Introduction to Exploratory Network Analysis for Archaeologists using visone

Workshop 5 @ Computer Applications and Quantitative Methods in Archaeology (CAA) 2015

Daniel Weidele, Tom Brughmans

March 30, 2015

Contents

Exercise -1: Installing visone

The goal of this exercise is to have visone running on your computer.

For detailed installation instructions please check out the official visone Wiki here:
`http://visone.info/wiki/index.php/Installation_%28trail%29`

Requirement:

- ☐ Java SE is installed on your operating system:
  `http://www.oracle.com/technetwork/java/javase/downloads/jre8-downloads-2133155.html`

Variant a) Webstart:

- ☐ Visit `http://visone.info/` with your web browser.
- ☐ Click `Webstart` to launch visone directly from the browser.

Variant b) Local Installation:

- ☐ Download the current stable visone version here:
  `http://visone.info/html/download.html`
- ☐ Run the downloaded visone JAR file by *double-clicking* the downloaded file, or executing the following command:
  `java -jar visone-x.x.x.jar`

  Tip: You can run visone with more memory by specifying e.g. -Xmx2g for 2GB of memory:
  `java -Xmx2g -jar visone-x.x.x.jar`

Result

- ☐ You see the graphical user interface of visone .

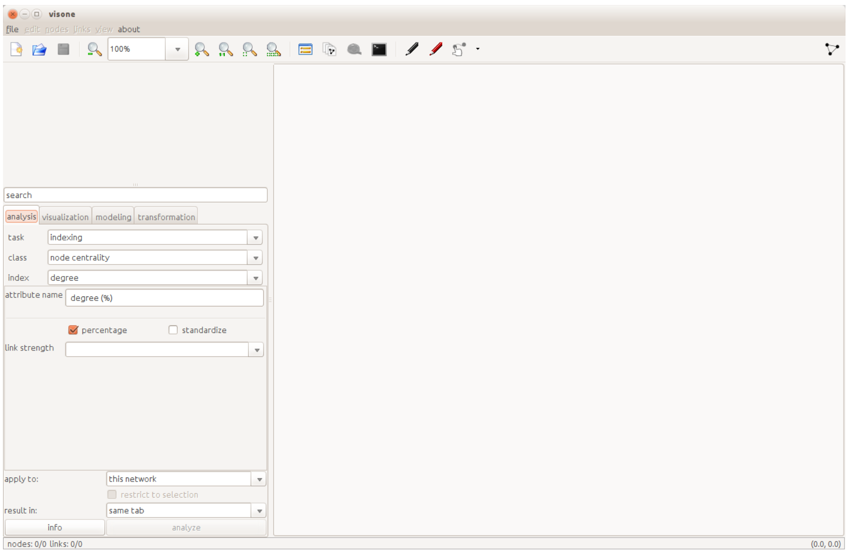


Figure 1: Graphical user interface of visone after startup.

You accomplished the exercise :-)

Now you are ready for the tutorial!

Chapter 1

Introduction

1.1 Welcome to visone

So, you finally decided to get into the basics of visone . That's great! visone is developed by the members of the Algoritmics Group of the Department of Computer and Information Science at the University of Konstanz. It is a free research tool and does not aim for any kind of commercialization. visone allows you to apply various algorithms and methods in order to analyze and visualize networks. This tutorial is meant for beginners who have little or no experience with Network Science.
You can find more advanced tutorials and further information here:
`http://visone.info/wiki/index.php/Main_Page`

Also, in case you experience any trouble, the official visone news group might be of interest to you:
`https://groups.google.com/forum/#!forum/visone-users`

The tutorial is built up in the following way:

- We will first have a general look at the visone graphical user interface.
- The next chapters address Network Science topics, where within a chapter
  - first, we explain methodological basics, and
  - then you can actively experience visone throughout exercises.

  This way of approaching the topic shall emphasize the importance of understanding the principles behind the methods offered in visone *before* applying them, to not eventually end up with misleading interpretation of the data.
- The last chapters point out more Network Science software resources and advanced reading material.

1.2 Tutorial Overview

Throughout this tutorial you will create 2 network visualizations of a Maya obsidian network. You will get to know more details on the data set at the beginning of exercise 1. Figure 2 provides an overview about the methodological pipeline you will run through within 10 exercises.

The general steps are as follows:

- ☐ Loading the data set from CSV files (Exercise 1 & 2).
- ☐ Filtering and editing the network (Exercise 3 to 5).
- ☐ Computation of centrality measures (Exercise 6).
- ☐ Clustering of nodes (Exercise 7).
- ☐ Visualizations (Exercise 8 to 10).

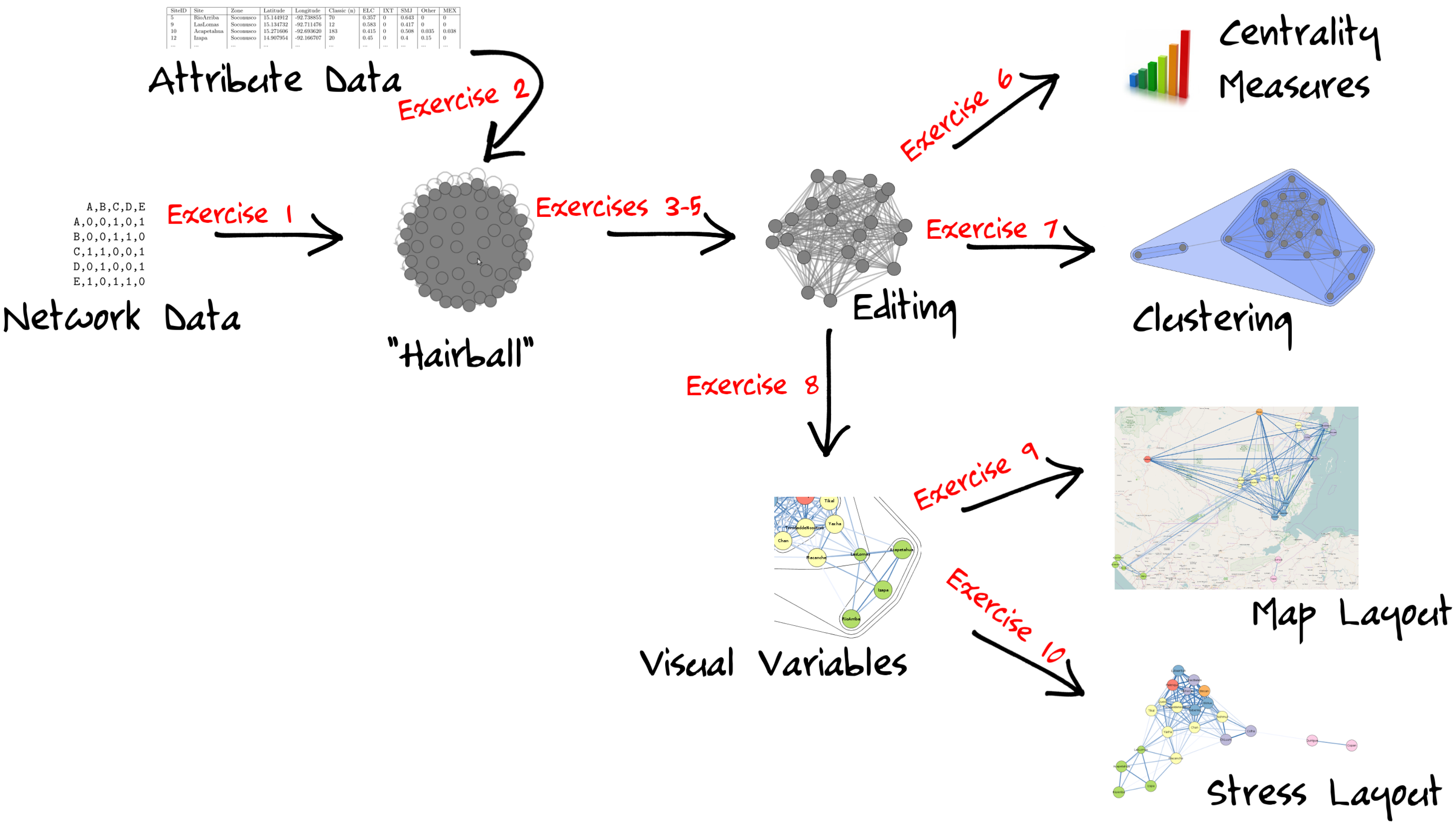


Figure 2: Tutorial Overview. Throughout 10 exercises you will learn how to load a Maya obsidian network in visone. In next steps you will reveal more structure of the network by filtering nodes and edges from the *hairball*. Centrality measures and clustering will provide further insight into the network. In the last steps you will see how to produce visualizations by mapping attribute data to visual variables and applying meaningful layout techniques.

1.3 visone Interface

Figure 3 shows the interface of visone .

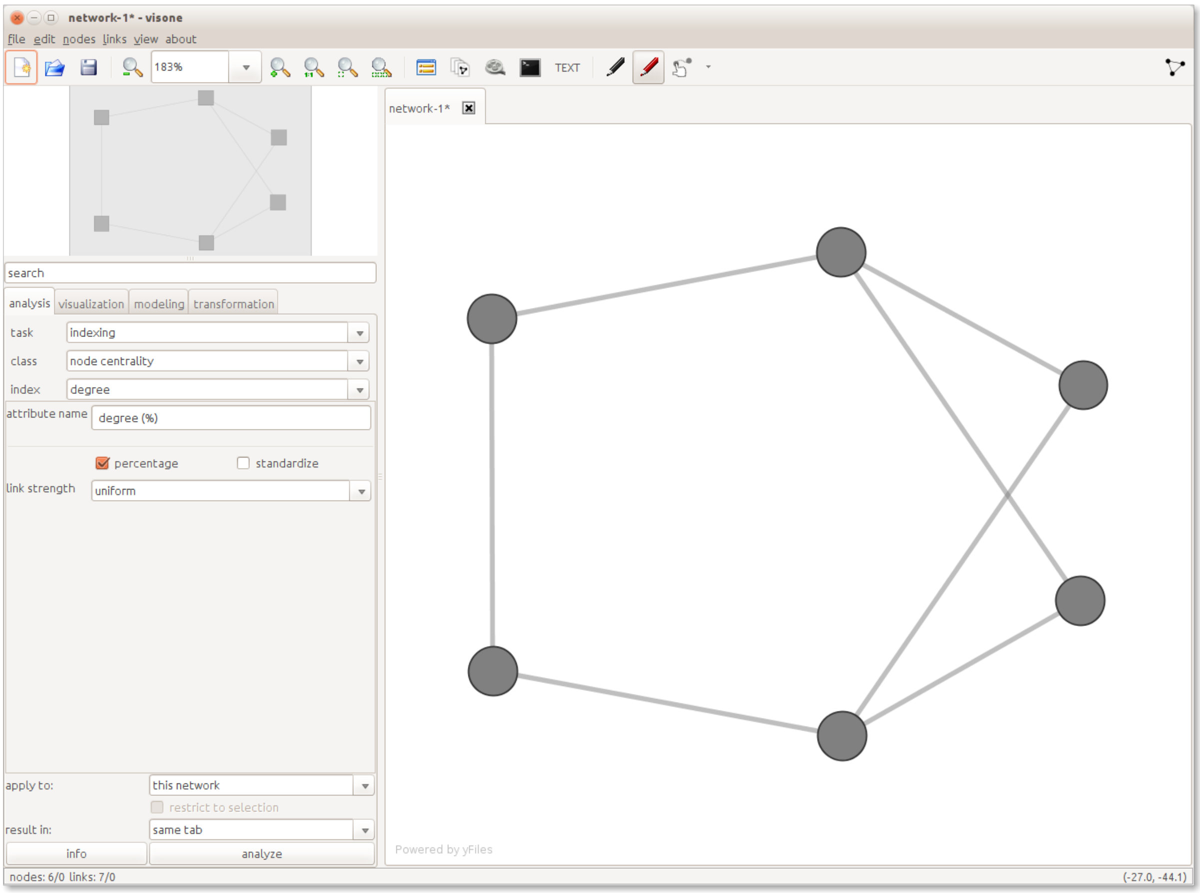


Figure 3: Graphical User Interface of visone .

- There is a *menu bar* at the top of the window containing the menus *file, edit, nodes, links, view* and *about*.
- Below the menu bar you find a *tool bar* offering quick access to general functions.
- On the right you find the *network panel* showing a basic network with 6 nodes and 7 edges.
- On the left below the tool bar there is a *network overview panel* which provides an overview about the network and highlights the viewport of the network panel in grey.
- Below the network overview panel you can find a *search box* to quickly access algorithms and methods.
- Below the search box there are 4 *tabs* each grouping methods for *analyzing, visualizing, modeling* and *transforming* a network.
- Within a *tab panel* you can select and configure a method, e.g. in figure 3 there is selected a method to compute the degree of the network nodes.
- Below the tab panel there is a *control panel* where you can choose a network to apply a method on, get some *info* on the method itself and execute a method by clicking on *analyze*.

1.3.1 Menu Bar

The menu bar is structured in 6 menus:

File menu

- *new* - create a new network in the network panel
- *open* - load a network from file
- *create* - generate a network by with network generation algorithms
- *create copy* - create a copy of the currently displayed network in the network panel
- *save* - save the currently displayed network
- *save as...* - open the save dialog to save the currently displayed network with different name or format
- *export* - export the currently displayed network as image, PDF or other special file formats
- *print* - print out the currently displayed network
- *options* - change general settings of visone and setup extensions (e.g. connection to R)
- *close network* - close the currently displayed network
- *exit* - exit visone

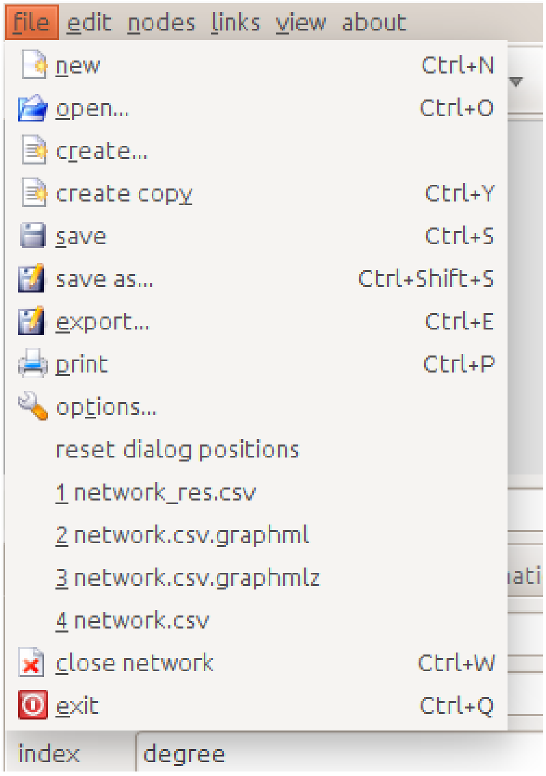


Figure 4: File menu of the visone menu bar.

Edit menu

- *select all* - select all nodes and edges
- *deselect all* - deselect any selected nodes or edges
- *invert selection* - invert the selection of selected nodes and edges
- *cut* - cut the currently selected nodes and edges
- *copy* - copy the currently selected nodes and edges
- *paste* - paste the nodes and edges that have been cut or copied before into the currently displayed network

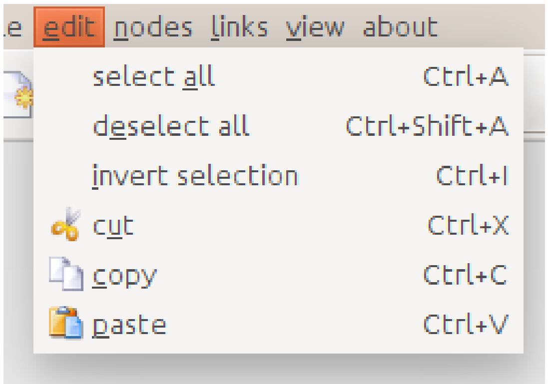


Figure 5: Edit menu of the visone menu bar.

Nodes menu

- *delete nodes* - delete the currently selected nodes
- *delete isolates* - delete the nodes that are not connected to any other node (called *isolates*)
- *delete all group nodes* - delete all nodes that represent a group of nodes (called *groups*)
- *select all* - select all nodes
- *deselect all* - deselect all nodes
- *select neighbors* - expand the current selection of nodes by additionally selecting their directly connected nodes (called *neighbors*) and edges
- *select incident links* - expand the current selection of nodes by additionally selecting any direct links between the selected nodes
- *invert selection* - invert the selection of selected nodes
- *select groups* - select all group nodes
- *templates* - opens the node template dialog
- *properties* - opens the node properties dialog

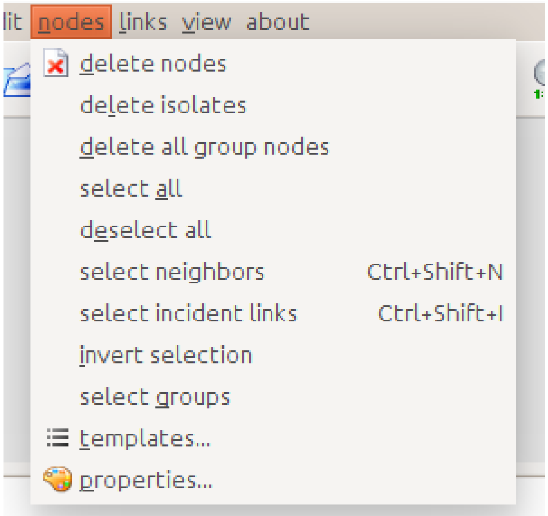


Figure 6: Nodes menu of the visone menu bar.

Links menu

- *delete nodes* - delete the currently selected links
- *delete loops* - delete links from a node to itself (called *loops*)
- *delete bends* - delete anchor points of links (called *bends*)
- *reverse direction* - reverse the direction of directed links
- *make directed* - make undirected links directed
- *make undirected* - make directed links undirected
- *select all* - select all links
- *deselect all* - deselect all links
- *invert selection* - invert the selection of selected links
- *templates* - opens the link template dialog
- *properties* - opens the link properties dialog

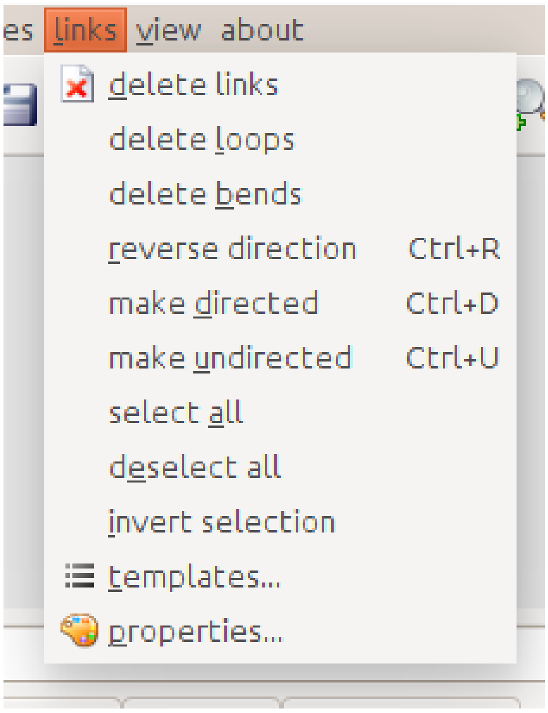


Figure 7: Links menu of the visone menu bar.

View menu

- *zoom in* - increase the zoom level of the network panel
- *zoom out* - decrease the zoom level of the network panel
- *zoom 100%* - zoom the network panel to 100%
- *fit network* - fit the whole network into the network panel
- *fit all networks* - fit all opened networks into their respective network panels
- *refresh* - refresh the network panel
- *draw multi-links* - allow the network panel to display more than one link between two nodes (called *multi-links*)
- *clear background* - remove the background of the network panel
- *modes* - change the interaction mode with the network panel (*analysis, edit* and *stress mode*).

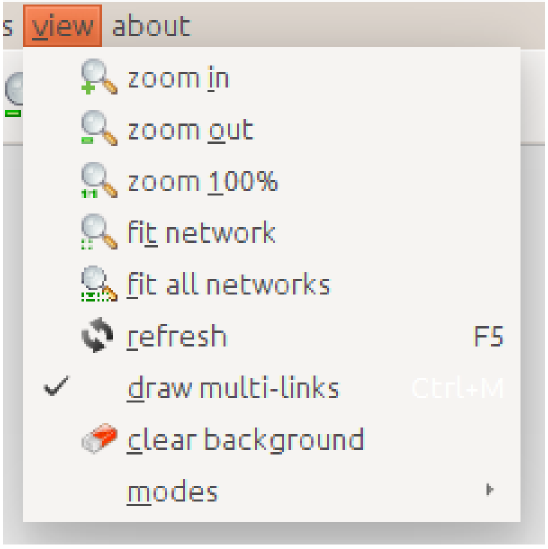


Figure 8: View menu of the visone menu bar.

About menu

- *about visone* - get some information about visone (version number, link to website, licenses)

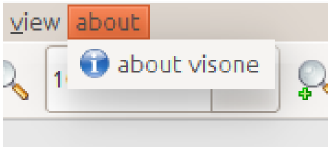


Figure 9: About menu of the visone menu bar.

1.3.2 Tool Bar

We quickly outline the functionality of the buttons in the visone tool bar *from left to right*:

Figure 10: visone tool bar.

- *new empty network* - create a new empty network tab in the network panel
- *open network from file* - load an existing network from file
- *save network to file* - save the currently displayed network in the network panel to a file
- *zoom out* - decrease the zoom level of the network panel
- *zoom level combo box* - select a specific zoom level
- *zoom in* - increase the zoom level of the network panel
- *zoom to 100%* - set the zoom level to 100%
- *zoom to fit network* - automatically adjust the zoom level to perfectly fit your network
- *zoom to fit all* - automatically adjust the zoom level of all open networks to a common viewport
- *attribute manager* - opens the attribute manager

1.3.3 Interaction Modes

In the tool bar you can switch between three different interaction modes to interact with the network panel.

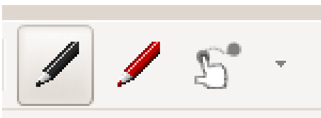


Figure 11: Interaction modes in the tool bar.

- *analysis mode*

  In analysis mode you can select nodes and edges, or drag nodes to change their position.

- *edit mode*

  In edit mode you can select nodes and edges, or drag nodes to change their position.

- *stress mode*

  In stress mode you can improve local stress while dragging nodes, which is useful in the exploration of large graphs.

1.4 Reminders

Before we start out with the first exercise make yourself aware of **3 important reminders** while working with visone .

1. There is **no undo / redo** functionality.

   This feature would require extensive memory usage, which is not practical for most Desktop environments. Instead, we recommend to save intermediate results to separate files, as well as to make use of the `Result in New Tab` feature that can be found at the bottom left of the visone window.

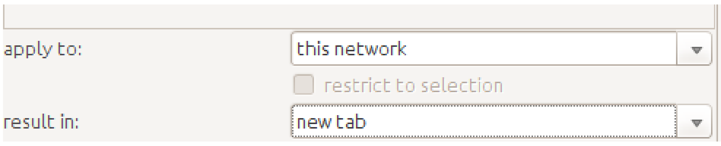


Figure 12: Instead of overwriting the active network you can show the result in a new tab whenever applying changes to your network.

2. There is a **Quick Layout Button** at the top right of the visone window.

   The quick layout button provides a default layout of your network. This is meant for getting quick insight, but should not replace in-depth thinking about which layout method to choose when producing actual visualizations.

Figure 13: Use the `Quick Layout` button to get a default layout of your network.

3. Use the **search box** if you forgot where to find a certain function or to quickly access methods.

   The search box on the left of the visone window provides fast access to the features.

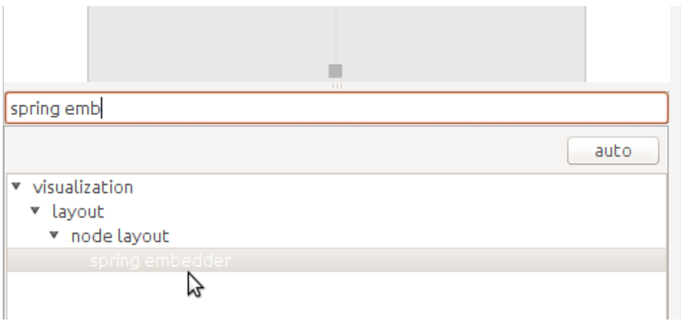


Figure 14: Example of a `search box` query.

Exercise 0: Drawing a New Network

The goal of this exercise to get familiar with the visone interface and the interaction with the network panel.

Tasks

- ☐ Create a new network panel by opening the `file` menu and selecting `new`.

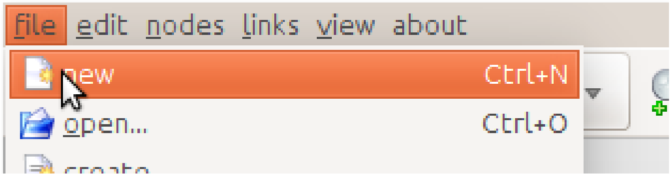


Figure 15: Create a new network panel.

- ☐ Switch to `edit mode` by clicking the red pencil in the `tool bar`.

Figure 16: Switch to edit mode.

- ☐ Click on the `network panel` to create some new nodes.

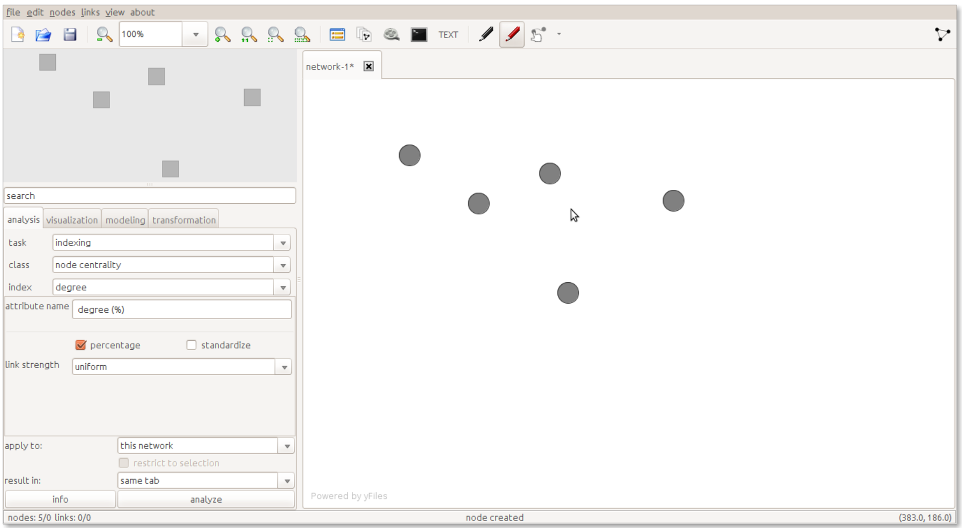


Figure 17: Create some nodes.

☐ Click a node (the link *source*) to start a new link, and click another node (the link *target*) to end the link.

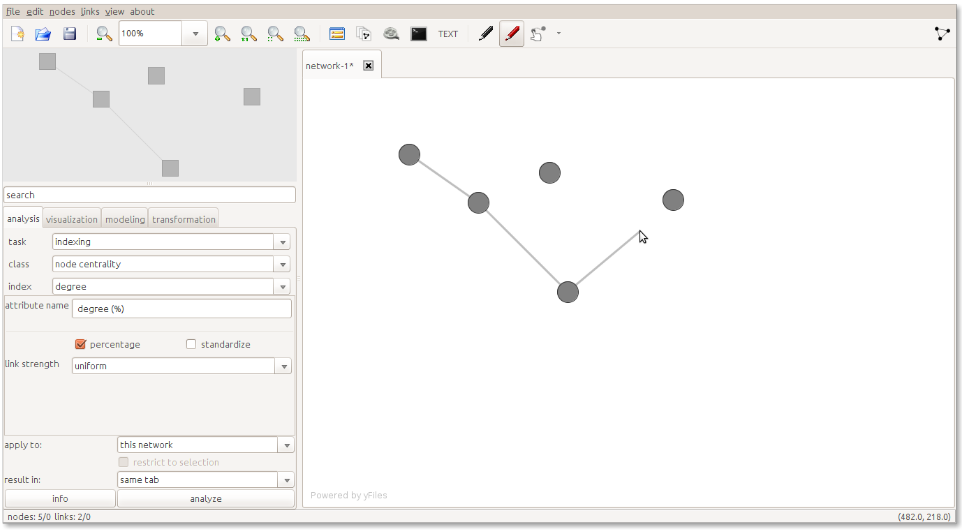


Figure 18: Create some links.

☐ Click a node to start a new link, and click *the same* node again to make a loop to the node itself.

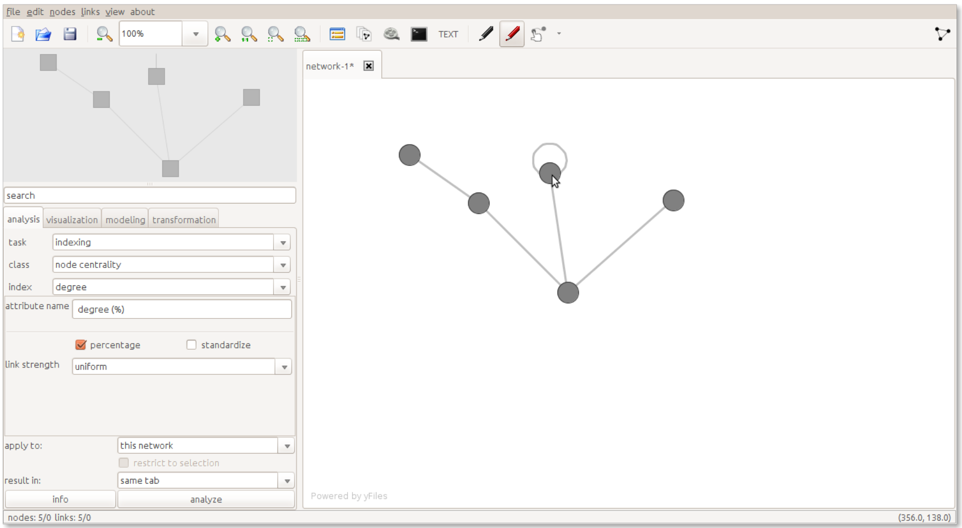


Figure 19: Create a loop.

☐ Show the direction of the links by selecting `make directed` in the `links` menu.

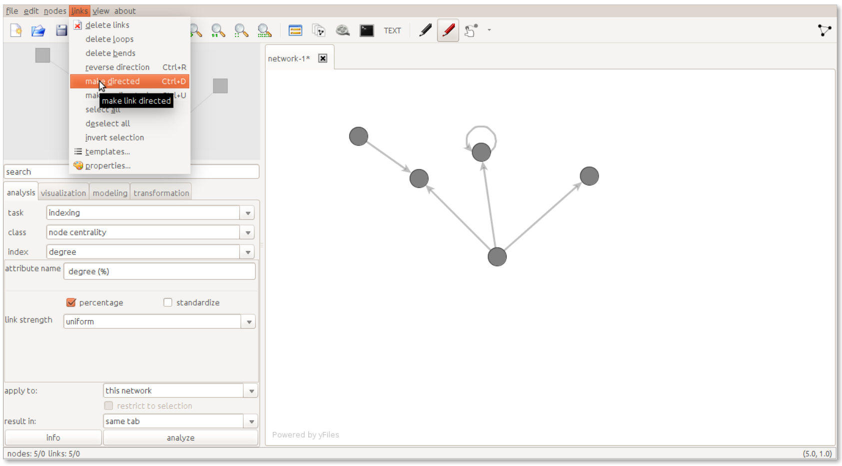


Figure 20: Make the network directed.

☐ Switch to `analyze mode` by clicking the black pencil in the `tool bar`.

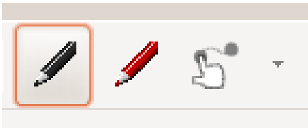


Figure 21: Switch to analyze mode.

☐ Select some nodes by clicking and holding (called *dragging*) the left mouse button and drawing a rectangle around them.

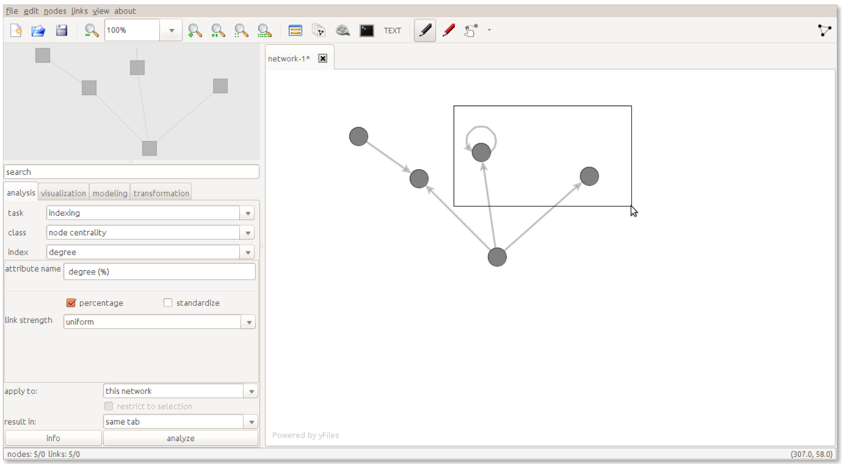


Figure 22: Multiple selection of nodes.

□ Change the position of the selected nodes by dragging any of the selected nodes to another position.

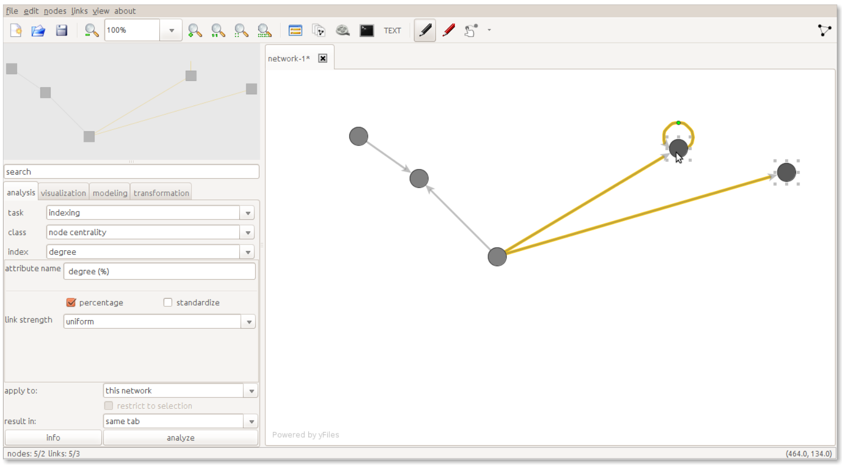


Figure 23: Change position of selected nodes.

□ Right click any of the selected nodes and `group` them.

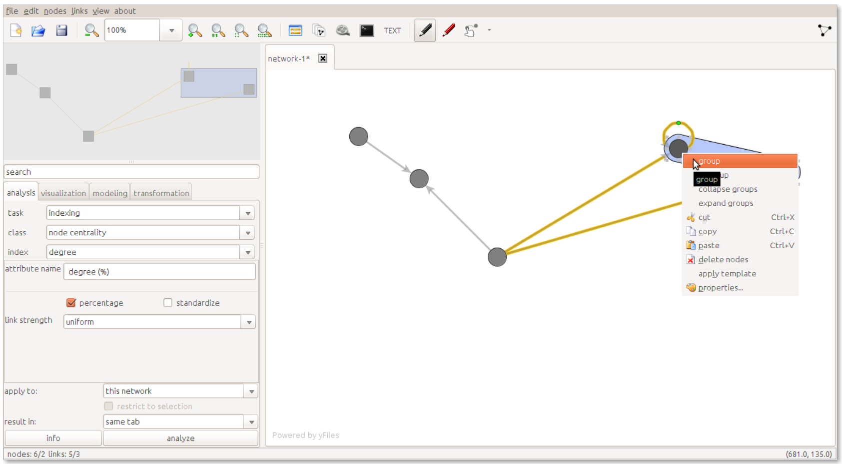


Figure 24: Group selected nodes.

□ Click the `file` menu and `export` your network to a folder of your choice as a PNG image using the `export dialog`.

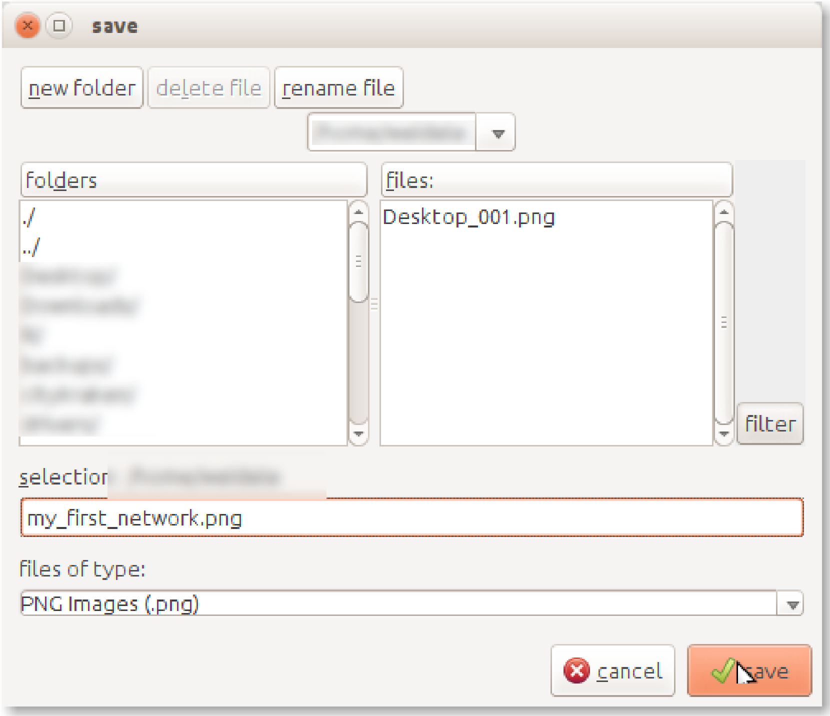


Figure 25: Export your network as PNG image.

Result

□ You see the exported PNG file of your network in your chosen folder.

You accomplished the exercise :-)

You are now familiar with the visone interface and the basic interaction with the network panel.

Chapter 2

CSV-based File Formats

Of course you do not want to manually redraw your real world data sets by hand. Therefore visone supports various input file formats.
In general, the transformation of your raw data sets into visone file formats is done in e.g. Microsoft Excel, LibreOffice Calc, or also scripting languages such as R, Python, etc.
Throughout the exercises you will be provided with a preprocessed data set that you can just directly import. However, we still want to give a short overview about typical file formats, so that in the future you can decide on your own which is the easiest format into which you can transform your data sets.

Consider the following examplary network of 5 nodes with labels A, B, C, D, E and 6 edges:

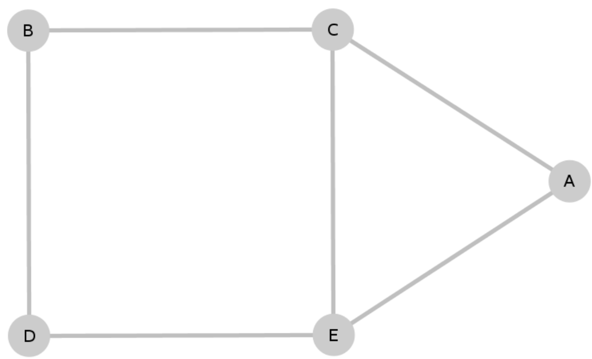


Figure 26: Examplary network consisting of 5 nodes and 6 edges.

2.1 Adjacency Matrix

Assume you have 5 nodes in the network as given in figure 26. The *adjacency matrix* is then a table of 5 rows and 5 columns with the node labels as row and column header.
Now whenever there is a link between 2 nodes, say between node A and node C, the corresponding cell value at row A and column C is set to 1. When there is no link between two nodes the respective cell value is set to 0.
The adjacency matrix of the network in figure 26 would then look like the following:

```
 A,B,C,D,E
A,0,0,1,0,1
B,0,0,1,1,0
C,1,1,0,0,1
D,0,1,0,0,1
E,1,0,1,1,0
```

Note that in the case of undirected networks this matrix is symmetric over the diagonal, whereas in directed networks this does not need to be the case. Instead of 1's you may also specify a decimal value as weight for the edges.

2.2 Adjacency List

An alternative way of formatting your data is the *adjacency list*. The adjacency list is a list of rows where the node specified in the first column is adjacent (that means *connected via a link*) to all the remaining nodes of the columns of the same row. Considering the example network in figure 26 where node A is connected to nodes C and E, we would for example create a row

```
A,C,E
```

in the adjacency list, meaning that A (first column) is adjacent to C and E (remaining columns). The full adjacency list of the network in figure 26 would therefore look like the following:

```
A,C,E
B,C,D
C,A,B,E
D,B,E
E,A,C,D
```

Note that in the case of undirected networks it is not necessary to specify adjacency twice in reversed direction, so the above adjacency matrix could be further simplified to

```
A,C,E
B,C,D
C,E
D,E
```

2.3 Edge List

In the *edge list* format you can further specify attributes for the links between nodes, e.g. weight of the link, or type of the link. Compared to the adjacency list, you cannot put multiple target nodes in one row, but need to specify one row for each pair of source and target node. We make up additional link attributes `weight` (e.g. between 1 and 5) and `type` (e.g. one of X, Y, Z) for the network in figure 26, and thus would receive the following edge list:

```
Source, Target, Weight, Type
A,C,5,X
A,E,3,X
B,C,5,Y
B,D,1,Z
C,E,4,X
D,E,5.Y
```

2.4 Other File Formats

You can check out further supported file formats (e.g. GraphML, UCINET, Pajek, Siena, ...) in the visone Wiki here:
`http://visone.info/wiki/index.php/Supported_network_formats`

Exercise 1: Loading the Data Set

The goal of this exercise is to load a data set to work with during the upcoming exercises.

Preliminaries

We prepared a Maya Obsidian data set kindly provided by [Golitko et al., 2012]. The original data set consists of ≈ 120 sites for which obsidian distributions from 5 different sources has been assembled over 4 periods. To provide you an overview of the geographic situation figure 27 shows a map of eastern Mesoamerica where the sites are displayed as nodes. We colored the site nodes following the classification into geographical zones by [Adams and Culbert, 1977].

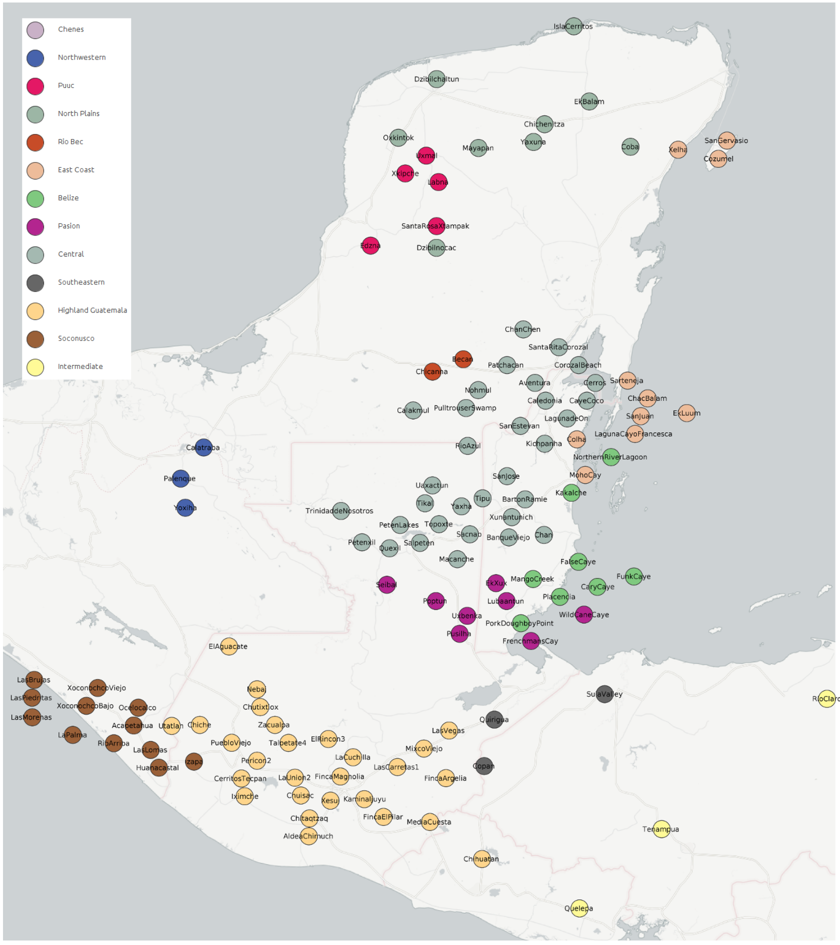


Figure 27: Map of eastern Mesoamerica showing the sites contained in the data set as nodes. The node colors encode the zone of the site as defined in [Adams and Culbert, 1977].

Throughout the exercises we will only make use of the data of the first period (Classic). For each site the data set contains the total amount of sourced obsidian and the distribution of obsidian according to the 5 sources `El Chayal (ELC)`, `Ixtepeque (IXT)`, `San Martin Jilopeque (SMJ)`, `Other (Other)` and `Mexican (MEX)`. The data set looks like the following:

SiteID	Site	Zone	Latitude	Longitude	Classic (n)	ELC	IXT	SMJ	Other	MEX
5	RioArriba	Soconusco	15.144912	-92.738855	70	0.357	0	0.643	0	0
9	LasLomas	Soconusco	15.134732	-92.711476	12	0.583	0	0.417	0	0
10	Acapetahua	Soconusco	15.271606	-92.693620	183	0.415	0	0.508	0.035	0.038
12	Izapa	Soconusco	14.907954	-92.166707	20	0.45	0	0.4	0.15	0
...	...	...	...	...	...	...	...	...	...	...

Table 2.1: Samples from the Maya Obsidian Data Set

Based on this data we related each two sites by computing the Brainerd-Robinson similarity [Robinson, 1951, Brainerd, 1951] between their respective obsidian distribution with the software `R`. The result is an adjacency matrix where the cell value indicates the similarity between the two sites in the respective row and column - the higher the value, the more similar are the two sites according to the obsidian in their assemblages.

Tasks

- ☐ Click the `open` button in the visone tool bar.

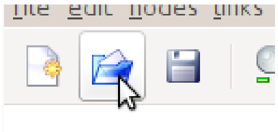


Figure 28: Click the `open` button in the visone tool bar.

- ☐ Make sure you selected `.csv` as the file type and open the adjacency matrix file `maya_network.csv` that you can find in the workshop supplementary material.

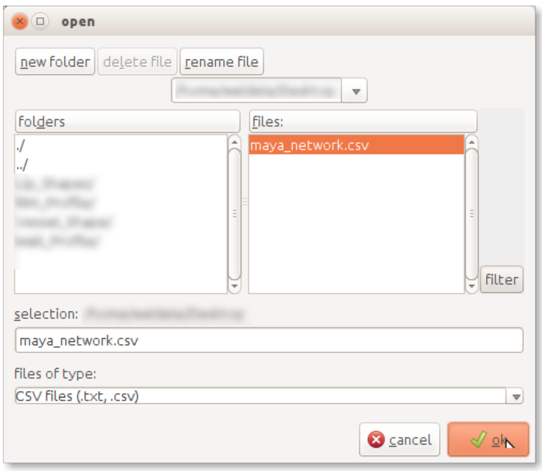


Figure 29: Open the adjacency matrix file `maya_network.csv` that you can find in the workshop supplementary material.

☐ In the `import dialog` select `adjacency matrix` as the data format, make sure you have the same settings as given in figure 30 and click `ok`.

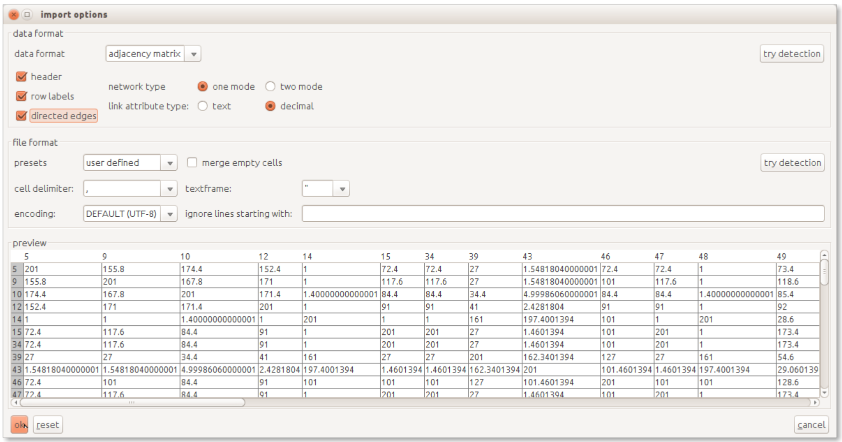


Figure 30: Import dialog

Result

☐ The network panel shows the loaded network of Maya sites as nodes. Links hold the Brainerd-Robinson Similarity between each two connected sites. Since there is a link from any node to any other node (called a *complete network*) we only see a meaningless *hairball*. Throughout the next chapter you will learn ways of gaining more insight into the structure of the network.

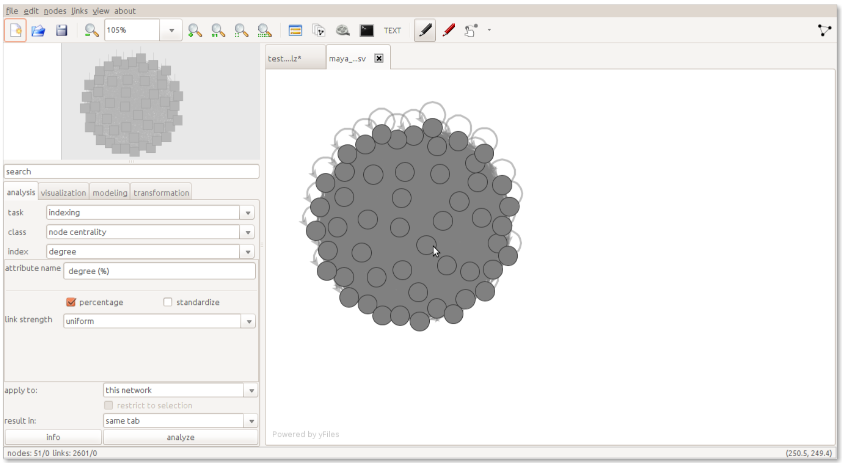


Figure 31: Network loaded in the `network panel`

You accomplished the exercise :-)

You are now able to load networks in visone .

Chapter 3

Network Editing

In this chapter we will make use of the `attribute manager` of visone . The attribute manager is a powerful tool that allows you to

- browse and edit node or link attribute values,
- filter for nodes or links based on attribute values,
- configure node or link attributes,
- calculate with node and link attributes,
- import additional node or link data.

3.1 Visone Attribute Manager

The attribute manager can be opened via the toolbar as shown in figure 32.

Figure 32: Opening the attribute manager via the tool bar.

At the top of the attribute manager dialog there are 4 buttons (*node, link, dyad, network*). In most of the cases you will choose either *node* or *link* here, so we will ignore *dyad* and *network* in this tutorial. The 6 edit operations at the left of the dialog provide access to different functions for editing node or link *attributes*. We will shortly outline the possibilities covered by the edit buttons before you can actively use them throughout another exercise.

- *Show & Edit* - the Show & Edit button opens a view where you can take a look at the node or link attributes. Also, when you activate the *allow editing* checkbox at the bottom left of the view, you can directly edit attribute values.

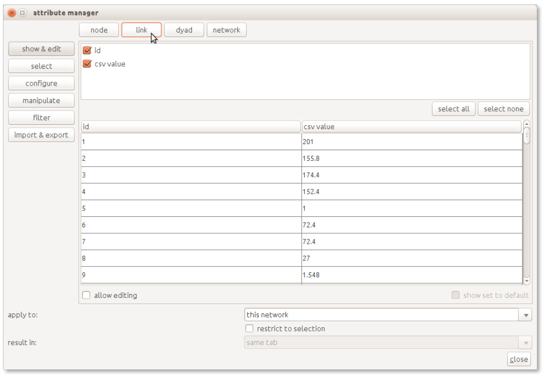


Figure 33: Show & Edit View of visone attribute manager.

- *Select* - the Select button allows you to select e.g. links based on a certain attribute value. In figure 34 we selected all links for which the attribute *csv value* is 201. The selected links are highlighted in the visone network panel.

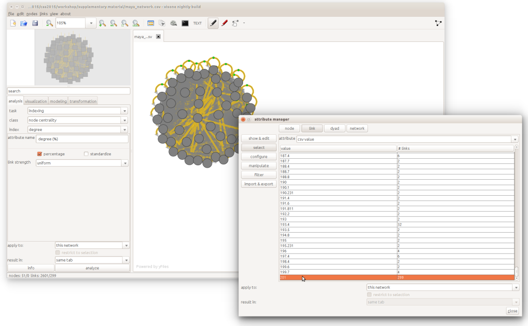


Figure 34: Select View of visone attribute manager.

- *Configure* - the Configure button opens a view where you can modify the specification of attributes. For example, you can rename attributes, or create and specify the attribute type as indicated for links in figure 35.

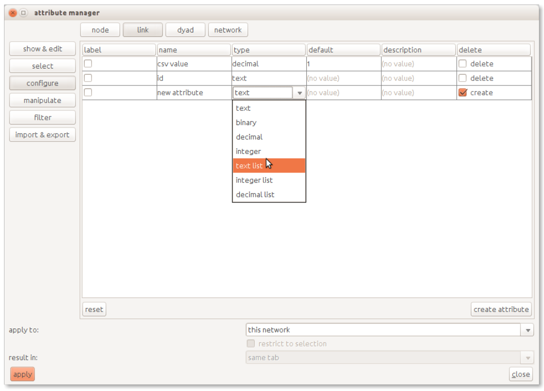


Figure 35: Configure View of visone attribute manager.

- *Manipulate* - the Manipulate button opens a view where you can manipulate attribute values with different kinds of operations. For example, in figure 36 we `normalize` the attribute *csv value* (to the range $[0, 1]$) and write the result to a new attribute called *normalized value*. In the manipulate view you can also compute the mean of an attribute, add or multiply values, etc.

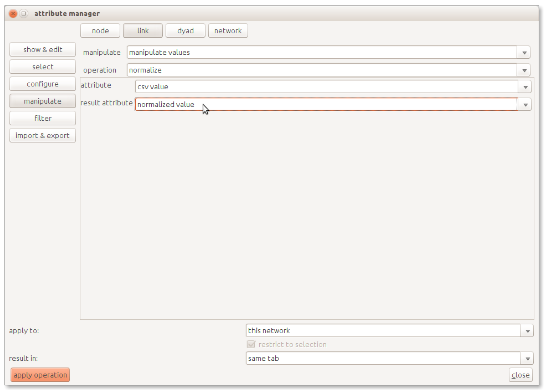


Figure 36: Manipulate View of visone attribute manager.

- *Filter* - the Filter button provides access to a view where you can specify simple queries to filter for certain nodes or links. Figure 37 shows how to select all links for which the *csv value* is `higher than` 150. With the radio button group at the bottom of the view you can further specify whether the filter result should `replace`, `add` to, be `removed` from or be `interesected` with the current selection. For example, you could first select all links for which the *csv value* is higher than 150, and then `intersect` with all links for which the *csv value* is lower than 170, in order to obtain a selection of all links with *csv value* between 150 and 170.

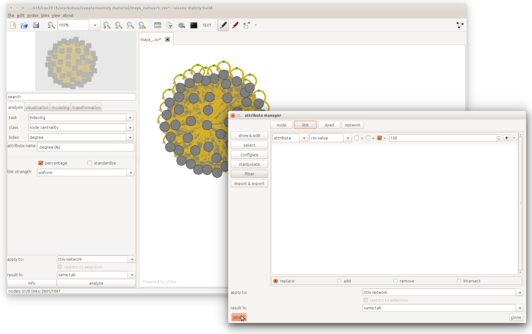


Figure 37: Filter View of visone attribute manager.

- *Import & Export* - the Import & Export button opens a view where you can import additional (or export existing) attributes that are attached to nodes or links of your network. When you load an adjacency matrix as done before throughout Exercise 1, visone only creates *identifier* attributes (called *ID*) for nodes and links, and the *csv value* attribute for links which represents the weight (i.e. the strength of the connection). The *Import & Export* view allows to load additional information, e.g. other link attributes, or, as in our Maya obsidian network, information about the site geography and obsidian distribution as indicated in figure 38.

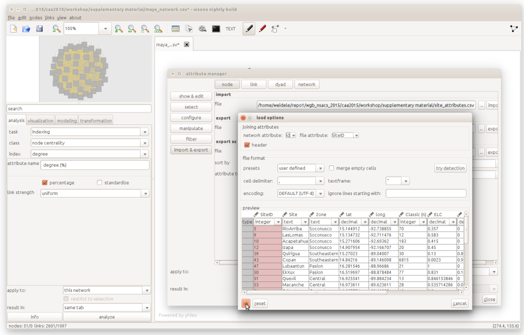


Figure 38: Import & Export View of visone attribute manager with Node Import Dialog in the foreground.

Exercise 2: Importing additional Site Attributes

The goal of this exercise is to import additional site attributes that have not been encoded in the adjancency matrix so far.

Tasks

□ Click the `attribute manager` button in the visone tool bar.

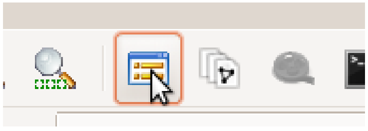


Figure 39: Click the `attribute manager` button in the visone tool bar.

□ Select the `node` button at the top of the attribute manager, and click `import & export` on the left of the attribute manager. You see the `import & export view`.

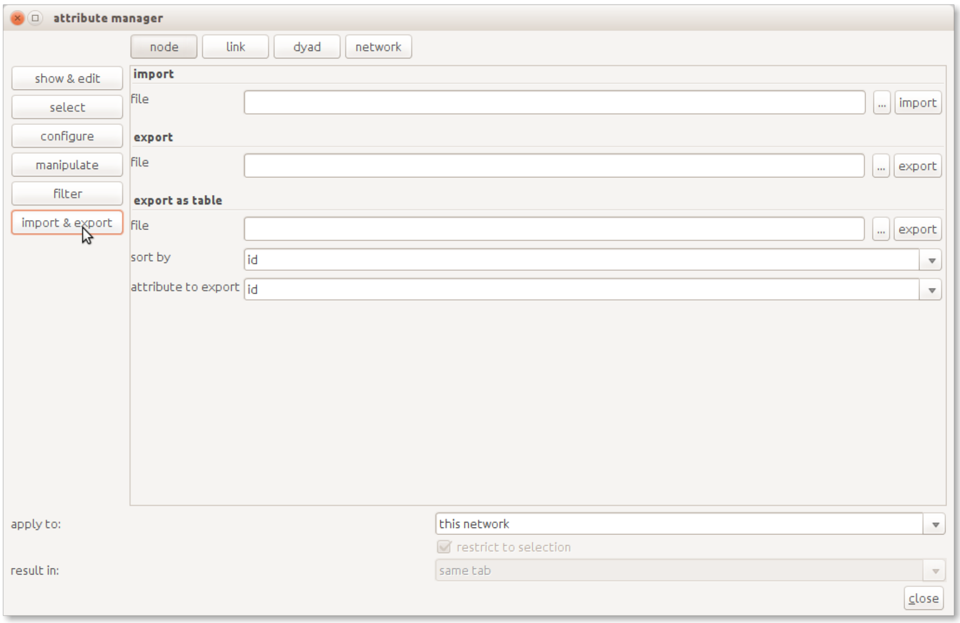


Figure 40: Import & Export view of the visone attribute manager.

- ☐ Click the '...'-button left of the import button in the `import & export view` of the attribute manager as shown in figure 41. A `file import` dialog will appear.

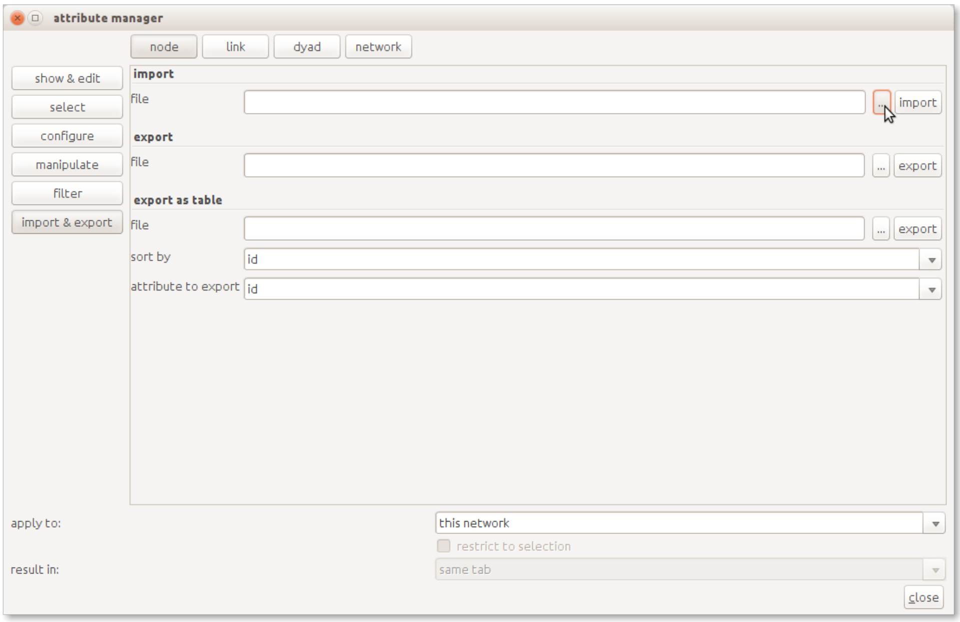


Figure 41: Click the '...'-button left of the import button.

- ☐ Choose the file `site_attributes.csv` which you can find in the `supplementary material` of this tutorial and click `ok`.

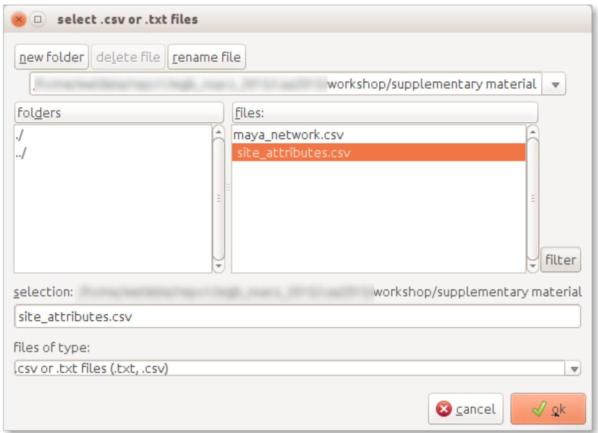


Figure 42: Choose the `site_attributes.csv` file contained in the supplementary material of this tutorial and click the `ok` button.

☐ In the `load options` dialog, have a look at the settings and make sure to configure everything as shown in figure 43. Click `ok` to finally import the node attributes.

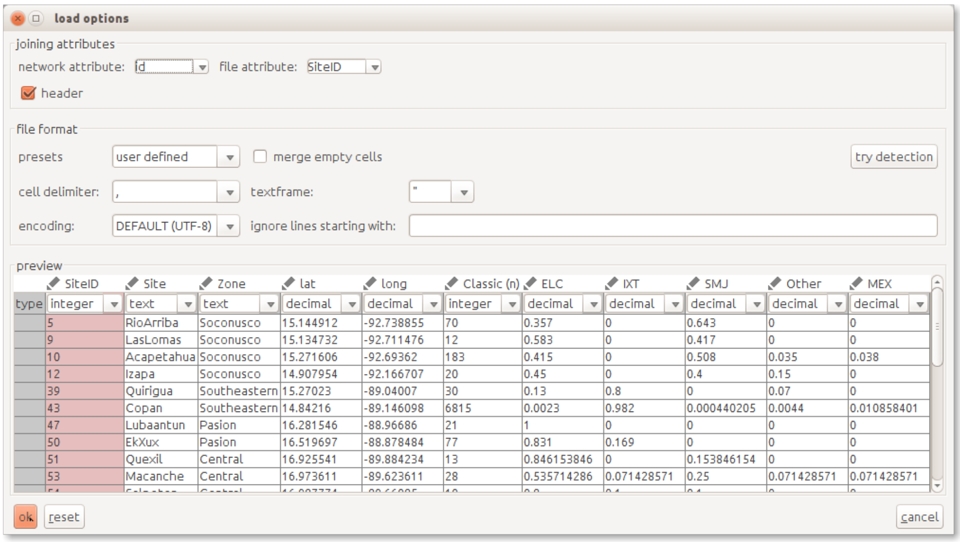


Figure 43: Make sure to use these settings in the `load options` dialog and click `ok`.

Result

☐ The `show & edit` view for `nodes` in the attribute manager now contains all the node attributes that have been imported.

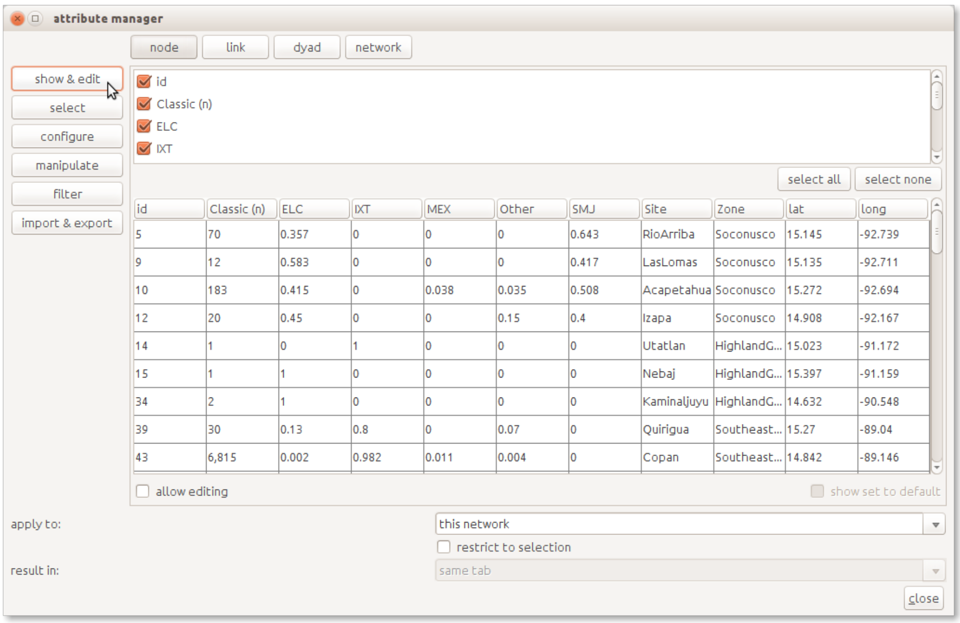


Figure 44: `Show & Edit` view for `nodes` now contains all the imported node attributes.

You accomplished the exercise :-)

You are now able to import additional attributes in visone .

Exercise 3: Filtering and Deleting Nodes

The goal of this exercise is to remove unreliable sites that have too little observations and therefore are not meaningful enough for the analysis.

Tasks

- ☐ Open the `Filter` view for `nodes` in the attribute manager and select the attribute `Classic (n)`.
- ☐ Activate the `smaller than` and `equal` checkboxes right to the attribute selection, and specify the value 10 as shown in figure 45.
- ☐ Click on select in the lower left of the attribute manager, which will now select all the sites that have less or equal 10 obsidian observations in the classic period.

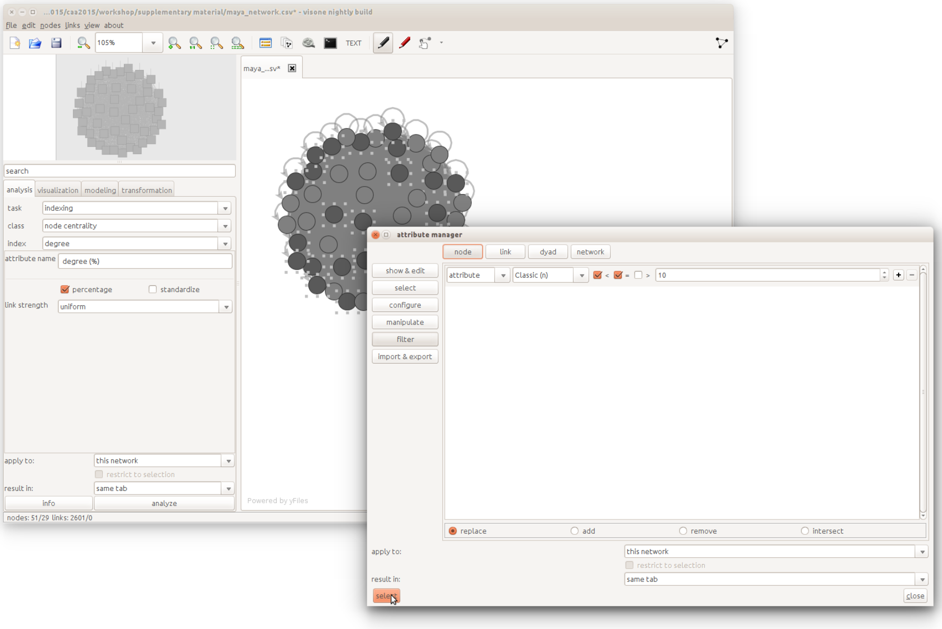


Figure 45: `Filter` view in the attribute manager

- ☐ Close the attribute manager.
- ☐ Delete the selected nodes by clicking on `nodes` in the visone menu bar, and then selecting `Delete nodes` from the menu as shown in figure 46.

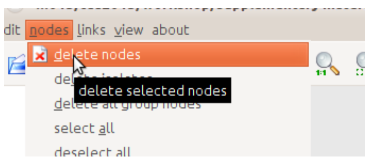


Figure 46: Delete the selected nodes via the menu bar.

Result

- ☐ The network panel shows your network without the nodes you just deleted. You see a `directed` network still containing `self-loops` from nodes to themselves.

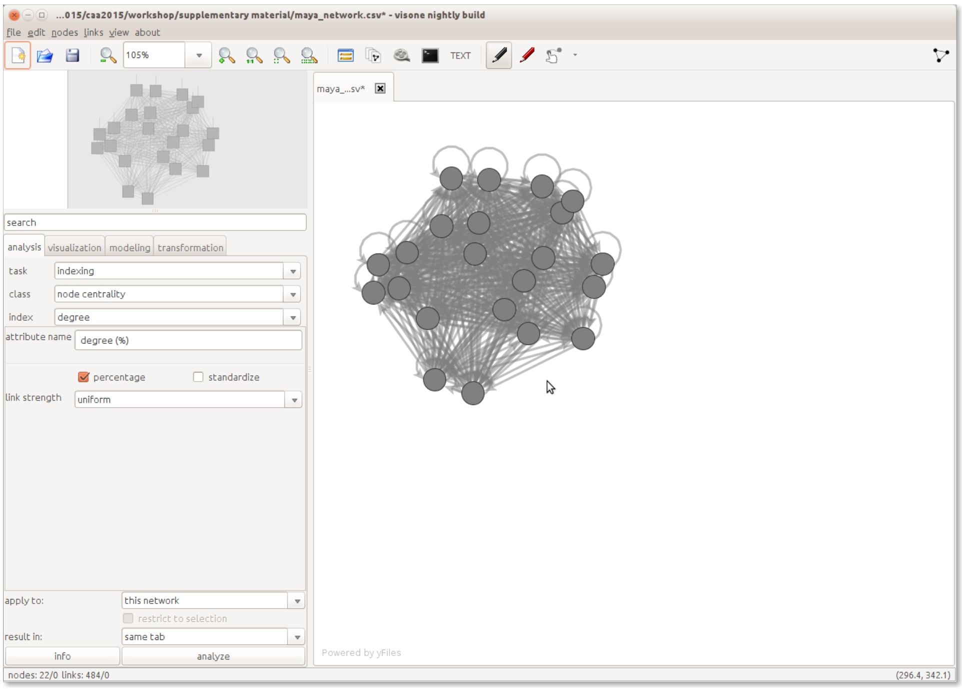


Figure 47: The network panel shows your network without the nodes you just deleted

You accomplished the exercise :-)

You are now able to filter and delete nodes.

Exercise 4: Configuring and Manipulating Link Attributes

While for a *similarity measure* (such as the Brainerd-Robinson Similarity) a high value represents a strong relationship between nodes, a high value in a *distance measure* indicates that two nodes are more distant, i.e. less similar. Since many of the algorithms in visone are based on distance, the goal of this exercise is to convert the Brainerd-Robinson *Similarity* into the Brainerd-Robinson *Distance* by substracting each link weight from the Brainerd-Robinson Similarity maximum value, which is `201` in our case. Thus, for any two nodes A and B in our network, we will transform the Brainerd-Robinson Similarity $BR^S_{A,B}$ into the Brainerd-Robinson Distance $BR^D_{A,B}$ via the following equation:

$$BR^D_{A,B} = 201 - BR^S_{A,B}$$

Remember that the Brainerd-Robinson Similarity $BR^S_{A,B}$ has so far been captured in the *csv value* link attribute.

Tasks

- ☐ Open the attribute manager again, and select the `Configure` view for **link**s.

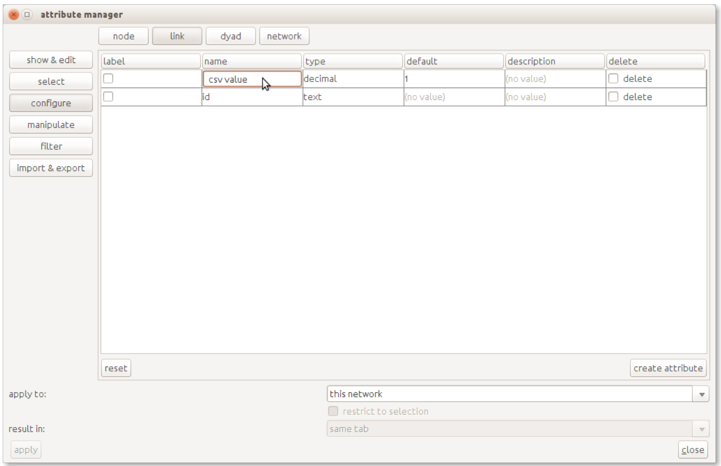


Figure 48: Open the Configure View in the attribute manager.

☐ Click into the textfield `csv value` and rename the attribute to `br_similarity`.

☐ Apply the rename operation by clicking on `apply` in the lower left of the attribute manager.

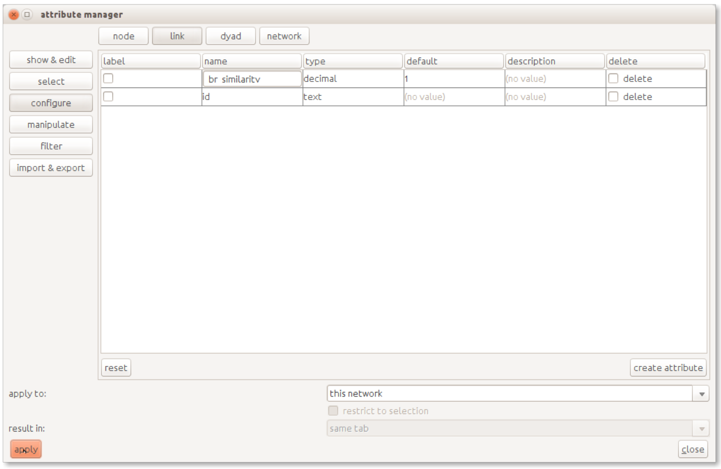


Figure 49: Rename the attribute `csv value` to `br_similarity`

☐ Open the `Manipulate` view for `links` in the attribute manager.

☐ Choose `manipulate values` and the `scale` operation as shown in figure 50.

☐ Select `br_similarity` as the attribute to scale by `-1` and write `br_distance` for the result attribute as shown in figure 50.

☐ Apply the operation by clicking on `apply` in the lower left of the attribute manager.

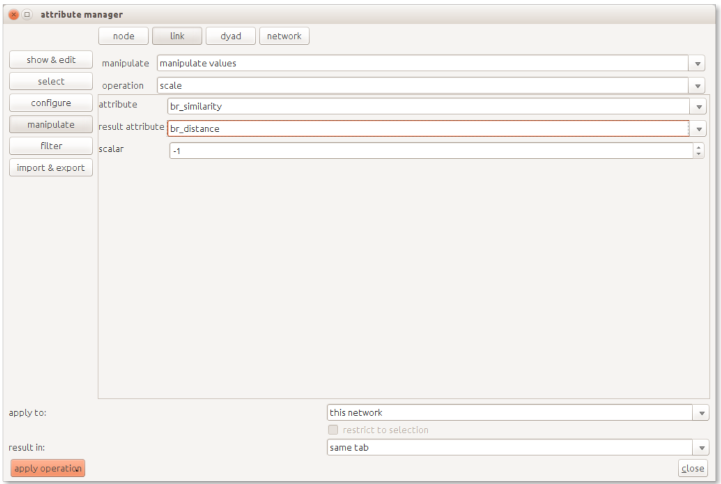


Figure 50: `Scale` the attribute `br_similarity` by `-1` and save the result to a new attribute `br_distance`

- ☐ Next, instead of `scale` choose `add` as the operation.
- ☐ Set both the attribute and the result attribute to `br_distance`, and write `201` for the offset. Compare your settings to figure 51.
- ☐ Apply the operation by clicking on `apply` in the lower left of the attribute manager.

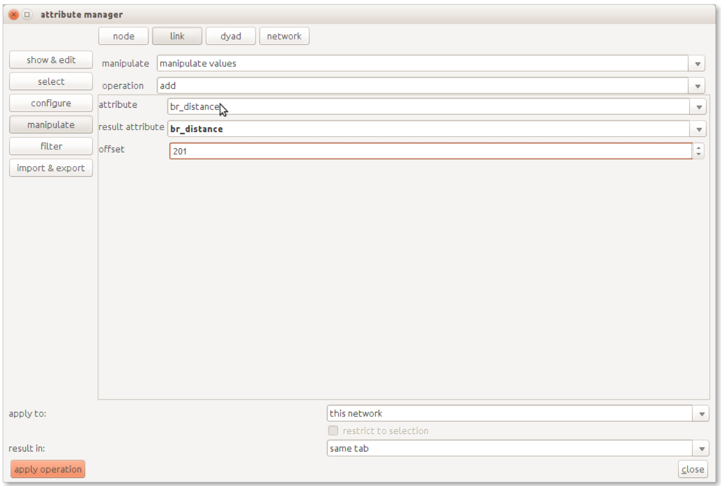
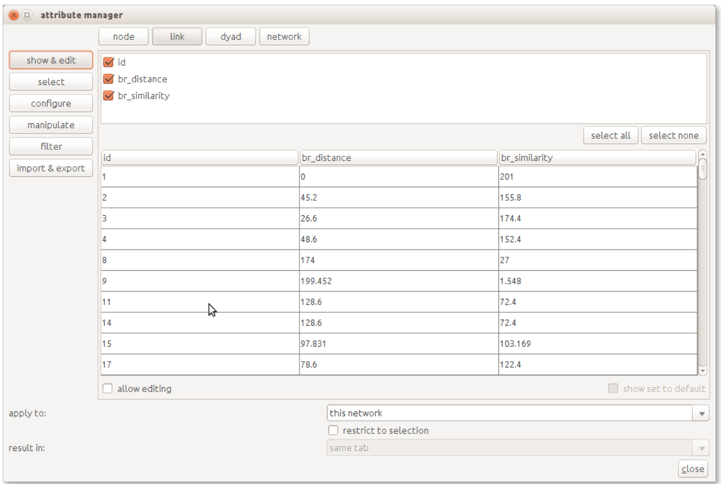


Figure 51: Add `201` to the new `br_distance` attribute

Result

- ☐ The `Show & Edit` view of the attribute manager shows the new `br_distance` attribute. You can cross check the values: for each link, `br_distance` plus `br_similarity` should add up to `201`.



id	br_distance	br_similarity
1	0	201
2	45.2	155.8
3	26.6	174.4
4	48.6	152.4
8	174	27
9	199.452	1.548
11	128.6	72.4
14	128.6	72.4
15	97.831	103.169
17	78.6	122.4

Figure 52: New `br_distance` attribute in the `Show & Edit` view of the attribute manager.

You accomplished the exercise :-)

You are now able to configure and manipulate attributes.

Exercise 5: Transforming Links

The goal of this exercise is to merge the directed links between two nodes into one and make the network undirected, since the similarity / distance of two nodes is the same no matter from which point of view.

Tasks

- ☐ First of all, we want to get rid of the self-loops. Open the `links` menu in the visone menu bar and click on `delete loops`.

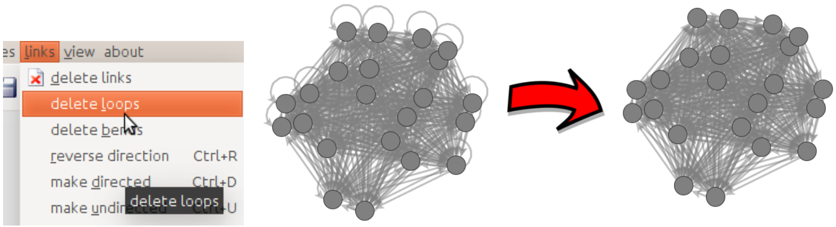


Figure 53: Delete the links from a node to itself.

- ☐ Next, switch to the `transformation tab` at the left of the network panel.
- ☐ Select `links` as the level and choose `merge` as the operation.
- ☐ Accept the default settings and click `transform`.

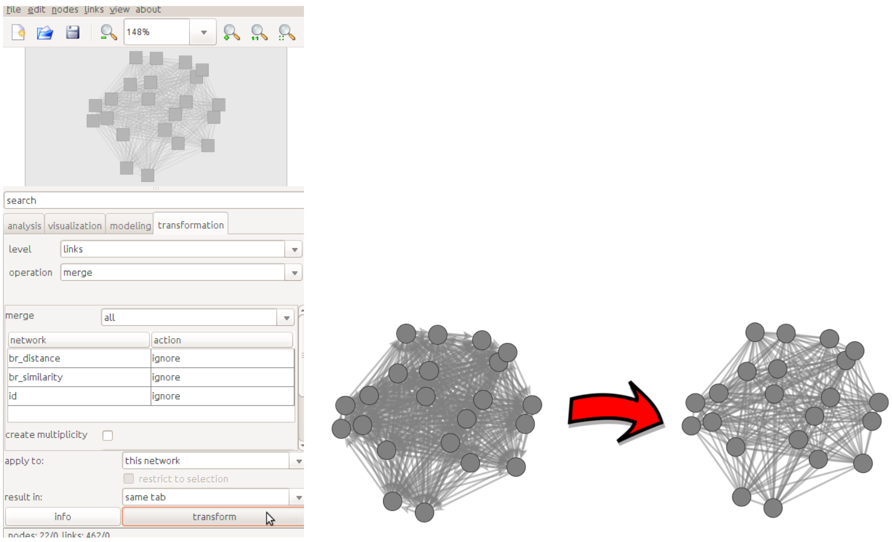


Figure 54: Merge the two directed links between any two nodes into one undirected link.

Result

- ☐ The network panel shows the undirected, loop-free network.

  You accomplished the exercise :-)

  You are now able to transform links.

Chapter 4

Network Analysis

4.1 Centralities

The general goal of a centrality measures is to provide a ranking of the nodes or edges, i.e. a node centrality is a function that assigns a value to each node, so that the nodes can be ordered according to this value. Centrality measures usually exploit the structure of the network and capture the *structural embedding* of nodes or edges within the network. Centrality measures are also used in network clustering, where the goal is to provide a grouping of nodes or edges.
To give basic example the *degree centrality* assigns to each node the number of links this node has with other nodes, e.g. if a node has 10 links the node has degree centrality 10.
We will briefly outline the definition of two further example centrality measures, after which you will compute them for your network through another exercise in visone .

Betweenness
The Betweenness of a node is defined as the number of times a node acts as a bridge on the shortest path between two other nodes. Thus, nodes with high betweenness can be regarded as important waypoints on the connections between other nodes, but also as bottlenecks in the network.

Closeness
The Closeness of a node is defined as the inverted sum of the node's shortest paths to all other nodes in the network. This means that nodes with a high closeness can reach all other nodes of the network in fewer steps than nodes with a low closeness.

Exercise 6: Computation of Betweenness and Closeness

The goal of this exercise is to compute node betweenness and closeness centrality for the sites of your Maya network.

Tasks

- ☐ Switch to the `analysis tab` of visone and choose the task `indexing` for `node centralities`. Select `betweenness` as the index to be computed in `percent`. Make sure you set `br_distance` as the link length as indicated in figure 55.

- ☐ Click the `analyze` button at the bottom of the analysis tab to run the computation.

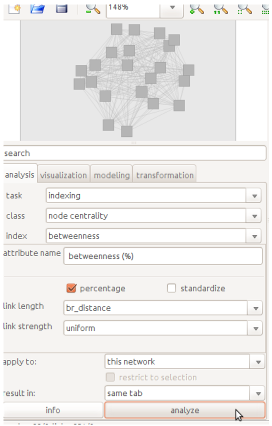


Figure 55: Compute the node `betweenness` centrality based on `br_distance` as link length in the `analysis tab` of visone .

- ☐ Next, again in the `analysis tab` of visone choose the index `closeness` to be computed in `percent`. Make sure you set `br_distance` as the link length as indicated in figure 56.

- ☐ Click the `analyze` button at the bottom of the analysis tab to run the computation.

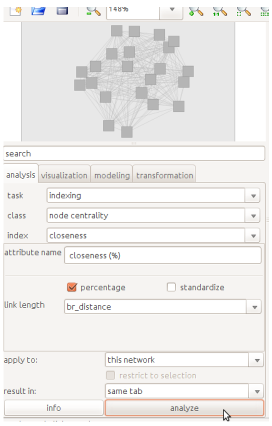


Figure 56: Compute the node `closeness` centrality based on `br_distance` as link length in the `analysis tab` of visone .

Result

□ The `Configure` view of the attribute manager for nodes shows the two computed attributes `betwenness (%)` and `closeness (%)` containing the centrality values.

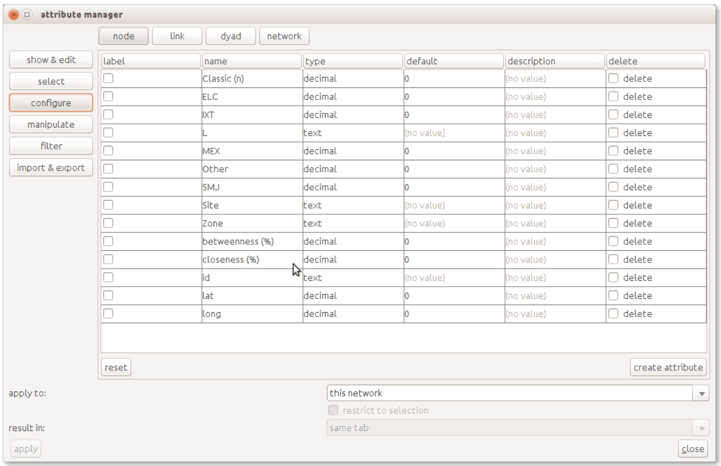


Figure 57: The `Configure` view of the attributes manager for nodes shows two new attributes `betwenness (%)` and `closeness (%)` containing the centrality values.

You accomplished the exercise :-)

You are now able to compute node centralities.

4.2 Clustering

`Clustering` in network analysis usually means to find a grouping of nodes based on the network structure, such that there are *many links within a group* (high cohesion), but only *few links between the groups* (low coupling). The methods typically assign each node to a cluster by internally

- optimizing a cluster quality function,
- determining the clusters according to specific properties, or
- revealing clusters by continuously editing the network (e.g. adding or removing links).

We will shortly outline **Girvan-Newman Clustering**, a clustering algorithm that makes use of the *edge betweenness* centrality, which is similar to the *node betweenness* you already got to know before. The algorithm can be described in 4 steps:

1. The *edge betweenness* of each edge in the network is determined, i.e. each edge is assigned with the number of times the edge is part of a shortest path between any two nodes.
2. The edge with highest betweenness is removed from the network.
3. For the remaining edges the edge-betweenness is recomputed.
4. Steps 2. and 3. are repeated until there are no more edges left in the network.

The result of this process is a top-down dendrogram, i.e. starting from one cluster containing all nodes, the clusters split while edges are being deleted until each node falls into a *singleton cluster* when no more edges are left. Thus Girvan-Newman Clustering is a *hierarchical clustering algorithm*.

Exercise 7: Girvan-Newman Clustering

The goal of this exercise is to produce a *hierarchical clustering* by applying the `Girvan-Newman Clustering (GNC)` algorithm in visone . Since the current implementation is not able to consider the weights of links (in our case the Brainerd-Robison similarity) but only the pure presence or absence of links, and our network is *complete*, we will first filter out some links with low similarity before running the GNC algorithm.

Tasks

- ☐ Open the `Filter` view for `links` in the attribute manager of visone .
- ☐ Set a filter for all links that have the attribute `br_similarity lower than 122`.
- ☐ Click on `select` in the lower left of the attribute manager to apply the selection.

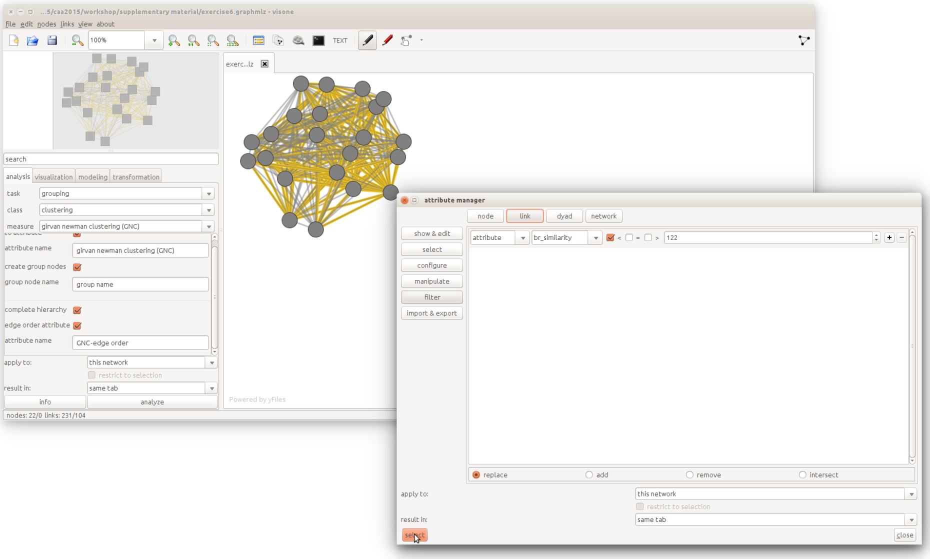


Figure 58: Select all links for which the `br_similarity is lower than 122`.

- ☐ Delete the selected links by clicking `Delete links` in the `links` menu of the visone menu bar.

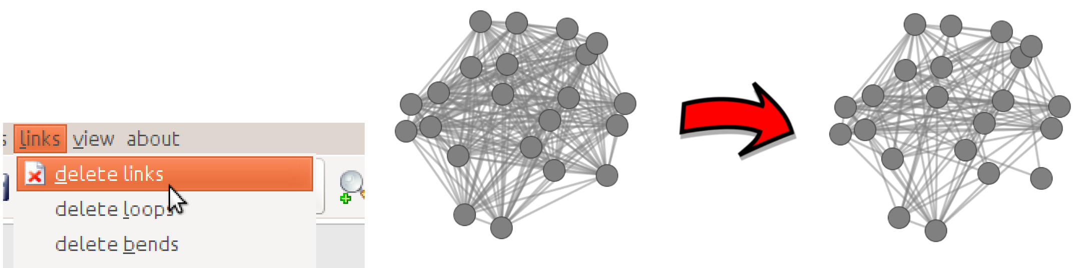


Figure 59: Delete the selected links from the network.

- ☐ Now switch to the `analysis tab` and choose the task `grouping` and class `clustering`.
- ☐ Select `girvan newman clustering (GNC)` as the measure and make sure to have the same settings as shown in figure 60.

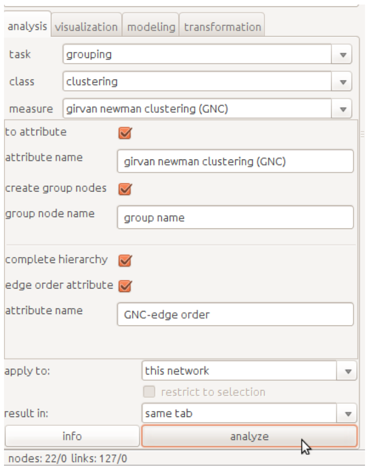


Figure 60: Running the Girvan-Newman Clustering algorithm.

- ☐ Click the `Quick Layout` button at the *top right of the* visone *window.*

Figure 61: Perform the quick layout of visone .

Result

- ☐ The network panel shows a layout of your network containing the hierarchical clusters (light blue) produced by the Girvan-Newman clustering algorithm.

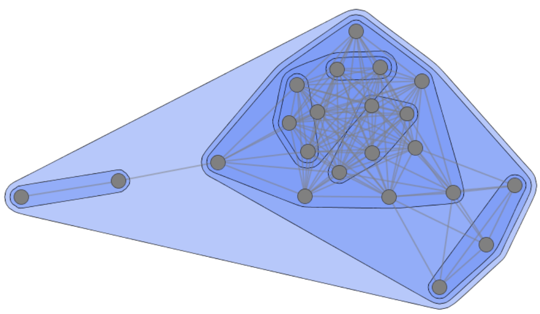


Figure 62: Layout of your network with hierarchical clusters.

You accomplished the exercise :-)

You are now able to run clustering algorithms.

Chapter 5

Network Visualization

So far you have seen how to load, edit, analyze and cluster networks. Another important block throughout the exploratory analysis of networks is a proper visualization of your network and its attributes, especially how to position nodes and how to map the attributes to visual variables of the visualization.

5.1 Mapping to Visual Variables

In general *Visual Variables* of a visualization are properties of the visualization elements which can encode attribute values of the data. In network visualization typical visual variables are

- color of nodes and links
- brightness of nodes and links
- size of nodes
- width of links
- node positions
- node or link labels
- shape of nodes
- line style of links
- ...

You will now see how to map attributes of your network to visual variables throughout another exercise.

Exercise 8: Mapping to Visual Variables

The goal of this exercise is to map the

- site names to the node labels,
- total number of sourced obsidian in a site to the size of the node,
- zone of a site to the color of the node, and
- Brainerd-Robison similarity to the color, width and z-Layer of links.

Tasks

- ☐ Switch to the `visualization tab` of visone and choose the category `mapping` for type `label` and property `node label`.
- ☐ Select `Site` as the attribute, and apply the mapping by clicking on the `visualize` button at the bottom right of the visualization tab.

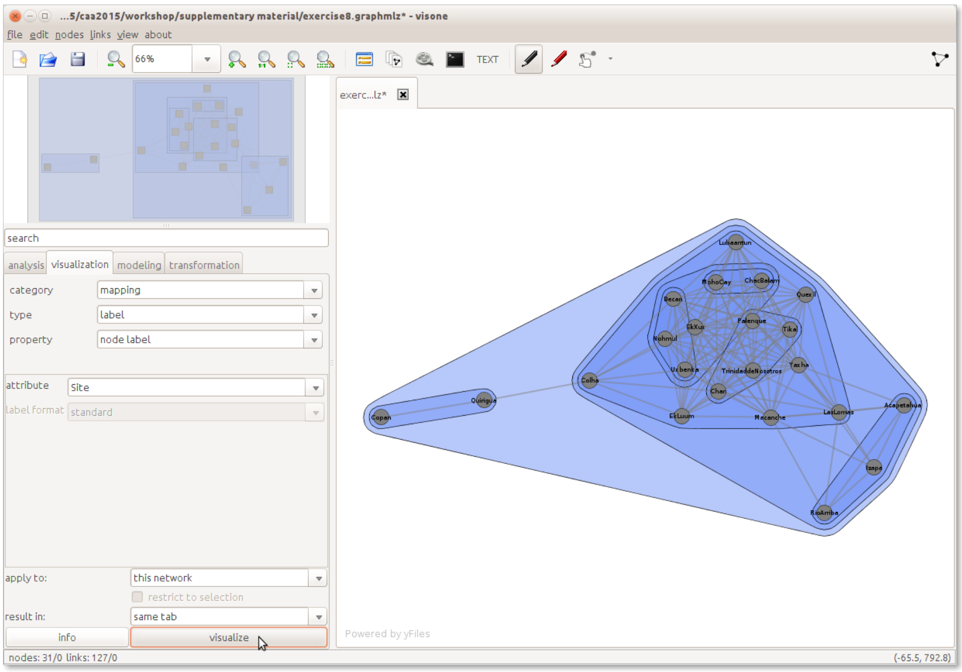


Figure 63: Mapping the `Site` name to the node label.

- ☐ Now we want to map the total number of sourced obsidian per site (stored in the node attribute *Classic (n)* of the data set) to the size of the nodes. Therefore set `size` as the type in the visualization tab, and choose `node area` as the property.
- ☐ Set the attribute to `Classic (n)`, and perform the mapping by again clicking on `visualize`.

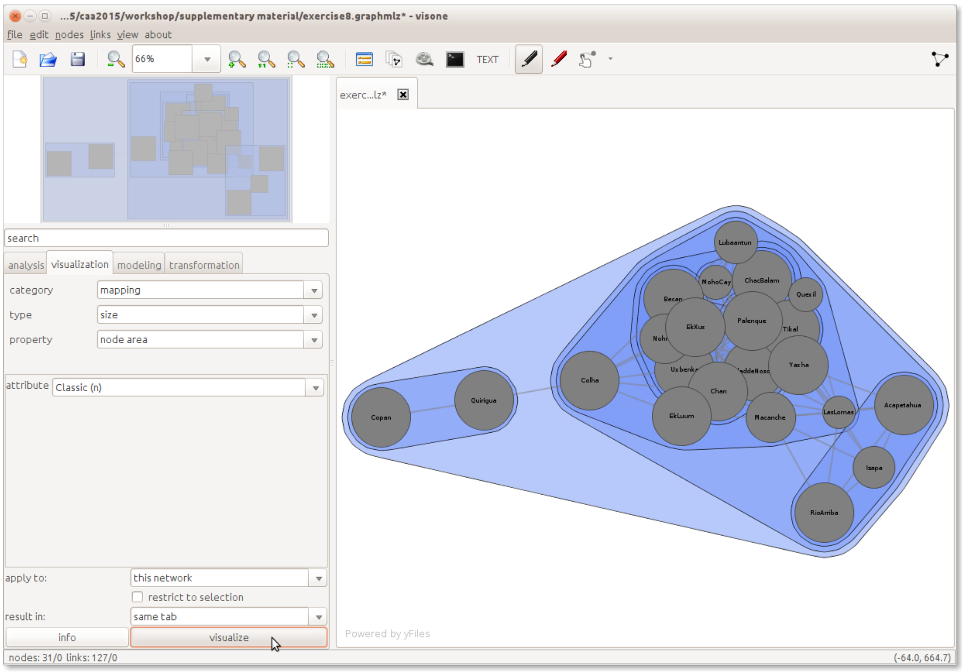


Figure 64: Mapping the `total amount of sourced obsidian per site` to the node size.

- ☐ In order to map the zone of the site to the color of the nodes choose the type `color` and property `node color` in the visualization tab.
- ☐ Set the attribute to `Zone` and select the method `color table`.
- ☐ Click on the color field of the first entry of the color table.

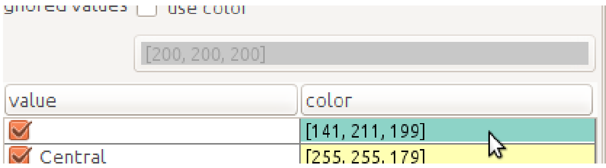


Figure 65: Open the color chooser dialog for the first entry of the color table.

☐ Switch to the tab `RGB` and set the `alpha channel` to `0`.

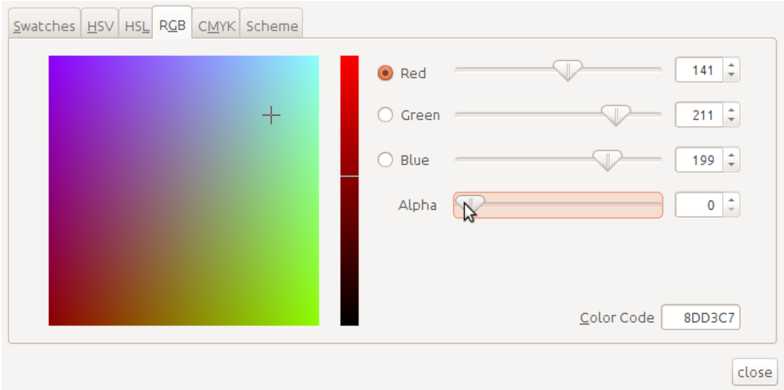


Figure 66: Set the `alpha channel` to 0 in the `RGB` tab of the color chooser dialog.

☐ Close the color chooser dialog, compare your settings to figure 67 and click `visualize` to apply the mapping.

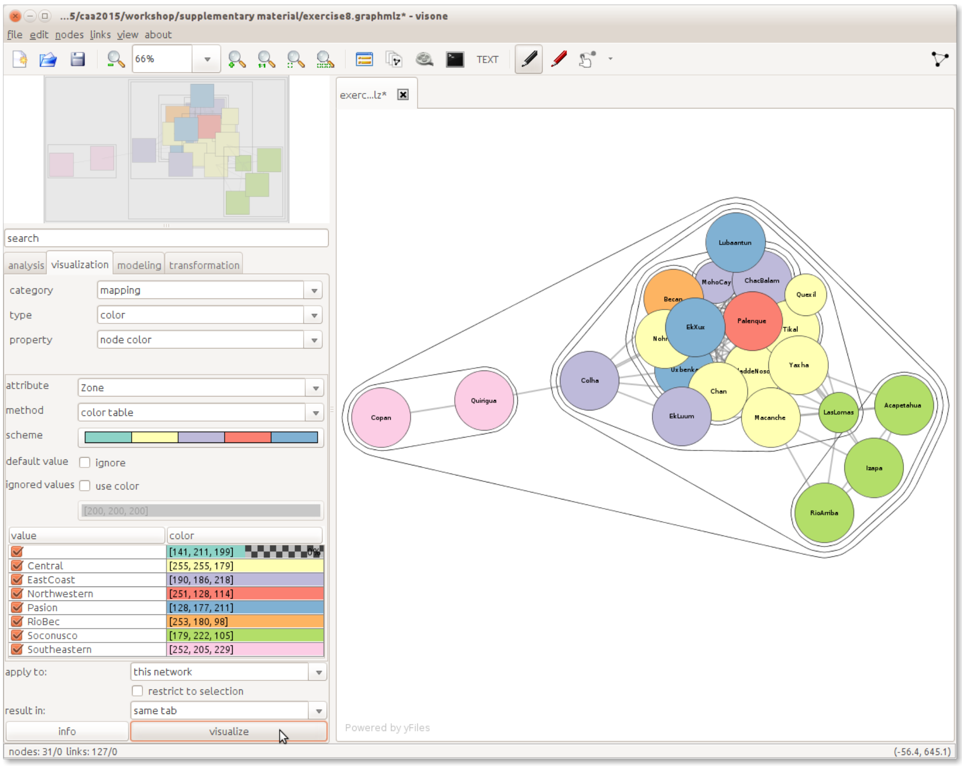


Figure 67: Apply the color mapping by clicking the visualize button.

- ☐ Now for the link color set the property to `link color` in the visualization tab.
- ☐ Select the attribute `br_similarity` and choose `interpolation` as the method.
- ☐ Click the `visualize` button to perform the mapping.

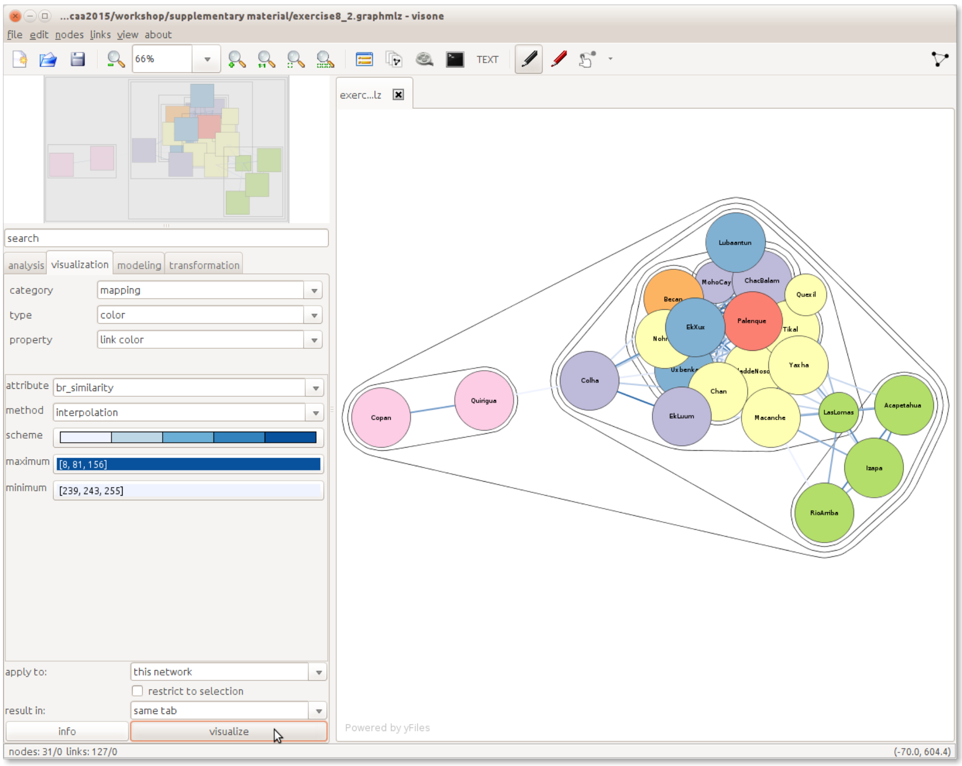


Figure 68: Mapping the Brainerd-Robinson Similarity to color of links: the more intense the color, the higher the similarity value.

- ☐ *Press and Hold* the `CTRL` key on your keyboard and *scroll down* with your mouse-wheel while the mouse pointer is within the network panel. This allows you to scale the node sizes keeping their relative size in order to make the links more visible.

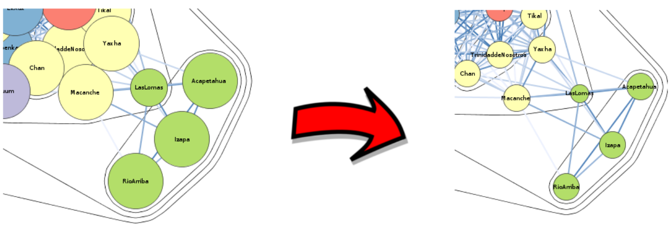


Figure 69: Scale down the size of the nodes by holding the `CTL` key on your keyboard and scrolling down with your mouse-wheel in the network panel.

- ☐ In a similar way you can also scale the size of the node labels (hold `CTRL + SHIFT` instead) or link widths (hold `ALT` instead).

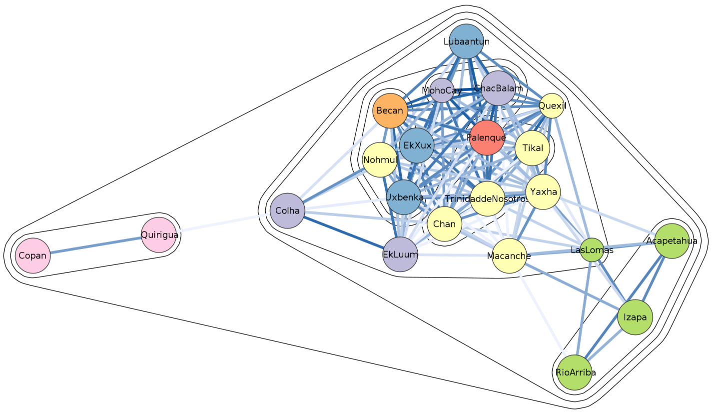


Figure 70: Network with scaled nodes sizes, scaled node label sizes and scaled link width.

- ☐ In order to scale the link width according to Brainerd-Robison similarity choose `size` as type and `link width` as property in the visualization tab.
- ☐ Set the attribute to `br_similarity` and apply the mapping by clicking on the visualize button.
- ☐ Next we want to bring links with higher similarity to the front, so choose `z-Layer` as type and `link z-Layer` as property in the visualization tab.
- ☐ Again set the attribute to `br_similarity` and apply the mapping by clicking on the visualize button.

Result

- ☐ The network panel shows your hierarchically clustered network with attribute mappings to visual variables and links representing higher similarity drawn on top of links with lower similarity.

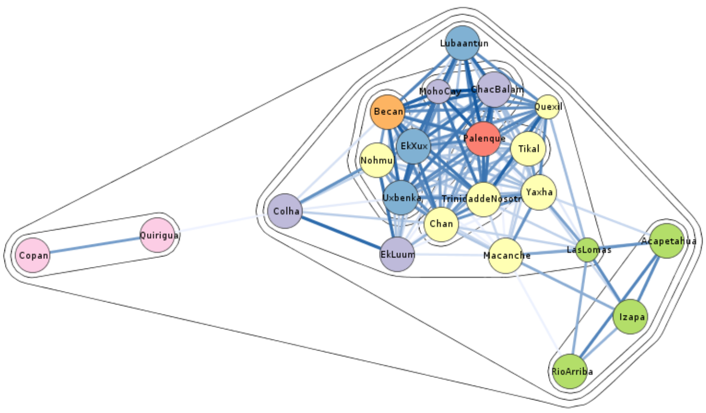


Figure 71: The network panel shows the clustered network with attribute mappings.

You accomplished the exercise :-)

You are now able to map attributes to your visualization.

5.2 Node Positioning

Positioning nodes is probably the most important aspect of network visualization, since this strongly affects the readability of the data. Bad node positioning will lead to meaningless edge crossings or node overlapping that will disturb the viewers perception. Also distances between nodes can serve as another visual variable that can encode for example the weight of links. You will explore two ways of node positioning through the final two exercises.

Exercise 9: Map Layout and Overlap Removal

The goal of this exercise is to create a map layout where we place sites according to their geographic position on a map. We will further apply a technique to reduce overlap where sites are too close and would hide each other in the visualization.

Tasks

- ☐ Since the map layout in visone can currently not cope with group nodes as produced before by the Girvan-Newman Clustering algorithm in Exercise 7, we need to remove them from the network.
- ☐ Click `delete all group nodes` in the `nodes` menu of the visone menu bar in order to do so.

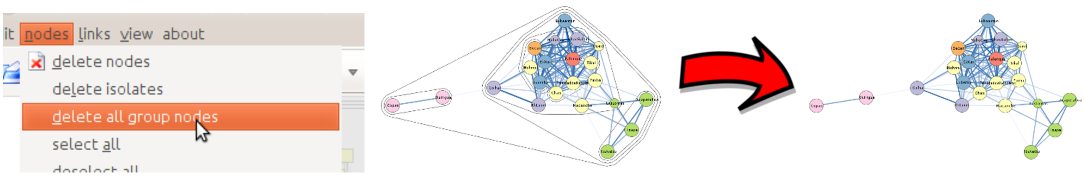


Figure 72: Remove all group nodes from the network.

- ☐ Switch to the `visualization tab` of visone and choose the category `mapping` for type `coordinates` and property `geopgraphic (mercator)`.
- ☐ Select `longitude` and `latitude` as provided in our Maya obsidian network and click `visualize` at the bottom of the visualization tab.

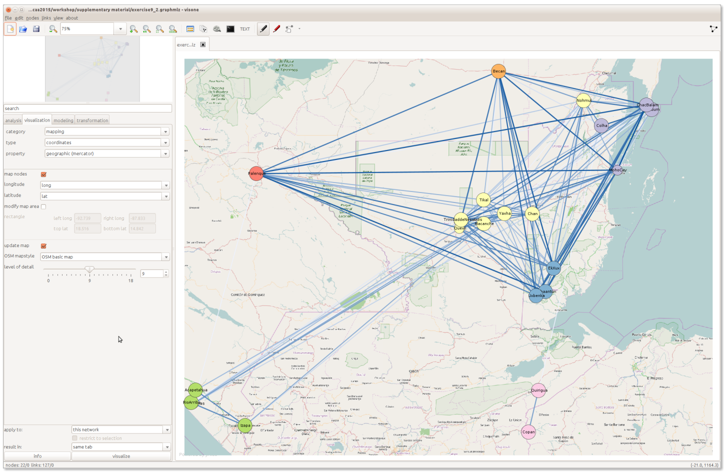


Figure 73: Apply the map layout to place the site nodes according to their geographic position and display a map view in the background.

☐ As you can see in figure 73 some sites are overlapping, so that it becomes hard to see their labels. Therefore we will apply an overlap removal technique that slightly moves the nodes to avoid overplotting.

☐ Set the category to `layout` in the `visualization` tab and choose `overlap removal` for `node layout`.

☐ Select to `append` the node labels as shown in figure 74 and click the `layout` button at the bottom of the tab in order to run the procedure.

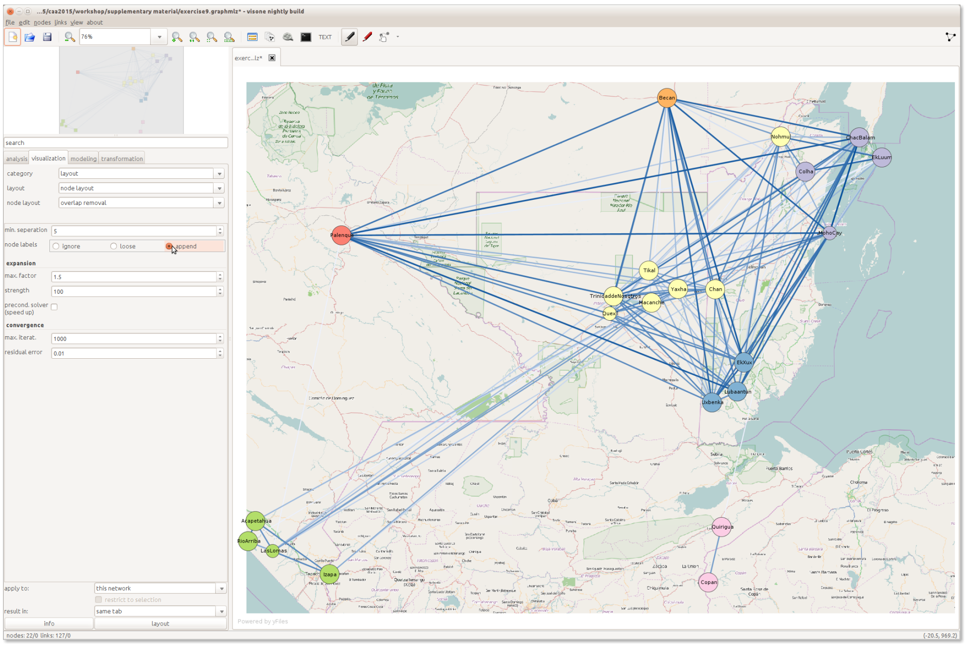


Figure 74: Run the overlap removal technique.

Result

☐ The network panel shows a map layout of your network where some sites are slightly shifted from their geographic location to avoid overlap.

You accomplished the exercise :-)

You are now able to create a map layout for networks with geographic node attributes.

Exercise 10: Stress Minimization

In the previous layout the node positions were predefined by their geographic location. This implicitly determined the length of the links as the *Euclidean distance* between sites in the map projection.
We will now do it the other way round: we specify the Brainerd-Robinson distance as the desired distance between nodes, and need to find a positioning of nodes to preserve the specified distances as good as possible.
To achieve this we apply *Stress Minimization*, a process where the node positions are optimized towards preserving the desired link lengths.

Tasks

- ☐ Clear the map in the background by clicking `clear background` in the `view` menu of the visone menu bar.

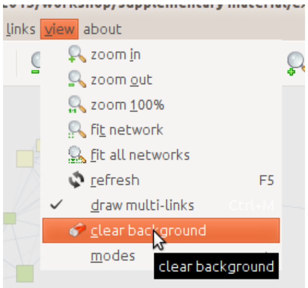


Figure 75: Clear the map in the background.

- ☐ In the visualization tab select the `node layout` as the `layout` category and choose the method `stress minimization`.
- ☐ Choose `attribute value` as link length scheme and select the `br_distance` attribute of your node data.
- ☐ Adjust the minimum and maximum link length to e.g. `100` and `300` as shown in figure 76.
- ☐ Click `layout` at the bottom to run the algorithm.

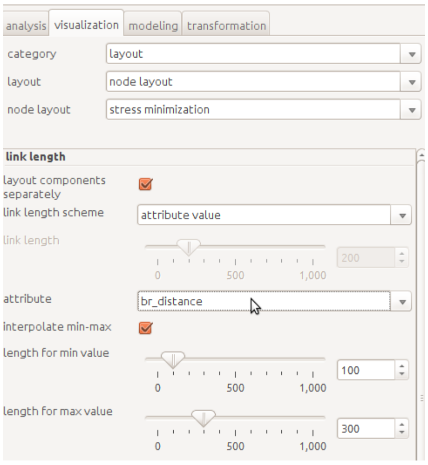


Figure 76: Running Stress Minimization in visone .

Result

- ☐ The network panel shows a layout of your network where nodes are positioned such that the distance between the nodes represents the Brainerd-Robinson distance.

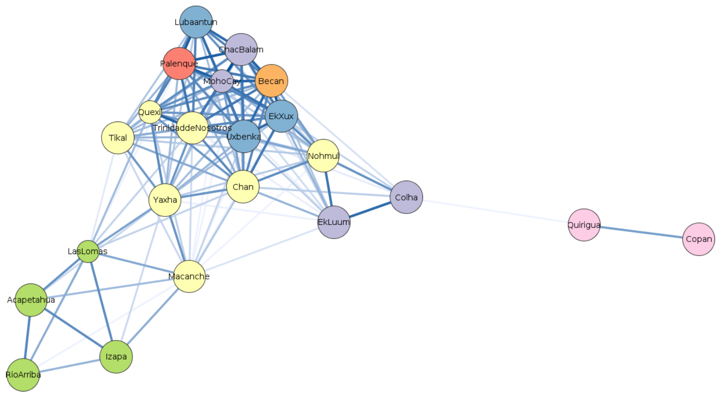


Figure 77: Network layout by Stress Minimization on Brainerd-Robinson distance.

You accomplished the exercise :-)

You are now able to create distance-based network layouts.

Chapter 6

Export

In exercise 1 you already exported a network to `.PNG` image via the `export` button in the visone menu bar. `.PNG` (and `.JPG`) are formats that save your network to a pixel raster. The resolution of your image has to be configured, and is then fixed. That means, when you zoom in, or when you include these images in higher resolution publications these images can become pixelated.

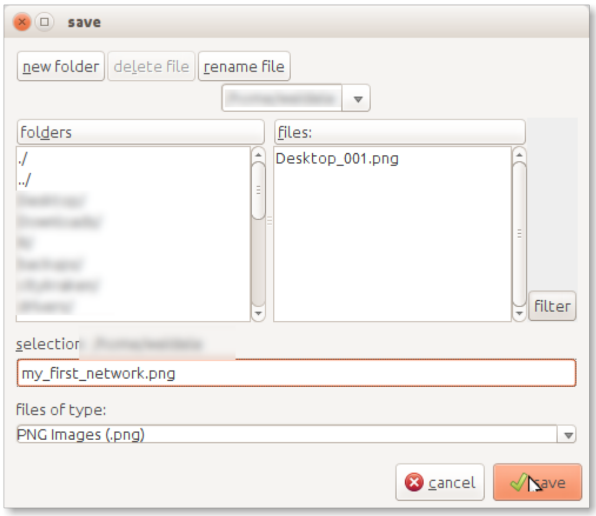


Figure 78: Exporting a network to raster graphics (here: `.PNG`).

Therefore we recommend exporting your images into vector graphic formats, such as *.PDF* or *.SVG*, such that the images keep a dynamic resolution and can scale with the final media (e.g. publication, higher-resolution displays, etc.).

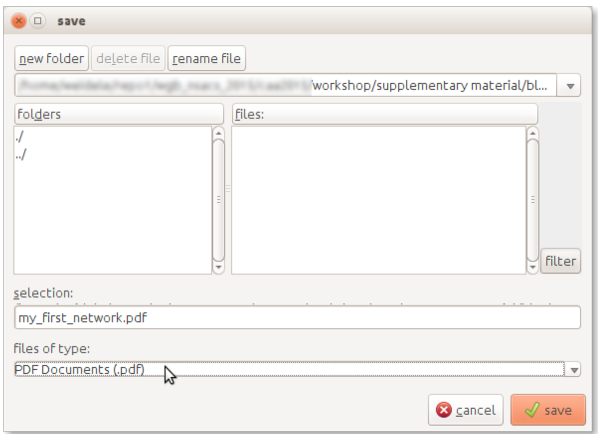


Figure 79: Exporting a network to vector graphics (here: `.PDF`).

Chapter 7

Discussion

Figure 80 shows the network layout where node positions are determined such that link lengths represent the Brainerd-Robison distance between sites (obtained by Stress Minimization). The black borders show the hierarchical clusters produced by Girvan-Newman clustering. While pink and most of the green sites nicely fall into separate clusters, the situation for the remaining sites is a bit more fuzzy. However, the layout suggests that the green site *Las Lomas* rather clusters with the yellow sites, thus being more similar to these and probably an important factor in bridging the two clusters. Other interesting sites are the blue *Lubaantun* and yellow *Nohmul* since they do not cluster with their geographic neighbors as reflected by the color. The geographic layout in Figure 81 suggests, that *Nohmul* might be falsly classified as yellow, since its geographic location is clearly closer to the violet sites. However, it remains an open question why *Lubaantun* is highly similar to 2 out of 4 violet sites, while not even being connected to the other 2. We could theorize that *Lubaantum* made use of sea routes to interact with allied violet coastal sites. A similar hypothesis could be made for the other two blue sites allying with *Nohmul, Ek Luum* and *Colha*. The overall impression is that there is high interaction among northern sites, while south east and south west don't show any similarities, possibly due to the sprawling mountain landscape between that did not allow for interaction.

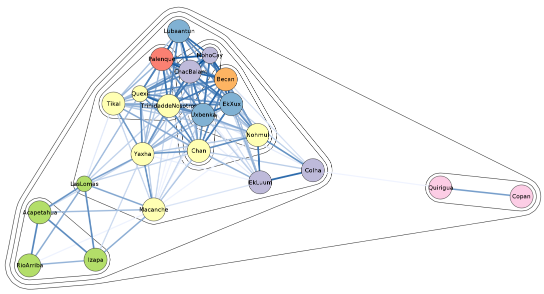


Figure 80: Node positioning by Stress Minimization. Clusters are computed with the Girvan-Newman clustering algorithm. Node distances represent Brainerd-Robinson distance between the respective sites. Node are colored by geographic zones. Node size encodes the absolute number of sourced obsidian. Link width and color intensity represent Brainerd-Robinson similarity.

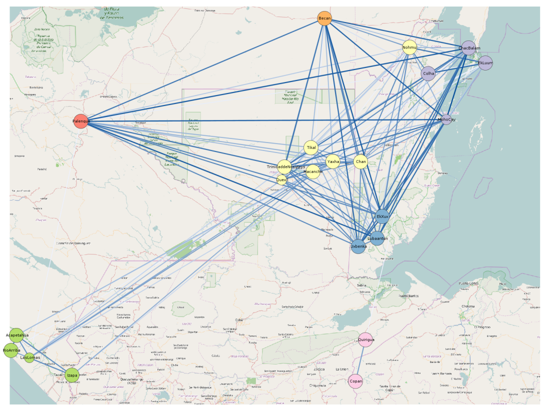


Figure 81: Node positioning by geographic coordinates of the respective sites. Node size encodes the absolute number of sourced obsidian. Link width and color intensity represent Brainerd-Robinson similarity.

Chapter 8

Other Software Resources

Pajek (http://pajek.imfm.si/doku.php)
Up to date set of analysis techniques, can handle large networks, good manual and supporting documentation, less easy to use than UCINET, includes some specific features not included in UCINET (Triad counts; nice matrix graphs; graphs automatically separating components).

UCINET (http://www.analytictech.com/ucinet/)
Up to date set of analysis techniques, good supporting documentation, great for converting to different network data formats. See handout network analysis practical 2.

Cytoscape (http://www.cytoscape.org/)
Poor documentation, user-friendly interface, easy analysis. See handout network analysis practical 1.

Gephi (http://www.gephi.org/)
Verry pretty visualisation, poor documentation, user-friendly interface, manual modification of layout algorithm settings.

Grass and **ArcGIS** (networkAnalyzer)
Have some network features. Can be used to produce visibility networks, least-cost paths, etc.

Processing (http://www.processing.org/)
visualization and animation, programming skills needed.

Mathematica (http://www.wolfram.com/mathematica/new-in-8/graph-and-network-analysis/index.html)
TODO

Matlab (http://www.levmuchnik.net/Content/Networks/ComplexNetworksPackage.html)
TODO

R (http://igraph.sourceforge.net/doc/R/00Index.html)
You can do basically anything you want if you can be bothered to code it, some great network analysis libraries (network, sna, Rnetworks, igraph, ergm, networkDynamic, Rsiena, statnet, tnet).

Sci2 (https://sci2.cns.iu.edu/)
TODO

Network workbench (http://nwb.cns.iu.edu/)
TODO

Excel and NodeXL (http://nodexl.codeplex.com/)
Everyone knows how Excel works, now you can use it to make networks.

Python (http://networkx.lanl.gov/, http://igraph.sourceforge.net/)
TODO

TRACER (http://mbuechler.e-humanities.net/tracer/)
A text re-use tracing software

Blocks
free separate program for blockmodeling - very good at it, does nothing else. Result pages somewhat cumbersome, but very good documentation

Chapter 9

Further Reading

Introductions

[1] Hennig, M. and Brandes, U. and Borgatti, S.P. and Pfeffer, J. and Mergel, I., 2012. Studying Social Networks: A Guide to Empirical Research. Campus Verlag.

[2] Scott, J., 2013. Social Network Analysis. SAGE Publications Ltd.

[3] Borgatti, Stephen P., Martin G. Everett, and Jeffrey C. Johnson, 2013. Analyzing social networks. SAGE Publications Ltd.

[4] Wasserman, S. and Faust, K., 1994. Social network analysis: Methods and applications. Cambridge university press.

[5] Brandes, U., Robins, G., McCranie, A., & Wasserman, S. 2013. What is network science? Network Science 1(01): p.1–15.

Networks in Archaeology

[6] Bentley, R.A. & Maschner, H.D.G., 2003. Complex systems and archaeology, Salt Lake City: University of Utah Press.

[7] Bentley, R.A. & Shennan, S.J. 2003. Cultural Transmission and Stochastic Network Growth. American Antiquity, 68(3), pp.459-485.

[8] Bevan, A., & Wilson, A. 2013. Models of settlement hierarchy based on partial evidence. Journal of Archaeological Science 40(5): p.2415–2427.

[9] Brughmans, T. 2013. Thinking through networks: A Review of Formal Network Methods in Archaeology. Journal of Archaeological Method and Theory.

[10] Brughmans, T. 2013. Networks of networks: a citation network analysis of the adoption, use and adaptation of formal network techniques in archaeology. Literary and Linguistic Computing, The Journal of Digital Scholarship in the Humanities.

[11] Brughmans, T. 2010. Connecting the dots: towards archaeological network analysis. Oxford Journal of Archaeology, 29(3), pp.277-303.

[12] Collar, A., Coward, F., Brughmans, T., & Mills, B., 2015. The Connected Past: critical and innovative approaches to networks in archaeology. A special issue of the Journal of Archaeological Method and Theory 22 (1).

[13] Collar, A., Coward, F., Brughmans, T., & Mills, B. J., 2015. Networks in Archaeology: Phenomena, Abstraction, Representation. Journal of Archaeological Method and Theory. DOI: 10.1007/s10816-014-9235-6

[14] Collar, A. 2007. Network Theory and Religious Innovation. Mediterranean Historical Review, 22(1),pp.149-162.

[15] Collar, A. 2013. Religious networks in the Roman Empire. The spread of new ideas. Cambridge: Cambridge University Press.

[16] Coward, F. 2010. Small worlds, material culture and ancient Near Eastern social networks. Proceedings of the British Academy, 158, pp.449-479.

[17] Coward, F. & Gamble, C., 2008. Big brains, small worlds: material culture and the evolution of the mind. Philosophical transactions of the Royal Society of London. Series B, Biological sciences, 363(1499), pp.1969-79.

[18] Graham, S. 2006a. Networks, Agent-Based Models and the Antonine Itineraries: Implications for Roman Archaeology. Journal of Mediterranean Archaeology, 19(1), pp.45-64.

[19] Graham, S. 2006b. EX FIGLINIS, the network dynamics of the Tiber valley brick industry in the hinterland of Rome, BAR international series 1486. Oxford: Archaeopress.

[20] Knappett, C. 2013. Network analysis in archaeology. New approaches to regional interaction. Oxford: Oxford University Press. [and contributions therein]

[21] Knappett, C. 2011. An archaeology of interaction: network perspectives on material culture and society. Oxford: Oxford University Press.

[22] Knappett, C., Evans, T. & Rivers, R., 2008. Modelling maritime interaction in the Aegean Bronze Age. Antiquity, 82(318), p.1009–1024.

[23] Knappett, C., Evans, T. & Rivers, R. 2011. The Theran eruption and Minoan palatial collapse: new interpretations gained from modelling the maritime network. Antiquity, 85(329), pp.1008-1023.

[24] Mills, B.J. et al. 2013. Transformation of social networks in the late pre-Hispanic US Southwest. Proceedings of the National Academy of Sciences of the United States of America: p.1–6.

[25] Schich, M. & Coscia, M., 2011. Exploring Co-Occurrence on a Meso and Global Level Using Network Analysis and Rule Mining. In Proceedings of the ninth workshop on mining and Learning with Graphs (MLG '11). San Diego: ACM.

[26] Sindbæk, S.M. 2007a. Networks and nodal points: the emergence of towns in Early Viking Age Scandinavia. Antiquity, 81(311), pp.119-132.

[27] Sindbæk, S.M. 2007b. The Small World of the Vikings : Networks in Early Medieval Communication and Exchange. Norwegian Archaeological Review, 40, pp.59-74.

Networks in History

[28] Bergs, A., 2005. Social Networks and Historical Sociolinguistics. Studies in Morphosyntactic Variation in the Paston Letters (1421-1503). (Topics in English Linguistics 51), Berlin/New York: Mouton De Gruyter.

[29] Düring, M., & Keyserlingk, L. 2011. Netzwerkanalyse in den Geschichtswissenschaften. Historische Netzwerkanalyse als Methode für die Erforschung von historischen Prozessen. In R. Schützeichel & S. Jordan (eds) Prozesse – Formen, Dynamiken, Erklärungen, Wiesbaden: VS Verlag für Sozialwissenschaften

[30] Düring, M., & Stark, M. 2011. Historical Network Analysis. In G. Barnett & J. G. Golson (eds) Encyclopedia of Social Networking, London: Sage

[31] Düring, M. 2011. Hilfe für Verfolgte während des Nationalsozialismus: Ein systematischer Vergleich von Egonetzwerken. In M. Schönhuth, M. Gamper, M. Kronenwett, & M. Stark (eds) Vom Papier zum Laptop? Perspektiven elektronischer Tools zur partizipativen Visualisierung und Analyse sozialer Netzwerke, Bielefeld

[32] Habermann, J. 2011. Verbündete Vasallen. Die Netzwerke von Grafen und Herren am Nordwestharz im Spannungsgefüge zwischen rivalisierenden Fürstgewalten (ca. 1250-1400). Norderstedt: Books on demand GmbH.

[33] Haggerty, J., & Haggerty, S. 2011. Temporal social network analysis for historians: a case study. Proceedings of the International Conference on Visualization Theory and Applications (IVAPP 2011), Algarve, Portugal, 5 - 7 March: p.207–217.

[34] Krempel, L., & Schnegg, M. 2005. About the Image : Diffusion Dynamics in an Historical Network. Structure and Dynamics 1(1).

[35] Lemercier, C., 2010. Formal network methods in history: why and how? In G. Fertig, ed. Social Networks, Political Institutions, and Rural Societies. Turnhout: Brepols publishers.

[36] Lemercier, C. 2005. Analyse de Réseaux et histoire de la famille: une rencontre encore à venir? Annales de démographie historique 109(1): p.7–31.

[37] Malkin, I. 2011. A small Greek world: networks in the Ancient Mediterranean. Oxford - New York: Oxford University Press.

[38] Malkin, I., Constantakopoulou, C., & Panagopoulou, K. 2009. Greek and Roman networks in the Mediterranean. Routledge.

[39] Padgett, J. & Ansell, C., 1993. Robust Action and the Rise of the Medici, 1400-1434. American Journal of Sociology, 98(6), pp.1259-1319.

[40] Padgett, J.F. & McLean, P.D., 2006. Organizational Invention and Elite Transformation: The Birth of Partnership Systems in Renaissance Florence. American Journal of Sociology, 6(5), pp.1463-1568.

[41] Pitts, F.R. 1965. A graph theoretic approach to historical geography. The Professional Geographer 17(5): p.15–20.

[42] Popovic, M. St. 2013. Networks of Border Zones : A Case Study on the Historical Region of Macedonia in the 14th Century AD. In Understanding different geographies, 227–241.

[43] Preiser-Kapeller, J. 2011. Calculating the Synod ? A network analysis of the synod and the episcopacy in the register of the patriarchate of Constantinople in the years 1379-1390. In C. Gastgeber, E.

[44] Mitsiou, & J. Preiser-Kapeller (eds) Das Patriarchatsregister von Konstantinopel. Eine zentrale Quelle zur Geschichte und Kirche im späten Byzanz, Vienna

[45] Ruffini, G.R., 2008. Social networks in Byzantine Egypt, Cambridge: Cambridge University Press.

[46] Schor, A.M. 2011. Theodoret's people, social networks and religious conflict in Late Roman Syria. Berkeley - Los Angeles - London: University of California Press.

[47] Wetherell, C. 1998. Historical Social Network Analysis. International Review of Social History 43: p.125–144.

References

[Adams and Culbert, 1977] Adams, R. E. and Culbert, T. (1977). The origins of civilization in the Maya lowlands. *The origins of Maya civilization*, pages 3–24.

[Brainerd, 1951] Brainerd, G. W. (1951). The place of chronological ordering in archaeological analysis. *American antiquity*, 16(4):301–313.

[Golitko et al., 2012] Golitko, M., Meierhoff, J., Feinman, G. M., and Williams, P. R. (2012). Complexities of collapse: the evidence of maya obsidian as revealed by social network graphical analysis. *Antiquity*, 86(332):507–523.

[Robinson, 1951] Robinson, W. S. (1951). A method for chronologically ordering archaeological deposits. *American antiquity*, 16(4):293–301.