

Appendix 4 – Update to Pajek 1.21

Since the publication of this book, Pajek software has been expanded. Additions and changes are listed in the history document (`history.htm`) that is shown when Pajek is being installed. Most of the changes are self-explanatory or very specialized. Concise explanations are provided in the manual (`PajekMan.pdf`) which is constantly being updated. The manual can be downloaded from the Pajek website. Some changes, however, are quite fundamental and expand the applications of Pajek considerably even to novice users. In this Appendix, we briefly describe these changes.

1 Interface

Networks
Partitions
Vectors
Permutations

Over all, the Pajek interface has not changed much. To facilitate working with more than one network, partition, vector, or permutation at the same time, Pajek's Main screen can now display two or three dropdown lists for these data objects (Figure 1). The number of dropdown lists visible can be changed by pressing the buttons labeled *Networks*, *Partitions*, *Vectors* or *Permutations*. It is no longer necessary to select the first and second network, partition, vector, or permutation in the *Nets*, *Partitions*, *Vectors* and *Permutations* menus when you want to execute a command. Just make sure that the right networks, partitions, etcetera, are selected in the right dropdown lists.

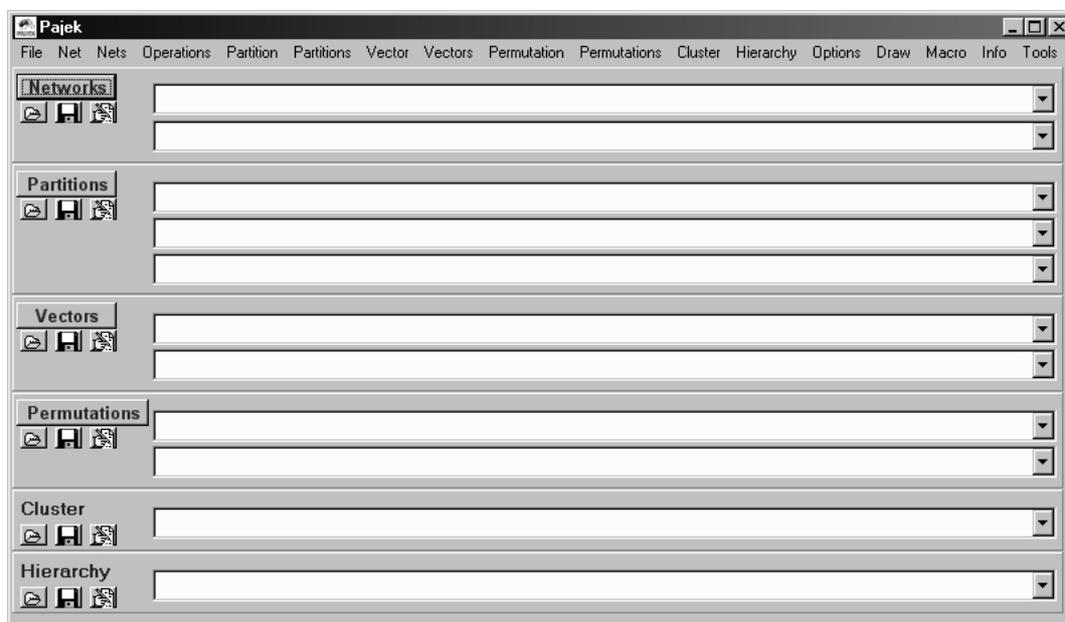


Figure 1 - Main screen with multiple dropdown lists.

File>Network
>Change Label

Another very useful extension is the option to rename Pajek objects (networks, partitions, and etcetera). During the analysis process, many networks, partitions and so on are created. Pajek assigns names to these objects that reflect where they come from. Most of them are merely intermediate results that are not very interesting. The ones that are important can now be renamed by the user, getting a name exactly showing the contents

or importance. The name or label of a network can be changed with the *File>Network>Change Label* command. Similarly, the other submenus in the *File* menu contain *Change Label* commands.

2 Multiple relations

The main addition to Pajek with respect to data structure is the concept of multiple relations. As of Pajek version 1.02, it is possible to assign a relation number to a set of lines. If we have different relations on the same set of vertices, e.g., we have data on imports of cereals between countries in addition to imports of miscellaneous manufactures of metal (Chapter 2), we can join them in one network, identifying one as relation number 1 and the other as relation number 2. Thus, we can draw and analyze the two relations in conjunction.

Let us first have a look at the structure of a multirelational network data file in Pajek. Figure 2 shows an excerpt from the data file *Imports_2relations.net* (*Worldtrade_2relations.paj*), which contains 80 countries and import relations on metal manufactures and cereals. As we can see, relation numbers and relation names or labels are added to the **Arcs* (or **Edges* or **Matrix*) statements. For each relation, a separate list of arcs or edges is created or a separate matrix.

File>Network>Edit

In the Draw screen, the relation number of a line can be manually changed by right clicking a vertex incident with the line. In the Editing Network screen that appears, right clicking a line allows the user to change the line value and the line's relation number. Note that this screen can also be accessed through the Edit Network button on the Main screen or *File>Network>Edit* command.

```
*Vertices      80
   1 "Algeria"          0.8281   0.7178   0.5000
   2 "Argentina"       0.3063   0.3063   0.5000
   3 "Australia"      0.4152   0.7755   0.5000
<...>
*Arcs :1 "manufactures of metal"
   78    25    556
   24    25  23023
   26    25  12714
<...>
*Arcs :2 "cereals"
   78    25    244
   28    25   1319
   46    25    184
<...>
```

Figure 2 - Excerpt from *Imports_2relations.net*.

[Draw]
Options>Colors>Edges
>Relation Number
Options>Colors>Arcs
>Relation Number
Options>Colors
>Relation Colors
Options>Lines>Draw
Lines>Relations

A relation number is a property of a line, quite like line values, line color, etcetera. It is best regarded as a discrete property of lines, assigning sets of lines to groups or categories, just like partitions assign vertices to classes. Therefore, it is natural to represent the relation number of a line by line color, which can be done in the Draw screen with the options *Options>Colors>Edges>Relation Number* and *Options>Colors>Arcs>Relation Number*. You can determine the color for each relation number in the *Options>Colors>Relation Colors* dialog screen. Finally, it is possible to draw the lines of just one relation by selecting a particular relation number in the *Options>Lines>Draw Lines>Relations* dialog box.

Info>Network>Multiple
Relations

A relation number is a property of lines, so information on the distribution of relation numbers over lines can be obtained from the *Info>Network>Multiple Relations* command, just like information on line values are available through the *Info>Network>Line Values* command.

Net>Transform
>Multiple Relations
>Extract Relation(s)
Nets>Union of Lines

Operations on multiple relations are collected in the *Net>Transform>Multiple Relations* submenu because they change the network. First of all, it is possible to extract one or more relations from a multiple relations network with the (*Net>Transform>Multiple Relations>Extract Relation(s)*) command. This will produce a new network for each of the selected relation numbers, preserving the relation number. If you want to obtain a network containing two or more relations, you have to join two networks at a time with the *Nets>Union of Lines* command.

Net>Transform
>Multiple Relations
>Change Relation
Number

In addition, the *Net>Transform>Multiple Relations* submenu has commands for recoding relation numbers and changing relation labels (*Net>Transform>Multiple Relations> Change Relation Number*). Because it is possible to select more than one original relation number at a time, this command also allows the fusing of relations. For instance, selecting relations 1 and 2 in the first dialog box issues by this command, will recode both relation numbers into one new number, thus merging the two relations.

Net>Transform
>Multiple Relations
>Line Values
-> Relation Numbers

Finally, it is possible to translate relation numbers into line values and vice versa (*Net>Transform>Multiple Relations> Line Values->Relation Numbers* and *Net>Transform>Multiple Relations> Relation Numbers->Line Values*). Basically, this is changing a continuous property of vertices expressed by line values, to a discrete property represented by relation number or the other way around. It is especially useful when the original network data is stored in a format that can represent line values but not multiple relations.

Net>Transform
>Multiple Relations
>Relation Numbers
->Line Values

Nets>Fragment (First in
Second)>Options

Net>Transform
>Remove
>Multiple Lines

Operations
>Shrink Network

There are no specific calculations for multirelational networks. In general, multiple relations are treated as multiple lines, e.g., in degree calculations. Sometimes, it is possible to pay special attention to relation numbers, e.g., in the options to the *Find Fragments* command. However, some operations do take into account the relation numbers of lines. For instance, removing multiple lines from a network (*Net>Transform>Remove>Multiple Lines*), only multiple parallel lines belonging to the same relation are replaced by a single line. It is important to note that the resulting network may still contain multiple lines after this command, that is, parallel lines with different relation numbers. Shrinking a multirelational network (*Operations>Shrink Network*) also respects relation numbers, so lines with different relation numbers are not shrunk into a single line.

Nets>Intersection
Nets>Difference

Because a multirelational network represents different relations on the same set of vertices, one is usually interested in the degree of overlap between relations. In Pajek, the *Intersection* and *Difference* commands in the *Nets* menu select the lines that are present in respectively both networks or just one of the two networks that are being compared. Splitting the multirelational network into networks with single relations, as discussed above, yields the networks that can be compared. It is important to note that the *Intersection* and *Difference* commands take into account relation numbers, that is, they only select lines as appearing in both networks if these lines have the same relation number.

Nets>Cross-Intersection
Nets>Cross-Difference

This may be interesting when comparing two multirelational networks, it is not to the point in a comparison of two single relation networks derived from a multirelational network. In the latter case, each derived single relation network contains the original relation number. Comparing networks with different relation numbers, the *Intersection* command does not yield any intersecting lines; an empty network. Similarly, the *Difference* command selects a line from the first network if there is not a line with the same relation number in the second network. In the case of two networks, each containing another relation number, *Difference* will include all lines from the first network and it will yield a network that is identical to the first network. If you want to disregard relation numbers, which is what you should do in this case, you should use the *Cross-Intersection* and *Cross-Difference* commands.

3 *Exporting to statistical software*

Exporting data from Pajek to statistical software is another functionality not covered in the first edition of the book. It is a very flexible procedure allowing for easy statistical analysis of data or results obtained in Pajek. The basic principle is that it is possible to send networks as vertices by vertices matrices and partitions or vectors as vertices by variables matrices to statistical software packages, presently R and SPSS. A Pajek command exports selected data objects in a format that can be read by the statistics package, opens the package, and makes it read the data objects so the user may start statistical analyses right away. Afterwards, the user may apply any statistical analysis to the data available in the software package. Because the packages offer a comprehensive set of statistical analyses, there is no need to implement them in Pajek now that it is easy to export from Pajek to these packages.

Tools>Program R
>Locate R
Tools>Program SPSS
>Locate SPSS

First of all, the user needs to install the statistical software on his or her computer and identify the location of the software to Pajek. The latter is accomplished by the *Tools>R>Locate R* and *Tools>SPSS>Locate SPSS* commands. The user must look for the folder containing the appropriate program file (*rgui.exe* or *rterm.exe* for R and *runsyntx.exe* for SPSS), which are usually located in the Program Files directory of the computer. Any time a new version of Pajek is installed, the statistical software must be located again.

Tools>Program R
>Send to R
Tools>Program SPSS
>Send to SPSS

Once this is done, networks and vectors can be exported with the *Send to R* or *Send to SPSS* commands. Either the network, partition, or vector currently selected in the first dropdown list is exported, or all networks, partitions, or vectors listed in the currently active cluster are exported, or all networks, partitions, or/and vectors. A cluster is used to

select particular vectors or partition in the following way: create an empty cluster (*Cluster>Create Empty Cluster*) and manually edit it, adding ‘vertices’ with the sequential numbers of the partitions or vectors that you want to export, e.g., ‘vertices’ 1 and 4 if you want to export partitions 1 and 4. Finally, select the *Partitions selected by Cluster* or *Vectors selected by Cluster* command in the appropriate *Send to* submenu. Partitions can be exported to SPSS, not to R. SPSS accepts only one network and it cannot combine a network with partitions or vectors. Finally, note that SPSS only accepts partitions and vectors of the same length (dimension).

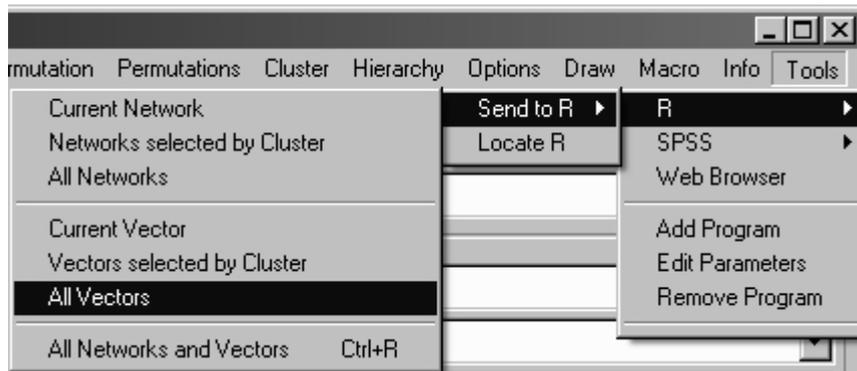


Figure 3 - Sending data objects to R from Pajek.

After exporting data objects to R and automatically opening the graphical user interface to R (*rgui.exe*), R shows a window like the one depicted in Figure 4. In this case, two vectors were exported from Pajek to R, containing normalized input degree and betweenness centrality in the multirelational world trade network. These two vectors may now be manipulated and analyzed with commands in R. In addition, some functions are displayed that can be used to save results in R in a format that can be read by Pajek: a vector or variable can be saved as a *.vec* file, so it can be read by Pajek. Both 1-mode and 2-mode matrices can be saved in Pajek network file format for further processing or display in Pajek. Finally, two commands are provided for reading additional vectors or networks from Pajek files. Note that all of these commands are available in R only if R is opened by the *Send to R* command in Pajek.

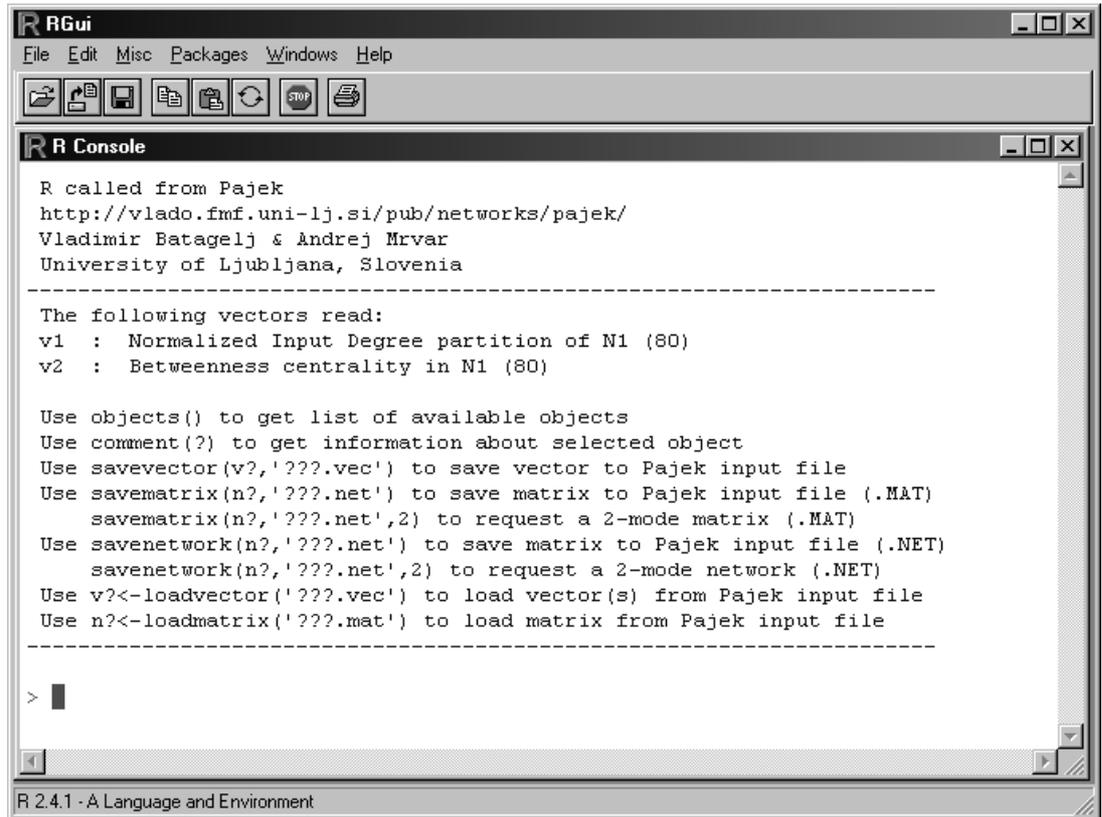


Figure 4 - R interface with special Pajek functions.

Figure 5 shows the SPSS datamatrix containing the countries of the multirelational world trade data set and their input degree (column C1), normalized input degree (column V1) and betweenness centrality (column V2). Exporting partitions or vectors to SPSS, the vertices are the cases and each partition and vector is represented by a variable. Vertex labels are automatically stored as an alphanumerical (string) variable provided that the number of vertices in the active network matches the dimension of the exported partitions and vectors. The vertex number is also stored as a variable. The name of the partition or vector as it appears in the dropdown list in Pajek is stored as the variable label. In SPSS, there is no option or function to save data in Pajek format, so it is not possible to easily send data from SPSS to Pajek.

1 : V1 0,278481012658228

	vertxlab	vertexID	C1	V1	V2	var
1	Algeria	1	22	,2785	,0000	
2	Argentina	2	24	,3038	,0197	
3	Australia	3	34	,4304	,0289	
4	Austria	4	25	,3165	,0097	
5	Barbados	5	18	,2278	,0009	
6	Bangladesh	6	16	,2025	,0000	
7	Belgium /Lux.	7	22	,2785	,0055	
8	Belize	8	17	,2152	,0000	

SPSS Processor is ready

Figure 5 - Degree and betweenness of countries in multirelational world trade data.

4 *Layout extras*

*Layout>Circular
>using Partition*

Pajek is known for its capabilities for visualizing networks. We will present some additions to Pajek that are quite powerful but not directly evident. First, the *Layout>Circular* submenu has the new command *using Partition* when a network and partition are drawn. This command creates a circle for each class of the partition, which provides a layout suitable for exploring ties within and among classes. The size of the circle is related to the number of vertices in the class and the circle's center is located at the average of the coordinates of the vertices belonging to the class.

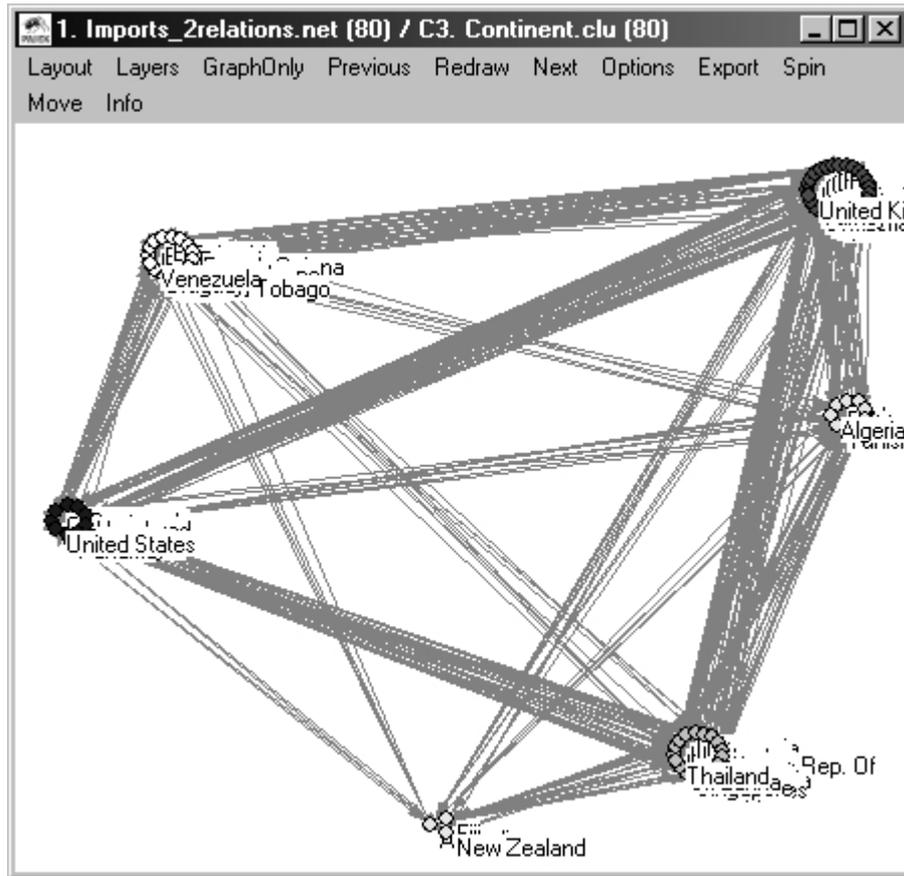


Figure 6 - World trade countries arranged by continent.

Options>Transform
>Fit Area
>max(x),max(y),max(z)

Figure 6 shows the layout obtained for the multirelational world trade data. Usually, circles are too small to see the pattern of ties within a class of vertices. Circles may also overlap. It is fairly easy, however, to move the circles by left clicking near the circle and dragging it away if the network and partition are drawn. One may increase the size of circles in two steps: (1) move the circles towards the center creating an empty border, and (2) issuing the *Options>Transform>Fit Area>max(x),max(y),max(z)* command. Figure 7 shows the results.

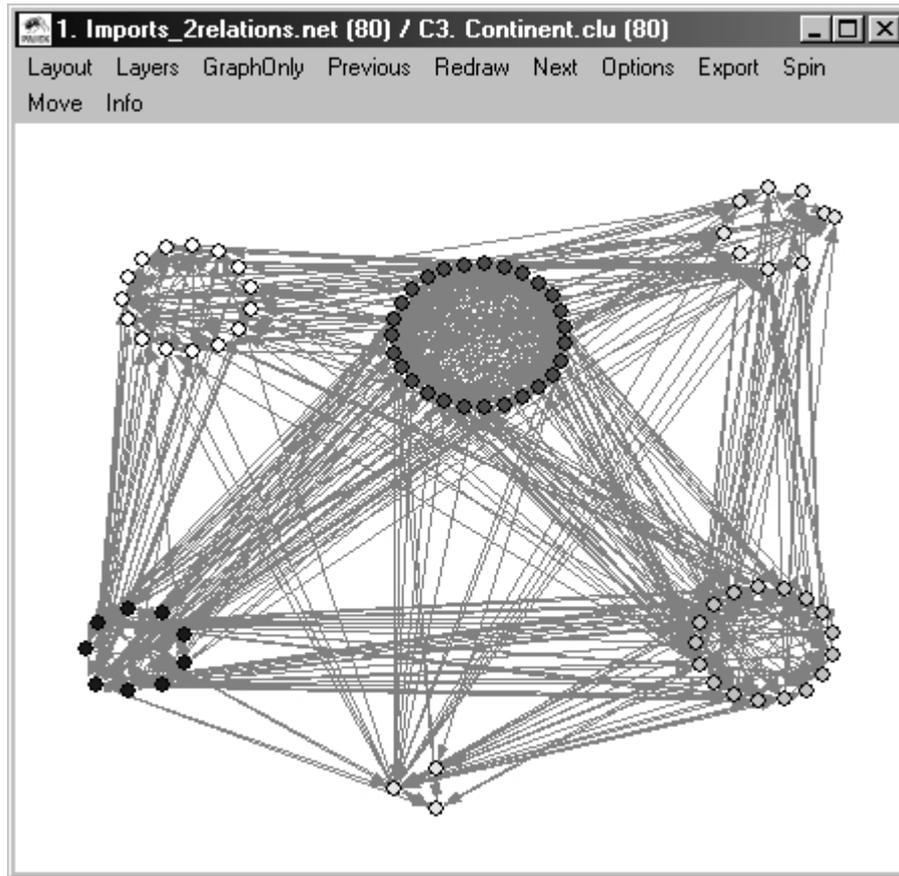


Figure 7 - World trade countries arranged by continent (enlarged circles).

Options>Lines
>Different Widths

Another very helpful command is sorting lines according to their line values before drawing them with line width corresponding to line value. Particularly when line width is representing line value, it may be helpful to have the thinner lines on top if one wants to be able to see them or, alternatively, have the widest lines on top if one wants to stress the largest line values in the visualization. In Figure 8, the first approach is used. It shows the import relations of metal manufactures (black arcs) and cereals (grey arcs) among continents. Because both types of import are usually present between two continents, the import with lower total value, which is represented by a thinner arc, must be drawn on top to be visible. The export of metal manufactures from North America to Asia, for instance, clearly involves a much lower amount of money than the export of cereals. The former is visible because it is drawn on top.

Net>Transform
>Sort Lines
>Line Values
>Ascending,
Descending

Lines can be sorted with the commands in the *Net>Transform>Sort Lines>Line Values* submenu. If you select *Ascending*, lines with the lowest values will be put on top of the network data list and lines with the highest values will be placed at the end. Because Pajek draws lines in the order in which they appear in the network list, lines with lower line values will be drawn underneath lines with higher line values in the Draw screen. Exports in (Encapsulated) PostScript (EPS/PS) and Scalable Vector Graphics (SVG) format also draw lines on top of each other in the order in which the lines are specified in the data file. In the network images, the lines with highest line values will come out on top. If you want the opposite results, as in our example (Figure 8), you must select the *Descending* command in the submenu.

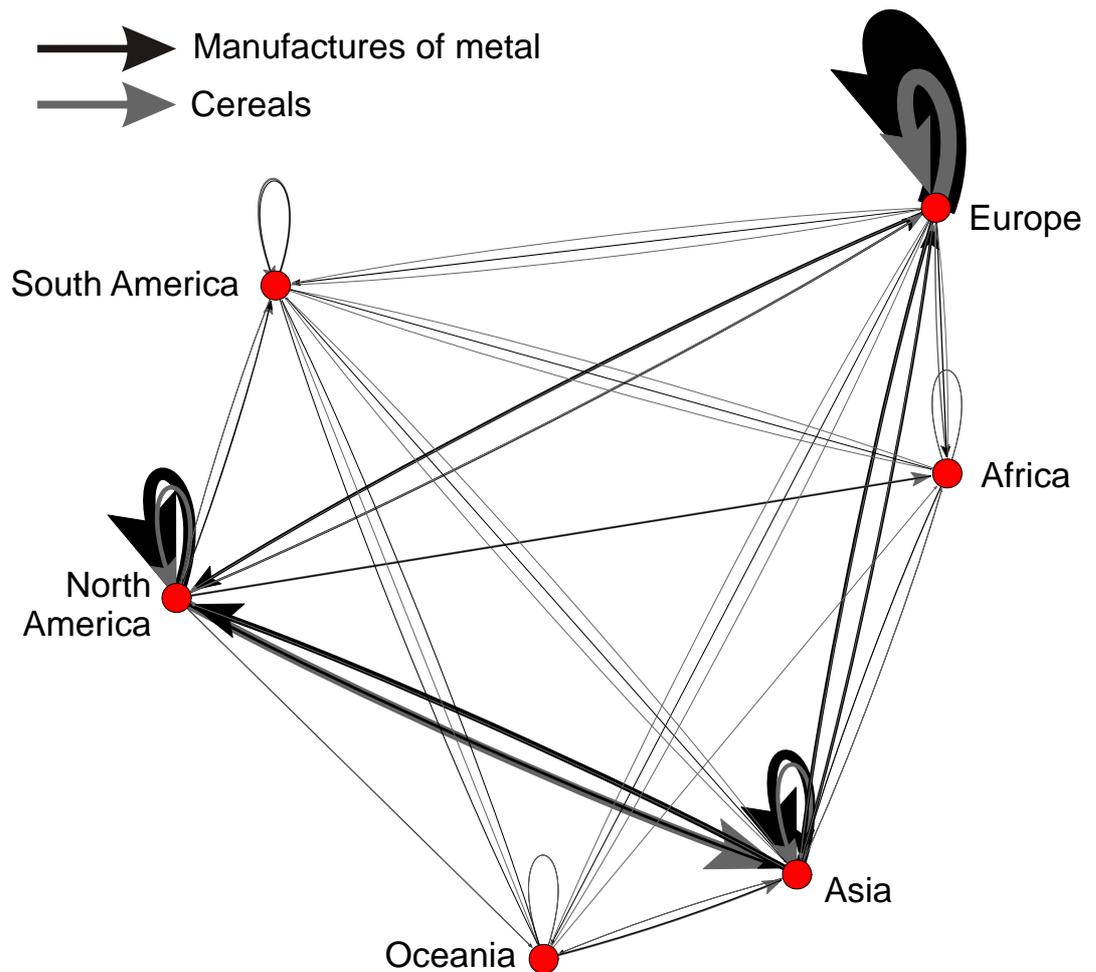


Figure 8 - Imports of metal manufactures and cereals among continents.

Draw>Draw-2Vectors

It has always been possible to express vector values by the size of vertices in the Draw screen. In addition, it is now possible to have vertex sizes reflect two vectors: one vector specifying the horizontal (or x direction) size of vertices, the other determining the vertical (or y direction) size. For instance, we may calculate the indegree of countries on each of the two import relations. Then we can use the *Draw>Draw-2Vectors* command to use the normalized indegree on imports of metal manufactures for horizontal size of vertices and the normalized indegree on imports of cereals for the vertical size. A country with relatively many import ties on metal manufactures are now identified by 'flat' vertices, whereas 'upright' vertices signal countries with relatively many import relations regarding cereals. Figure 9 shows the result.

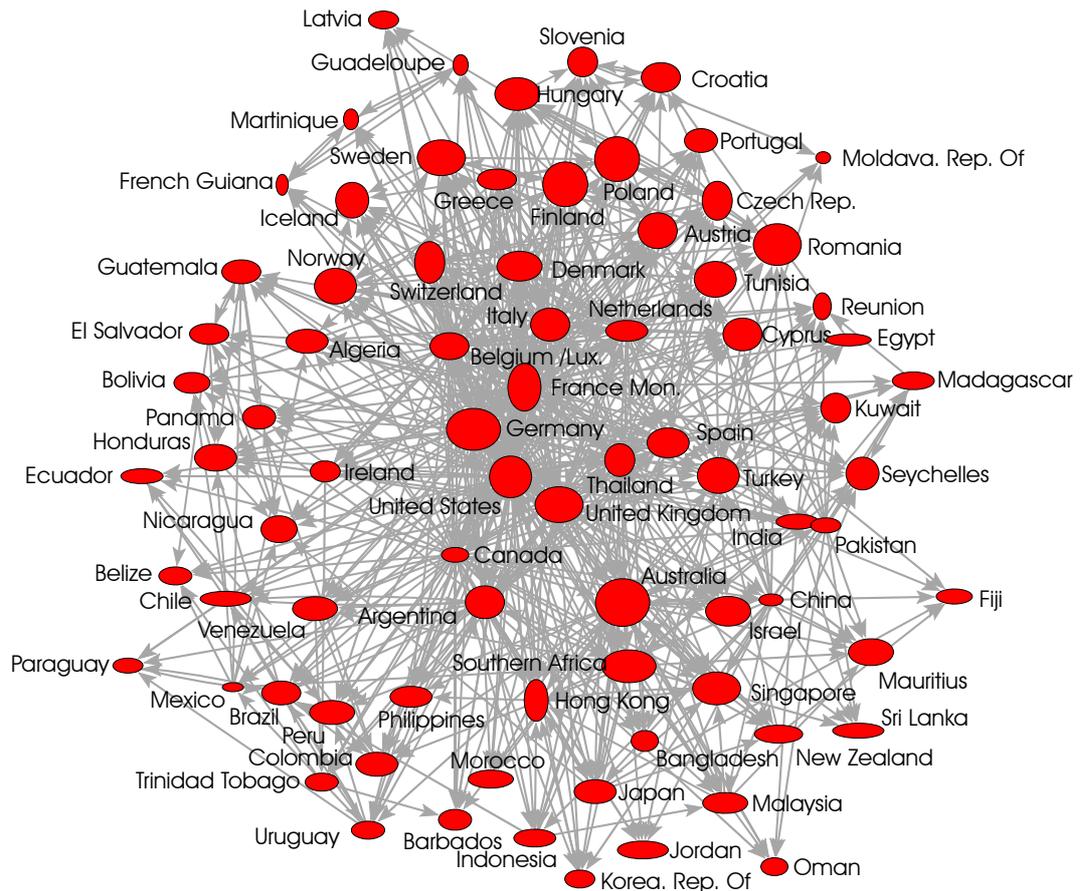


Figure 9 - Imports of cereal, vertices showing indegree on metal manufactures (x) and cereals (y).

5 And much more...

Finally, we will just enumerate some new features of Pajek that broaden the scope of applications. We advise to play around with the new commands and consult the digital manual to Pajek (`Pa jekMan.pdf`) for basic information.

Nets>*Multiply First* *
Second

Operations>*Vector*
 >*Network*Vector*

First, efficient algorithms for multiplication of networks have been added in the *Nets* menu. This command executes matrix multiplication. In conjunction with commands in the *Operations* menu, especially *Operations*>*Vector*>*Network*Vector*, a lot of matrix algebra can be performed in Pajek.

Net>*Partitions*> *Core*,
Valued Core
Operations
 >*Blockmodeling*

Second, there are several new procedures for direct analysis of 2-mode networks. The *Core* and *Valued Core* algorithms in the *Net*>*Partitions* submenu have been extended to include the identification of cores in 2-mode networks. In addition, the blockmodeling command now allows for blockmodeling 2-mode networks.

Net>*Hierarchical*
Decomposition
 >*Clustering with*
Relational Constraint

Third, it is now possible to execute hierarchical clustering with relational constraint. In this hierarchical clustering method, which takes a network of dissimilarities as its input, only dissimilarities among vertices that are linked are taken into account. Thus, it is possible to find hierarchical clusters in large networks.

Finally, some new export formats have been implemented: GraphML for network data and X3D format for visualizations in 3D.