory (Box A), then click the **Export button**.

3.10. Exporting triplets

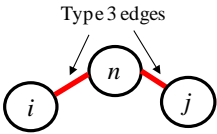
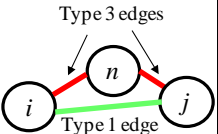
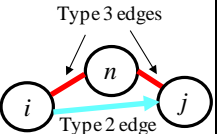
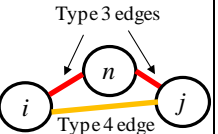
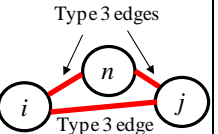
To export the triplets found in the network (*complete*), go into the **Analysis** → **Export triplets** submenu. This will open a new popup windows.



To export the triplets (see below for a description), first select the output directory (Box A). Second, select the output format: either tab-separated values (TSV) or comma-separated values (CSV) (Box B). Finally, click on the **Save button**.

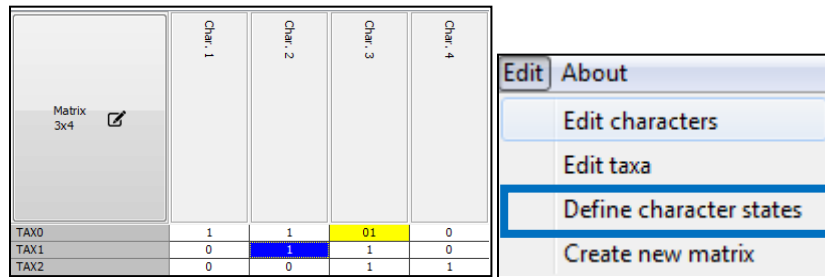


The different types of identified triplets are describe in the following table.

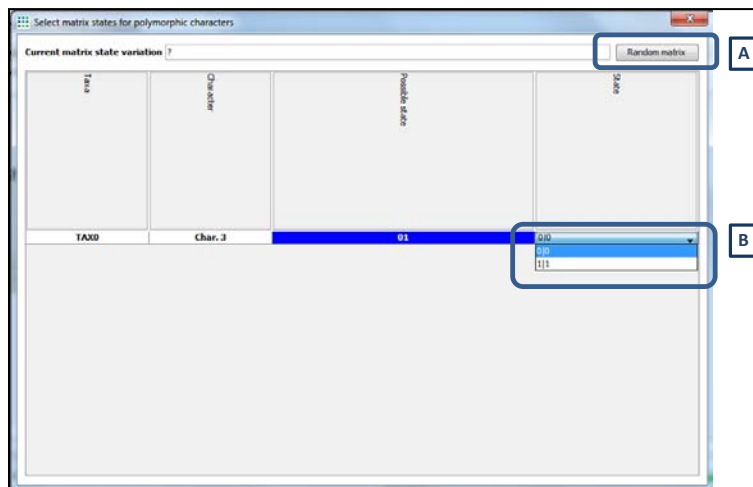
Triplet type A	Triplet type B	Triplet type C	Triplet type D	Triplet type E
				
Nodes i and j are not connected. Central node n is linked with type 3 edges.	Nodes i and j are connected by a type 1 edge. Central node n is linked with type 3 edges	Nodes i and j are connected by a type 2 edge. Central node n is linked with type 3 edges	Nodes i and j are connected by a type 4 edge. Central node n is linked with type 3 edges	Nodes i and j are connected by a type 4 edge. Central node n is linked with type 3 edges (e.g. it form a triangle)

3.11. Polymorphic characters

If the analyzed matrix contains polymorphic (*multiple states for one character*) characters (indicated in yellow in the matrix), it is necessary to define the character that will be used in the analysis. To do so, either go into the **Edit → Edit characters** submenu or click the **Define states button** in the main interface view.

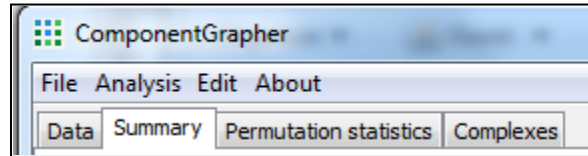


A new popup windows will be displayed. In this windows, either click the **Random matrix button**(Box A). This will generate a random variation of each possible states. Otherwise, for each taxon and character combination, you will need to select one of the state (Box B).



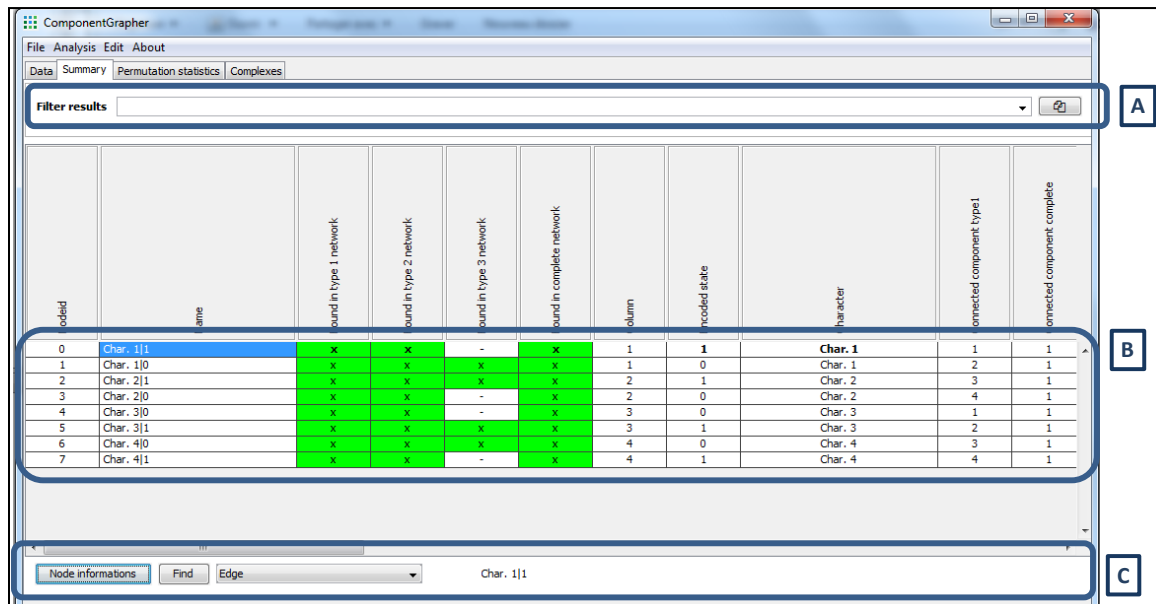
4. Results in GUI mode

Once an analysis is completed, the following results tabs are available in the main interface: **Summary** (section 4.1), **Permutation statistics** (section 4.2), **Complexes** (section 4.3). They will each be describe in the following sections.



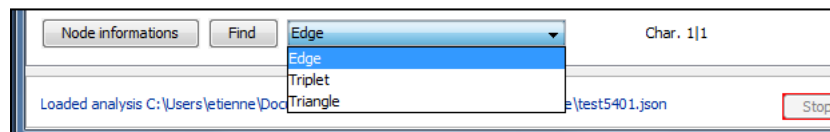
4.1 Summary tab

The summary tab displays a summary of each node statistics in the generated network. To filter for a particular node, use the filter results box (Box A) to enter your search parameters (e.g. the node identification or associated taxa) then click Enter. A summary of each node is displayed in box B.



Note: In the summary display (Box B), an 'x' indicates the presence of a node in a specific network, while a '-' denote its absence.

To find **Edges**, **Triplets** or **Triangles** (see section 3.10) associated with a particular node, **select the node** (Box B) then click the **Find button** (Box C). This will open a new windows



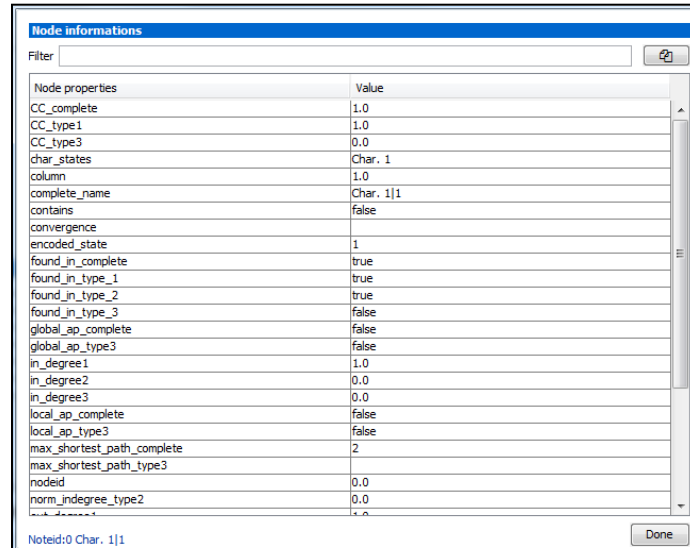
Important note: Triplets and Triangles are searched only if the node is present in the type 3 network.

Important note: For Triplets and Triangles, a file named tmp.txt is created in the main ComponentGrapher directory. Results will be presented as follow:

node1	central node*	node3	triplet type
6	1	2	Type A
6	1	2	Type B

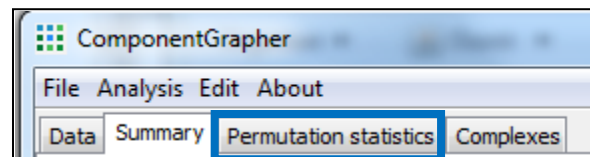
*Selected node prior to search.

To get more information for a particular node, select the node (Box B) then click the **More information button** (Box C). A new popup windows will be displayed.

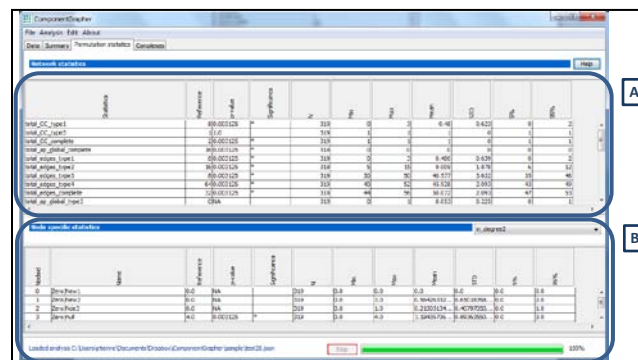


Note: this menu display the raw node information. Refer to section 7 for description of each variable.

4.2 Permutation statistics tab



The permutation statistics tab displays information about the **network statistics** (Box A, section 4.2.1) or **individual node statistics'** (Box B, section 4.2.2) after the permutation test has been completed.



4.2.1 Network statistics

Statistics	Reference	p-value	Significance	N	Min	Max	Mean	STD	5%	95%
total_CC_type1	8	0.003125	*	319	0	3	0.48	0.623	0	2
total_CC_type3	1	1.0		319	1	1	1	0	1	1
total_CC_complete	2	0.003125	*	319	1	1	1	0	1	1
total_ap_global_complete	16	0.003125	*	319	0	0	0	0	0	0
total_edges_type1	8	0.003125	*	319	0	3	0.486	0.639	0	2
total_edges_type2	16	0.003125	*	319	5	15	9.009	1.878	6	12
total_edges_type3	8	0.003125	*	319	30	50	40.577	3.632	35	46
total_edges_type4	64	0.003125	*	319	40	52	45.928	2.093	43	49
total_edges_complete	32	0.003125	*	319	44	56	50.072	2.093	47	53
total_ap_global_type3	0	NA		319	0	1	0.053	0.225	0	1

A B C D E F G H I J K

The network statistics panel display the following information:

Columns	Descriptions
A	Statistics (refer to section 7)
B	Reference value. The value associated with the computed original network.
C	The calculated p-value
D	Significance of the p -value. This indicate if the p -value should be considered significant. If no star (*) is present, the value should not be considered statistically different. Legend: * $p < 0.05$ ** $p < 0.01$ *** $p < 0.001$
E	Number of permutation performed.
F	Minimum value found in the permuted dataset.
G	Maximum value found in the permuted dataset.
H	Mean value found in the permuted dataset.
I	Standard deviation found in the permuted dataset.
J	5% percentile value found in the permuted dataset.
K	95% percentile value found in the permuted dataset.

Note: p -values are not calculated if the reference value is either absent (NA) or equal to zero.

4.2.2 Node statistic's

Node specific statistics

in_degree2

Nodeid	Name	Reference	p-value	Significance	N	Min	Max	Mean	STD	5%	95%
0	Zero New1	0.0	NA		319	0.0	0.0	0.0	0.0	0.0	0.0
1	Zero New2	0.0	NA		319	0.0	3.0	0.56426332...	0.65018768...	0.0	2.0
2	Zero Node3	0.0	NA		319	0.0	1.0	0.21003134...	0.40797050...	0.0	1.0
3	Zero null	4.0	0.003125	*	319	0.0	4.0	1.19435736...	0.89363550...	0.0	3.0

A B C D E F G H I J K

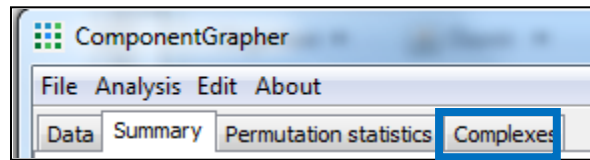
For the node statistic's, the specific statistic can be selected using the dropdown menu (Box H).

The node specific statistics panel then displays the following information:

Columns	Descriptions
A	Node ID and node name
B	Reference value. The value associated with the computed original network.
C	The calculated p-value
D	Significance of the p -value. This indicate if the p -value should be considered significant. If no star (*) is present, the value should not be considered statistically different. Legend: * $p < 0.05$ ** $p < 0.01$ *** $p < 0.001$
E	Number of permutation performed.
F	Minimum value found in the permuted dataset.
G	Maximum value found in the permuted dataset.
H	Mean value found in the permuted dataset.
I	Standard deviation found in the permuted dataset.
J	5% percentile value found in the permuted dataset.
K	95% percentile value found in the permuted dataset.

Note: p -values are not calculated if the reference value is either absent (NA) or equal to zero.

4.3 Complexes



The **network complexes tab** displays information for each connected component (complexes) found in the different networks. Networks can be selected using the dropdown menu in the top right corner (Box A). Once selected, the different complexes are listed (Box B) and their associated taxa.

The screenshot shows the full ComponentGrapher window. The 'Complexes' tab is active. In the top right corner, a dropdown menu (Box A) is open, showing options: 'Complete network', 'Complete network', 'Type 1 network', 'Type 2 network', and 'Type 3 network'. Below this, a table (Box B) lists the complexes. The table has four columns: 'Complexes', 'Characters states', 'Number of nodes', and 'Taxa'. The data is as follows:

Complexes	Characters states	Number of nodes	Taxa
1	Zero New1 One2 A	2	TAX1
2	Zero New2 One2 B	2	TAX2,TAX4,TAX3
3	Zero Nde3 One2 C	2	TAX6,TAX5
4	Zero null One2 null	2	TAX10,TAX9,TAX8,TAX7
5	Char. 3 null Char. 4 null	2	TAX2,TAX1,TAX3
6	Char. 3 C Char. 4 C	2	TAX6,TAX5,TAX4
7	Char. 3 B Char. 4 B	2	TAX8,TAX7
8	Char. 3 A Char. 4 A	2	TAX10,TAX9

At the bottom of the window, a status bar shows 'Loaded analysis C:\Users\etienne\Documents\Dropbox\ComponentGrapher\sample\test28.json', a 'Stop' button, and a progress bar at 100%.

5. Miscellaneous

5.1 External node annotation file

If needed, you can supply a list of character-state annotation if none is provided in the NEXUS matrix or if you use a Phylip matrix file. To do so, include a list of node character-states with the option [-nodeid=].

Example command-line with the sample.phy matrix

```
java -jar COMPONENT-GRAPHER.jar sample.phy -nodeid=sample_nodeid.txt
```

This file should be a **tab-separated value** with four columns: column number (character position starting at 1), encoded state, new character label, new state label:

Column	State	Character label	State label
--------	-------	-----------------	-------------

sample_nodeid.txt

1	0	Mouth opening orientation	Terminal
1	1	Mouth opening orientation	Ventral
2	0	Pre-oral chamber	Absent
2	1	Pre-oral chamber	Present
3	0	Radially arranged circumoral structures	Absent
3	1	Radially arranged circumoral structures	Present
4	0	Differentiated circumoral structures	undifferentiated plates
4	1	Differentiated circumoral structures	three or four enlarged plates
5	0	Pharynx differentiated from midgut	not differentiated
5	1	Pharynx differentiated from midgut	differentiated
6	0	Pharynx reversible	permanently inverted
6	1	Pharynx reversible	completely reversible

These annotations will be reflected in the **node identification file** (_id.txt) and in the **summary file** (_summary.txt).

sample.phy_id.txt (before)

Node id	complete_name	char_label	state_label	...
0	char. 1 1	char. 1	1	...
1	char. 1 0	char. 1	0	...
2	char. 2 1	char. 2	1	...
3	char. 2 0	char. 2	0	...

--	-----	-----	-----	-----

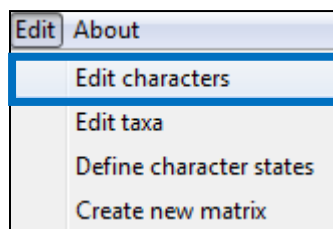


sample.phy_id.txt (after provided annotations)

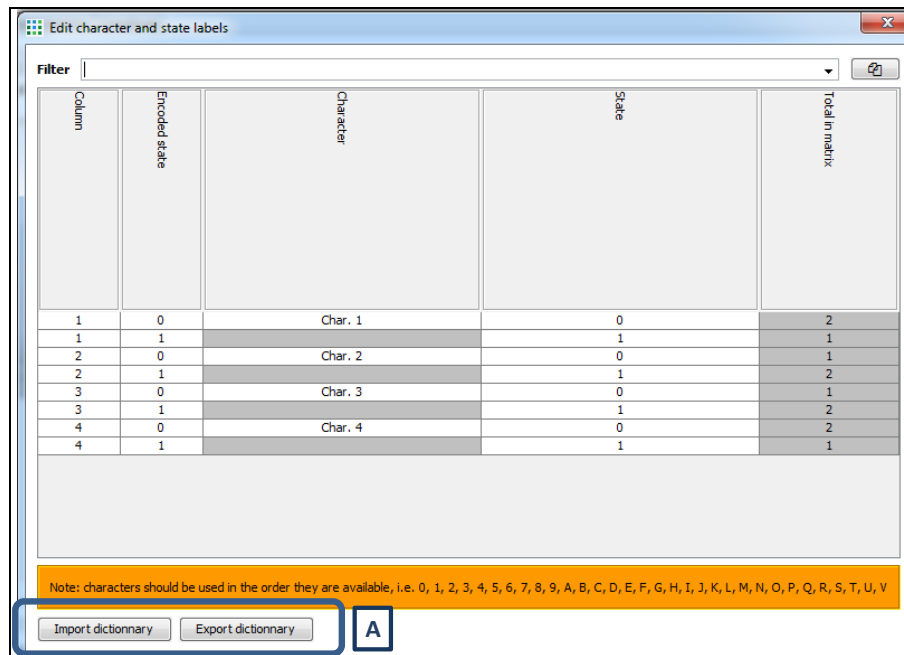
Node id	complete_name	char_label	state_label	...
0	Mouth opening orientation ventral	Mouth opening orientation	ventral	...
1	Mouth opening orientation terminal	Mouth opening orientation	terminal	...
2	Pre-oral chamber present	Pre-oral chamber	present	...
3	Pre-oral chamber absent	Pre-oral chamber	absent	...
4	Radially arranged circumoral structures absent	Radially arranged circumoral structures	absent	...
5	Radially arranged circumoral structures present	Radially arranged circumoral structures	present	...
6	Differentiated circumoral structures three or four enlarged plates	Differentiated circumoral structures	three or four enlarged plates	...
7	Differentiated circumoral structures undifferentiated plates	Differentiated circumoral structures	undifferentiated plates	...
8	Pharynx differentiated from midgut differentiated	Pharynx differentiated from midgut	differentiated	...
9	Pharynx differentiated from midgut not differentiated	Pharynx differentiated from midgut	not differentiated	...
10	Pharynx reversible permanently inverted	Pharynx reversible	permanently inverted	...
11	Pharynx reversible completely reversible	Pharynx reversible	completely reversible	...

5.2 External node annotation file in GUI mode

To save or import external nodes annotation in GUI mode, first go into the **Edit → Edit characters** submenu.

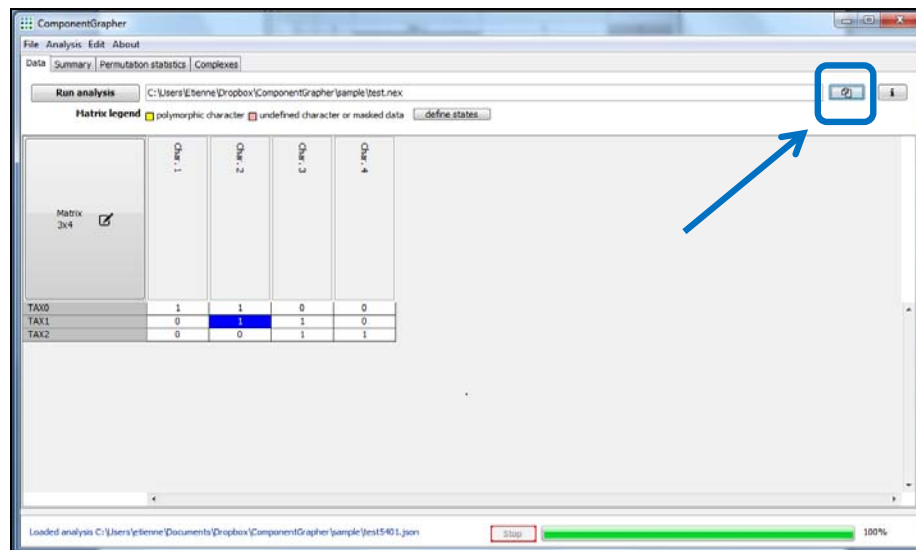


This will open a new popup windows. In this windows, use the **Import dictionary** or **Export dictionary** buttons (Box A).



5.3 Copy matrix from / to Excel

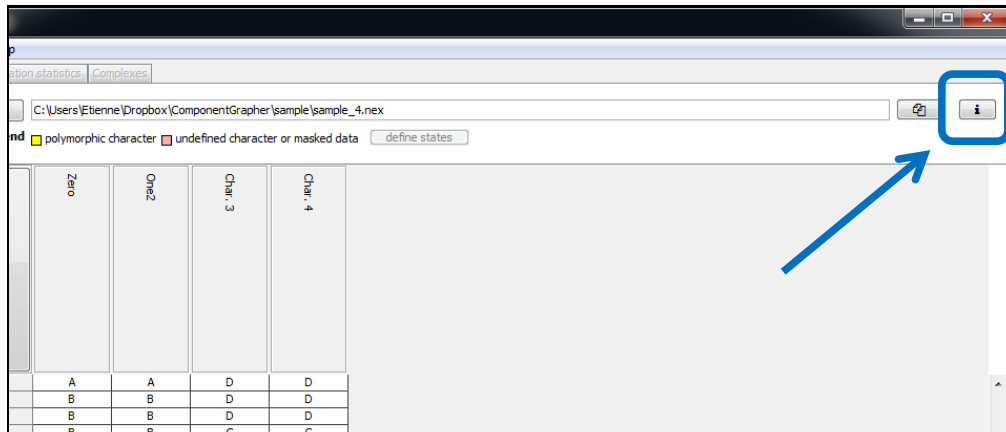
It is possible to copy/paste the current matrix from Microsoft Excel™. To copy the complete current matrix to Excel, use the Copy button located in the top right corner of the main interface (data tabs). To copy a single column, select the associated cells, then use the Ctrl-C combination. To paste data from Excel, use the Ctrl-V key combinations.



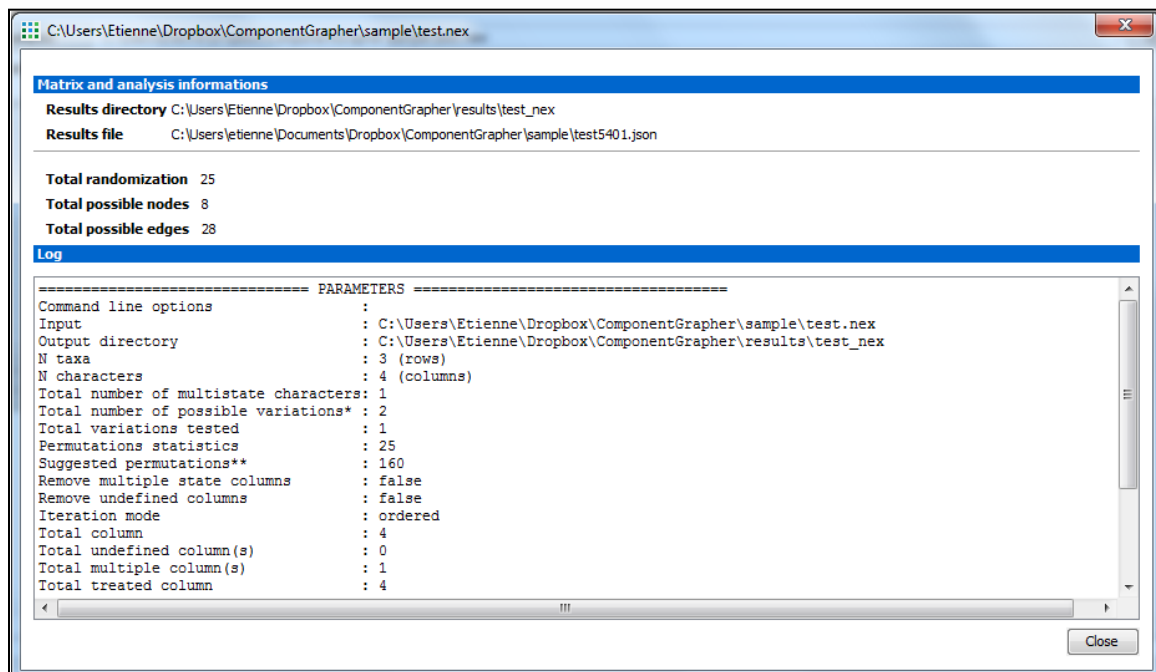
taxa/char	Char. 1	Char. 2	Char. 3	Char. 4
TAX0	1	1	0	0
TAX1	0	1	1	0
TAX2	0	0	1	1

5.4 Information about the current analysis

To display information about the current analysis *e.g.* number of nodes, output directory, etc. **click the Information button** located in the right top corner of the main interface view (data tabs).



This will open a new window with details about the results directory, results file, total number of permutation (*randomization*), total number of edges, etc.



5.4 Compiling the source code

The java source code provided is a **Netbeans project** (<https://netbeans.org>). To compile the source code, download the Netbeans IDE, open the project file and build the project. The required libraries are listed below:

SSJ(<https://github.com/umontreal-simul/ssj>)

The SSJ library is used for BitVector calculations and for computing random uniform distribution. The `ssj.jar` archive (version 3.2.0) is included in the COMPONENT-GRAPHER distribution and it must be in the CLASSPATH environment variable.

Apache Commons Mathematics Library (<http://commons.apache.org/proper/commons-math/>)

The Apache common mathematics library is used for distribution and p -value calculations. The `commons-math3-3.6.1.jar` archive is included in the COMPONENT-GRAPHER distribution and it must be in the CLASSPATH environment variable.

Google gson (<https://github.com/google/gson>)

The Google gson library is used for serialization/deserialization of the datasets. The `gson-2.6.2.jar` archive is included in the COMPONENT-GRAPHER distribution and it must be in the CLASSPATH environment variable.

5.5 Citation

Lord E, Pathmanathan, J, Corel, E, Makarenkov V, Lopez P, Bouchard F, Bhattacharya D, Antoine PO, Le Guyader H, Lapointe FJ, Baptiste E. (2017). Palaeontological trait networks identify fluidity in organismal evolution. (submitted to Nature Communications)

6. References

- [1] Felsenstein, J. 2005. PHYLIP (Phylogeny Inference Package) version 3.6. Distributed by the author. Department of Genome Sciences, University of Washington, Seattle.
- [2] Maddison DR, Swofford DL, Maddison WP. 1997. NEXUS: an extensible file format for systematic information. *Syst Biol.* 46(4):590-621.
- [3] Bastian M., Heymann S., Jacomy M. 2009. Gephi: an open source software for exploring and manipulating networks. International AAAI Conference on Weblogs and Social Media.

7. Abbreviations

Some abbreviations can be found in the software outputs.

Abbreviations	Descriptions
N	Number of permutations (replicates)
%triplet_type3	Number of time (in percent) that this node is central in a triplet
between_type3	Node betweenness in type3 network
between_type3'	Betweenness centrality of node (type 3 network)
CC_complete	Number of complexes (complete network composed of type 1, 2, and 3 edges)
CC_type1	Number of complexes (type 1 network)
CC_type3	Number of complexes (type 3 network)
closeness_type3	Closeness centrality of node (type 3 network)
den_complete	Network density (complete network composed of type 1, 2, and 3 edges)
density_type1	Network density (type 1 network)
density_type2	Network density (type 2 network)
density_type3	Network density (type 3 network)
density_type4	Network density (type 4 network)
in_degree1	Degree of nodes (type 1 network, undirected)
in_degree2	In-degree of nodes (type 2 network)
in_degree3	Degree of nodes (type 3 network, undirected)
n_edges_complete	Number of edges (complete network composed of type 1, 2, and 3 edges)
n_edges_type1	Number of edges (type 1 network)
n_edges_type2	Number of edges (type 2 network)
n_edges_type3	Number of edges (type 3 network)
n_edges_type4	Number of edges (type 4 network)
out_degree2	Out-degree of nodes (type 2 network)
prop_triangle	Proportion of triangles (type 3 network, over the total number of possible triangles in a complete graph)
sign.	Significance of p-value from the permutation statistics and Bonferronicorrection for multiple tests (* p<0.05, ** p<0.01, *** p<0.001)
time/net. (ms)	Time to process a network (in milliseconds)
total_ap_compl.	Number of global articulation points (complete network composed of type 1, 2, and 3 edges)
total_ap_type3	Number of global articulation points (type 3 network)
local_ap_compl.	Number of local articulation points (complete network composed of type 1, 2, and 3 edges)
local_ap_type3	Number of local articulation points (type 3 network)
triangle_type3	Number of triangles (type 3 network)
triplet_type3	Number of non-transitive triplets (type 3 network)
triplet_typeA	Number of triplets of type A. For a node n, the triplet is in the form i -3- n -3- j where i and j are other nodes connected to n with a type 3 edge and i and j are not connected)

triplet_typeB	Number of triplets of type B. For a node n, the triplet is in the form $i-3-n-3-j$ where i and j are other nodes connected to n with a type 3 edge and i and j are connected with a type 1 edge)
triplet_typeC	Number of triplets of type C. For a node n, the triplet is in the form $i-3-n-3-j$ where i and j are other nodes connected to n with a type 3 edge and i and j are connected with a type 2 edge)
triplet_typeD	Number of triplets of type D. For a node n, the triplet is in the form $i-3-n-3-j$ where i and j are other nodes connected to n with a type 3 edge and i and j are connected with a type 4 edge)
triangle	Triangle where node n is present (type 3 network)