1 Introduction

This section provides basic information of GEOMI-2 software.

1.1 Overview

GEOMI-2 is a visual analysis and exploration tool for the visualisation and analysis of large and complex networks such as social networks and biological networks. It is an extended version of GEOMI, supporting more input file formats and providing more plugins that allow users to visualize and analyse different types of networks. GEOMI-2 provides cutting edge visual analytic tools combining network visualisation techniques with network analysis methods to enable exploring and discovering such networks for critical insights.

To understand the system architecture of GEOMI-2, see “Dwyer. (2005), Extending the WilmaScope 3D graph visualisation system – software demonstration” and “Ahmed., et al (2005) GEOMI: GEOmetry for Maximum Insight”.

1.2 Getting Started

GEOMI-2 can run on both Windows and Linux. To start GEOMI-2 in Windows, open the batch file GEOMI-2.bat. Ensure that the directories that GEOMI-2 is placed do not contain any spaces in their name. The user can use Eclipse to run the GEOMI-2 source code. Import the GEOMI-2 project into Eclipse, put the file WILMA_CONSTANTS.properties into the "src" folder then run the main class geomi.gui.Geomi.

To start GEOMI-2 on a Mac/Linux machine, open the GEOMI-2.sh script at the terminal:

bash$ sh /GEOMI-2.sh

Prerequisites:

- Java JDK 1.6 and Java3D 1.5.2 need to be installed on your computer for GEOMI-22 to work.
- Java JDK is available from: http://java.sun.com/j2se.
- For Java3D, get it from: https://java3d.dev.java.net/binary-builds.html.
- A 3-button mouse is required to rotate (left button), zoom (middle/scroll button) and translate (right button) the displayed graph.

1.3 GEOMI-2 GUI

The figure below shows the GEOMI-2 GUI. It consists of three parts. In the top left part users can find some menus that allow user interaction such as file loading and graph editing. The left center is the graph display area. In the right part users can choose different plugins to generate, layout, modify or analyse a graph.
1.4 Plugins

GEOMI-2 provides four types of plugins that allow users to visualize and analyse different types of data:

- Graph Generators: create a graph
- Graph Layouts: apply graph layout algorithms
- Graph Modifiers: modify a graph layout
- Graph Analyses: analyse a graph based on node/edge properties

1.5 GEOMI-2 Visualization Steps

The following steps show how users can use GEOMI-2 to visualize data. The details of each step are described in next few sections.

1. Create a graph based on input data (see section 2)
2. Select appropriate from layout plugins (see section 5) or
3. Select a modifier to filter, group or cluster the graph (see section 6) or
4. Select an analyse plugin to analyse the graph properties (see section 7)
2 Input

Users have three ways to create the input graph with GEOMI-2:

- Create a file and loaded into GEOMI-2
- Use GEOMI-2 generate plugins to create a graph
- Use GEOMI-2 interaction tool to create a graph

2.1 File Input

An easy way for users to visualize their own data with GEOMI-2 is to convert the data into a file format that GEOMI-2 understands and then simply to load them into GEOMI-2. The file menu allows users to control their files.

- New: Clear all existing graph elements, ready for a new visualization
- Open: prompts you with a file selection dialog and then opens up the file
- Save: saves the graph to a file
- Save As: saves the graph to another file
- Screen Capture: capture the current visual display in the graph drawing panel
- Screen Capture (animated): captures a series of visual display in an animation

The file formats that GEOMI-2 supports are:

- XML Wilma Graph (.xwg)
- Graph modelling language (.gml)
- LEDA graph (.lgr)
- NET graph (.net)
- GraphML graph (.graphml)
- Clustered graph (.cgf)

2.1.1 XML Wilma Graph (.xwg):

XML Wilma Graph (XWG) file format is the internal format of Wilmascope. Based on XML, XWG files are easily parsed and generated by XML libraries and tools. XWG is based on a fairly straightforward DTD which can be found in data/WilmaGraph.dtd file. Note that this file should be present in any directory from which you wish to load XWG files into GEOMI-2.

An example of XWG file format:
Users can specify node and edge properties in the file.

Users can define a clustered graph in this format. The figure below shows how to define a cluster.
Loaded into GEOMI-2:

Following this way, a clustered graph can be created and shown in the figure below:

### 2.1.2 Graph Modelling Language (.gml)

Graph Modelling Language (GML) is a hierarchical ASCII-based file format for describing graphs. It has been also named Graph Meta Language. GEOMI-2 currently supports a very limited subset of the GML file format. Users can use “GML Graph Generator” to load the file.

An example of GML file format supported by GEOMI-2:
2.1.3 LEDA graph (.lgr)

GEOMI-2 currently supports a limited subset of the LEDA Graph File Format (LGR).

An example of LGR file format supported by GEOMI-2:
2.1.4 NET graph (.net)

NET file format is used in Pajek

An example of NET file format supported by GEOMI-2:

```
*Vertices 6
 1 I1
 2 I3
 3 W1
 4 W2
 5 W3
 6 W4

*Edges
 1 5
 3 5
 3 6
 5 6
 2 3
 4 5
```

Loaded into GEOMI-2:
2.1.5 GraphML graph (.graphml)

GraphML is an XML-based file format for graphs. The GraphML file format results from the joint effort of the graph drawing community to define a common format for exchanging graph structure data. It uses an XML-based syntax and supports the entire range of possible graph structure constellations including directed, undirected, mixed graphs, hypergraphs, and application-specific attributes.

An example of GraphML file format supported by GEOMI-2:

```xml
<?xml version="1.0" encoding="UTF-8"?>
<graphml xmlns="http://graphml.graphdrawing.org/xmlns"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://graphml.graphdrawing.org/xmlns http://graphml.graphdrawing.org/xmlns/1.0/graphml.xsd">  
    <graph id="G" edgedefault="undirected">  
        <node id="n0"/>  
        <node id="n1"/>  
        <edge id="e1" source="#n0" target="#n1"/>  
    </graph>  
</graphml>
```

Loaded into GEOMI-2:

![GraphML Diagram]

The user can define a clustered graph in GraphML format and use the "Graphml Graph Reader" to load it.

An example of clustered graph in GraphML format:
2.1.6 Clustered graph (.cgf)

The CGF file simply contains the file names for the GML files. The GML files must be in the same directory as the CGF file.
2.2 Input Using Generate Plugins

Users can use GEOMI-2 generate plugins to create an input graph. Details are given in Section 4.

2.3 Input Using Interaction Tools

GEOMI-2 provides interaction tools that allow users to create their own graph using mouse click.

Graph generator plugins create different types of graph input. They are two types of generator plugins: generator and data reader. When users click “edit”, they will see a drop-down list like this:

![Drop-down list](image)

2.3.1 Create a Node

Users can click to choose from the menus to create a node, create an edge or create a cluster.

Create a new node:

![Create a new node](image)

Now if the user right clicks on the new node a menu will appear:
There are quite a few things the user can choose such as change a node's radius, color and show a node's details.

### 2.3.2 Create an Edge

If the user wants to create an edge, he needs to choose two nodes to connect. Select the two nodes (one at a time). In the figure, four mouse icons were added to illustrate the mouse clicks needed to do (from the top to the bottom).

Once the user selected the two nodes and click OK (at the bottom right corner of the window), a new edge is created connecting the two selected nodes.

Now if the user right clicks on the new edge a menu will appear:

There are quite a few options for the user to choose such as changing edge's color, weight, etc.

### 2.3.3 Create a Cluster

Given a graph, users can group nodes into clusters using the "Add Cluster" button, then select nodes to add into this cluster. Each time the user chooses a node, its color is changed. When the user finishes node selection, click on "OK" button on the bottom right corner. A new cluster will appear and contain all the nodes selected. The new cluster is shown with a light yellow bubble around the clustered nodes.
The figure below shows a graph containing two clusters.

Users can right click a cluster and choose the options on the menu to change properties of the cluster.

For example, if users choose “Select Type”, they can set different views for the cluster. The figure below shows spherical view of a cluster.

2.3.4 Other Interactions

Users can do other interactions such as adjust lighting, set parallel projection or set background color. These can be done through the drop-down list of the “view” menu.
For example, Users can change the background color of the display area:
3 Output

Users can use “File-Save” menu to save an existing graph and create an output file. Click “File” menu and choose “Save” or “Save as”, then specify the file name and location to create the output file. The output file records the information of layout, node and edge properties. It is very useful for users who want to preserve the node and edge positions after applying a particular layout, such as force directed layout.

Users can specify the output file format, see the figure below. For the file formats supported by GEOMI-2, see Section 2.1.
4 Graph Generators

4.1 Generator

A generator can create a graph (randomly) that will be satisfied by the specified class of graph. Users can specify also the number of nodes, the number of edges or the probability of connections of nodes, in the result graph.

When users click on the “Generators” tab in the plugins panel, a menu of generators and readers will appear. They can choose a generator from the generator category.

Users can choose a generator from the list.

- Random Graph
- Tree
- Grid Graph
- HTML Graph Generator
- Triangulated Planar Graph
- Scale Free Network
- General Clustered Graph
- Tree-Structured Clustered Graph
- Planar-Structured Clustered Graph

4.1.1 Random Graph

This plugin creates a random graph. Users can set the number of nodes and edges.
Users can apply many general layouts to this graph, see 5.1 General Layout.

4.1.2 Tree

This plugin creates a tree graph. Users can set the number of nodes, node degree, root degree and height.

Users can apply tree layouts to this graph, see 5.4 Tree Layout.
4.1.3 Grid Graph

This plugin creates a grid graph. Users can set width and height.

Users can apply the Force Directed Layout to this graph, see 5.1.1 Force Directed Layout.
4.1.4 HTML Graph generator

This plugin creates a graph of HTML page links. Users input the url and the depth, the result graph will show the connection of HTML pages. The figure below shows the graph of the InfoVis group website page links (http://www.it.usyd.edu.au/~visual/).

Users can apply general layouts to this graph, see 5.1 General Layout.

4.1.5 Triangulated Planar Graph

This plugin creates a triangulated planar graph. Users can set the number of nodes.
Users can apply Planar Layout to this graph, see 5.3 Planar Layout.

4.1.6 Scale Free Network

The scale-free network generator can generate a scale-free network. Users can specify the initial number of nodes, the target number of nodes and edge connection probability.
Users can apply Scale-Free Layout or Force Directed Layout to this graph, see 5.1.1 Force Directed Layout and 5.1.3 Scale-Free Layout.

4.1.7 General Clustered Graph

This plugin creates a general clustered graph. Users can set the number of nodes in each cluster, the number of clusters, the number of inter-cluster and intra-cluster edges.

Users can apply General Clustered Layout to this graph, see 5.5 General Clustered Graph Layout.

4.1.8 Tree-Structured Clustered Graph

This plugin creates a clustered graph with tree structure. Users can set the number of nodes in each cluster, the number of clusters, the number of inter-cluster and intra-cluster edges. Moreover, users can also set the average number of children, balanceness and depth of the graph.
Users can apply Tree-Structured Clustered Graph Layout to this graph, see 5.7 Tree-Structured Clustered Graph Layout.

4.1.9 Planar-Structured Clustered Graph
This plugin creates a clustered graph with planar-structure. Users can set the number of nodes in each cluster, the number of clusters, the number of inter-cluster and intra-cluster edges.
Users can apply Planar-Structured Clustered Graph Layout to this graph, see 5.6 Planar-Structured Clustered Graph Layout.

### 4.2 Data Reader

The data reader plugins provide similar functions as “File-Open” button, but can support more data formats. Normally users can convert their data file into one of the format described in Section 2 and use “File-Open” button to load the data. However, if a developer wants to use GEOMI-2 to visualize a data file in other formats, he/she can write a data reader plugin to achieve it. When users click on the "Generate" tab in the plugins panel, a menu of generators and readers will appear. They can choose a reader from the reader category.

#### 4.2.1 Tree Reader

The tree reader reads a tree. The input file format is shown below. Each node should have the property of name, parent and depth.

```
# Node, Parent, Depth
N1, root, 1
N2, N1, 2
N3, N1, 2
N4, N1, 2
N5, N2, 3
N6, N2, 3
N7, N3, 3
N8, N3, 3
N9, N4, 3
N10, N4, 3
N11, N4, 3
N12, N4, 3
N13, N2, 4
N14, N2, 4
N15, N2, 4
N16, N2, 4
```

Click “Tree Reader”, choose a file, and click "Generate" to build the tree graph.

Users can apply tree layouts to this graph, see 5.4 Tree Layout.
4.2.2 Simple Time Series Reader

It reads a time series data. The input graph should contain information of time step and weight. An example input data file is shown below:

```
#time series
#number of nodes in each time series
5
#year,name,weight
1995,n1,18
1995,n2,98
1995,n3,90
1995,n4,210
1995,n5,20
1996,n1,18
1996,n2,18
1996,n3,280
1996,n4,110
1996,n5,60
1997,n1,18
1997,n2,182
1997,n3,40
1997,n4,210
1997,n5,70
1998,n1,18
1998,n2,180
1998,n3,210
1998,n4,100
1998,n5,100
```

See 5.1.6 Stream Graph Layout for details.
5 Graph Layouts

When users have a graph input, they can choose a layout plugin to apply a layout for the graph. There are a number of different layouts, which can be characterised by their characteristics and their functionality such as force-directed layouts, tree layouts, clustered graph layouts, circular layouts, etc. Some of the layouts can be applied for any general graph, and others require a particular graph input.

Embedded layouts: Some graph generators already embed a graph layout such as a force-directed layout. In these cases, the specified layout will be applied automatically after the graph is constructed. However, user may still can choose a different graph layout and apply it to the current graph.

For graph generators that don’t include a graph layout, user may then need to select a layout to apply. The default layout in GEOMI-2 is force-directed.

List of Layout Plugins:

a) General Layout:
   - Force Directed Layout
   - Circular Layout
   - ScaleFree Layout
   - Graphscape
   - Matrix Layout
   - Stream Graph Layout

b) Hierarchical Graph layout:

c) Planar Graph Layout

d) Tree Layout:
   - Free Tree Layout
   - Cone Tree Layout
   - Rod Tree Layout
   - Radial Tree Layout with Interaction
   - Walker Tree Layout
   - Treemap Layout
   - Sunburst Layout
   - Tree Dendrogram
   - Radial Dendrogram

e) General Clustered Graph Layout:
   - Clustered Force-Force Layout
   - Clustered Circular-Force Layout
   - Clustered Force-Circular Layout
   - Clustered Circular-Circular Layout
   - Clustered Circular-Matrix Layout

f) Planar-Structured Clustered Graph Layout

g) Tree-Structured Clustered graph layout:
   - Clustered Graph with FreeTree-Force Layout
   - Clustered Graph with ConeTree-Force Layout
5.1 General Layout

5.1.1 Force Directed Layout

This layout uses force-directed algorithm to draw a graph.

**Input:** General graph.

When users generate a graph, they can click the "Layout" tab and choose "Force Directed" to apply this layout. The example below shows the force directed drawing of a graph.

When users choose force-directed layout in the Layout panel, a control panel will appear to allow them to adjust the forces affecting the layout of the graph, see below.

These sliders allow users to navigate and select the forces. Users can adjust individual force by using the corresponding slider.
There are several types of forces as follows:

- **Spring**: spring force applied to the edges
- **Repulsion**: repulsive force between pairs of nodes
- **Origin**: a force pulling all nodes towards the origin, or centre of the graph. This avoids disconnected graph elements from flying off into space (and disappear)
- **Directed Field**: a "magnetic force field" which causes directed edges to be aligned with the field.
- **Planar**: a force which squashes everything into a plane.

Users can also adjust velocity, friction coefficient and balanced threshold in the control panel.

### 5.1.2 Circular Layout

This plugin was implemented by Joshua Ho. This layout produces a circular (outerplanar) drawing of graphs.
**Input**: General graph.

Click "Layout" tab and choose "circular layout", then click start. A circular layout is produced.

### 5.1.3 Scale-Free Layout

This plugin was implemented by Colin Murray. This layout provides a fast force directed algorithm allowing for interactive modification of the force parameters.

**Input**: The input must be scale free graph.

Users can choose "Scale Free" under "Generate" tab (see Section 4.1.6) or load an input file from `data/Scale-freeNetworks` folder. Users can also create similar format of their own data as an input.
Click "Layout" tab and choose "Scale-free Layout", then click start to apply the layout.

Users can adjust force in the control panel. The edge force controls allow users to turn edge forces on or off. It also allows for the resilience of the edges to be increased or decreased. Other options include the ability to have edges repel instead of attract and to use the weights associated with an edge in calculating the edge force. The node force controls allow users to turn node forces on or off. It also allows the user to modify the extent of the repulsion force as well as the accuracy parameter of the algorithm, which allows for an accuracy vs speed tradeoff. An attractive force towards the origin can also be turned on or off and modified.

The degree layering controls allow users to restrict nodes to layers based on degree. The nodes can be restricted to either parallel planes or concentric spheres. The nodes are partitioned into layers such that nodes with degree greater than or equal to 10 are at the highest layer and nodes with degree less than 5 are on the lowest layer. Single degree nodes are placed on their neighbour's level. The user can also select to colour the different partitions differently and can select to have only incoming edges count towards the degree total.

Parallel planes layout:
5.1.4 Graphscape

This plugin was implemented by Andrew Cunningham. GraphScape adopts a landscape metaphor: the network is placed on a 2D plane, and each attribute is represented by a three dimensional surface,

**Input:** General graph.

Click “Layout” tab and choose “Node Landscape”, then click start to apply the layout.

The style controls allow users to choose the style of the graph view. There are four styles available: Smooth, Delaunay, Spherical and Nodes Pos Indication.

Smooth style:

Spherical style:
Users can tick "Wireframe" or "Cull backfaces" to change the display. Node degree can be differentiated by different radius or different color gradient. These parameters can be set in the section of "Height" and "Color" in the control panel.

Node degree indicated by radius:
The texture controls allow users to choose texture of the graph. The height of the nodes can adjusted in "Height" slide in the control panel.

5.1.5 **Matrix Layout**

This plugin is implemented by Hui Liu. This layout represents a graph as an adjacent matrix. Click the "Layout" tab and choose "Matrix Representation" to apply this layout.

**Input:** General graph.

5.1.6 **Stream Graph Layout**

This plugin is implemented by Hui Liu. It produces a stream graph to represent temporal data.

**Input:** The input graph should contain information of time step and weight, see Section 4.2.2. Click "Layout" tab and choose "Stream Graph" to apply this layout.
5.2 Hierarchical Layout

The Hierarchical Layout plug-in was implemented by Nikola S. Nikolov. It implements the algorithm for drawing directed graphs in three dimensions, a 3D extension to the Sugiyama method which includes an additional step after the layering step. It further partition the layer into a set of $k > 1$ subsets, called walls.

Input: General graph.

Click "layout" tab and choose "Hierarchical Layout", then click start to apply the layout.

Currently the following algorithms are available for each step of the algorithm:

- Layer-assignment algorithms: Longest-Path, Network Simplex, Minwidtt, Stretchwidth, 2-Line Bipartite Layering, k-Line Bipartite Layering
- Wall-assignment algorithms: Single Wall, 2-Wall Min-Cut, 2-Wall Zig-Zag, 2-Wall Dominating Split, K-Wall Balanced, K-Wall Barycentre
- Global ordering algorithms: Layer-By-Layer Sweep
- Coordinate assignment: Brandes-Kopf Algorithm, Longest-Path Layering Based.

The user can choose a method for each step of the algorithm in the control panel. Further, the user can also choose a colour scheme for the hierarchical layout.

A 2D layout using Single Wall assignment method and Longest-Path Layer assignment method:
A 3D layout using 2-Wall Zig-Zag wall assignment method and StretchWidth layer assignment:

A 3D layout using K-Wall Barycenter method:
5.3 Planar Layout

This plugin was implemented by Adel Ahmed. It uses a weighted version of the drawing algorithm for planar triangulated graphs of de Fraysseix, Pach, Pollack (FPP). Node weight is considered.

**Input:** The input graph must be triangulated, see Section 4.1.5.

Click "Layout" tab and choose "Planar Layout", then click start to apply the layout.
The control panel allows users to fold or compress the graph. When users tick the box "Enable Folding", select the folding stretch factor, and click "start" button, the existing graph is folded and a 3D drawing is produced. When users tick the box "Enable Compaction", and click "start" button, the existing graph will become compact.

5.4 Tree Layout

For all the tree layouts, the input must be a tree graph. Users can specify their data as a tree format and use the “tree reader” to read them (see “Tree Reader” in Section 3), or use “File-Open” menu to load them into GEOMI-2.

5.4.1 3D Free Tree Layout

This plugin was implemented by Joshua Ho. This produces a 3D free tree layout. Click the "Layout" tab and choose "3D Free Tree" to apply this layout.

Input: Tree.
5.4.2 3D Cone Tree Layout

This plugin was implemented by Joshua Ho. It produces a 3D cone tree layout. Click the "Layout" tab and choose "3D Cone Tree" to apply this layout.

**Input:** Tree
5.4.3 3D Rod Tree Layout

This plugin was implemented by Joshua Ho. It produces a 3D rod tree layout. It draws the highest child at the center, and the rest of the children around, make the graph looks like a rod. Click the "Layout" tab and choose "3D Rod Tree" to apply this layout.

**Input:** Tree

![3D Rod Tree Layout](image)

5.4.4 Radial Tree Layout with Interaction

This plugin was implemented by Tristan Manwaring. This layout draws a 2D radial tree with interactivity. Click the "layout" tab and choose "Radial Tree" to apply this layout.

**Input:** Tree
Users can interact with the graph by clicking and dragging on the endpoint of the red stick on the graph.

Users can adjust the value of bend frequency and radius separation on the control panel. Users can also choose to use straight edges to achieve a fast interaction, and tick "Momentum Based Rotation" to see the ratio of the graph.
5.4.5 Walker Tree Layout

This plugin was implemented by Nicolas Senechal. It displays a 2D walker tree of the graph. Check the paper “Improving Walker's Algorithm to Run in Linear Time” for details. Click the "layout" tab and choose "Walker Tree" to apply this layout.

**Input**: Tree
Users can adjust the value of vertical distance and dispersion factor in the control panel.

5.4.6 Treemap Layout

This plugin was implemented by Hui Liu. This layout uses squarified algorithm to draw a tree as a treemap. Click the "layout" tab and choose "Treemap" to apply this layout.

**Input**: Tree

![Treemap Layout Example](image)

Users can adjust the value of inclusion to make an inclusion tree.
5.4.7 Sunburst Layout

This plugin was implemented by Hui Liu. This uses sunburst layout to draw a tree. Nodes are placed in different concentric circles. Different subtrees are colored differently. Click the "layout" tab and choose "Sunburst Tree" to apply this layout.

**Input:** Tree
5.4.8 Tree Dendrogram
This plugin was implemented by Hui Liu. This layout uses dendrogram to represent a tree. Different subtrees are colored differently. Click the "layout" tab and choose "Tree Dendrogram" to apply this layout.

Input: Tree

5.4.9 Radial Dendrogram
This plugin was implemented by Hui Liu. This layout uses radial dendrogram to represent a tree. Nodes are placed in different layers in a radial layout. Different subtrees are colored differently. Click the "layout" tab and choose "Radial Dendrogram Tree" to apply this layout.

Input: Tree
5.5 General Clustered Graph Layout

For all the general clustered layouts, the input must be a clustered graph. Users can load a clustered graph file, or use "Clustered General Graph" under "Generate" tab to create a clustered graph. All clustered graph layout were implemented by Joshua Ho. See the paper "Ho., et al (2006) Drawing Clustered Graphs in Three Dimensions" for details of 3D clustered graph drawing algorithms.

The naming of general clustered graph layouts and tree-structured clustered graph layouts is in the format Clustered X-Y Layout, where X is the layout of supergraph and Y is the layout of each cluster.

GEOMI-2 also provides a new matrix layout for each cluster, namely Clustered Circular-Matrix Layout.

5.5.1 Clustered Force-Force Layout

This plugin uses force-directed method to draw a clustered graph. It uses the spring algorithm to draw the super graph. Each cluster is drawn on a 2D plane using spring algorithm. Click the "Layout" tab and choose "Clustered Force-Force Layout" to apply this layout.

Input: General clustered graph, see Section 4.1.7.
Users can adjust the value of spring force, repulsion force and origin force in the control panel. The "Enable ICOM" control allows applying inter-cluster occlusion minimization algorithm. The "Enable rotation" control allows allowing rotational transformation of plane. The "2D" control allows drawing only in 2D. The "Hide IC Edges" control allows hiding the inter-cluster edges.

5.5.2 Clustered Circular-Force Layout

This layout produces a clustered graph. It uses the circular algorithm to draw the super graph. Each cluster is drawn on a 2D plane using spring algorithm. Click the "Layout" tab and choose "Clustered Circular-Force Layout" to apply this layout.

Input: General clustered graph, see Section 4.1.7.
5.5.3 Clustered Force-Circular Layout

This layout produces a clustered graph. It uses the spring algorithm to draw the super graph. Each cluster is drawn on a 2D plane using circular layout algorithm. Click the "Layout" tab and choose "Clustered Force-Circular Layout" to apply this layout.

**Input:** General clustered graph, see Section 4.1.7.
Users can adjust the value of spring force, repulsion force and origin force in the control panel.

5.5.4 Clustered Circular-Circular Layout

This layout uses 3D circular layout algorithm to draw the super graph. Each cluster is drawn on a 2D plane using circular layout algorithm. Click the "layout" tab and choose "Clustered Circular-Circular Layout" to apply this layout.

**Input:** General clustered graph, see Section 4.1.7.
5.5.5 Clustered Circular-Matrix Layout

This plugin was implemented by Quan Hoang Nguyen. It uses 3D circular layout algorithm to draw the super graph. Each cluster is drawn on a 2D plane using matrix layout algorithm. Click the "layout" tab and choose "Clustered Circular-Matrix Layout" to apply this layout.

**Input:** General clustered graph, see Section 4.1.7.
5.6 Planar-Structured Clustered Graph Layout

This plugin was implemented by Adel Ahmed. It uses a weighted version of the Fraysseix, Pach, Pollack algorithm to draw the supergraph. Each cluster is drawn on a 2D plane using a standard spring algorithm. Click "Planar-Structured Clustered Graph Layout" under the "Layout" tab to apply.

**Input:** Planar-Structured clustered graph, see Section 4.1.9.
5.7 Tree-Structured Clustered Graph Layout

For all the clustered layouts with tree structure, the input must be a tree-structured clustered graph. Users can load a clustered graph file, or use "tree-structured clustered graph" under "Generate" tab to create a clustered tree. All clustered graph layout with tree structure plugins were implemented by Joshua Ho.

5.7.1 Clustered Graph with Free-Force Tree Layout

This layout uses 3D free tree algorithm to draw the super graph. Each cluster is drawn on a 2D plane using spring algorithm.

**Input:** Tree-structured clustered graph, see Section 4.1.8.

Click the "Layout" tab and choose "Clustered FreeTree-Force Layout" to apply this layout. Users can adjust the vertical distance between each subtree. The "Enable ICOM" control allows applying inter-cluster occlusion minimization algorithm. The "Enable rotation" control allows allow rotational transformation of plane. The "2D" control allows drawing only in 2D. The "Hide IC Edges" control allows hiding the inter-cluster edges.
5.7.2 Clustered Graph with Cone-Force Tree Layout

This layout uses 3D cone tree algorithm to draw the super graph. Each cluster is drawn on a 2D plane using spring algorithm. Click the "Layout" tab and choose "Clustered Graph with ConeTree-Force Layout" to apply this layout. Users can adjust the vertical distance and dispersion factor between each subtree.

Input: Tree-structured clustered graph, see Section 4.1.8.
5.7.3 Clustered Graph with Rod-Force Tree Layout

This layout uses 3D rod tree algorithm to draw the super graph. Each cluster is drawn on a 2D plane using spring algorithm. Click the "Layout" tab and choose "Clustered Graph with RodTree-Force Layout" to apply this layout. Users can adjust the vertical distance and dispersion factor between each subtree.

**Input**: Tree-structured clustered graph, see Section 4.1.8.
5.7.4 Clustered Graph with FreeTree-Circular Tree

This layout it uses 3D free tree algorithm to draw the super graph. Each cluster is drawn on a 2D plane using circular layout algorithm. Click the "Layout" tab and choose "Clustered Graph with FreeTree-Circular Layout" to apply this layout. Users can adjust the vertical distance between each subtree.

**Input:** Tree-structured clustered graph, see Section 4.1.8.
5.7.5 **Clustered Graph with ConeTree-Circular Layout**

This layout it uses 3D cone tree algorithm to draw the super graph. Each cluster is drawn on a 2D plane using circular layout algorithm. Click the "Layout" tab and choose "Clustered Graph with ConeTree-Circular Layout" to apply this layout. Users can adjust the vertical distance and dispersion factor between each subtree.

**Input:** Tree-structured clustered graph, see Section 4.1.8.

5.7.6 **Clustered Graph with RodTree-Circular Layout**

This layout it uses 3D rod tree algorithm to draw the super graph. Each cluster is drawn on a 2D plane using circular layout algorithm. Click the "Layout" tab and choose "Clustered Graph with RodTree-Circular Layout" to apply this layout. Users can adjust the vertical distance and dispersion factor between each subtree.

**Input:** Tree-structured clustered graph, see Section 4.1.8.
5.8 Clustered Graph Layout with Interaction

GEOMI-2 provides a new type of clustered graph layout that allows users to interact with them. Each cluster is shown with a light bubble around the clustered nodes. Users can change the view of each cluster.

These plugins were implemented by Quan Hoang Nguyen. Users can load a clustered graph file with XWG format, or use “Clustered General Graph for Interactive Layout” under the “Generate” menu as an input.

Users can choose “Clustered Circular-Adaptive Layout with Interaction”, “Clustered Circular-Circular Layout with Interaction” or “Clustered Force-Circular Layout with Interaction” under the “Layout” menu to apply these layouts.

Clustered Circular-Circular Layout with Interaction:
Users can set different views for different clusters. Right click on a cluster and select “Set Type”, then select one of the options from the list to set a view of the cluster (see Section 2.3.3).

A circular clustered layout with mixed cluster views is created:
6 Graph Modifiers

A graph modifiers plugin provides functions such as filtering, clustering and centrality comparison to a given graph. When users build an input graph, they can choose a modifier plugin to modify the graph. For example, they can group nodes, delete edges or rearrange the layout.

List of Modify Plugins:

a) Filter:
   - Node Degree Filter
   - Parallel Edge Filter
   - Biggest Component

b) Clustering:
   - K-Means Clustering

c) Centrality Comparison:
   - Hierarchical Centrality Comparison
   - Orbital Centrality Comparison
   - Parallel Coordinates Centrality Comparison

6.1 Filter

6.1.1 Node Degree Filter
This plugin was implemented by Kai Xu. Users set a node degree threshold values. This plugin can delete nodes with degree no greater than this value.

Input: General graph.

Click "Modify" tab and choose "Node Degree Filter", then click start to apply this plugin.

Input example:
After applying node degree filter, nodes with no more than 3 degrees have been deleted.

6.1.2 Biggest Component

This plugin was implemented by Nikola S. Nikolov. It displays the biggest connected subgraph. It deletes all nodes and edges which do not belong to the biggest connected component.
**Input:** General graph.

Click "Modify" tab and choose "Biggest Component", then click start to apply this plugin.

Input example:

Biggest component of the input graph:
6.2 Clustering

6.2.1 K-Means Clustering

This plugin was implemented by Tim Dwyer. It partitions the graph into clusters using k-means clustering algorithm. Click "Modify" tab and choose "K-Means Clustering", then click start to apply this plugin. Users can select the value of K and select the number of clusters to keep if their values are smaller than K. After applying this, users should select a layout (such as “Force Directed” or “Circular Layout”) under the “Layout” tab to apply the result of clustering.

**Input:** General graph.

Apply K-Means Clustering: 

![K-Means Clustering Plugin](image)

Apply Force Directed Layout to see the result:
Apply Circular Layout to see the result:
6.3 Centrality Comparison

6.3.1 Hierarchical Centrality Comparison

This plugin was implemented by Richard Webber. It uses a hierarchical layout to compare centralities. For each graph the layering is based on centrality values, i.e., nodes in the upper layer have larger centrality values than those in the lower layer. Within each layer nodes are ordered to reduce edge crossings. The “Centrality difference threshold” slider allows users to choose a percentage value. As a result, only the nodes whose centrality value changes over the percentage between different centrality measures are shown. Click "Modify" tab and choose "Hierarchical Centrality Comparison", then click start to apply this plugin.

Input: General graph

After applying “Hierarchical Centrality Comparison”, nodes are placed in different levels based on their centrality values.

6.3.2 Orbital Centrality Comparison

This plugin was implemented by Dirk Koschuetzki. It uses an orbital layout to compare centralities. The idea of orbit-based comparison of centralities can be summarised as follows: copies of the analysed network are stacked, every copy is used to visualise one centrality measure and within a copy the vertices are placed on concentric circles depending on the centrality value of the vertex. Click "Modify" tab and choose "Orbital Centrality Comparison", then click start to apply this plugin. The control panel allows users to select the centralities, the number of orbits and representation of inter-plane edges.

Input: General graph

Users can select one of the centralities in the control panel, for example, degree centrality. Nodes will be placed on the 2D plane based on their centrality values. The “Max.number of orbits” control allows users to select the number of concentric circles on the plane. Nodes with high centrality will be placed
on inner circles, and low centrality nodes will be placed on out circles. Users can also set the “Intra Plane Spring” and “Intra Plane Repulsion” that can apply forces into nodes.

Users can also select more than one centrality to compare. This will give a 2.5D layout that each plane shows one centrality comparison. Users can set intra-plane and inter-plane forces.
6.3.3 Parallel Coordinate Centrality Comparison

This plugin was implemented by Kai Xu. It produces 3D parallel coordinates layout based on different node centralities. Each vertical axis represents a centrality measure. Click the "Modify" tab and choose "Parallel Coordinates Centrality Comparison" to apply.

**Input:** General graph

After applying this plugin, nodes are placed in a set of vertical axes based on their centrality values.
7 Graph Analyses

A graph analyses plugin does not change the given graph. It provides analysis of graph properties such as node centralities.

When users create an input graph, they can choose a analyse plugin to analyse the graph properities.

List of Analyse Plugins:

   a) Centrality Analysis
   b) Biconnected Components
   c) Block Model

7.1 Centrality Analysis

This includes a number of plugins, such as "Closeness Centrality", "Betweenness Vitality" and "Degree Centrality". They were implemented by Tim Dwyer and Nicolas SENECHAL Click "Analyse" tab and choose one of them, then click "add" to apply.

Input: General graph.

For example, if users choose "Degree Centrality" and select visual mapping "Node Label", then click "Apply Mapping", the nodes will be labelled with their centrality values.

Users can also choose to apply "Node Size" instead, then nodes are sized based on their centrality values.
Furthermore, if the user chooses "Node Level Constraints", nodes are placed on different planes based on their centrality values.

If users click "chart", a car chart will display to illustrate the centrality value of the nodes.
7.2  Biconnected Components

This plugin was implemented by Tim Dwyer. It analyses biconnected components of a graph. Cut vertices are highlighted. Click "Analyse" tab and choose "Biconnected Components", then click "add" to apply. Users can choose different visual mappings to see cut vertices.

**Input**: General graph:

After applying Biconnected Components, cut vertices are shown in different colors:

![Diagram of a graph with cut vertices highlighted in different colors]

7.3  Block Model

This plugin was implemented by Richard Webber. It groups nodes according to the graph structure associated with them. After blockmodel analysis, the nodes that are structurally equivalent are assigned the same colour. Two nodes are structurally equivalent if they have the same neighbour set. Click "Analyse" tab and choose "Block Model", then click "Add", choose "Node Colour", and click "Apply Mapping" to apply this plugin.

**Input**: General Graph.
8 Summary

GEOMI-2 is a visual analysis and exploration tool that provides analysis methods, graph algorithms and layout methods to visualize and analyse different types of networks. GEOMI-2 provides four types of plugins:

Generate plugins: graph generator, data reader

Layout plugins: general layout, tree layout, clustered graph layout, clustered graph with tree structure

Modify plugins: graph filter, graph clustering, centrality comparison

Analyse plugins: centrality analyse, block model, biconnected components

Usability & Extensibility:

Users can use existing plugins to visualize and analyse their own data. They can also extend GEOMI-2 and write new plugins for graph visualization.
9 Reference:

