

Bipartite Network Graph

Description

The Bipartite Network Graph algorithm plots a bipartite network (a network with exactly two distinct types of nodes) where nodes have an attribute *bipartitetype*. For instance, in the images at Sample Visualizations section shows one type of nodes represents a person, and the other type represents an award. The [Extract Bipartite Network](#) algorithm is an easy way to generate networks that are graph-able by this plugin.

The nodes and edges can each be independently weighted. The algorithm chooses which column to put a node in based on a node attribute called "bipartitetype". For details on the input format, see Usage Hints below.

Pros & Cons

It's easy to see the network structure in bipartite networks, because of the obvious separation between the two different node types. However, medium to large networks (50+ nodes) may not look very good.

Menu Path

Visualization -> Networks -> Bipartite Network Graph

Input Parameters

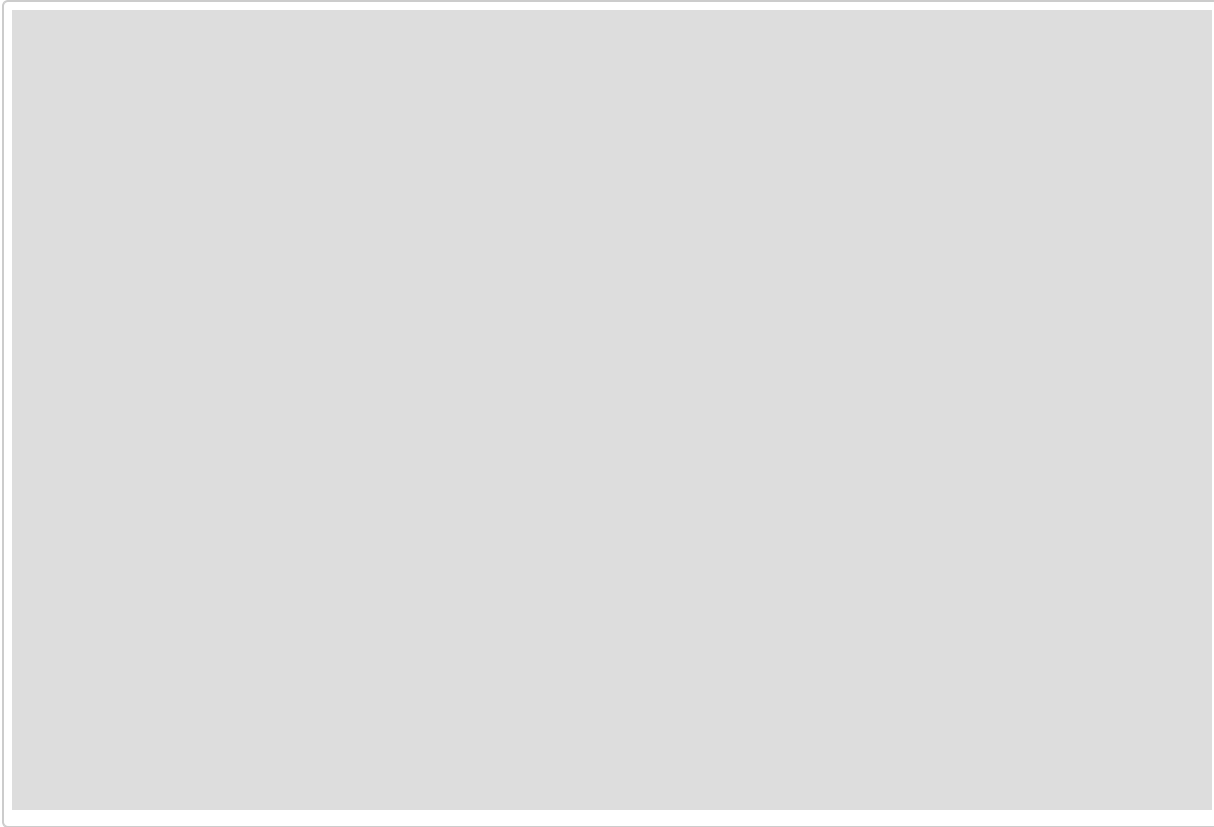
| Name | Description | Notes |
|------------------------|--|---|
| Layout Type | Layout the resulting visualization for print use or web use | <i>PRINT</i> selects a version that is better for use with printing and <i>WEB</i> selects larger, simpler version for use in a webpage |
| Left Side Node Type | The value from the <i>bipartitetype</i> node attribute that should be used for the left side | The other value will be used for the right side See #Usage Hints for details of <i>bipartitetype</i> |
| Node Size | The node attribute by which to size the nodes | If none is selected, the nodes will be sized equally |
| Edge Weight | The edge attribute by which to size and color the arrows | If none is selected, the edges will be equally sized and of the same color |
| Title for left column | The title to put above the left side | If blank, the value of the node attribute "bipartitetype" for the left side will be used |
| Title for right column | The title to put above the right side | If blank, the value of the node attribute "bipartitetype" for the right side will be used |

Outputs

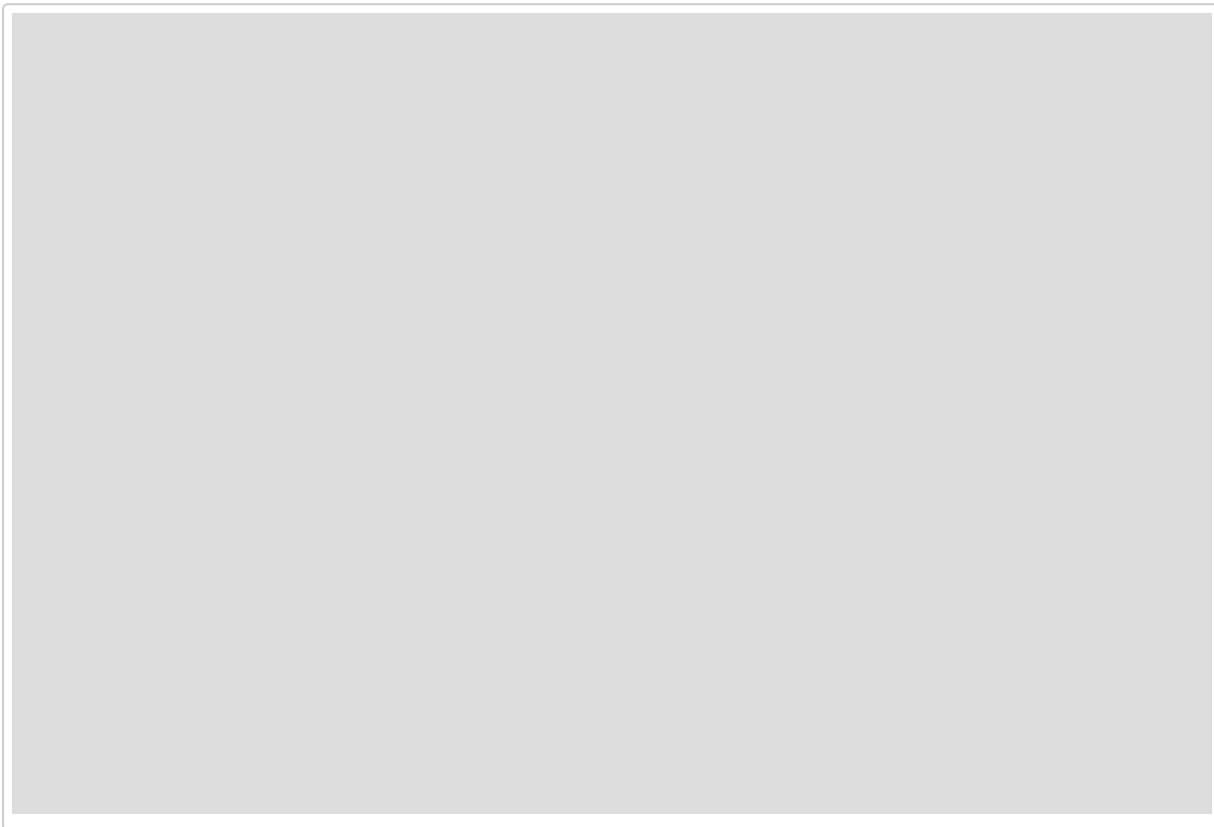
| Name | Description | Notes |
|----------------------------|--------------------------------|-------|
| Bipartite Network Graph PS | A PS file of the visualization | |

Sample Visualizations

*PRINT*Layout



Simplified Layout



Usage Hints

This is the network file ([NWB \(.nwb\)](#) format) that was used to generate the example graphs above:

```
*Nodes
id*int label*string totaldesirability*int bipartitetype*string
1 "Applicant's Proposal" 1 "Who"
2 "Kiss My Red Ruby Lips" 5 "What"
3 "Shoe My Pretty Little Feet" 10 "What"
4 "Glove My Hand" 9 "What"
5 "Be My Man" 4 "What"
6 "Papa" 9 "Who"
7 "Mama" 8 "Who"
8 "Sister" 4 "Who"
9 "No Man" 3 "Who"
*DirectedEdges
source*int target*int linkdesirability*int
1 4 1
1 2 1
1 3 1
1 5 1
6 3 9
7 4 8
8 2 4
9 5 3
```

The nodes are required to have an attribute called "bipartitetype". Note that in the above network, the two values of "bipartitetype" are "Who" and "What". These are also the titles of the columns in the graph. This attribute is used to determine which column the nodes should go in. You can generate your own network with this attribute, or you can use the Extract Bipartite Network algorithm.

The weights ("totaldesirability" and "linkdesirability") were generated using an aggregation function file with the Extract Bipartite Network algorithm. For more on this, see [Extract Bipartite Network](#) and the [Extract Co-Occurrence Network](#) page (for info on aggregation functions).

To generate the graph file above, I imported [no-man.csv](#) into Sci2, then ran the Extract Bipartite Network algorithm. I chose "Who" and "What" as the First and Second Column parameters, and used [aggfunc-man.txt](#) as the Aggregate Function File. I saved the resulting file as a NWB file.

Links

- [Source code](#) in SVN
- Also requires:
 - [JavaGeom](#)
 - [Geomap library](#) and its dependencies (see [Geospatial Visualization](#))
- [Test code](#)

See Also



The license could not be verified: License Certificate has expired! [Generate a Free license now.](#)