

The Network Analysis Profiler (NAP v2.0): a web tool for visual topological comparison between multiple networks

Mikaela Koutrouli¹, Theodosios Theodosiou², Ioannis Iliopoulos², Georgios Pavlopoulos¹✉

¹Institute for Fundamental Biomedical Research, BSRC “Alexander, Fleming”, Vari, Greece

²University of Crete, School of Medicine, Department of Basic Sciences, Crete, Greece

Competing interests: MK none; TT none; LP none; II none; GP none

Abstract

In this article we present the Network Analysis Profiler (NAP v2.0), a web tool to directly compare the topological features of multiple networks simultaneously. NAP is written in R and Shiny and currently offers both 2D and 3D network visualisation, as well as simultaneous visual comparisons of node- and edge-based topological features as bar charts or scatterplot matrix. NAP is fully interactive, and users can easily export and visualise the intersection between any pair of networks using Venn diagrams or a 2D and a 3D multi-layer graph-based visualisation. NAP supports weighted, unweighted, directed, undirected and bipartite graphs.

Introduction

Networks are key representations that can capture the associations and interactions between any kind of bioentity such as genes, proteins, diseases, drugs, small molecules and others (Pavlopoulos *et al.*, 2011, 2015; Pavlopoulos, Wegener, *et al.*, 2008; Pavlopoulos *et al.*, 2013; Kontou *et al.*, 2016; Koutrouli *et al.*, 2020). Gene co-expression networks, gene regulatory networks, protein-protein interaction networks (PPIs), signal transduction networks, metabolic networks, gene-disease networks, sequence similarity networks, phylogenetic networks, ecological networks, epidemiological networks, drug-disease networks, disease-symptom networks, literature co-occurrence networks, food webs, semantic and knowledge networks are the most widely known in the biomedical and biomedical-related fields (Koutrouli *et al.*, 2020). However, not all networks are the same in terms of structure and come with certain topological features. For example, Erdos–Rényi networks are random graphs with no specific structure (Bollobás, 2001), Watts–Strogatz networks are random graphs with small communities (Watts and Strogatz, 1998), and Barabási–Albert networks are random scale-free networks whose degree distribution follows a power law (Barabási and Albert, 1999). While basic topological network analysis is offered by widely used network visualisation applications (Pavlopoulos, Wegener, *et al.*, 2008) such as the Cytoscape (Shannon *et al.*, 2003) and

Gephi (Bastian *et al.*, 2009), in this article we present the Network Analysis Profiler (NAP v2.0), a complementary web-based tool designed to fill certain gaps and aid non-experts in not only analysing the topological features of a network, but also to visually perform direct comparisons across multiple network in an easy and user-friendly way.

The application

In its current version, NAP (Theodosiou *et al.*, 2017) supports weighted, unweighted, directed, undirected, simple and bipartite networks. It is implemented in R and Shiny and most of its backend calculations are based on the igraph library (Csardi and Nepusz, 2006). It accepts as input a tab-delimited file in which the first two columns indicate the connections between the nodes and the third column the weight between these edges. Users have the option to upload as many networks as they like, name them accordingly, and process them simultaneously. NAP v2.0 has four main functions: i) Basic Visualisation, ii) Topological analysis, iii) Node/Edge ranking and iv) Intersection network hosting the common vertices and edges between two selected networks.

Basic 2D/3D visualisation

Once a network has been uploaded and named, it is visualised with the use of visNetwork library. VisNetwork

Article history

Received: 19 August 2020

Accepted: 16 September 2020

Published: 2021

© 2021 Koutrouli *et al.*; the authors have retained copyright and granted the Journal right of first publication; the work has been simultaneously released under a Creative Commons Attribution Licence, which allows others to share the work, while acknowledging the original authorship and initial publication in this Journal. The full licence notice is available at <http://journal.embnet.org>.

offers a fully interactive visualisation as it allows network zooming, dragging, and panning. Nodes can be selected and placed anywhere on the plane, whereas the first neighbors of any node can be highlighted upon selection. This network view can show one network at a time and is automatically updated when a different network is selected. In this view, NAP supports the following igraph layouts (Pavlopoulos *et al.*, 2017):

- Fruchterman-Reingold (Fruchterman and Reingold, 1991): it places nodes on the plane using a force-directed layout.
- Random: it places the vertices on a 2D plane uniformly using random coordinates.
- Circle: it places vertices on a circle, ordered by name.
- Kamada-Kawai (Kamada and Kawai, 1989): it places the vertices on a 2D plane by simulating a physical model of springs.
- Reingold-Tilford (Reingold and Tilford, 1981): this is a tree-like layout and is suitable for trees, ontologies and hierarchies.
- LGL (Adai *et al.*, 2004): a force-directed layout suitable for larger graphs.
- Grid: this layout places vertices on a rectangular 2D grid.
- Sphere: this layout places vertices on a rectangular 3D-like sphere.

In addition to the 2D visualisation, NAP offers a fully interactive 3D network visualisation using a force-directed layout. Users can zoom-in and out and interactively drag and drop a node or the whole network and place it anywhere in space. This visualisation is based on the D3.js library and is sufficient for larger graphs, especially when the 2D view becomes overcrowded. An

example of a Yeast PPI (Gavin *et al.*, 2002) is shown in Figure 1.

The topological features

In its current version, NAP supports the following igraph-based topological features:

- Number of nodes: the number of vertices in the network.
- Diameter: the length of the longest geodesic. The diameter is calculated by using a breadth-first search like method. The graph-theoretic or geodesic distance between two points is defined as the length of the shortest path between them.
- Radius: the eccentricity of a vertex is its shortest path distance from the farthest other node in the graph. The smallest eccentricity in a graph is called its radius.
- Density: the density of a graph is the ratio of the number of edges divided by the number of possible edges.
- Average path length: the average number of steps needed to go from a node to any other.
- Clustering Coefficient: a metric to show if the network has the tendency to form clusters.
- Modularity: this function calculates how modular is a given division of a graph into subgraphs.
- Number of self-loops: the number of nodes connected to themselves.
- Average Eccentricity: the eccentricity of a vertex is its shortest path distance from the farthest other node in the graph.
- Average Eigenvector Centrality: the influence of a node in a network.

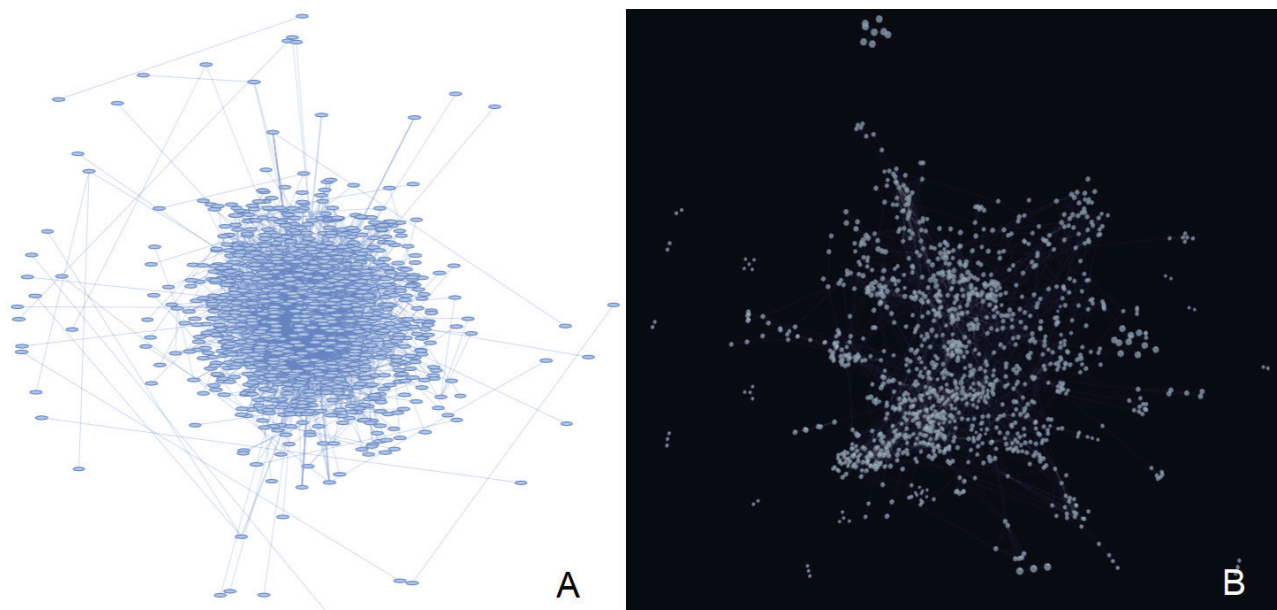


Figure 1. NAP's basic visualisation.

A) 2D visualisation of a Yeast PPI (Gavin *et al.*, 2002) using the Kamada-Kawai layout. **B)** The same network visualised in 3D.

- Assortativity degree: the assortativity coefficient is positive if similar vertices (based on some external property) tend to connect or negative otherwise.
 - Directed acyclic graph: it shows if a graph has cycles or not.
 - Average number of Neighbors: the number of neighbors each node of the network has on average.
 - Centralization betweenness: an indicator of a node's centrality in a network. It is equal to the number of shortest paths from all vertices to all others that pass through that node. Betweenness centrality quantifies the number of times a node acts as a bridge along the shortest path between two other nodes.
 - Centralization closeness: measures the speed with which randomly walking messages reach a vertex from elsewhere in the graph.
 - Centralization degree: defined as the number of links incident upon a node.
 - Graph mincut: calculates the minimum st-cut between two vertices in a graph. The minimum st-cut between source and target is the minimum total weight of edges needed to remove to eliminate all paths from source to target.
 - Motifs-3: searches a graph for motifs of size 3 (Koutrouli *et al.*, 2020).
 - Motifs-4: searches a graph for motifs of size 4 (Koutrouli *et al.*, 2020).
- While users can select and visualise each topological measure in a numeric form, one can select several of the uploaded networks and directly compare their topological features in different bar charts. Figure 2 shows an example of a direct comparison between two Yeast PPI networks (Gavin *et al.*, 2002, 2006) (generated in 2002 and 2006 respectively) and a random scale-free Albert-Barabasi network consisting of 1000 nodes (generated by NAP's automatic network generators). Bar charts are fully interactive and are produced with the use of the plotly library.

Topological feature pairwise comparison and node/edge ranking

As explained in NAP's v1.0 article (Theodosiou *et al.*, 2017), nodes can be ranked by centralisation degree, centralisation betweenness, clustering coefficient, page rank, eccentricity, eigenvector and subgraph centrality, whereas edges can be ranked by betweenness centrality only. An all-versus-all scatterplot matrix can be generated to show the pairwise correlations between any of the selected topological features (Figure 3). The upper



Figure 2. Direct comparison of fourteen topological features across three different networks.

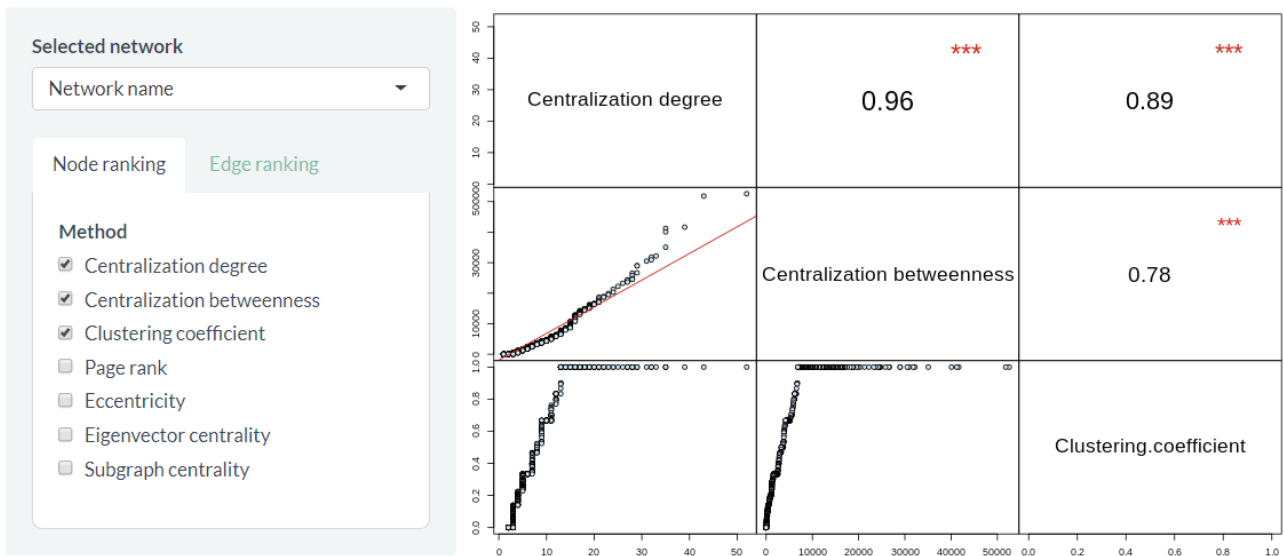


Figure 3. Intra-network pairwise topological feature comparison.

part of the matrix shows the correlation between any pair of features in a numerical form, whereas its lower part shows these correlations in a scatterplot. If only one option has been selected, the viewer will generate a chart showing the values of the selected topological feature in a histogram.

Network intersection

With NAP, users can automatically extract, export, and visualise the common edges and nodes between any selected pair of networks. Common node and edge names will be initially reported in interactive tables as text. In contrast, Venn diagrams are used to show the node/edge union and intersection between the two networks. Venn

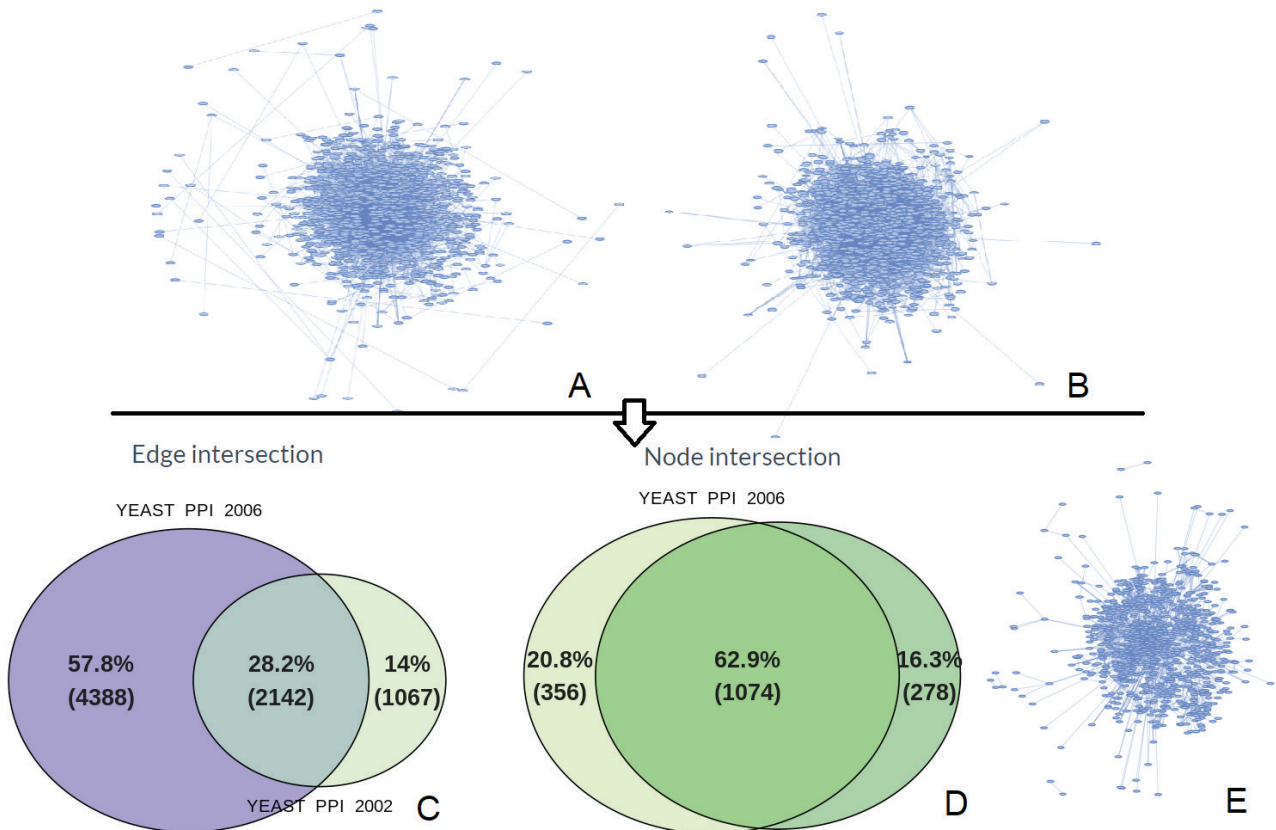


Figure 4. Automatic generation of common edges and nodes between two selected networks. **A)** A Yeast PPI network generated in 2002 (Gavin *et al.*, 2002). **B)** A Yeast PPI network generated in 2006 (Gavin *et al.*, 2006). **C)** 2142 common edges shown in a Venn diagram. **D)** 1074 common nodes shown in a Venn diagram. **E)** The generated network consisting of the common edges between the 2002 and 2006 yeast PPI networks.

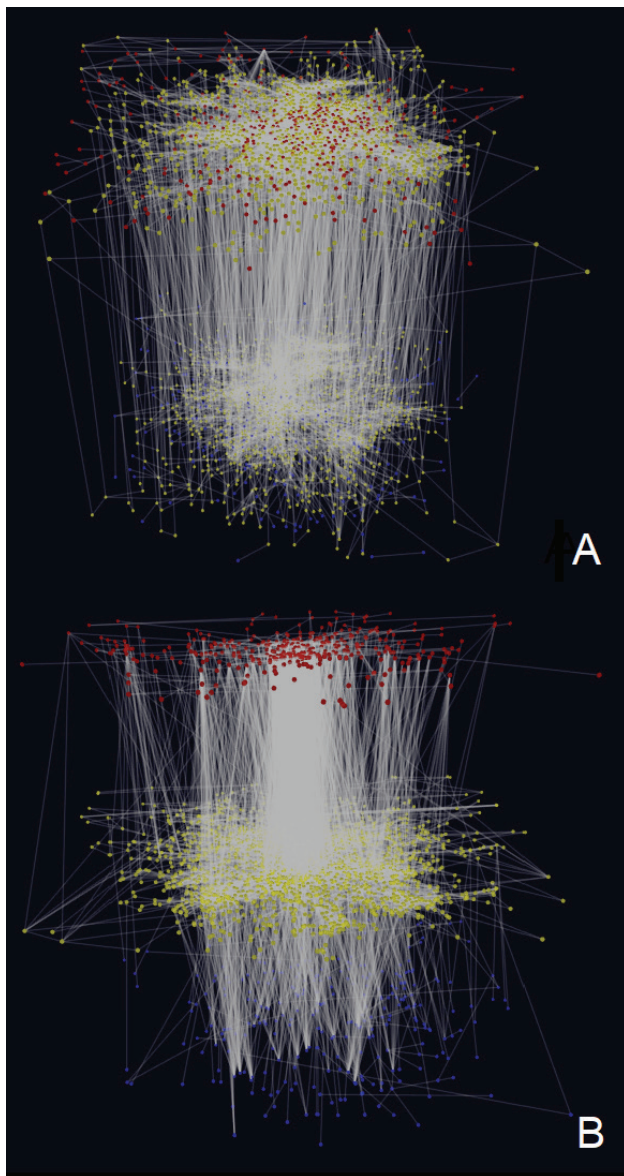


Figure 5. Visualisation of common nodes between two networks using a 3D multi-layer graph. A) Two Yeast PPI networks (2002 and 2006 accordingly) are placed in two different layers and their common nodes are marked yellow. B) Alternatively, common nodes are separated and placed on a third middle layer.

diagrams are generated with the use of Venn diagrams library whereas VisNetwork library is used to visualise the network's intersection in an interactive 2D view. In Figure 4, a comparison between two Yeast PPI networks (Gavin *et al.*, 2002, 2006), generated in 2002 and 2006 respectively is shown.

In addition to the 2D view, NAP gives the option to visualise the common parts between two selected networks using a 3D multi-layer graph implemented in D3.js. Nodes of the first network are placed on a layer and are colored in blue, whereas nodes from the second network are placed on a different layer and are colored in red. Nodes that belong to the two different networks but

have the same name are considered as common and are colored in yellow, whereas edges are drawn to connect these nodes across the two layers. Notably, users can use a 3-layer representation to place the common nodes on a third middle layer for a more comprehensive view (not always better).

To minimize the crossovers between the lines across layers, a layout can be initially applied on the whole network and nodes can be separated on their two distinct layers upon completion by adjusting their height coordinate. The layouts that are currently supported by NAP for this view are the random, circular, fruchterman-reingold, fruchterman-reingold grid, kamada-kawai, spring, and LGL force-directed algorithms.

The multi-layer 3D graph is fully interactive, and users can zoom in/out and drag and rotate each node or the whole network in 3D space for easier exploration. In addition, users can export the network in a text file in order to be processed by more advanced third-party 3D visualisers like, for example, Arena3D (Pavlopoulos *et al.*, 2008; Secrier *et al.*, 2012). The whole concept is schematically shown in Figure 5.

Availability

NAP is available at bib.fleming.gr:3838/NAP/¹ and its code can be found at github.com/PavlopoulosLab/NAP/²

Key Points

- Exploration of network topological features.
- Node/edge ranking by topological properties.
- Simultaneous comparison of topological features between several networks.
- Distribution plotting of any topological feature.
- Usage of various layouts to visualise a network in using interactive 2D/3D views.

Acknowledgments

GAP and MK were supported by the Hellenic Foundation for Research and Innovation (H.F.R.I) under the “First Call for H.F.R.I Research Projects to support Faculty members and Researchers and the procurement of high-cost research equipment grant”, GrantID: 1855-BOLOGNA.

References

1. Adai AT, Date SV, Wieland S, Marcotte EM (2004) LGL: creating a map of protein function with an algorithm for visualizing very large biological networks. *J. Mol. Biol.* **340** (1), 179–190. <http://dx.doi.org/10.1016/j.jmb.2004.04.047>.
2. Barabási A-L and Albert R (1999) Emergence of Scaling in Random Networks. *Science* **286** (5439), 509–512. <http://dx.doi.org/10.1126/science.286.5439.509>.
3. Bastian M, Heymann S, Jacomy M (2009) Gephi: An Open Source Software for Exploring and Manipulating Networks.
4. Bollobás B (2001) *Random graphs* Cambridge University Press, Cambridge ; New York, 2nd ed.

¹<http://bib.fleming.gr:3838/NAP/>

²<https://github.com/PavlopoulosLab/NAP/>

5. Fruchterman TMJ and Reingold EM (1991) Graph drawing by force-directed placement. *Softw. Pract. Exp.* **21** (11), 1129–1164. <http://dx.doi.org/10.1002/spe.4380211102>
6. Csardi G and Nepusz T (2006) The igraph software package for complex network research. *InterJournal Complex Systems*, 1695.
7. Gavin A-C, Aloy P, Grandi P, Krause R, Boesche M, *et al.* (2006) Proteome survey reveals modularity of the yeast cell machinery. *Nature* **440** (7084), 631–636. <http://dx.doi.org/10.1038/nature04532>
8. Gavin A-C, Bösch M, Krause R, Grandi P, Marzioch M, *et al.* (2002) Functional organization of the yeast proteome by systematic analysis of protein complexes. *Nature* **415** (6868), 141–147. <http://dx.doi.org/10.1038/415141a>
9. Kamada T and Kawai S (1989) An algorithm for drawing general undirected graphs. *Inf. Process. Lett.* **31** (1), 7–15. [http://dx.doi.org/10.1016/0020-0190\(89\)90102-6](http://dx.doi.org/10.1016/0020-0190(89)90102-6)
10. Kontou PI, Pavlopoulou A, Dimou NL, Pavlopoulos GA, and Bagos PG (2016) Network analysis of genes and their association with diseases. *Gene* **590** (1), 68–78. <http://dx.doi.org/10.1016/j.gene.2016.05.044>
11. Koutrouli M, Karatzas E, Paez-Espino D, Pavlopoulos GA (2020) A Guide to Conquer the Biological Network Era Using Graph Theory. *Front. Bioeng. Biotechnol.* **8**, 34. <https://doi.org/10.3389/fbioe.2020.00034>
12. Pavlopoulos GA, Iacucci E, Iliopoulos I, Bagos P (2013) Interpreting the Omics 'era' Data. In: Tsihrintzis GA, Virvou M, and Jain LC (eds) *Multimedia Services in Intelligent Environments*. Springer International Publishing, Heidelberg, Heidelberg, Vol 25, pp. 79–100
13. Pavlopoulos GA, Malliarakis D, Papanikolaou N, Theodosiou T, Enright AJ, *et al.* (2015) Visualizing genome and systems biology: technologies, tools, implementation techniques and trends, past, present and future. *GigaScience* **4**, 38. <http://dx.doi.org/10.1186/s13742-015-0077-2>
14. Pavlopoulos GA, O'Donoghue SI, Satagopam VP, Soldatos TG, Pafilis E, *et al.* (2008) Arena3D: visualization of biological networks in 3D. *BMC Syst. Biol.* **2**, 104. <http://dx.doi.org/10.1186/1752-0509-2-104>
15. Pavlopoulos GA, Paez-Espino D, Kyrpides NC, Iliopoulos I (2017) Empirical Comparison of Visualization Tools for Larger-Scale Network Analysis. *Adv. Bioinforma.* 2017, 1278932. <http://dx.doi.org/10.1155/2017/1278932>
16. Pavlopoulos GA, Secrier M, Moschopoulos CN, Soldatos TG, Kossida S, *et al.* (2011) Using graph theory to analyze biological networks. *BioData Min.* **4**, 10. <http://dx.doi.org/10.1186/1756-0381-4-10>
17. Pavlopoulos GA, Wegener A-L, Schneider R (2008) A survey of visualization tools for biological network analysis. *BioData Min.* **1**, 12. <http://dx.doi.org/10.1186/1756-0381-1-12>
18. Reingold EM and Tilford JS (1981) Tidier Drawings of Trees. *IEEE Trans. Softw. Eng.* SE-7 (2), 223–228. <http://dx.doi.org/10.1109/TSE.1981.234519>
19. Secrier M, Pavlopoulos GA, Aerts J, Schneider R (2012) Arena3D: visualizing time-driven phenotypic differences in biological systems. *BMC Bioinformatics* **13**, 45. <http://dx.doi.org/10.1186/1471-2105-13-45>
20. Shannon P, Markiel A, Ozier O, Baliga NS, Wang JT, *et al.* (2003) Cytoscape: a software environment for integrated models of biomolecular interaction networks. *Genome Res.* **13** (11), 2498–2504. <http://dx.doi.org/10.1101/gr.1239303>
21. Theodosiou T, Efstathiou G, Papanikolaou N, Kyrpides NC, Bagos PG, *et al.* (2017) NAP: The Network Analysis Profiler, a web tool for easier topological analysis and comparison of medium-scale biological networks. *BMC Res. Notes* **10** (1), 278. <http://dx.doi.org/10.1186/s13104-017-2607-8>
22. Watts DJ and Strogatz SH (1998) Collective dynamics of 'small-world' networks. *Nature* **393** (6684), 440–442. <http://dx.doi.org/10.1038/30918>