

7.4 Graph Layout Algorithms

Cellware provides two different graph layout algorithms to import SBML files. Due to the non-existence of graph layout information in SBML Level 1, these algorithms help in laying out the various components of the network in the workspace automatically.

Graph layout algorithms are one of the traditional problems studied in computer science. As such literature provides various algorithms to draw a graph. As the general problem of the layout of a graph is NP-complete, different algorithms concern graphs with specific properties. In fact, the notion of having a good layout is based on aesthetic criteria.

For example, the bending of the edges can be seen as a good criterion for some graphs, but for other graphs the edges will only be straight. Some algorithms compute the bend of the edges while others only take care of the position of the nodes. We have tried to address this issue, but more needs to be done in the future.

As Cellware deals with reaction pathways, typical graphs contain linear cascades with branches and cyclic loops. So it seems sensible to draw the cycles in circles and try to straighten the linear parts. Of course, basic criteria are to minimize the crossing of edges and avoid the overlapping of nodes. Also, the bending of the edges is not considered and only the placement of the nodes in the given area is computed. The two algorithms that Cellware includes - force-directed and hierarchical - fulfil a part of the requirements.

More details on the two algorithms are as below:

- **Force-Directed Methods**

The version used in Cellware is based on Fruchterein [3], which is further based on the previous work of Eades [4]. This method represents the graph as a physical system whereby nodes are taken as molecules whereas reaction links are taken as forces between molecules. The method aims to find the equilibrium state of the system for which total forces on each molecules is equal to zero. The force-directed method is computational intensive but is capable of producing highly aesthetic layouts.

- **Hierarchical Method**

This algorithm [5] determines a hierarchy between the nodes. In other words, the nodes are placed on different layers from top to bottom, depending on direction of the edges. Once the nodes are attributed to a layer, order of the nodes is rearranged within the layers to minimize the crossings of the edges.

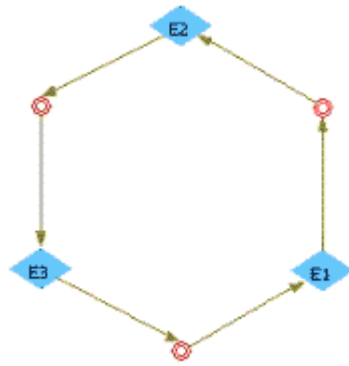


Figure 7-2: Layout using Force-Directed Algorithm

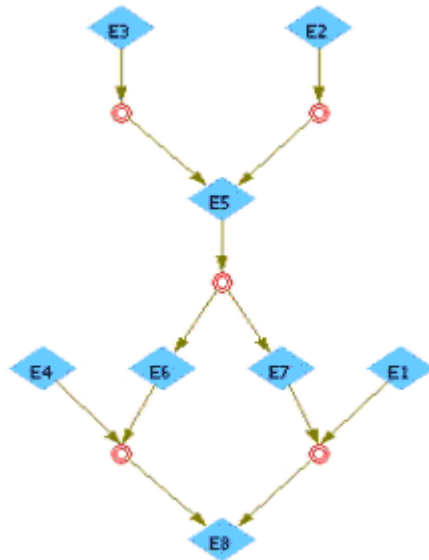


Figure 7-3: Layout using Hierarchical Algorithm

On the whole, the force-directed algorithm is better to draw a graph with cycles (*Figure 7-2*) and the hierarchical algorithm is better for the linear and branched parts (*Figure 7-3*). For a graph with these two types of parts, if there are few nodes, the hierarchical algorithm can give nice results, but with more nodes the force-directed one is better. (*Figure 7-2*) and (*Figure 7-2*) show the layout of a 27-node network using force directed and hierarchical algorithms, respectively.

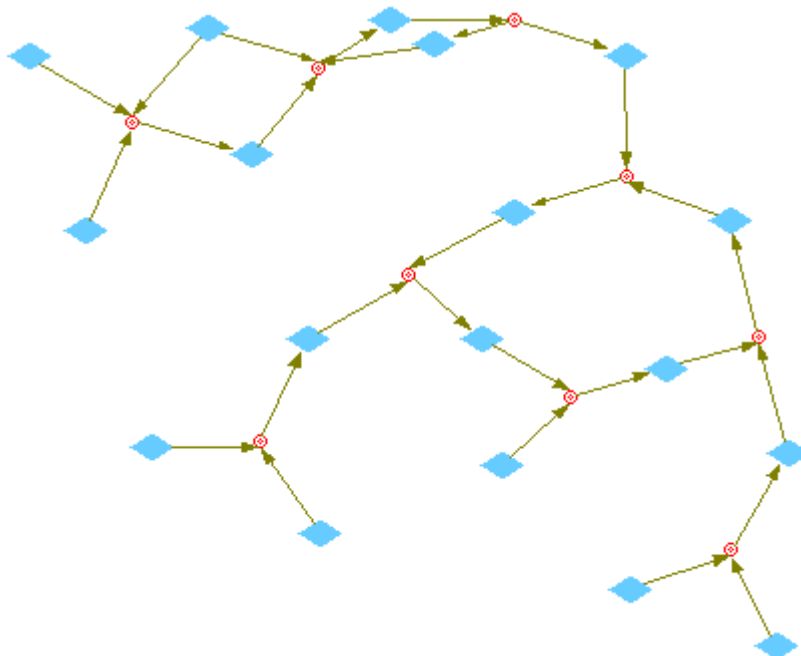


Figure 7-4: Layout of a 27 Node Network Using *Force-Directed* Layout Algorithm

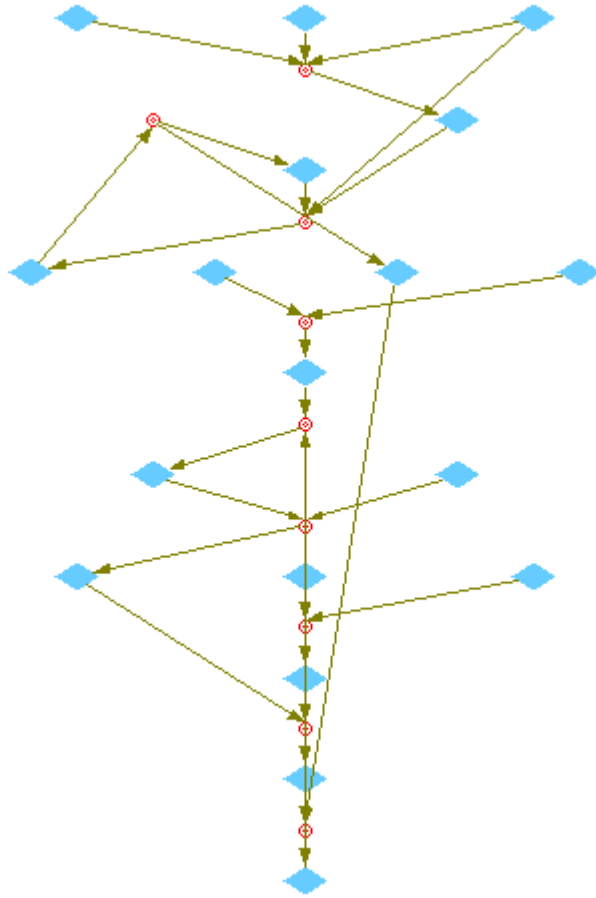


Figure 7-5: Layout of a 27 Node Network Using *Hierarchical* Layout Algorithm

[Back to top](#)