Dimitar Garkov[1] / Karsten Klein[1] / Christian Klukas[2] / Falk Schreiber[1,3]

Mental-Map Preserving Visualisation of Partitioned Networks in Vanted

[1] Department of Computer and Information Science, University of Konstanz, 78464 Konstanz, Germany, E-mail: dimitar.garkov@uni-konstanz.de. https://orcid.org/0000-0002-2134-1686.
[2] Digitalization of Research and Development, BASF SE, 67056 Ludwigshafen am Rhein, Germany
[3] Faculty of Information Technology, Monash University, Clayton, Victoria 3800, Australia

Abstract:
Biological networks can be large and complex, often consisting of different sub-networks or parts. Separation of networks into parts, network partitioning and layouts of overview and sub-graphs are of importance for understandable visualisations of those networks. This article presents *NetPartVis* to visualise non-overlapping clusters or partitions of graphs in the Vanted framework based on a method for laying out overview graph and several sub-graphs (partitions) in a coordinated, mental-map preserving way.

Keywords: Network visualization, Network analysis, Mental map preservation, Data clustering, Graph drawing

DOI: 10.1515/jib-2019-0026

Received: April 5, 2019; **Revised:** April 24, 2019; **Accepted:** April 29, 2019

1 Introduction

Biological systems are characterised by complex, interwoven processes which comprise thousands of elements such as different genes, transcripts, proteins and metabolites. There are different representations of these processes, for example, textual descriptions or mathematical equation systems. Often biological systems are also represented as networks or graphs, see Figure 1. More details regarding the representation of biological networks as graphs, typical layout methods for different biological networks as well as generic layout methods can be found in articles such as [5], [6], [7], [8].

Biological networks can be separated or partitioned in different ways. For example, there exists a large collection of network analysis algorithms including many graph clustering methods [9], [10], partly also for the computation of non-overlapping clusters (partitions). It is also possible to separate the highly interconnected processes into different process classes, such as gene regulatory processes, protein interaction and metabolic processes among several other processes. Independent of the base for partitioning, the result are several non-overlapping clusters in the graph, as well as an overview graph where each node represents a cluster of the graph. This can be formalised as follows:

A (non-overlapping) clustered graph $C = (G, G_O)$ consists of a base graph G and an overview graph G_O, such that the nodes of G_O represent the clusters of G, each node of G belongs to exactly one cluster, and there is an edge between nodes in G_O if there are edges between nodes in the respective clusters in G, see Figure 2. Note that there are several definitions of clustered graphs, for example, a tree-based definition given by Eades defines a clustered graph $C = (G, T)$ as consisting of a base graph G and a rooted tree T, such that the leaves of T are exactly the nodes of G [11]. Let $G = (V_G, E_G)$ be the base graph. The hierarchy is defined by the tree $T = (V_T, E_T)$, with the leaves $L(T) = V_G$. A view is defined as a subset of V_T that induces a partition of V_G [12]. In the remainder of this article we will use the simple definition $C = (G, G_O)$ with an overview graph. Here we also do not want to distinguish between clustered and heterogeneous graphs [13] and therefore consider heterogeneous graphs also clustered graphs where, for example, each cluster contains a specific graph type (graph types can be undirected graphs, directed graphs or graphs with specific node types such as to represent metabolic networks, to name a few examples).

Dimitar Garkov is the corresponding author.

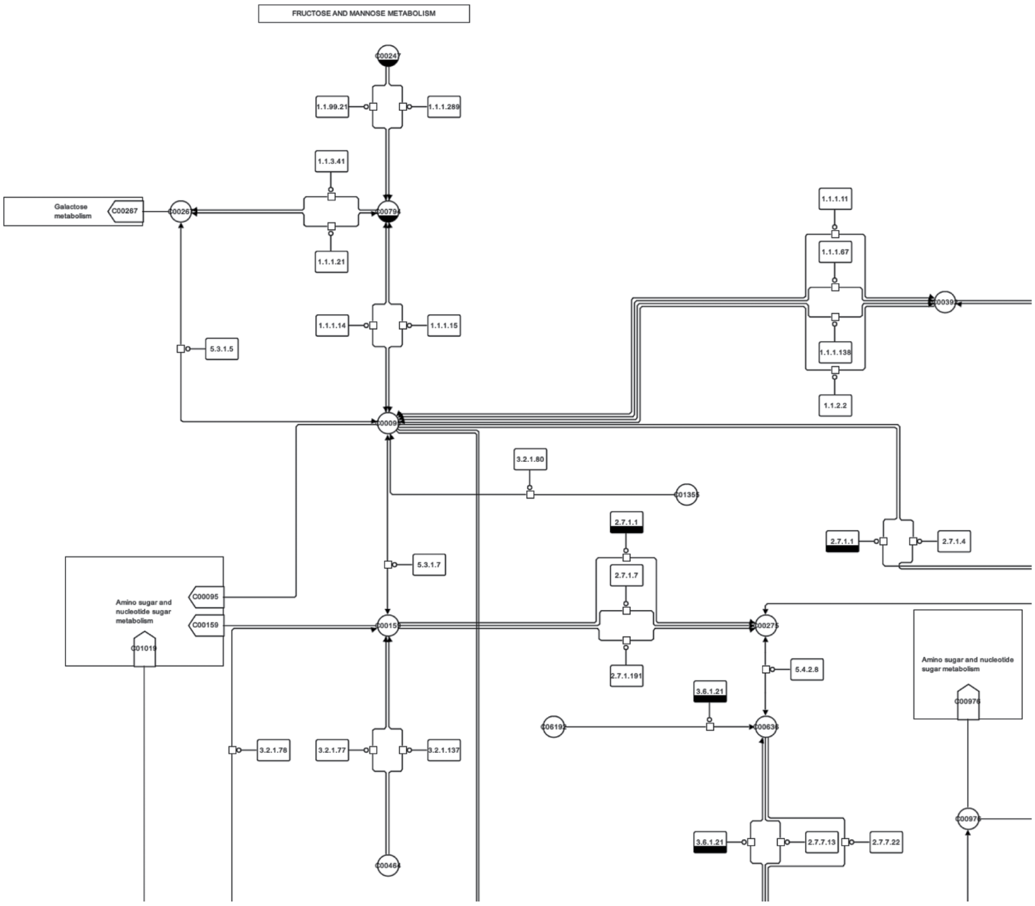


Figure 1: Typical visualisation of a metabolic network, in this case a cutoff of a metabolic pathway derived from KEGG [1]. The network is shown in the SBGN [2] representation computed by SBGN-ED [3], including automatic layout based on the original KEGG layout [4].

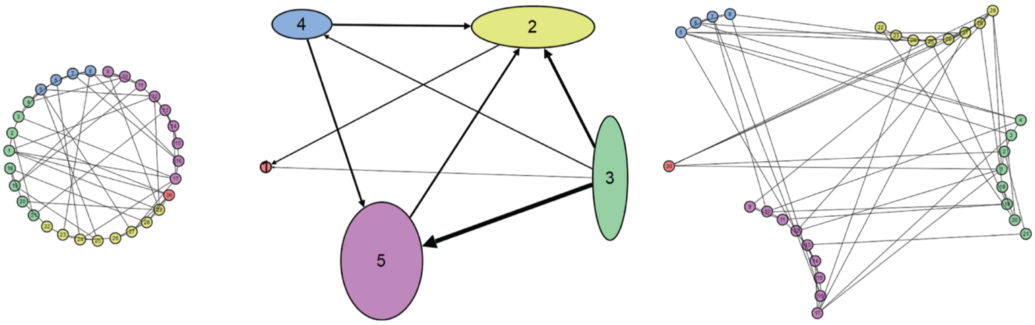


Figure 2: Graph G (with clusters marked by different colours), overview graph G_O with one node for each cluster (in the respective colour) and mental-map preserving layout of G (preserving the structure of the overview graph within the graph G), in this case also without re-layout of the different clustered sub-graphs (from left to right).

Graphs are a mathematical concept for expressing structural relationships between elements of a certain system. An adequate layout of a graph helps to visually recognise its substructures. Layout and visualisation of clustered graphs (including graph structures such a planar clustered graphs and compound graphs) have been a research area in graph drawing for more than 30 years. Early work includes multi-level visualisation of clustered graphs [11], clustering and visual abstraction [14], structured layouts that separate zones for sub-graphs [15], orthogonal grid drawings of clustered graphs [16], drawing of compound graphs [17] and convex drawings of planar clustered graphs [18]. While there have been approaches to apply the planarisation concept to clustered graphs in order to minimise crossings [19], [20], the complexity of the general decision problem is still unknown [21], and current solutions for sub-classes of clustered graphs are far from practical application [22]. There also exist specific approaches for clustered biological networks such as the layout of biological compound graphs [23]. To some extent visualisation solutions which compare several graphs and build a comparison tree [24] can also be seen as layouts of clustered graphs.

Mental-map preserving layouts, introduced by Misue et al. [25], play an important role for the understanding of layout changes. The mental map of a graph is the abstract representation of this graph in the user's brain which is then used to quickly navigate through the graph visualisation when changes occur. Here we will use the term *mental-map preserving* for coordinated views where the layout of the overview graph is easily visible in the spatial relations of the different clusters – the mental map regarding the overview graph is preserved.

Vanted [26] is an open source framework for the analysis and visualisation of biological networks and related experimental data. An Add-on mechanism allows for simple extension of the Vanted core, and several extensions exist, for instance, to compute network centralities [27] and to visualise fluxes in networks [28]. Vanted supports the visualisation of all kinds of biological networks and supports graphical standards for biological networks, in particular SBGN PD [29], SBGN ER [30] and SBGN AF [31]. A typical workflow in Vanted is presented in [32] – in this workflow metabolic maps in the SBGN standard are constructed, enriched with different kinds of *omics data and exported to clickable websites. Other workflows can be found in [33].

The visualisation and exploration of networks is an important part of the Vanted workflow, therefore suitable methods to support these tasks are needed. Here we present *NetPartVis*, a method for layout and explorative visualisation of clustered graphs in Vanted which enables users to lay out an overview graph and the comprising sub-graphs (partitions) in a coordinated, mental-map preserving way. Due to the broad usability of such visualisations we decided to add *NetPartVis* to the main functionality of Vanted (core), therefore no additional loading of an Add-on is necessary.

2 *NetPartVis* – Layout and Visualisation of Clustered Graphs in Vanted

2.1 Construction of Graphs

Let G be the source (base) graph, G_O the overview graph (initially empty) and $G_1, \dots, G_k$ a set of graphs representing the clusters 1, ..., k. Each node of G contains an ID (integer) from 1, ..., k, representing the cluster this node belongs to.

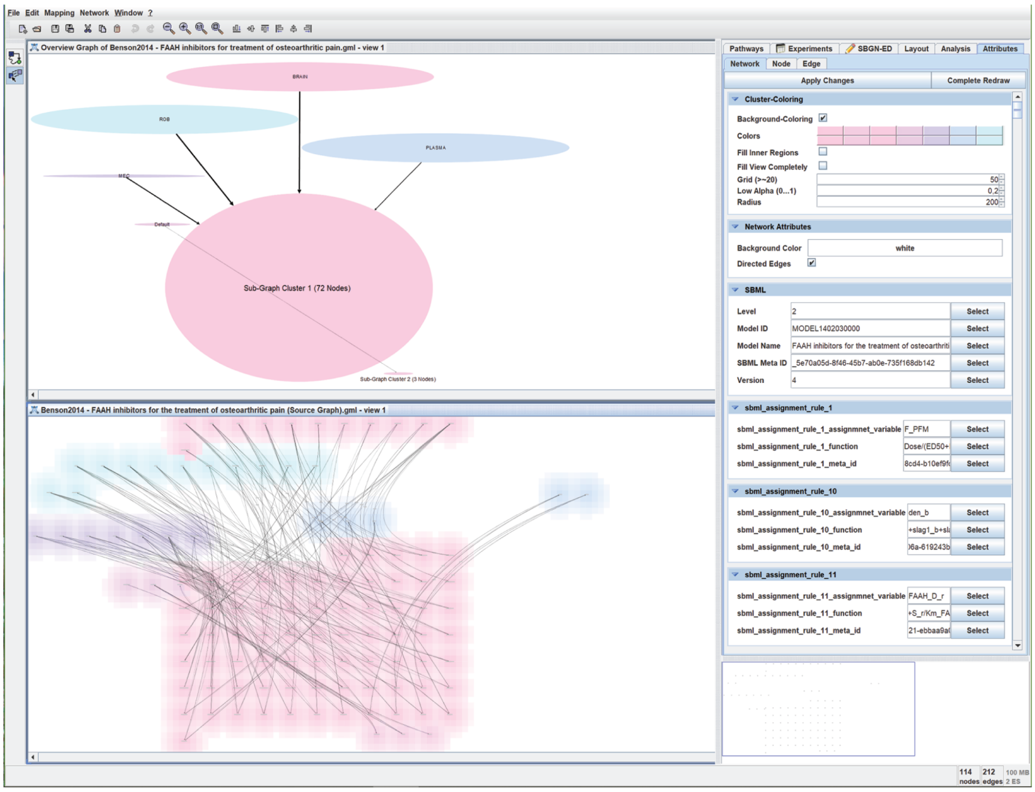
The method for mental-map preserving visualisation of partitioned networks is based on the following data property: The source graph contains nodes that all have cluster IDs assigned. An overview graph and clustered sub-graphs are created from the source graph as follows:

1. For each distinct cluster ID that is assigned to at least one node in the source graph G, a new node representing a cluster is created in the overview graph G_O.
2. For each edge in the source graph G, a new edge connecting nodes in G_O is created, if and only if the cluster IDs of the source and target nodes of that edge in G are different. To avoid duplicated connections of nodes in the overview graph G_O, in case of an existing edge no new edge is created, but a dedicated edge attribute, *edgecount*, is increased by one. Thereby the information on the number of connections between clusters in the source graph G remains available in the overview graph G_O.
3. The sub-graphs $G_1, \dots, G_k$ are then defined as follows: For $i = 1, \dots, k$ graph G_i contains all nodes with ID i and all edges, whose end nodes have cluster ID i.

To assign cluster IDs to nodes and edges in the source graph G, users are offered several options in Vanted. Users can

1. Enter a cluster ID for the current selection of graph elements;
2. Copy cluster IDs from the labels of the selected graph elements;
3. Let cluster IDs be determined from connected sub-graphs;
4. Sort graph elements into different clusters each with a distinct cluster ID, based on a given attribute, such as size, colour and position among others.

To make different clusters better distinguishable in the source graph G and in the overview graph G_O, nodes are colour-coded. Additionally, clusters can have their surrounding background also colour-coded. This can be particularly useful for very large clustered graphs. By changing the colour of the clusters, the colour of the corresponding nodes in the source and overview graphs, and the colour of the cluster backgrounds in the source graph, are modified, respectively. See Figure 3 for an example.



Overview Graph of Benson2014 - FAAH inhibitors for treatment of osteoarthritic pain.gml - view 1

Benson2014 - FAAH inhibitors for the treatment of osteoarthritic pain (Source Graph).gml - view 1

Cluster-Coloring

Background-Coloring ☑

Colors

Fill Inner Regions □

Fill View Completely □

Grid (>~20) 50

Low Alpha (0...1) 0,2

Radius 200

Network Attributes

Background Color white

Directed Edges ☑

SBML

Level 2

Model ID MODEL1402030000

Model Name FAAH inhibitors for the treatment of osteoarthriti

SBML Meta ID _5e70a05d-8f46-45b7-ab0e-735f168db142

Version 4

sbml_assignment_rule_1

sbml_assignment_rule_1_assignmnet_variable F_PFM

sbml_assignment_rule_1_function Dose/(ED50+

sbml_assignment_rule_1_meta_id 8cd4-b10ef9fc

sbml_assignment_rule_10

sbml_assignment_rule_10_assignmnet_variable den_b

sbml_assignment_rule_10_function +slag1_b+sla

sbml_assignment_rule_10_meta_id)6a-619243b

sbml_assignment_rule_11

sbml_assignment_rule_11_assignmnet_variable FAAH_D_r

sbml_assignment_rule_11_function +S_r/Km_FA

sbml_assignment_rule_11_meta_id 21-ebbaa9a(

114 nodes 212 edges 100 MB 2 ES

Figure 3: *NetPartVis* uses an overview graph G_O (shown at the top, created from the source graph, shown at the bottom), to influence the placement of clusters in the source graph G. Here, the source graph has been downloaded from the BioModels database (https://www.ebi.ac.uk/biomodels/BIOMD0000000512) using Vanted 2.6.5. In addition to the pre-existing clusters given by BioModels (BRAIN, ROB, PLASMA, MEC and Default), the rest has been determined via the connected sub-graphs feature. Cluster background colouring (set from the Network Attributes Tab on the right) and edge bundling have been applied to the source graph.

2.2 Layout of Graphs

To account for the size of the clusters and the connections between different clusters, the nodes and edges of the overview graph G_O are modified:

1. For all nodes of G_O holds: The size of a node n_i in the overview graph is determined as described in the layout process below.
2. The thickness of an edge in G_O is used to visualise the number of connections between clusters in the source graph G and for each edge of G_O yields: The thickness of an edge is determined by its dedicated edge attribute *edgecount*.

The layout process consists of the following steps, illustrated in Figure 4:

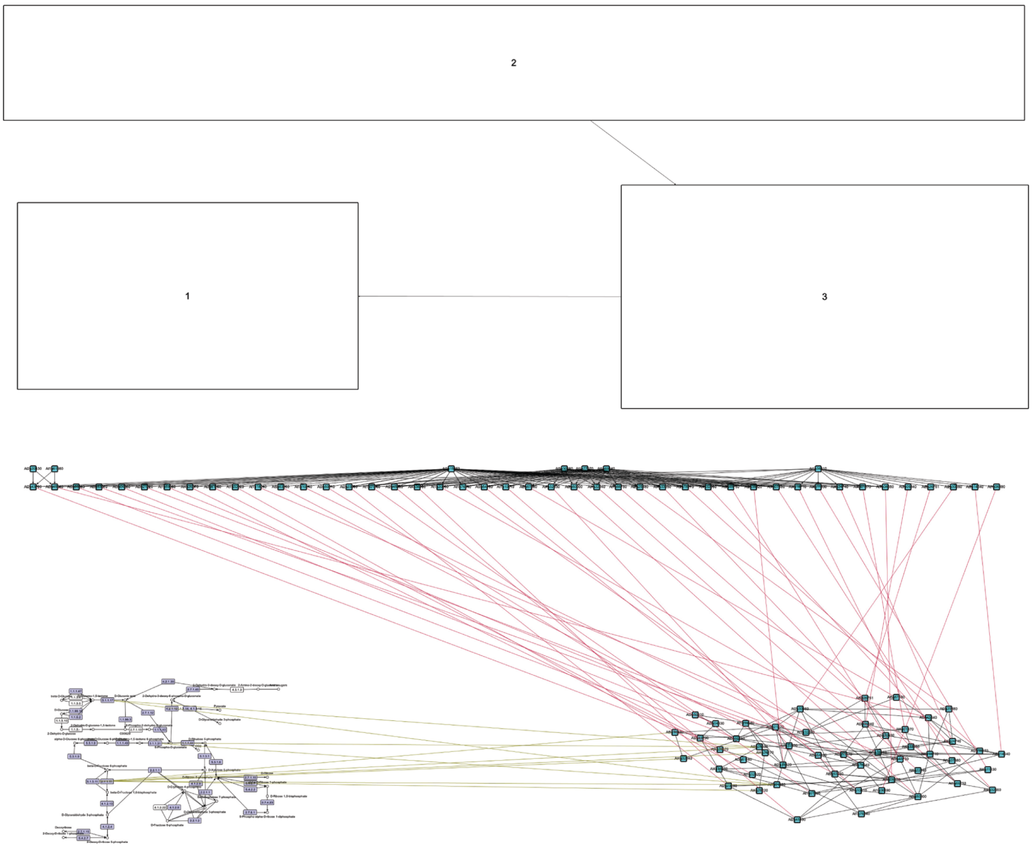


Figure 4: Overview graph (top) and clustered graph (bottom) for heterogeneous networks as an example. In contrast to previous figures where the size of nodes of the overview graph depends on the number of represented nodes, here the overview graph shows the bounding boxes of the layouts of the clustered sub-graphs.

1. Layout sub-graphs $G_1, \dots, G_k$: Any layout method available can be used, alternatively the existing layouts can be kept unmodified to preserve the layout of the clustered sub-graphs. The result of this step is that each sub-graph is laid out. Finally for each graph $G_1, \dots, G_k$ the size of the bounding box is computed, and the size of the nodes $n_1, \dots, n_k$ of G_O is set to the size of the bounding box of the respective graph.

2. Layout overview graph G_O: Any layout method available can be used. As the size of the nodes in G_O represents the bounding box of the clustered sub-graphs, it is advisable that the layout takes node size into account or that the distance between nodes is large enough such that nodes do not overlap.

3. The positions of the nodes $n_1, \dots, n_k$ of G_O are used to layout G. The coordinates of nodes in G are based on the position of the nodes in G_O and an offset based on the layout of the graphs $G_1, \dots, G_k$. For node n with cluster ID i the position is given by the position of n_i of G_O plus the position of n in G_i.

Thanks to the framework implementation of Vanted, *NetPartVis* provides additional features to work with and analyse graphs, such as to select and modify nodes which belong to a given cluster or set of clusters. All discussed methods are available in Vanted (http://www.vanted.org).

3 Conclusion

We presented a method for visualising complex large graphs by graph partitioning and laying out an overview graph and several sub-graphs (partitions) in a coordinated, mental-map preserving way. *NetPartVis* is part of the Vanted system for the analysis and visualisation of experimental data in the context of biological networks. However, Vanted is also a general graph editor which can be used for graphs or networks from many other domains. Networks can be imported using several standards, such as GML [34], GraphML [35], SBGN-ML [36] and others, and be exported in the respective format, or alternatively as images and clickable web pages. Thus, the presented method, as well as its implementation is of broad use for network visualisation and exploration.

Funding

Deutsche Forschungsgemeinschaft (Funder Id: http://dx.doi.org/10.13039/501100001659, SFB-TRR 161).

Conflict of interest statement: Authors state no conflict of interest. All authors have read the journal's publication ethics and publication malpractice statement available at the journal's website and hereby confirm that they comply with all its parts applicable to the present scientific work.

References

[1] Kanehisa M, Goto S, Hattori M, Aoki-Kinoshita KF, Itoh M, Kawashima S, et al. From genomics to chemical genomics: new developments in KEGG. Nucleic Acids Res 2006;34:D354–7.

[2] Le Novère N, Hucka M, Mi H, Moodie S, Schreiber F, Sorokin A, et al. The systems biology graphical notation. Nat Biotechnol 2009;27:735–41.

[3] Czauderna T, Klukas C, Schreiber F. Editing, validating, and translating of SBGN maps. Bioinformatics 2010;26:2340–1.

[4] Czauderna T, Wybrow M, Marriott K, Schreiber F. Conversion of kegg metabolic pathways to sbgn maps including automatic layout. BMC Bioinformatics 2013;14:250.

[5] Bachmaier C, Brandes U, Schreiber F. Biological networks. In: Handbook of graph drawing and visualization. Boca Raton: Taylor & Francis, 2014:621–651.

[6] Di Battista G, Eades P, Tamassia R, et al. Graph drawing: Algorithms for the visualization of graphs. New Jersey: Prentice Hall; 1999.

[7] Kohlbacher O, Schreiber F, Ward MO. Multivariate networks in the life sciences. In: Multivariate network visualization. Cham: Springer, 2014:61–73.

[8] Schreiber F, Dwyer T, Marriott K, Wybrow M. A generic algorithm for layout of biological networks. BMC Bioinformatics 2009;10:375.

[9] Brandes U, Erlebach T. Network analysis: Methodological foundations. volume 3418 of LNCS. Berlin, Heidelberg: Springer, 2005.

[10] Junker BH, Schreiber F. Analysis of biological networks. Wiley series on bioinformatics, computational techniques and engineering. New Jersey: Wiley, 2008.

[11] Eades P, Feng QW. Multilevel visualization of clustered graphs. In: Proceedings Graph Drawing, LNCS. volume 1190, Berlin, Heidelberg: Springer Verlag, 1996;101–112 1190.

[12] Buchsbaum AL, Westbrook JR. Maintaining Hierarchical Graph Views. 11th ACM-SIAM Symposium on Discrete Algorithms; 2000.

[13] Schreiber F, Kerren A, Börner K, Hagen H, Zeckzer D. Heterogeneous networks on multiple levels. In: Multi-variate network visualization. Cham: Springer, LNCS 8380, 2017:175–206.

[14] Quigley A, Eades P. Fade: Graph drawing, clustering, and visual abstraction. In: Graph Drawing 2000, volume 1984 of LNCS. Berlin, Heidelberg: Springer, 2000:197–210.

[15] Wang X, Miyamoto I. Generating customized layouts. In: Brandenburg FJ, ed. Graph Drawing. Berlin Heidelberg: Springer, 1996:504–515.

[16] Eades P, Feng Q-W, Nagamochi H. Drawing clustered graphs on an orthogonal grid. J Graph Algorithms Appl 1997;3:3–29.

[17] Sugiyama K, Misue K. Visualization of structural information: Automatic drawing of compound digraphs. IEEE Trans Syst Man Cybern Syst 1991;21:876–92.

[18] Feng Q-W, Cohen R F, Eades P. How to draw a planar clustered graph. In: Du D-Z, ed. Proceedings International Conference on Computing and Combinatorics (COCOON'95), volume 959 of LNCS. Berlin, Heidelberg: Springer Verlag, 1995:21–30.

[19] Chimani M, Gutwenger C, Jansen M, et al. Computing maximum c-planar subgraphs. In: Tollis IG, Patrignani M, eds. Graph Drawing. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009:114–120.

[20] Di Battista G, Didimo W, Marcandalli A. Planarization of clustered graphs. In: Mutzel P, Jünger M, Leipert S, eds. Graph Drawing. Berlin, Heidelberg: Springer Berlin Heidelberg, 2002:60–74.

[21] Cortese PF, Patrignani M. Clustered planarity = flat clustered planarity. In: Biedl T, Kerren A, eds. Graph Drawing and Network Visualization. Cham: Springer International Publishing, 2018:23–38.

[22] Chimani M, Di Battista G, Frati F, Klein K. Advances on testing c-planarity of embedded flat clustered graphs. Int J Found Comput Sci 2019;30:197–230.

[23] Dogrusoz U, Giral E, Cetintas A, et al. A compound graph layout algorithm for biological pathways. In: Pach J, ed. Proceedings International Symposium on Graph Drawing (GD'04), LNCS, 2004:442–447.

[24] Brandes U, Dwyer T, Schreiber F. Visual triangulation of network-based phylogenetic trees. In: Deussen O, Hansen C, Keim D, Saupe D, eds. Proceedings Joint Eurographics – IEEE TCVG Symposium on Visualization (VisSym'04). Eurographics Association, 2004:75–84.

[25] Misue K, Eades P, Lai W, Sugiyama K. Layout adjustment and the mental map. J Visual Lang Comput 1995;6:183–210.

[26] Junker BH, Klukas C, Schreiber F. VANTED: A system for advanced data analysis and visualization in the context of biological networks. BMC Bioinformatics. 2006;7:109.

[27] Gräßler J, Koschützki D, Schreiber F. Centilib: comprehensive analysis and exploration of network centralities. Bioinformatics 2012;28:1178–9.

[28] Rohn H, Hartmann A, Junker A, Junker BH, Schreiber F. Fluxmap: a vanted add-on for the visual exploration of flux distributions in biological networks. BMC Systems Biology 2012;6:33.1–9.

[29] Moodie SL, Le Novère N, Demir E, Mi H, Villéger A. Systems biology graphical notation: Process description language level 1 version 1.3. J Integr Bioinform 2015;12:263.

[30] Sorokin AA, Le Novère N, Luna A, Czauderna T, Demir E, Haw R, et al. Systems biology graphical notation: Entity relationship language level 1 version 2. J Integr Bioinform 2015;12:264.

[31] Mi H, Schreiber F, Moodie S, Czauderna T, Demir E, Haw R, et al. Systems biology graphical notation: activity flow language level 1 version 1.2. J Integr Bioinform 2015;12:e265.

[32] Junker A, Rohn H, Czauderna T, Klukas C, Hartmann A, Schreiber F. Creating interactive, web-based and data-enriched maps using the systems biology graphical notation. Nat Protoc 2012;7:579–93.

[33] Rohn H, Junker A, Hartmann A, Grafahrend-Belau E, Treutler H, Klapperstück M, et al. VANTED v2: A framework for systems biology applications. BMC Systems Biology 2012;6:139.

[34] Himsolt M. GML: A portable graph file format. Technical report, University of Passau, 1996.

[35] Brandes U, Eiglsperger M, Lerner J, et al. Graph markup language (GraphML). In: Tamassia R, ed. Handbook of graph drawing visualization, discrete mathematics and its applications. Boca Raton: CRC Press, 2013:517–541.

[36] Van Iersel MP, Villéger AC, Czauderna T, Boyd SE, Bergmann FT, Luna A, et al. Software support for sbgn maps: Sbgn-ml and libsbgn. Bioinformatics 2012;28:2016–21.