

Network Analysis and Visualization Tool: Pajek

Arshpreet Singh, Chaitanya Bisht, R Prathmesh, Vaibhav Arora
Network Science (CS552) [2023-24-W]



February 15, 2024

History Lesson

- Pajek was developed by Vladimir Batagelj and Andrej Mrvar, who are both professors at the University of Ljubljana in Slovenia. The development of Pajek started in the mid-1990s, with the first public release in 1996.
- The name "Pajek" is derived from the Slovenian word for spider, reflecting its ability to analyze and visualize complex networks.
- The Slovenian pronunciation is "Pa-yek", but we will be referring to the tool as Pajek only.

Uses of Pajek

Initially, Pajek was created to analyze large social networks, particularly those derived from sociology and anthropology research. Over time, its capabilities expanded, and it became widely adopted in various fields, including computer science, biology, economics, and more.

Some common use cases of Pajek are as follows -

- Social Network Analysis
- Biological Network Analysis
- Infrastructure Network Analysis

and many more

Suggested Readings


As such, there is no documentation available for Pajek but the following books may be referred -

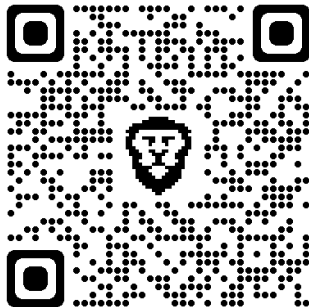
- 1 Pajek Manual
- 2 Exploratory Social Network Analysis with Pajek

Installation

Website

<http://mrvar.fdv.uni-lj.si/pajek/> [Click here](#)

	Ver.	32 bit	64 bit
Sep 1, 2023	5.18	Web Start <i>Pajek P₂ Pajek</i> Install Shield Install-Zip Portable	Web Start <i>Pajek P₂ P₃ Pajek</i> Install Shield Install-Zip Portable
2018	Pajek Book Edition 3		
pajek@list.fmf.uni-lj.si		Datasets	



Pajek can be installed relatively easily in Windows and Linux

Windows and Linux

- Pajek is readily available on Windows and can be downloaded through the website. Just follow the steps in the installation wizard.
- However, for Linux, you first need to install Wine, a Windows emulation program in Linux

Wine Installation

<https://pastebin.com/3ppAEeYE> 

Enable 32-bit programs

```
sudo dpkg --add-architecture i386
```

Add the Repository

```
sudo mkdir -pm755 /etc/apt/keyrings
```

```
sudo wget -O /etc/apt/keyrings/winehq-archive.key
```

```
https://dl.winehq.org/wine-builds/winehq.key
```

Update package repository

```
sudo apt update
```

Install Wine

```
sudo apt install --install-recommends winehq-stable
```



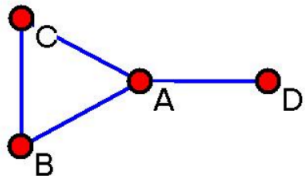
Basics - The .net file structure

*Vertices 4

1 "A"
2 "B"
3 "C"
4 "D"

*Edges

1	2	1
1	3	1
2	3	1
1	4	1



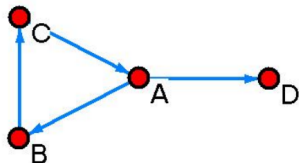
Basics - The .net file structure

*Vertices 4

1 "A"
2 "B"
3 "C"
4 "D"

*Arcs

1	2	1
2	3	1
3	1	1
1	4	1



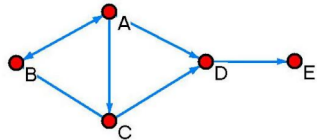
Basics - The .net file structure

*Vertices 5

1 "A"
2 "B"
3 "C"
4 "D"
5 "E"

*Arcslist

1	2	3	4
2	1		
3	4	2	
4	5		



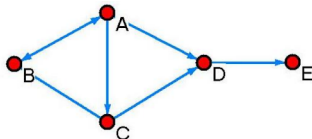
Basics - The .net file structure

*Vertices 5

1 "A"
2 "B"
3 "C"
4 "D"
5 "E"

*Matrix

0	1	1	1	0
1	0	0	0	0
0	1	0	1	0
0	0	0	0	1
0	0	0	0	0



Basics - The .net file structure

*Vertices 26

1	"Ada"	0.1646	0.2144	0.5000
2	"Cora"	0.0481	0.3869	0.5000
3	"Louise"	0.3472	0.6969	0.5000

...

*Arcs :1 "*Dining-table partner choice*"

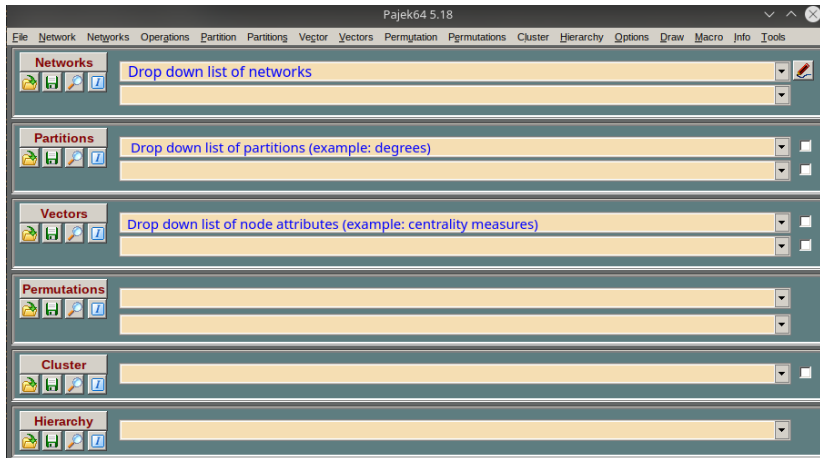
1	3	2
1	2	1
2	1	1

...

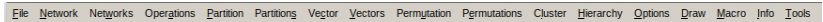
*Edges :2 "*Cooperation*"

1	2	1
2	4	1
1	4	1

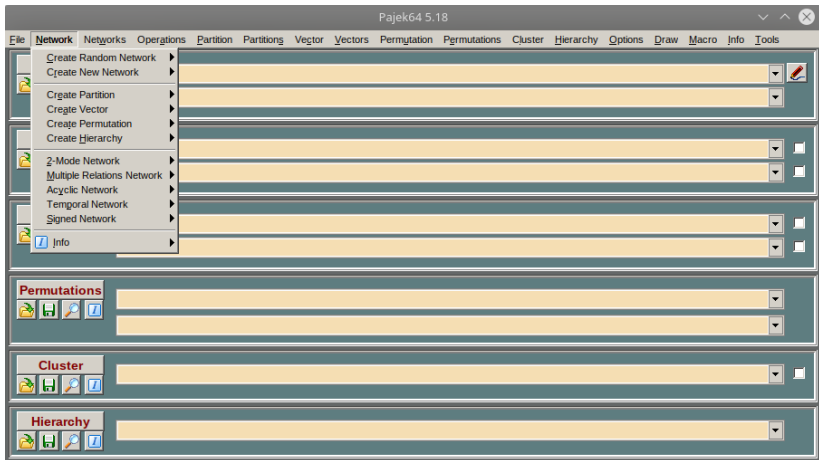
Interface



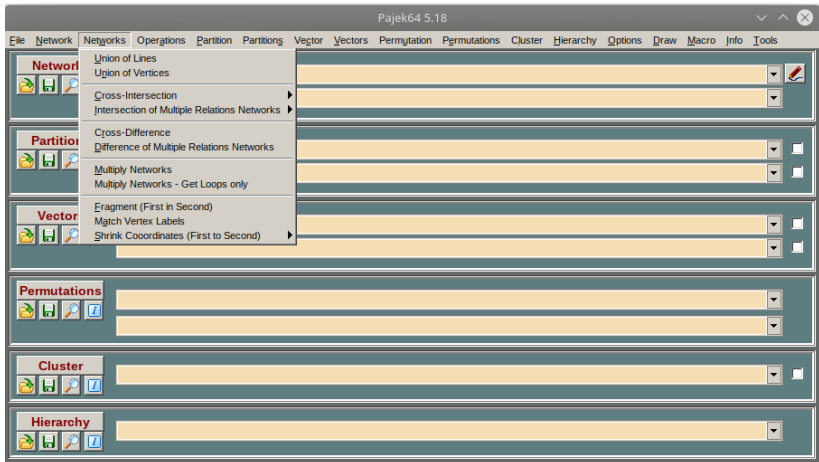
Interface



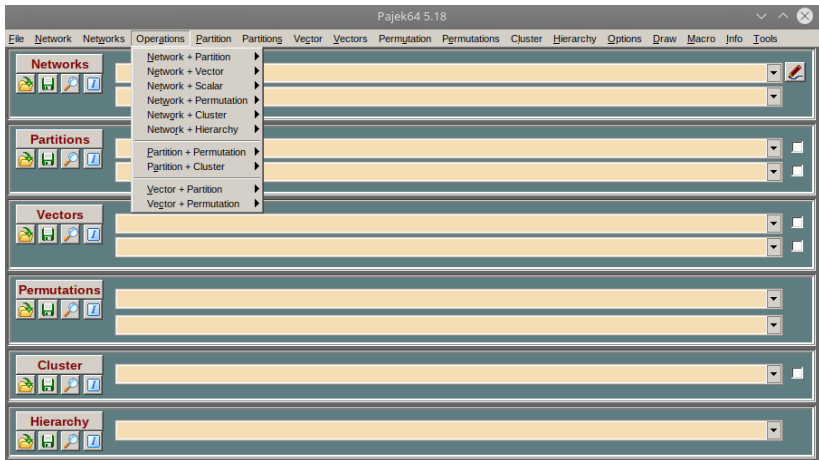
Interface



Interface

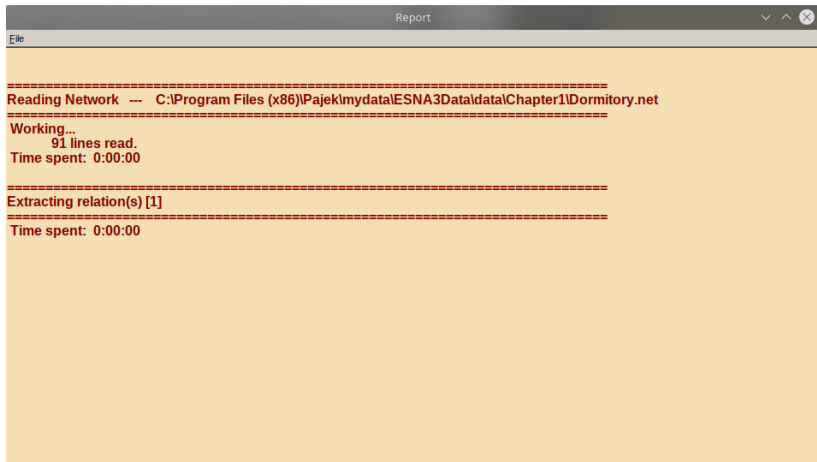


Interface



Report Window

Info > Child Windows > Report Window > Show

A screenshot of the 'Report' window in the Pajek software. The window has a title bar with 'Report' and standard window controls. Below the title bar is a menu bar with 'File'. The main area has a light orange background and displays text in a red monospaced font. The text is organized into sections separated by lines of equals signs. The first section is titled 'Reading Network' and shows the file path 'C:\Program Files (x86)\Pajek\mydata\ESNA3Data\data\Chapter1\Dormitory.net'. The second section is titled 'Working...' and shows '91 lines read.' and 'Time spent: 0:00:00'. The third section is titled 'Extracting relation(s) [1]' and shows 'Time spent: 0:00:00'.

```
=====  
Reading Network --- C:\Program Files (x86)\Pajek\mydata\ESNA3Data\data\Chapter1\Dormitory.net  
=====  
Working...  
    91 lines read.  
Time spent: 0:00:00  
  
=====  
Extracting relation(s) [1]  
=====  
Time spent: 0:00:00
```

Creating a network from Pajek

We can start with a blank network, having just vertices and no connections.

Network > Create New Network > Empty Network

Enter the number of vertices = 10

File > Network > View/Edit

Creating a network from Pajek

We can also create a random network in Pajek. In this presentation, we'll focus on two types of networks

A small-world network is a graph characterized by a high clustering coefficient and low distances.

A scale-free network is a network whose degree distribution follows a power law, at least asymptotically.

$$P(k) \sim k^{-\gamma}, \text{ where } 2 < \gamma < 3$$

Vectors

Data object used to define the values for each node.

Centralities

- Betweenness centrality
- Closeness centrality
- Eigenvector centrality
- Degree centrality

Format of .vec file

*Vertices n

<Value1>

<Value2>

...

Example

*Vertices 5

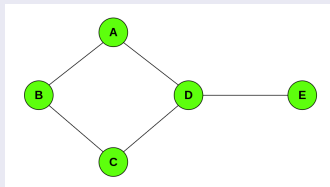
3

2

2

1

3



Finding betweenness centrality

Finding betweenness centrality

Create a vector for betweenness centrality

Network > Create Vector > Centrality > Betweenness

Finding closeness centrality

Finding closeness centrality

Create a vector for closeness centrality

Network > Create Vector > Centrality > Closeness

There are three options

- 1 Input: centrality of each vertex according to distances of other vertices to selected vertex
- 2 Output: centrality of each vertex according to distances of selected vertex to all other vertices.
- 3 All: forget direction of lines – consider network as undirected.

Finding eigenvector centrality

Finding eigenvector centrality

Create a vector for eigenvector centrality

Network > Create Vector > Centrality > Hubs-Authorities

Finding degree centrality

Finding degree centrality

Create a vector for degree centrality

Network > Create Vector > Centrality > Degree

There are three options

- 1 Input: Number of lines into vertices.
- 2 Output: Number of lines out of vertices.
- 3 All: Number of neighbors of vertices.

Partitions

Data object which classifies the vertices/nodes in a network into clusters. Each vertex is assigned to to exactly one class/cluster

- Connected Component
- Distance from a vertex
- Degree

Format of .clu file

*Vertices n

<Value1>

<Value2>

...

Example

*Vertices 5

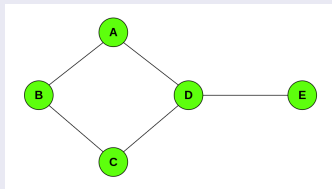
3

2

2

1

3



Finding all the connected components

Finding all the connected components

Create a partition for the connected components

Network > Create Partition > Components

There are three options

- 1 Weak: Weakly connected components
- 2 Strong: Strongly connected components
- 3 Strong-period: Will not be discussed here

Finding the distance of all nodes from a given node

Finding the distance of all nodes from a given node

Create a partition for distance

Network > Create Partition > k-Neighbours

There are three options

- 1 Input: Takes only incoming edges into account
- 2 Output: Takes only outgoing edges into account
- 3 All: Takes both into account

Finding the degree of each vertex (using a Partition)

Finding the degree of each vertex (using a Partition)

Create a partition for degree

Network > Create Partition > Degree

There are three options

- 1 Input: Takes only incoming edges into account
- 2 Output: Takes only outgoing edges into account
- 3 All: Takes both into account

Understanding Permutations

In Pajek, a permutation typically refers to a reordering of the nodes or vertices in a network. They are typically used for the following tasks -

- BFS and DFS (Both weak and strong are supported!)
- Graph Drawing
- Network Comparison

The list goes on and on...

Format of .per file

*Vertices n

<Value1>

<Value2>

...

Example

*Vertices 5

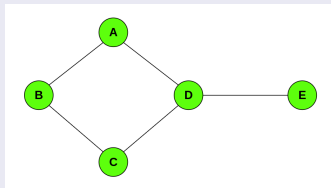
2

1

4

3

5



Breadth First Search

Since BFS is merely a breadth wise reordering of the nodes, the functionality comes under the permutation section. To perform BFS on a graph, the following steps should be performed -

- 1 Go to 'Network' menu.
- 2 Under 'Create Permutation', select the breath first option.
- 3 Choose from strong and weak based on your requirements.

Depth First Search

To perform DFS on a graph, the following steps should be performed -

- 1 Go to 'Network' menu.
- 2 Under 'Create Permutation', select the depth first option.
- 3 Choose from strong and weak based on your requirements.

Limitations of Pajek

Since it was made back in 1996, Pajek has a few limitations -

- 1 Overwhelming UI for beginners
- 2 Performance issues for large networks
- 3 Limited Export features and analysis features
- 4 Non-programmable
- 5 Limited community support and less frequent updates

