Imperial College London
Department of Computing

Analysing Directed Network Data

Anida Sarajlić

December 2015

Supervisor: Dr. Nataša Pržulj

Submitted in part fulfilment of the requirements for the degree of
Doctor of Philosophy in Computing of Imperial College London
and the Diploma of Imperial College London

Declaration of Originality

I herewith certify that that the original work submitted in this dissertation is my own and that all material in this dissertation which is not my own work has been properly acknowledged.

Anida Sarajlić

Copyright Declaration

Abstract

The topology of undirected biological networks, such as protein-protein interaction networks, or genetic interaction networks, has been extensively explored in search of new biological knowledge. Graphlets, small connected non-isomorphic induced sub-graphs of an undirected network, have been particularly useful in computational network biology. Having in mind that a significant portion of biological networks, such as metabolic networks or transcriptional regulatory networks, are directed by nature, we define all up to four node directed graphlets and orbits and implement the directed graphlet and graphlet orbits counting algorithm. We generalise all existing graphlet based measures to the directed case, defining: relative directed graphlet frequency distance, directed graphlet degree distribution similarity, directed graphlet degree vector similarity, and directed graphlet correlation distance. We apply new topological measures to metabolic networks and show that the topology of directed biological networks is correlated with biological function. Finally, we look for topology–function relationships in metabolic networks that are conserved across different species.

I dedicate this dissertation to my parents and my brother for all their love and support.

Acknowledgements

I wish to thank my supervisor, Dr. Nataša Pržulj, for giving me the opportunity to join her research group at Imperial College London and to work in this exciting research environment. I am grateful for Dr. Pržulj's guidance and support on my research, as well as for her passion for science, motivating me to always strive for more. With this guidance, I developed research skills which I hope to always be proud of. I also wish to thank Dr. Nöel Malod-Dognin, Dr. Pržulj's post-doctoral researcher, for all insightful discussions and guidance during my work on this dissertation. Also, I would like to thank members of my viva examination committee, Prof. Dr. Jan Baumbach and Prof. Dr. Murray Shanahan, for all the useful comments which led to the improved final version of this manuscript.

I am especially grateful to all former and current fellow PhD students: Dr. Vuk Janjić, Vladimir Gligorijević, Dr. Ömer Nebil Yaverŏglu and Dr. Kai Sun, for genuine friendship, fruitful discussions and all their support.

I sincerely thank Prof. Djordje Radak, Prof. Charles Coombes and Dr. Aleksandra Filipović for the collaborative work on several journal publications during my research as a member of Dr. Pržulj's group.

I acknowledge European Research Council (ERC) Starting Independent Researcher Grant 278212 (2012-2017) for the funding of my studies.

Finally, I am very grateful for my dearest, my family, their understanding, love and support.

Contents

List of Tables

List of Figures

1 Introduction

1.1 Motivation

A *network* (also called a *graph*) is a common model of a set of objects and their interactions for describing and analysing data in numerous research areas. Different graph theoretic approaches are used for analysing network data.

Computational network biology is an emerging research area that complements traditional biology and medicine. Computational analysis on ever-growing amounts of available biological data offers new insight into research on the development of living organisms, research on human diseases and the discovery of new drug targets. A number of large-scale data sets were generated as a result of recent advances in high-throughput techniques. These molecular data include information on interactions between biological macromolecules, such as protein–protein interactions (PPI), genetic interactions (GI), enzyme-substrate relationships and pathway maps. The concept of networks has been introduced in systems biology as it accurately captures the inner workings of many complex biological systems and reduces the complexity of biological data that is required for performing computational analyses. Also, the fact that a specific network topology comes as a direct consequence of biological processes occurring between the elements of the underlying system, highlights the importance of the topology as a valuable source of new biological knowledge. Graph theoretic approaches help to identify topological properties which differ from the wiring which can be expected at random, revealing the connection between a specific topological characteristic and a related biological function or phenotype, such as disease.

The majority of current publicly available biological networks are undirected networks. For example, PPI networks, where nodes correspond to proteins and edges are placed between two proteins if they physically interact, are networks with a highly explored topology. It was shown that proteins which are close in the PPI network are more likely to perform the same function [1] which was used for inferring functions of unannotated proteins: the direct neighbourhoods of proteins [1], n–neighbourhoods of proteins [2], and shared neighbours of proteins [3] were examined looking for the most

common functions among annotated direct neighbours. Several other methods have shown that PPI network topology around proteins is a predictor of their function or their involvement in a disease [4–6].

Graphlets, small connected non-isomorphic induced sub-graphs of an undirected network, first introduced by Pržulj *et al.* [7], have been particularly useful: the local topology around a protein in a PPI network was summarised into a topological "signature" of a protein – *graphlet degree vector (GDV)* [4] and the similarity of these protein "signatures" is a good indicator that proteins belong to the same protein complex, perform similar biological functions, are coexpressed, involved in the same diseases, and are part of the same sub–cellular components [4]. This topological measure of similarity was used for predicting new melanogenesis related genes that were phenotypically validated [5] and for identifying key cardiovascular disease genes [8]. In addition, Gligorijević *et al.* [9] developed an integrative model for gene ontology (GO) reconstruction and gene function prediction. They show that the GDV similarity between nodes contains complementary information to the connectivity patterns, and integrating these two sources of information boosts the quality of the integrative model by increasing the performance of gene function and GO term association predictions.

In recent years, there is a growing trend in completing and exploring biological networks that are directed by nature, such as transcriptional regulatory networks, metabolic networks, and effective connectivity brain networks [10–12]. Various network properties and measures exist for topological analysis of directed networks, such as properties based on nodes' degrees or network spectra. However, directed graphlets are still not defined. In this dissertation, we define directed graphlets and directed graphlet-based heuristics for analysis of directed networks. Furthermore, we show that our measures outperform common existing measures for directed networks comparison. Finally, we apply these new topological measures to show that topology of directed metabolic networks is correlated to biological information.

As was the case with the undirected graphlets [13], application of directed graphlets and derived measures is certainly not limited to computational biology. For instance, sociology, economy and technology are just some of the many research areas in which the underlying complex interactions can be modelled using directed networks: social interaction networks, world trade networks, citation networks, autonomous system networks etc. The set of new measures that we propose in this dissertation opens up a window of opportunities for exploring these research areas from a new perspective.

1.2 Dissertation Outline

In the remainder of this section, we introduce undirected and directed networks and and give an overview of graph-theoretic measures for local network topology analysis, network comparison and network modelling. In addition we provide a brief description of different types of biological networks, as all applications presented in Chapter 4 of this dissertation involve biological networks.

We use Chapter 2, to show that it is possible to tackle open problems in biology and medicine using graph theoretic approaches on undirected biological networks. We first give a short review of network-based approaches in research on complex diseases, in particular cardiovascular diseases (CVD), as they are a leading health problem worldwide [14]. We then present two case studies where we analyse CVDs using undirected biological networks: (1) In the first study, we apply graphlet-based measures to the human PPI network to identify key genes involved in CVDs.(2) In the second study, we use human PPI and genetic networks to explore the reasons behind the protective role of diabetes against the development of aneurysm. Using a topological measure of brokerage - a measure that identifies "weak" points in a network, we find kinases that, conditioned with diabetes pathways, can influence disruption of pathways responsible for the development of aneurysm.

Motivated by the growing amounts of available directed biological network data, and the usefulness of topological analysis in biological research described in Chapters 1 and 2, in Chapter 3 we introduce our new methodology for the analysis of directed networks. We define directed graphlets and generalise the following undirected graphlet-based measures to the directed case: relative graphlet frequency distance, graphlet degree distribution similarity, graphlet degree vector similarity, and graphlet correlation distance. Using synthetic networks and model network clustering, we then show that directed graphlet-based measures outperform commonly used measures for comparison of directed networks. In addition, in case of directed networks without anti-parallel pairs of arcs, we find orbits that are redundant among up-to-four node graphlets.

In Chapter 4, we demonstrate the use of the new network measures introduced in Chapter 3 by applying them to directed metabolic networks. Namely, we evaluate the quality of topology-based clustering of metabolic networks of eukaryotic species according to their taxonomic classification and confirm that graphlet-based measures outperform other common measures for directed network comparison. Further on, we show that similar local topologies around genes in the human metabolic network correspond to similar biological functions. Motivated by this finding we use directed graphlets to

further explore metabolic networks of several eukaryotic species and find conserved relationships between topology around genes and their biological functions across different species. We also show the predictive power of a gene's directed graphlet signature in a metabolic network for annotating the gene with biological functions.

Finally, in Chapter 5, we conclude the dissertation with a summary of our contributions, and discuss future work.

1.3 Networks and Network Properties

A network, also called a graph, is a mathematical object denoted as a pair $G = \{V, E\}$, where V is a set of vertices (nodes) and E is a set of edges that connect pairs of nodes according to some relationship between them [15]. In an *undirected* graph, edges are unordered pairs of vertices. In a *directed* graph, edges are ordered pairs of vertices, often called arcs, directed edges, or arrows. A directed edge or arc $e = (x, y)$ is considered to be directed from x to y, where y is called the head and x is called the tail of the arc, y is a direct successor of x, and x is a direct predecessor of y.

A graph can be represented as the $|V| \times |V|$ dimensional adjacency matrix A. In an undirected graph, the entry A_{ij} from matrix A takes a non-zero value or a zero value if the nodes i and j are connected with an edge or not, respectively [15,16]. In a directed graph, A_{ij} is the number of arcs from node i to node j. In a weighted graph the values in the matrix can be used to represent the edge weights. A graph can also be presented in the format of an adjacency list: it is a $|V| \times 2$ dimensional array representing nodes in the network, with each node linked to a list of nodes that it is connected to. In case of a weighted network an additional list of edge weights is necessary for each node.

The choice of network representation depends on computing requirements and the type of the network. For example, for more sparse networks the adjacency list is more memory efficient then the adjacency matrix. Also operations of adding or deleting nodes from a network have a high computational cost in case of adjacency matrix because the size of the matrix changes and it needs to be allocated again. However, operations on edges are more computationally efficient if performed on adjacency matrices because they require only a change in the value of the existing matrix elements. Another advantage of matrix representation is that additional network information can be encoded in the matrix. For example, the *Laplacian matrix* of a network contains information on nodes degrees in the diagonal elements. It is calculated as the difference between: (1)the matrix containing only nodes' degrees as the diagonal elements and (2)the adjacency matrix.

Depending on the type of edges - whether they have an orientation or not - networks (graphs) can be directed, undirected, or mixed. A network is weighted if values are assigned to network edges. Note that in the case of an undirected graph the adjacency matrix is symmetric, whereas in the case of a directed or mixed graph it is not. Here, we list common network concepts:

- *Multiple edges*, also called parallel edges, are two or more edges with the same pair of endpoints. In directed networks, multiple edges are edges with the same ordered pair of endpoints [17].
- A *loop* is an edge whose endpoints are equal.
- A *simple graph* is an unweighed graph containing no loops or multiple edges. A directed graph is simple if each ordered pair of vertices is the head and tail of at most one edge [17].
- A *neighbourhood* of node i is a set of nodes adjacent to node i.
- A *path* is a simple graph whose vertices can be ordered so that two vertices are adjacent if and only if they are consecutive in the list [17]. A directed path is a simple directed graph whose vertices can be linearly ordered so there is an edge with tail u and head v if and only if v immediately follows u in the vertex ordering [17].
- *The shortest path* between nodes, also called a *geodesic path* [18], is such that no shorter path between these nodes exists in the network.
- A graph is *connected* if each pair of vertices in the graph belongs to a path, otherwise, the graph is disconnected [17]. A directed graph is *weakly connected* if its underlying undirected graph is connected. A directed graph is *strongly connected* if for each ordered pair (u, v) of vertices, there is a directed path from u to v [17].
- A *cycle* is a path that starts and ends in the same node.
- An *arc(x,y) inverted* in a directed graph is $arc(y, x)$.
- An *anti - parallel pair of arcs* is a pair of arcs such that one's head/tail is the other's tail/head (*e.g.* arcs (x, y) and (y, x)).
- A *sub-graph* of graph $G(V, E)$ is a graph $G'(V', E')$ such that $V' \subseteq V$ and $E' \subseteq E$.

- An *induced sub-graph* of graph $G(V, E)$ is a sub-graph $G'(V', E')$ such that it contains all edges in G between vertices V'.

- A *partial sub-graph* of graph $G(V, E)$ is a sub-graph $G'(V', E')$ such that it does not contain all edges in G between vertices V'.

- An *isomorphism* of graphs G and H is a bijection f between the nodes of G and H such that any two vertices i and j from G are adjacent in G if and only if $f(i)$ and $f(j)$ are adjacent in H.

- A *sub-graph isomorphism problem* is a task where for given networks G and H, it has to be determined whether graph G contains a sub-graph that is isomorphic to H. This problem is NP-complete [19] which means that there are no polynomial time exact solutions. NP denotes a set of all decision problems whose solutions can be verified in polynomial time. P denotes a set of all decision problems whose solutions can be found in polynomial time. A problem p is NP-complete if it is in NP and if every problem in NP is reducible to p in polynomial time [20]. Note that there is still no proof whether NP-complete problems are solvable in polynomial time ($P = NP$) and this is one of the great unsolved problems of mathematics [21, 22]. In the case that $P = NP$, the sub-graph isomorphism problem would be solved in polynomial time.

Below are listed the properties which summarise the topological characteristics of networks. First we address the *global network properties*, which give an overview of the network with respect to all its nodes and edges. Then we address the *local network properties*, which describe the network topology using sub-graphs, namely motifs [23] or graphlets [7]. We also list all network distance measures that are based on the discussed properties. The distance measures address the network topology comparison problem and quantify the topological correspondence between the two networks or between the local topologies around nodes in the network.

Note that in this dissertation we do not address the network alignment problem and thus we do not discuss existing network alignment algorithms. Network alignment algorithms are another approach for the network comparison problem with the goal of producing a mapping between nodes of two networks such that the correspondence between the edges of the compared networks is maximised.

1.3.1 Global Network Properties

- **Degree distribution**. The *degree* k of a node is the number of edges attached to that node. The degree distribution $d(k)$ shows the probability that a randomly selected node has degree k, for all $k \geq 0$ [16]. Nodes with the highest degree values are called network *hubs*. Note that this is a very simplistic notion of a hub and that identification of hubs in real world networks is often based on multiple network attributes [24]. For example, in a brain network with vertices corresponding to brain regions and edges representing inter-regional pathways, hub brain regions were identified based on node degree, motif fingerprint, betweenness and closeness centrality of nodes in the network [25]. Another example is a directed network of web-pages [26], where nodes are ranked based on the relevancy of information they contain. In that network, a hub was defined as a vertex that points to highly ranked vertices [26, 27].

  The degree distribution captures only one aspect of network topology; networks with completely different topologies can have the same degree distributions [28], as illustrated in Figure 1.1. In social networks the degree of a node is sometimes referred to as *degree centrality*, to emphasise its use as a centrality measure [18]. The centrality measures will be discussed further later on. The *average degree* of a network is the arithmetic average of the degrees of all nodes in the network.

  In directed networks there exist two different types of degrees for a node: (1) the *in-degree* of a node is defined as the number of edges that are pointing to the node, (2) the *out-degree* is defined as the number of edges that are pointing from the node. The total degree of the node in the directed network is given as the sum of its *in-* and *out-* degrees.

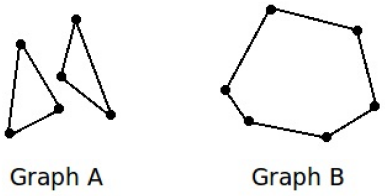


Figure 1.1. Different topologies with the same degree distributions. The topologies of two graphs shown in the figure differ: Graph A consists of two connected components (two triangles), while Graph B is a single connected component. Still, both graphs have the same number of nodes (six), the same number of edges (six), and the same degree distribution (each of the six nodes in the graph has the degree of two).

When comparing networks using their degree distributions, the most common approach is computing the Euclidian distance between the distributions of the two networks. If d_i and d_j are the degree distributions of the two networks I and J being compared, then the Euclidian distance $E_{dist}(d_i, d_j)$ is given as:

$$E_{dist}(d_i, d_j) = \sqrt{\sum_{k=0}^{max(k_i,k_j)} (d_i(k) - d_j(k))^2;}. \tag{1.1}$$

where k_i and k_j are maximum degrees in networks I and J, respectively. It is possible to differently weight or normalise distributions before computing the Euclidian distance in order to reduce or emphasise the significance of some elements in the distribution.

- **Clustering spectrum, Average clustering coefficient**. The *clustering coefficient* is the probability that two nodes j and k, connected to node i, are also connected among themselves [29, 30]. The clustering coefficient of a node i is defined by

$$C_i = \frac{2K_i}{k_i(k_i - 1)}, k \geq 2; \tag{1.2}$$

where K_i denotes the number of edges between neighbours of the node i and k_i denotes the degree of the node i. $C_i = 0$ for $k < 2$. *The average clustering coefficient* of a network is calculated as the average value of clustering coefficients over all nodes in the network [31]:

$$\bar{C} = \frac{1}{n} \sum_{i=1}^{n} C_i, \tag{1.3}$$

where n denotes number of nodes in the network. *Clustering spectrum* of the network, $C(k)$, is the distribution of the averages of clustering coefficients of all nodes of degree k in the network, over all k.

- **Average path length**. The shortest path length l_{ij} between nodes i and j is the minimum number of edges that form a connected path between these nodes. The *average path length* in the network is defined as

$$\langle l \rangle = \frac{2}{N(N-1)} \sum_{\langle i,j \rangle} l_{ij}; \tag{1.4}$$

where the sum is over all different pairs of i, j [32].

- **Network diameter and network radius**. In a connected graph, the *eccentricity of a node* is the maximum distance between the node and any other node in the graph. The maximum eccentricity is denoted as the *network diameter*. The minimum graph eccentricity is denoted as the *network radius*. Note that the diameter of the network is also defined as the maximum distance within the network (described above): $D = max(l_{ij})$. Here we also mention the small-world properties of a network [33]. In small world networks, the shortest distance between two randomly chosen nodes grows proportionally to the logarithm of the number of nodes in that network [29]. Social networks and some biological networks exhibit small-world network characteristics—they have much smaller diameters than would be expected at random [29].

- **Centrality Measures**
  - **Degree centrality**. As mentioned above, degree centrality is equivalent to the degree of a node. In terms of centrality, the degree relates to the *importance* of a node, based on the number of neighbouring nodes.
  - **Eigenvector centrality**. This is an extension of degree centrality: the *importance* of the vertex increases based on the importance of its neighbouring nodes [18]. Relative scores are assigned to nodes based on the concept that having high-scoring nodes as neighbours contributes more to the score of the node than having low-scoring neighbours. Eigenvector centrality of a node i can be calculated using [34]:

$$x_i = k_1^{-1} \sum_j A_{ij} x_j; \tag{1.5}$$

    where A is the adjacency matrix of the network, and k_1 is the largest of the eigenvalues of matrix A. The eigenvector centrality of a node has a higher value if the node has many neighbours, and/or his neighbours are *important*. Another generalisation of degree centrality is *Katz centrality* [35] of a node, which measures the number of all nodes that can be connected with the node through a path. The contributions of more distant nodes are penalised using an attenuation factor $\alpha \in (0, 1)$. The centrality measure with the trade name *PageRank* [36] is used as a central part of Google's web ranking technology. It can be observed as a variation of Katz centrality where the centrality that a node derives from its neighbours is proportional to their centrality divided by their out-degree [18] (a network of webpages being a directed network).

Spectral network theory analyses the topology of a network by using the eigenvalues and eigenvectors of network matrices. If X is a matrix describing the network (Laplacian or adjacency matrix) [37], then the eigendecomposition of X is given as $X = \phi\lambda\phi^T$, where $\lambda = diag(\lambda_1, \lambda_2, ..., \lambda_n)$ is the diagonal matrix with the sorted eigenvalues as elements and $\phi = (\phi_1|\phi_2|...|\phi_n)$ is the matrix of columns containing sorted eigenvectors. The *graph spectrum* is defined as the set of eigenvalues $s = \lambda_1, \lambda_2, ..., \lambda_n$, where $\lambda_1 \leq \lambda_2 \leq ... \leq \lambda_n$. Note that the eigenvalues of a matrix are real numbers in the case of a symmetric matrix, i.e. $A = A^T$, which means that the spectra of undirected networks are real numbers. Two networks are called *cospectral* if they have the same eigenvalues.

Networks can be compared based on their spectra by calculating the *spectral distance*. If s^1 and s^2 are network spectra of two graphs G and H, then the spectral distance is defined as the Euclidian distance $d(G, H)$ between the spectra s^1 and s^2 [37]:

$$d_s(G, H) = \sqrt{\sum_i (\lambda_i^1 - \lambda_i^2)^2}; \tag{1.6}$$

If the lengths of the spectra are different, 0 valued eigenvalues are added to the smaller spectrum while preserving the correct magnitude ordering. Graph spectrum can be computed using Laplacian matrix, normalised Laplacian matrix, adjacency matrix, shortest path length matrix etc. It was shown that the spectral distance computed using Laplacian matrices is the most appropriate for classification and clustering experiments [37]. Thus, in our experiments we use spectral distance based on Laplacian matrices and denote it simply as spectral distance.

– **Closeness centrality**. The *farness* of a node is defined as the sum of its shortest paths to all other nodes, and its *closeness* is defined as the inverse of the farness [38]. For node v it is calculated as:

$$C_c(v) = \frac{1}{\sum_{u \in V} dist(u, v)}; \tag{1.7}$$

where $dist(u, v)$ is the distance between nodes u and v and V is the set of nodes in the network. Therefore, the more central the node is in a network, the lower its total distance to all other nodes is.

– **Betweenness centrality**. This quantifies the number of times a node lies on the shortest path between two other nodes in the network [39]. Betweenness centrality of a node i can be calculated using [18]:

$$x_i = \sum_{s \neq i, t \neq i, s \neq t} \frac{n_{st}^i}{g_{st}}, \tag{1.8}$$

where n_{st}^i is the number of shortest distances between s and t that pass through node i, and g_{st} is the total number of shortest distances between nodes s and t. The convention is that $\frac{n_{st}^i}{g_{st}}$ equals 0 if g_{st} is 0. Equation 1.8 can be normalised by dividing it with the total number of node pairs in the network.

– **K-shell decomposition**. K-shell decomposition of a network is conducted by iteratively removing and grouping nodes based on their degrees. The steps of the algorithm are:

1. All nodes of degree ≤ 1, along with their edges, are removed from the network. All removed nodes form the 1-shell of the network;
2. In the resulting network, all nodes of degree ≤ 2, along with their edges are removed from the network, forming the 2-shell;
3. The decomposition process is repeated until all nodes are assigned to one of the k-shells.

The largest value of k for which the resulting network is not empty is called k_{max}, and the corresponding sub-network is called k_{max}-*core*, or the *core* of the network. Nodes corresponding to higher degree shells are more central in the network, but are not necessarily hubs [40].

1.3.2 Local Network Properties

- **Network Motifs**. A network motif is a pattern that occurs at a statistically significant frequency in the network [23]. Motifs are partial sub-graphs. The process of finding motifs is as follows: (1) occurrences of different patterns of interest in a network are counted, (2) the network is randomised conserving the nodes degrees, (3) the frequencies of the patterns are counted in the randomised network, where the null model is Erdös-Renyi (ER) random network model (random network models will be described in more detail in section1.4), (4) steps 2 and 3 are repeated to find the frequency distribution for topological patterns in the

set of randomised networks, (5) statistical significance of the frequency of each sub-graph in the original network is determined from the frequency distributions obtained in (4) using the Z-score as follows [41]:

$$Z_i = \frac{(N_{real_i} - < N_{rand_i} >)}{std(N_{rand_i})}, \tag{1.9}$$

where N_{real_i} is the number of times the sub-graph appears in the original network, and $< N_{rand_i} >$ and $std(N_{rand_i})$ are the mean and standard deviation of its appearances in the randomised networks, respectively. The normalised Z-score of the sub-graph in question is called the sub-graph's *significance profile* SP and for a sub-graph i it is calculated as:

$$SP_i = \frac{Z_i}{(\sum Z_i^2)^{\frac{1}{2}}}. \tag{1.10}$$

This normalisation emphasises the relative significance of sub-graphs, rather than the absolute. Normalisation enables the comparison of networks of different sizes because the motifs in large networks tend to have higher Z values than those in smaller networks. It is possible to group networks according to similar motif-spectra [41]. Obviously, network motifs can be identified both for directed and undirected networks and indicate the main organisational principles within a network. An example is the feed forward loops that are shown to be over-represented in signalling networks [42] which is in line with how the signals are propagated though such networks. A drawback of network motifs is that they are dependent on the choice of a network null model [43]

- **Graphlets**. Graphlets are the small, connected, non-isomorphic induced sub-graphs of a network first introduced by Pržulj *et al.* [7]. Recall that an induced sub-graph of a network G is a sub-graph that contains all edges between its nodes which are present in G; this is different to a partial sub-graph that contains only some of these edges (above defined network motifs are partial sub-graphs). They can appear in the network at any frequency and thus are not dependent on a null model. All 30 two to five node graphlets, denoted by G_0 to G_{29} are shown at the top of Figure 1.2. Three highly sensitive measures of network local structural similarities are based on graphlets: the *Relative Graphlet Frequency Distance* (RGF distance) [7], *Graphlet Degree Distribution Agreement* (GDD agreement) [44] and *Graphlet Correlation Distance* (GCD) [13]. Also, the *Graphlet*

Degree Vector (GDV), or node signature, captures the topology of a node's neighbourhood. Comparing the signatures of two nodes provides a highly constraining measure of local topological similarity between them - *Graphlet Degree Vector* similarity [4]. As a large part of this dissertation deals with the generalisation of all graphlet-based measures to a directed case, in the following section we elaborate upon graphlet-based network properties in more detail.

1.3.3 Graphlet-based Measures for Analysing Network Topology

The **Graphlet Degree Vector** [4] (GDV) is a generalisation of the degree of a node and it counts the number of all two to five node graphlets that the node touches, taking into account different "symmetry groups" within each graphlet (numbered from 0 to 72 in the top panel of Figure 1.2). These symmetry groups are called automorphism orbits (detailed in [44]). For example, it is topologically relevant whether a node touches graphlet G_4 at the middle node, or at one of the end nodes (top of Figure 1.2). These counts are coordinates in the 73-dimensional *Graphlet Degree Vector (GDV)* of a node. An illustration of a GDV of node v is given in the bottom panel of Figure 1.2.

The similarity between GDVs of nodes u and v in graph G is computed as follows [4]. If u_i is the i^{th} coordinate in the GDV of node u, and v_i is the i^{th} coordinate in the GDV of node v, then the distance between these two coordinates is computed as:

$$D_i(u,v) = w_i \times \frac{|log(u_i+1) - log(v_i+1)|}{log(max(u_i,v_i)+2)}. \tag{1.11}$$

In formula (3.1), w_i represents the weight of coordinate i, which takes into account dependencies between orbits, as described in [4]. Namely, the occurrence of some orbits is dependent on the occurrence of other orbits. For example, the difference in the number of orbits 3 that a node touches implies the difference in the number of orbits that contain orbit 3, such as orbits 14 and 72. This observation is applied to all orbits and the distinction is established between "more important" and "less important" with higher and lower values of w_i respectively. To compute w_i, each orbit i is assigned a value o_i that denotes the number of orbits that affect orbit i. Each orbit also affects itself. For example, $o_{15} = 4$ because orbit 15 is affected by orbits 0, 1, 4, and itself. Finally w_i is computed as follows:

$$w_i = 1 - \frac{\log o_i}{\log 73}. \tag{1.12}$$

The total distance between the GDVs of nodes u and v, normalised in $[0,1]$ range, is

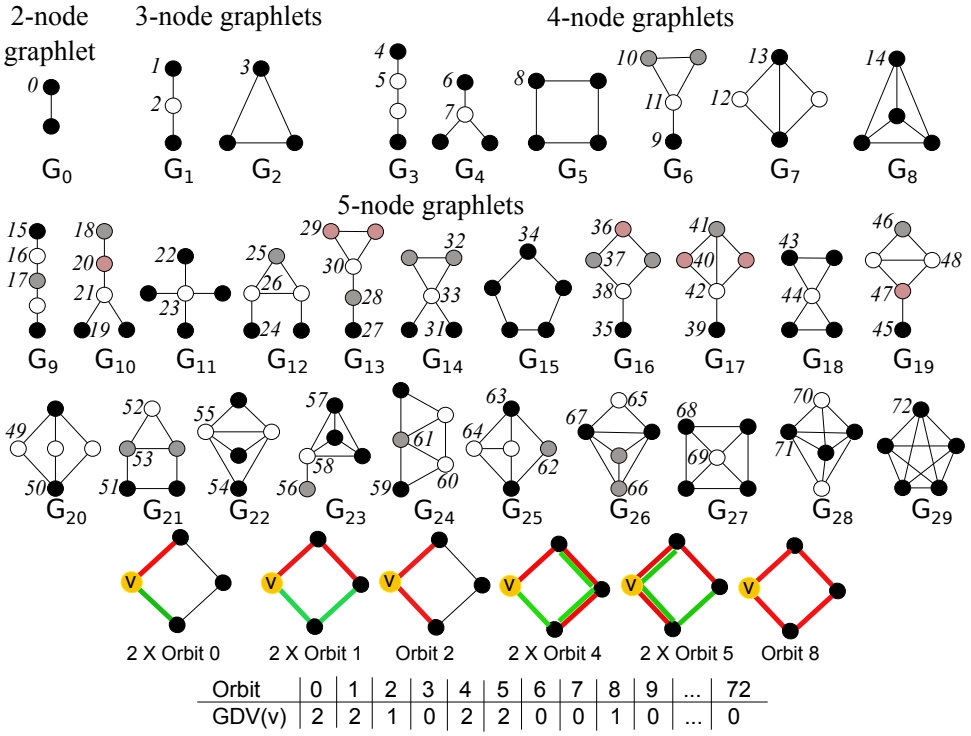


Figure 1.2. 73 Graphlets and graphlet degree vector (GDV) of a node. Top: Graphlets with up to five nodes, denoted by $G_0, G_1, G_2, \ldots G_{29}$. They contain 73 "symmetry groups," denoted by $0, 1, 2, \ldots, 72$. Within a graphlet, nodes belonging to the same symmetry group are of the same shade [44]. Bottom: An illustration of the GDV of node v. $GDV(v) = (2, 2, 1, 0, 2, 2, 0, 0, 1, 0 \ldots, 0)$, meaning that v is touched by two edges (orbit 0, illustrated in green and red in the first panel), two times as end-node of one graphlet G_1 (orbit 1, illustrated in the second panel), the middle node of one graphlet G_1 (orbit 2, illustrated in the third panel), two times as end-node of graphlet G_3 (orbit 4, illustrated in the fourth panel), two times as a middle node of graphlet G_3 (orbit 5, illustrated in the fifth panel), and touched by graphlet G_5 (orbit 8, as illustrated in the most right panel).

calculated as:

$$D(u,v) = \frac{\sum_{i=0}^{72} D_i}{\sum_{i=0}^{72} w_i}. \tag{1.13}$$

Finally, the **GDV similarity** of the two nodes is computed as:

$$S(u,v) = 1 - D(u,v). \tag{1.14}$$

The GDV similarity between proteins in the human PPI network has already been used to successfully predict protein function and involvement in disease [4–6, 45].

Relative graphlet frequency is defined as $\frac{N_i(G)}{T(G)}$, where $N_i(G)$ is the number of graphlets of type $i, i \in 1, ..., 29$ in the network G, and $T(G) = \sum_{i=1}^{29} N_i(G)$ is the total number of graphlets in G [7]. The relative graphlet frequency distance $D(G, H)$ for the two graphs G and H is defined as:

$$D(G, H) = \sum_{i=1}^{29} |F_i(G) - F_i(H)|, \tag{1.15}$$

where $F_i(G) = -log(\frac{N_i(G)}{T(G)})$ [7]. The logarithm of graphlet frequency is used to avoid dominance of the most frequent graphlets in the networks over less frequent ones. Relative graphlet frequency was used to compare PPI networks with different types of random networks and to show that PPI networks are closest to geometric random graphs with respect to this parameter [7]. The model networks will be described further in Section 1.4.

Graphlet Degree Distribution (GDD) is analogous to degree distribution: for each of the 73 automorphism orbits (Figure 1.2), the distribution of nodes that are touching a particular graphlet at the node belonging to a particular orbit is calculated (for a particular orbit we count the number of nodes touching a graphlet at that orbit). This results in spectrum of 73 graphlet degree distributions, where the degree distribution is one of them (the first one). Networks can be compared based on the GDD agreement measure which is defined as follows: Let d_G^j be GDD for the j^{th} automorphism orbit in network G. The normalised distribution for the network G is defined as [44]:

$$N_G^j(k) = \frac{S_G^j(k))}{T_G^j}; \tag{1.16}$$

where d_G^j is scaled as $S_G^j(k) = \frac{d_G^j(k))}{k}$ to decrease the contribution of larger degrees in GDD, and then the distribution is normalised with respect to its total area:

$$T_G^j = \sum_{k=1}^{\infty} S_G^j(k). \tag{1.17}$$

The distance between normalised j^{th} distributions for the two networks G and H is [44]:

$$D^j(G,H) = \frac{1}{\sqrt{2}} \left(\sum_{k=1}^{\infty} [N_G^j(k) - N_H^j(k)]^2 \right)^{\frac{1}{2}}, \tag{1.18}$$

where the resulting value is between 0 and 1, 0 meaning that the j^{th} GDDs are identical. Further, j^{th} GDD agreement is obtained as:

$$A^j(G,H) = 1 - D^j(G,H). \tag{1.19}$$

Finally, the **GDD agreement** between two networks is defined either as the arithmetic or geometric mean of all GDD agreements over 73 automorphism orbits [44]. This measurement was used to show that PPI networks are best modelled by geometric random graphs [44]. More on the topic of network modelling is covered in the following section.

Graphlet Correlation Matrix and Graphlet Correlation Distance. Recall that when the GDV similarity between two networks was calculated, the dependencies between orbit counts were observed. By exploiting these dependencies, a new concept, *Graphlet Correlation Matrix*, and a new measure for comparing the network topologies, *Graphlet Correlation Distance*, were introduced [13]. In addition, the redundant orbits were identified and eliminated. Namely, there exist 17 linear equations describing all redundancies amongst the 73 graphlet orbits, which means that only 56 orbits are non–redundant. Of 15 orbits for up to 4-node graphlets, 11 of them are non–redundant.

The dependencies (correlations) between non–redundant orbits over all nodes in a network are motivation for computing a Spearman's Correlation between graphlet degrees and constructing the Graphlet Correlation Matrix (GCM) of the network as follows [13]. First, it is observed that there are fewer dependencies between the 11 non–redundant orbits for up to 4-node graphlets, than between the 56 non–redundant orbits for up to 5-node graphlets. This means that up to 4-node graphlets introduce less noise in the new network statistic, so the new statistic takes into account only 11 non-redundant orbits on up to 4-node graphlets. For each node in a network a Graphlet Degree Vector, corresponding to the 11 non-redundant orbits, is constructed. Then a matrix containing rows of Graphlet Degree Vectors is formed. Its number of rows equals the number of nodes in the network and the number of columns equals 11 (number of orbits). For a given network N, the Spearman's Correlation coefficients between all pairs of columns of the above described matrix are computed and presented in a 11×11 symmetric ma-

trix of the network N, named GCM_N [13]. In this way, the topology of the network N is summarised into an 11×11 sized symmetric matrix with values in the interval [-1, 1]. Note that some graphlets, and hence orbits, may not appear in the network, which would result in an entire column of zeros. Spearman's correlation coefficient can not be calculated if all the values of one of the vectors are the same (zero is the only possibility in the case of GDV), and this problem is solved by introducing a dummy node in the network with a GDV vector containing only values that equal 1. This way the correlation between non-existing orbits is 1, and the correlation between a non-existing orbit and any other orbit, whose column has non-zero values, is close to 0.

The Graphlet Correlation Distance (GCD) between networks N_1 and N_2, characterised with GCM_{N_1} and GCM_{N_2}, is calculated using the Euclidean distance of the upper triangle values of GCM_{N_1} and GCM_{N_2}. GCD is free of redundancies and encodes information about local network topology. It outperforms other measures both on synthetic and real world networks [13] and has been used to track dynamics of the world trade network (WTN). It was also used to discover broker and peripheral roles of countries in WTN and the correspondence of these roles to economic prosperity or poverty, respectively [13].

GCD-11 denotes the graphlet correlation distance that is computed from the GCM of non-redundant 2- to 4-node graphlet orbits. Similarly, GCD-73 denotes the graphlet correlation distance that is computed from the GCM of all 2 to 5-node graphlet orbits.

1.4 Random Network Models

A network model is a random network with specific and predefined network properties. A network model that is well fitted to a real world network can provide better understanding of real world network data. For example, as we will discuss in Section 1.5, biological data are still incomplete and noisy due to sampling, biases in data collection and interpretation, and limitations in technology [46,47]. If it is possible to find an adequate theoretical network model that fits a network, *i.e.* precisely reproduces the network's structure and laws, then that model can be used to predict missing data. Also, a well-fitting model can provide easier computational manipulation of the network data and help understand the mechanisms of biological processes and evolution within the cell [48]. We evaluate our new measures for the comparison of directed networks by evaluating their performance on clustering model networks (see Chapter 3.2).

Here we present the rules for constructing network models commonly used for studying biological networks: Erdös-Rényi (ER) graphs, Scale-free (SF) random graphs,

Stickiness-index-based model, Small-world network model, and several types of Geometric graph (GEO) models. In Chapter 3 we go into further detail in regards to how we generate directed network models.

- **Erdös-Rényi (ER) graphs**. This is the earliest random graph model. An ER graph $G(N, M)$ is constructed so that M edges are randomly placed between N nodes with the same probability p [49]. These graphs have Poisson degree distributions (for small number of nodes in the network it is binormal), their average diameter is an order of $\log(n)$, they have low clustering coefficients for low p, and generally exhibit the small-world property for $p > \frac{2}{N(N-1)}$ [50]. Note that, according to Watts-Strogatz definition of small world networks [29], the small-world properties include short (logarithmically growing) average path lengths and high clustering coefficients. Real life networks, in general, are not well described using Erdös-Rényi model. Regarding biological networks, which have power-law degree distributions and high clustering coefficients, it was shown that ER model poorly captures their properties [7].

  However, random graphs can be generalised by constructing the generalised random model ER-DD graph, in the sense that edges are randomly chosen in the same way, but the degree distribution has to fit the degree distribution of the real network that is being modelled (this was applied to world wide web networks and networks of collaboration between scientists) [51]. An ER-DD graph is constructed in the following way: (1) the number of "stubs" (that will be filled by edges) is assigned to each node, based on the degree distribution of the real network to be modelled [24], (2) edges are created between pairs of nodes with stubs picked at random, and each time the number of stubs left available at the corresponding end nodes of the edges is decreased by one, (3) multiple edges between the same pair of nodes are not allowed.

- **Scale-free (SF) random graphs**. These networks have a power-low degree distribution [52]. They can be generated by iteratively adding nodes to a small seed network, so each new node is attached to existing nodes proportionally to their connectivity. This is *the rich get richer* principle, known as Barabási-Albert preferential-attachment model (SF-BA) [52]. The probability that a newly added node in the network will be connected to node i among n existing nodes is $p(v_i) = \frac{d_i}{\sum_{j=0}^{n} d_j}$, where d_i is the degree of the node i. Clustering coefficient and average diameter of SF-BA networks are low.

Another way of generating scale-free networks (when $N \to \infty$) is to duplicate existing edges in a way that they keep their existing interactions with the probability $0 < \pi < 1$ [53]. SF networks capture the degree distribution of PPI networks which follows a power-law [54]. Note that it has been shown that subsets of scale-free networks are not scale-free [55]. Since currently available PPI information is incomplete, the fact that current PPI networks have power-law degree distributions therefore does not guarantee that a complete PPI network would share the same property. Also, Vasquez *et al.* [56] proposed the *Scale-free gene duplication and divergence model* (SF-GD) that generates networks with power-low degree distribution, and fits PPI networks better than the preferential-attachment model, mentioned above. The principle for building SF-GD networks is as follows: (1) a newly added gene to the seed network inherits the same connections that a randomly chosen existing node has (a duplication step), (2) the new node and the selected node are connected with probability p, (3) in the mutation (divergence) step, each edge that the new node inherited is deleted with a probability q. This process is completed when the network reaches desired size and density.

- **Stickiness-index-based model (Sticky)**. This is a random graph model, where a connection between two proteins is inserted according to the degree, or "stickiness", of the two nodes involved [57]. This idea is based on the assumption that in a biological network, the proteins that partake in more interactions have many binding domains and it is highly likely that such proteins interact among themselves. The *stickiness index* of a node i is calculated as $S_i = \frac{k_i}{\sqrt{\sum_{j \in V} k_i}}$, where V is a set of nodes in the network, and k_i is the degree of the node i. Sticky networks mimic degree distribution, clustering coefficient and average diameter of real-worlds networks well.

- **Small-world model**. This model was proposed by Watts and Strogatz in [29]. It starts as a circle model (ring lattice) with n vertices in which every vertex has a degree of c, but for each node one of its edges is removed with a probability p and replaced with an edge that connects the node to a uniformly randomly chosen node in the network. The parameter p controls the interpolation between circle model and random graph (for $p = 0$ the circle model is present, while for $p = 1$ we have a random graph). This model captures both high clustering coefficient and the small-world effect of real networks. However, the small-world model does not mimic the degree distribution of real world networks well. An alternative to this model is not to remove any edges from the circle when adding random additional

edges between nodes [18].

- **Geometric graphs (GEO)**. For a set of nodes distributed in space and a constant value ε, two nodes are connected if the "distance" between the nodes is within the value of a distance threshold ε. The value of constant ε is chosen so that the resulting network would capture the number of edges in the network that is being modelled. If the nodes are distributed uniformly at random in space, these are called geometric random graphs [7]. These *uniform* geometric graphs have high clustering coefficients and a small average diameter, like real worlds networks, but differ in type of the degree distribution – GEO networks follow a Poisson degree distribution.

  It has been shown that PPI data are better fit by GEO than SF model [7,44]. Note that the choice of a network property that can be used to examine the fit of a model is non-trivial, and using different network properties can yield different results. In order to examine the fit of GEO model and PPI networks Memišević et al. in [48] created a "network fingerprint" that integrates several network properties: the average degree, the average clustering coefficient, the average diameter, and graphlet frequency. The results showed that the structure of PPI networks is most consistent with noisy GEO networks.

  The GEO model was later refined into a *Trained Geometric Model* (TGEO) [58], which learns the structure of a PPI network, and therefore captures most of the network's properties from the real data, instead of reproducing the properties. In this model, nodes are not distributed in a metric space at random, but the distribution $p_{learned}$ in the metric space is learned from the real data, which results in the power-law degree distribution of the TGEO network. Only the high confidence part of *S. cerevisiae* PPI network was used to train the model. A 3-dimensional Euclidian unit cube was chosen as a metric space, into which the nodes were embedded. The embedding algorithm that was used [59] is based on the premise that network connectivity information corresponds to Euclidian proximity (similarly to geometric random graphs). The model was evaluated by comparing it to PPI data networks using main global network properties, and it was outperformed only by the standard geometric random model. When using GDDA measure for comparison, this model outperformed others. Note that for more noisy networks, both GEO and TGEO models were outperformed by other models.

  Pržulj *et al.* [60] introduce two network models that use the principles of geometric graphs to model the evolutionary dynamics of PPI networks: *Geometric Model*

with Gene Duplication and Divergence (GEO-GD) expansion and *GEO-GD with the probability cut-off*. Both models govern growth of the network from a small seed network by adding new nodes in a way that imitates gene duplications (GD) and mutations. GEO-GD is a biologically motivated model for PPI networks: (1) genes and proteins exist in a multidimensional biochemical space, (2) the duplicated gene is placed at the same point in space as its parent and then, if not eliminated by natural selection, is slowly separated keeping some of the parent's interactions and gaining some new ones, (3) the difference in their properties is proportional to their distance in this abstract space. The process of generating GEO-GD networks imitates this scenario as follows: (1) A small number of nodes are distributed randomly in the space—seed network; (2) Each new node is introduced as a duplicated node of a randomly selected node in the network; (3) The new node is moved randomly from its *parent* node in the metric space. In the GEO-GD expansion (GDE) model, the new node moves a random distance within the value of $2 \times \varepsilon$. For a GEO-GD with the probability cut-off (GDP) model a node can move from its parent for a random distance of ε with the probability p or for a random distance $10 \times \varepsilon$ with the probability $1 - p$. Note that when generating GEO-DD networks we use the GDE approach.

GEO-GD networks have power-low degree distributions, high clustering coefficients and low network diameters. A GDD agreement measure was used to examine the fit of the GEO-GD model and PPI networks. It outperformed other networks models for high confidence parts of the yeast interactome, closely followed by scale-free duplication model from [56].

Finally, we distinguish between two different types of network models: descriptive models and network-driven models [60]. Descriptive models describe general properties of a particular type of network (e.g. PPI network), for example by reproducing the type of degree distribution characteristic for that type of network, or by modelling the principle (e.g imitating gene duplication). Such are, for example, Scale-free (SF) duplication model or GEO-GD model. On the other hand, Stickiness-index-based model, ER-DD model and TGEO model are network-driven because they need a particular network example to reproduce its structure.

1.5 Biological Networks

In a biological network, biological elements and the relations between them correspond to nodes and edges. For example, an edge in a protein network is placed between two proteins if they bind together to perform their biological function, which results in a protein-protein interaction (PPI) network. However, an edge between two proteins can also correspond to a common trait between two proteins, such as being targeted by the same drug, or causing the same disease–associations that exist in the scientific literature, thus resulting in a type of an association network. Other highly exploited biological networks are genetic interaction networks, metabolic networks and transcriptional regulation networks. In this section we provide further details on these networks.

Recent advances in high-throughput techniques have resulted in a number of large-scale biological data sets. Table 1.1 lists commonly used databases of biological knowledge. These databases contain the biological information necessary for building different types of biological networks: interactions and relationships among biological macromolecules and metabolites, such as protein-protein interactions (PPI), genetic interactions or enzyme-substrate relationships. Available data also include gene functional annotations, pathway maps, information on genetic disorders and disease associations. To give an example of the scale of available data, BioGRID currently[1] lists 771,245 combined (physical and genetic) raw interactions between 56,907 genes (proteins) across 56 species, while DRYGIN contains 5,482,948 genetic interactions for *S. cerevisiae*. A limiting factor regarding the reliability of the networks is certainly the quality of data. As discussed before, although large amounts of biological data are available, they are still noisy and incomplete [46, 47]. This is influenced by biases introduced by screening techniques used for obtaining the data - they may not be sensitive enough to detect all the changes in the system [61]. Also, the outcomes of experiments depend on the stringency of experimental conditions: overly stringent conditions can lead to false negative interactions, as opposed to false positive results obtained from experiments that were not stringent enough. Another bias is introduced by the focus of the research - in particular, some genes/proteins can be more interesting to scientists, thus their interactions are explored more often. An example of this is disease related genes. This can result in the existence of false hubs in the network, without reflecting the true network topology. In addition, not all biological processes can be accurately represented as interactions (edges in the network) between two elements because a biological process can require more than two elements and involves different types of interactions.

[1] June 2015

Database name	Type of data	Number of organisms	Ref.
BioGRID	PPI and genetic interactions	56	[62]
HPRD	PPI, disease associations, posttranslational modifications, tissue expression, subcellular localisation, and enzyme/substrate relationships	1 (H. sapiens)	[63]
DIP	Experimentally determined PPI	10	[64]
HomoMINT	PPI experimentally verified in model organisms for *H. sapiens*	1 (*H. sapiens*)	[65]
I2D	PPI	7	[66]
KEGG	Pathway maps, diseases, drugs, orthology groups, genes, relations within genes, metabolites, biochemical reactions and enzymes	3900	[67]
OMIM	Information on genes and genetic disorders	1 (H. sapiens)	[68]
DRYGIN	Genetic interactions	1 (S. cerevisiae)	[69]
RegulonDB	Transcriptional regulation information	1 (E.coli)	[70]
Reactome	Pathways data	19	[71]
SCOP	3D structure information for proteins	Not classified according to species	[72]

Table 1.1. Databases of molecular interaction data.

Here we list commonly analysed networks of interactions between different biomolecules.

- **Protein-protein interaction (PPI) networks**. Proteins are the main building blocks of a living organism. In PPI networks, proteins correspond to nodes in the graph, with an edge between any two nodes whose corresponding proteins interact. These interactions occur when proteins bind together to perform their biological function within a cell. PPIs are essential to many processes in the cell and therefore PPI networks have been a focus of research in systems biology. Advances in proteomics led to large quantities of PPI data. There are several methods for detecting protein-protein interactions. Most commonly used are Yeast Two-Hybrid (Y2H) screening [73] which results in binary data, and Mass Spectrometry (MS) [74] of purified complexes which results in co-complex data. These are high-throughput methods which, in contrast to small-scale techniques, result in less biased interactions. In the same way the sequence data provides an overview of the genome, the PPI data will hopefully give us an analogues view of the interactome [75]. Currently collected PPI networks are noisy and are just a sample of the complete networks [75], and incompleteness has an effect on over-

all network topology [76]. Still, as discussed throughout Chapters 1 and 2 of this dissertation, PPI network topology has provided insights into new biological knowledge. Note that many PPIs are undirected and represent "stable" interactions in which interacting partners stay bound (such as in protein complexes). However some interactions are "transient" which means that interacting partners are bonded at different times depending on conditions (this happens in signalling cascades). This means that ideally a PPI network would contain both directed and undirected edges and its topology would be time-dependent. This type of edge information is currently not available on systems-level scale, therefore PPI networks are represented as undirected static networks [77], with the exception of some studies which assign weights to the edges to include the information about confidence of the interactions [78].

- **Genetic interaction networks**. In a genetic network, genes correspond to nodes in the graph, while edges represent functional associations between genes. An interaction between two genes occurs when the observed phenotype that is a result of simultaneous mutations in the genes is not just an expected combination of phenotypes of single mutations. For, example two genes that do not cause lethality when individually mutated, can cause lethality if mutated simultaneously. Note that the expected phenotype of simultaneous mutations would be based on the multiplicative phenotype fitness model - when the combined deletion of two genes results in phenotype which is multiplication of effects caused after single deletions [79–81]. Genetic interactions are classified as negative, if the phenotype of double mutants is significantly worse than expected from the phenotypes of single mutants, or positive if the phenotype of double mutants is better [82]. Negative genetic interactions often do not correlate with PPIs or protein associations in protein complexes, because they often contain pairs of genes which are involved in parallel pathways [79, 83]. Genetic interactions can be identified using synthetic genetic array (SGA) experiments [84] or synthetic lethal analysis by microarray (SLAM) experiments [81].

- **Metabolic networks**. A series of successive biochemical reactions for a specific metabolic function forms a metabolic pathway. When representing metabolic pathways as a graph, nodes correspond to metabolites, and directed edges are metabolic reactions. A metabolic network then represents the union of all metabolic pathways within a cell and is a complex network of reactions and integrating processes that generate mass, energy, information transfer, and specify the fate of

the cell [85–87]. Edges in a metabolic network representing biochemical reactions (chemical conversions of metabolites from one form to another) can be directed or undirected, depending on the reversibility of a specific reaction. It is common practice that reactions in a metabolic network are replaced with enzymes that catalyse them (or genes/proteins that produce these enzymes). Thus, there are different representations of metabolic networks: bipartite networks in the form of metabolite–enzyme, metabolite–gene or metabolite–protein interactions. If the reaction nodes (enzymes, genes or proteins) or metabolite nodes are removed and the remaining nodes are connected under the condition that they were at a distance 2 in the original bipartite network, the result is a simple metabolic network containing genes, proteins, enzymes or metabolites. An example of such metabolic modelling is the network of nodes which represent enzymes and directed edges placed between the nodes (enzymes) if a product of one enzyme is the substrate of the other [88]. This means that interacting enzymes in the original bipartite network had the following roles: one enzyme catalysed the reaction whose product was a substrate for a reaction catalysed by the other enzyme.

In Chapter 4 we analyse metabolic networks in the form of gene–gene interactions (genes that encode for enzymes).

- **Transcriptional regulation networks**. Living cells are the product of gene expression, a process that regulates which genetic information will be turned into gene products. Gene expression programs depend on the recognition of specific promoter sequences by transcriptional regulatory proteins. How a collection of regulatory proteins associates with genes across a genome can be described as a transcriptional regulatory network [89]. Just as metabolic networks describe the potential pathways that may be used by a cell to accomplish metabolic processes; a network of regulator–gene interactions describes the potential pathways cells use to regulate global gene expression programs [89]. The nodes in a transcriptional regulation network represent genes and a directed edge exists if the product of one gene (a protein) regulates the transcription of another gene. In particular, this protein binds to regulatory DNA regions of a gene targeted with a directed interaction resulting in its over-expression or under-expression. These interactions are identified based on relative mRNA levels of the genes.

- **Signal transduction networks** Nodes in these networks correspond to proteins and the directed edges represent the signals propagated from one protein to another, encompassing the complex signalling mechanisms inside the cells [90].

These signals represent cellular responses by means of pathways as an answer to internal and external stimuli.

- **Protein Structure Networks**. A protein structure network represents a 3D structure of a protein. Proteins are linear polymers (polypeptides) built from amino acids. The amino acids in a polypeptide chain are linked by peptide bonds. Once linked in the protein chain, an individual amino acid is called a residue. In network representations of protein structures residues correspond to nodes and inter-residue interactions correspond to edges, forming residue interaction graphs (RIGs) [91]. The two amino acids are considered connected if the distance between them is less than $7.5\mathring{A}$ ($\mathring{A}$—Angstrom is 10^{-10} meters).

In addition to the above networks, there exists the notion of *disease networks*, which consist of biomolecules involved in a particular disease or group of diseases and are used for exploring relationships between different diseases. For example, Goh et al. [92] constructed a bipartite "diseasome" network, where one partition consisted of a set of diseases and the other of a set of disease genes (by definition of a bipartite network, all edges in the network go between the partitions). They used it to generate two network projections: the disease–gene network and the human disease network (which they found to be clustered according to major disorder classes). By exploring centrality and peripherality of genes in the gene network, they showed that contrary to essential human genes which encode hub proteins, the majority of disease genes do not encode hubs, and are localised in the periphery of the network [92]. Janjić and Pržulj [93] demonstrated the existence of a topologically and functionally homogeneous "core sub-network" of the human PPI network, which is enriched in disease genes, drug targets, and a small number of genes that have theoretically been proposed as absolutely necessary for tumour formation and that are usually referred to as "driver genes" [94]. They call this sub-network the "Core Diseasome" [93] and postulate it is the key to disease onset and progression; hence it should be the primary object of therapeutic intervention. They find this sub-network purely computationally by utilising the $k-$core decomposition algorithm [95,96] applied to the human PPI network. We will come back to this network in Chapter 2 during the study of key cardiovascular disease genes.

2 Undirected Biological Networks in Researching Complex Diseases: Cardiovascular Disease (CVD) Case Studies

In this chapter, we present two studies which show how the topology of biological networks can be successfully used to complement existing knowledge in biology and medicine; in particular in research of cardiovascular diseases. We choose cardiovascular diseases, a group of diseases of the heart and blood vessels, as an example, because they are currently a leading health problem worldwide [14] with more people dying every year from CVDs than from any other cause [97].

First, we give a short overview of the existing approaches which use topological properties of biological networks to tackle open questions in CVD research [98]. We then present our published study [8] that examines the PPI network wiring around genes involved in CVDs and identifies a subset of CVD-related genes that are statistically significantly enriched in drug targets and *driver genes* [94] - genes that have been proposed to drive onset and progression of a disease. Our identified subset of CVD genes has a large overlap with the Core Diseasome, which has been postulated to be the key to disease formation and hence should be the primary object of therapeutic intervention. This indicates that our approach identifies *key* genes responsible for CVDs. Thus, we use it to predict new CVD genes and validate over 70% of our predictions in the literature. Finally, we show that the predicted genes are functionally similar to currently known CVD drug targets, further confirming a the practical potential of biological network analysis in improving therapy choices for CVDs.

Finally, we present our published research [99] on the protective role of diabetes on the development of aneurysm, a phenomenon suggested in recent studies with still unknown biological mechanisms. We postulate the existence of genes that disrupt the pathways needed for the onset of aneurysm in the presence of diabetes. Motivated by the significance of genetic interactions for understanding disease-disease associations, we use

both protein-protein interaction and genetic interaction data. We use *brokerage* [100], a topological measure that identifies proteins in this sub-network which, if removed, severely affect the interconnectedness of their neighbourhood, enabling such proteins to disrupt the pathway they are in. We identify a set of proteins with statistically significant brokerage values and find this set to be enriched in biological functions that have already been suggested as possible causes of diabetes-aneurysm dissociation.

2.1 Review of Network-based Approaches in Researching Cardiovascular Diseases

Cardiovascular diseases (CVDs) cover a broad range of disorders which affect different parts of the cardiovascular system and include coronary diseases, carotid diseases, peripheral arterial diseases and aneurysms. They remain the leading health problem which affects more than 80 million individuals in the United States alone [14]. By 2020 it is expected that Brazil, Russia, India, and China will contribute significantly to a global increase of 4% in deaths caused by CVDs [101].

For addressing the complex nature of these diseases, integrative approaches that would take into account the co-action between multiple causes behind CVDs are methods of choice. This is why different systems biology approaches have been used in CVD research, which has recently been reviewed in [102–105].

2.1.1 Exploring Disease Through Network Topology

The topology of PPI networks has widely been explored and used for inferring the involvement of proteins in biological functions and processes, as discussed in Chapter 1 of this dissertation. This also applies to inferring involvement of proteins in diseases. The *guilt by association* approach [1] for inferring functions of unannotated proteins in PPI networks was used to associate genes with diseases using linkage methods (nomenclature adopted from [106]). In that sense, it has been shown that directly linked proteins in the human PPI network are more likely to cause similar diseases [107, 108] (simplified concept illustrated in Figure 2.1, panel A). A variant of the linkage method was successfully applied to discover genes related to Alzheimer's disease [109].

Graphlet-based methods have shown that the PPI network topology around a protein is a predictor of its involvement in disease [5, 6]. In particular, GDV similarity between proteins in the PPI network was used as a similarity measure for clustering proteins using a series of clustering methods, resulting in clusters significantly enriched in cancer

and disease related proteins. This lead to predictions of new melanogenesis-related genes purely from the topology of the human PPI network, and the predictions were phenotypically validated [5,6]. The concept of associating genes with the disease if they are in the same cluster of the disease related genes is illustrated in Figure 2.1, panel B. The clustering is based on the topological properties of the nodes in the network. Note that this is different from clustering the network by identifying its topological modules: locally dense neighbourhoods in the network which are called communities [110]. It is generally accepted that a subset of nodes is a good community if the induced sub-graph is dense, with relatively few connections between the included nodes and nodes that are in the remaining part of the graph [111]. These topological modules often correspond to *functional modules* (the aggregation of nodes similar in function) and *disease modules* (the set of components that contribute to a specific disease phenotype [106]).

Integrative approaches for identifying functional modular structures in biological networks were thoroughly reviewed by Mitra *et al.* [112]. Accordingly, module-based methods work with the assumption that nodes which belong to the same topological or functional module are likely to be involved in the same disease. The concept of associating genes with a disease if they are in the same community in the network is illustrated in Figure 2.1, panel C. These methods have often been applied in studies relating to cancer [113–115]. Another example of this principle are the modules identified using community discovery algorithm from [116], which led to the discovery of new links between Alzheimer's disease and CVDs [117]. Several module-based methods have been applied to research of CVDs, which will be elaborated upon in more detail later in this survey.

An interesting survey [118] on the different methods that use network topology for predictions of disease genes pointed out that many of the methods that rely on clustering algorithms or linkage-based inference are outperformed by random-walk-based methods. Random walkers diffuse along the network starting from disease involved nodes with the same probability of visiting any neighbouring node. The most visited genes are considered to be on the disease pathway and potentially involved in a particular disease. The concept of associating genes with a disease based on the random walk principle is illustrated in Figure 2.1, panel D. A method for prioritising candidate disease genes using random walk analysis was tested on 110 disease-gene families and significantly outperformed methods based on local distance measures, such as linkage-based methods or methods based on shortest paths to disease proteins [119].

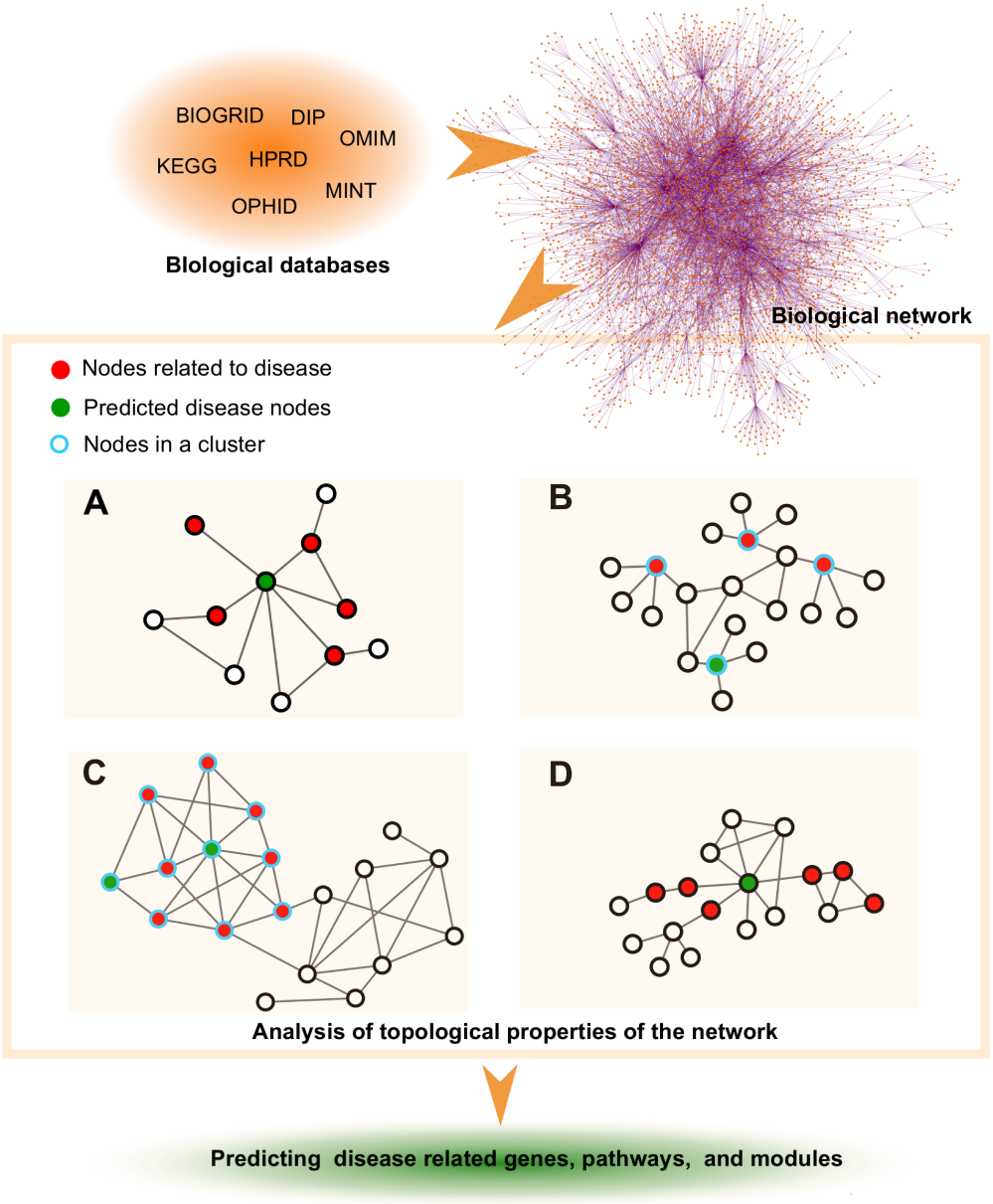


Figure 2.1. Using network topology to uncover elements involved in a disease. Panel A: Green nodes are associated with a disease based on their neighbouring disease nodes (shown in red). Panel B: Nodes with blue borders are part of the same cluster based on a similar topology around them. The green node is associated with disease based on the cluster's enrichment in disease nodes (shown in red). Panel C: Nodes with blue borders are part of the same community in the network. The green node is associated with disease based on the community's enrichment in disease nodes (shown in red). Panel D: Node shown in green is associated with the disease, as a common node on shortest paths between nodes related to disease (shown in red). Figure is taken from Sarajlić *et al.* [98].

2.1.1.1 Revealing new CVD knowledge through network topology

The emerging interest in the molecular interaction networks of various cardiovascular diseases has resulted in a number of association, gene-expression, PPI, and transcriptional regulatory networks being examined to study atherosclerosis [120], in-stent restenosis [121], heart failure, and CVDs in general [102, 103, 122, 123]. Many of these networks were constructed using experimental data combined with literature mining, with the aim of identifying a broader set of genes involved in a particular CVD. Such CVD networks are a valuable platform for exploring disease mechanisms.

Several approaches further explored the topological properties of CVD networks in search of new CVD knowledge. In Table 2.1, we give a short overview of such approaches: we specify the CVD network that was explored, the type of molecular data and interactions that were used, the type of topological analysis that was performed and the aims of the topological analysis. Note that the vast majority of the above-presented topological analyses focused on CVD sub-networks in isolation, rather than observing them as parts of a larger, more complete interaction network, such as the entire human PPI network. This may be a limiting factor when exploring the interplay between the genes involved in different CVDs, or when targeting genes that have previously not been linked to CVDs. The importance of observing the neighbourhood of disease genes in the entire PPI network was emphasised in one of the studies related to atherosclerosis [124] where a functional enrichment test performed only on differentially expressed genes failed to detect biological processes related to disease progression. However, the network that included both differentially expressed genes and genes that have high connectivity with them in the entire PPI network, was functionally enriched in relevant biological processes.

There are only few approaches from Table 2.1 that identify new genes relevant to CVDs relying solely on topological properties of entire PPI network. For example, Zhang et al. [125] introduced a computational method based on six network topological features (degree, neighbour count of disease genes, ratio of disease genes among neighbours, betweenness centrality, clustering coefficient, mean shortest path length to disease gene), and constructed a combined classifier to predict candidate genes for coronary artery diseases. There is huge potential in analysing CVD-related molecular sub-networks and their topology in the context of complete biomolecular interaction networks. Such approaches could give better insight into the interconnectedness of different CVDs. They could help discover novel CVD genes and the pathways responsible for the dependency between different disorders.

Network	Type of data/interactions in the network	Topological analysis performed on the data	Aims of topological analysis	Ref.
Heart failure (HF) network	HF relevant genes, genes differentially expressed in HF and dilated cardiomiyopathy (DCM), PPI data	Connectivity of nodes	Relationship between gene connectivity and gene co-expression levels and their biological functions	[126], [127]
Network of atherosclerosis	Literature associations, gene-expression data	Network modules identified based on closeness centr.	GO enrichment of network modules	[128]
Network of ischemic dilated cardiomyopathy (ICM)	Genes differentially expressed in ICM, cardiac myocytes proteins, PPI data	Number of edges between network clusters	Correlation between number of edges between network clusters and differential gene expression patterns	[129]
CVD "functional linkage network" (CFN)	CVD proteins, PPI data	Degree distribution, betweenness centr., modularity measure	Associating functional modules (highly connected sub-graphs) with diseases	[130]
Congenital hearth disease (CHD) network	Known CHD genes, genes differentially expressed in CHD, PPI data	Sub-networks based on shortest paths and current flow (network was modelled as an electrical circuit)	Functional sub-network analysis in search of key pathways of CHD	[131]
Networks for analysis of cardiac development, hypertrophy and failure	Gene co-expression data	Network modules based on hierarchical clustering and shared network neighbours	Identifying common modules in networks of different type of myocardial tissue	[132]
Human PPI network	PPI data	Node degree, neighbourhood enrichment, betweenness centr., clustering coef., shortest path length	Inferring coronary artery disease genes based on topological information	[125]
Human PPI network	PPI data	Clustering nodes based on graphlet degree vector similarity	Inferring new CVD genes based on clusters' enrichment in CVD genes	[8]

Table 2.1. Methods that explore the topology of biological networks in CVD research.

In the next section, we present two studies that use the topology of biological net-

works for inferring proteins' involvement in CVDs [8] and for explaining the mechanisms behind the interplay of CVDs and diabetes [99].

2.2 CVD Case Studies

2.2.1 Network Topology Reveals Key Cardiovascular Disease Genes

We explore the relationship between the wiring around proteins (we use terms protein and gene interchangeably) in the human PPI network and their involvement in CVDs. In particular, we find clusters of proteins with similar wiring to the proteins already known to be involved in CVDs and identify a consensus set of CVD genes from clusters that are statistically significantly enriched in CVD-related genes. Then, to validate potential gene candidates that might drive CVD onset and progression and are drug targets, we find that this consensus set of genes is enriched in drug targets and driver genes and that it has a large overlap with the Core Diseasome [93]. We also find that many of these genes are functionally similar to known CVD drug targets. Hence, we call this consensus set *Key CVD Genes* and we use the same methodology to predict new CVD gene candidates. We validate that the predicted genes are functionally similar to currently known CVD drug targets, indicating that our methodology may be used for finding new genes relevant for CVD therapy (see paragraph Therapeutic Properties of Key and Predicted CVD Genes).

2.2.1.1 Methods

Our methodology for identifying the key CVD and prediction of new therapeutically relevant CVD genes is shown on the flowchart in Figure 2.2). Below, we describe all the steps taken in more detail.

Data sets. We use the latest human PPI network data from I2D, version 2.0.0 [1], because I2D integrates most of the available PPI data [2]. We remove all self-interactions, as well as any low confidence (originating from only one source) and predicted interactions. To further reduce noise in the PPI network, we remove all proteins with degree lower than 4, since their low connectivity may be a result of a lack of experiments performed

[1] http://ophid.utoronto.ca/
[2] http://ophid.utoronto.ca/ophidv2.204/statistics.jsp

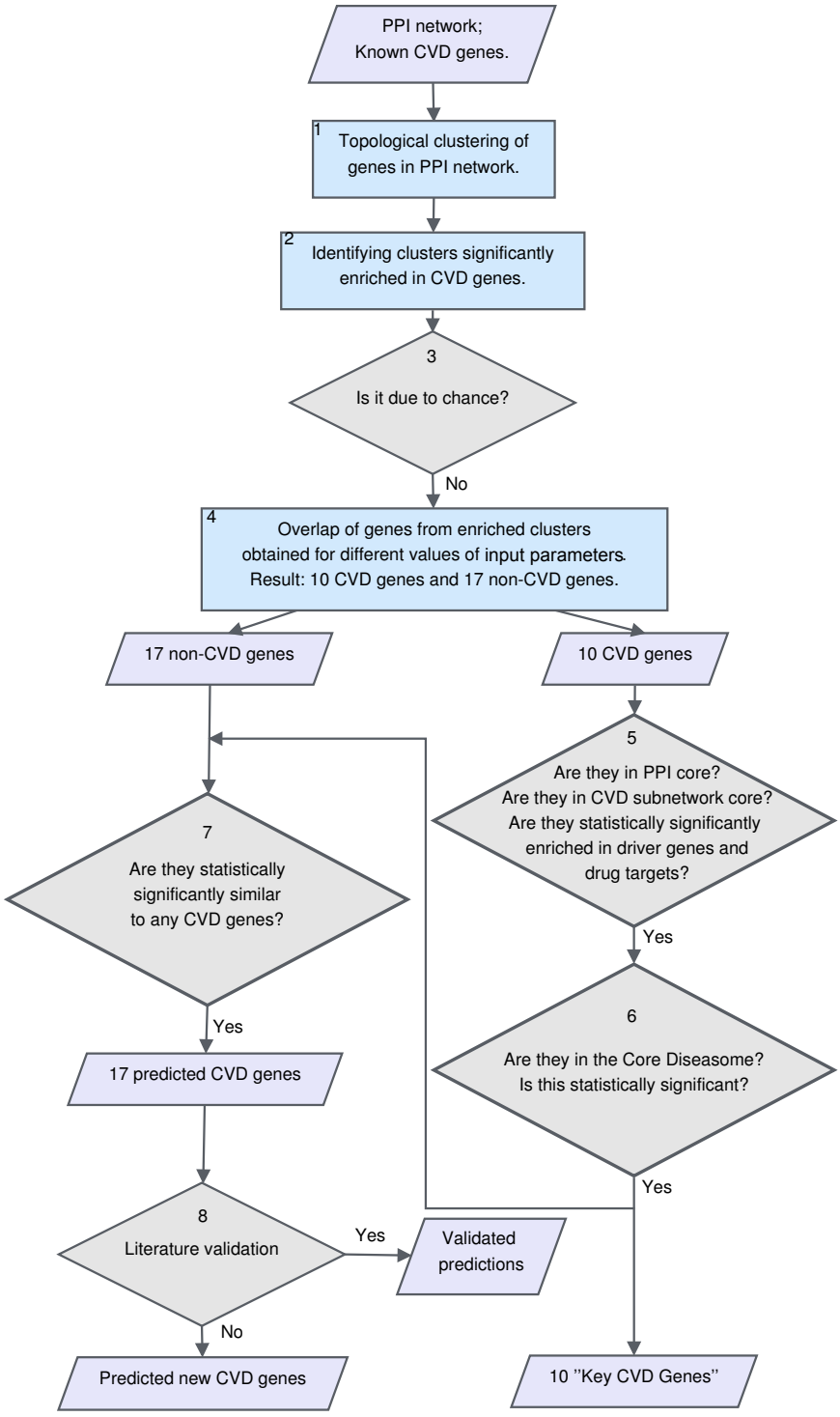


Figure 2.2. Flowchart of our approach. Parallelograms denote inputs and outputs. Rectangles denote analyses. Rhombuses denote choices to be made. Figure is taken from Sarajlić *et al.* [8].

for detecting their interactions, i.e. they may be involved in false negative interactions. The resulting human PPI network has 82,649 interactions between 7,551 proteins.

We obtain the list of genes involved in CVDs from two sources to increase coverage: (1) Disease Ontology (DO) Lite[3] [133] and (2) pathways from KEGG database, downloaded in September 2012. The list includes genes known to be involved in the following CVDs in DO: aortic-aneurysm, atherosclerosis, brain-ischemia, cardiovascular-disease, cerebrovascular disorder, heart-disease, heart-failure, intermediate-coronary-syndrome, ischemia, moyamoya-disease, pseudoxanthoma-elasticum (which later may result in the form of premature atherosclerosis), stroke, Takayasu's-arteritis, thrombophilia, thrombophlebitis, vascular-dementia, vascular-disease, and vasculitis. We obtain additional genes from the following KEGG pathways: hypertrophic cardiomyopathy, arythmogenicright ventricular cardiomyopathy, dilated cardiomyopathy, and viral myocarditis. This results in the set of 656 CVD-related genes, out of which we analyse 423 genes that are present in human PPI network.

We download the drug target data from DrugBank[4]: there are 1,245 drug targets in our PPI network, among which 199 are known CVD genes.

Similarity measure. We use GDV similarity [4] to measure the topological similarity between two proteins in the PPI network. As mentioned in the Chapter 1, GDV similarity between proteins in the human PPI network has already been used to successfully predict protein function and involvement in disease [4–6,45]. Here, we examine its usability for predicting CVD-related genes. We use it to make clusters of proteins with similar wiring in the PPI network as described below.

Clustering methods. By using the above described GDV similarity between proteins in the human PPI network, we obtain clusters of proteins with similar wiring around them in the PPI network. Clustering is a hard problem and a major research area in its own right. Some clustering methods, such as K-nearest neighbours (KNN), produce overlapping clusters, while others, such as K-medoids, or Hierarchical clustering, produce clusters with non-overlapping sets of elements. We use a method that produces non-overlapping clusters to avoid enrichments in clusters that are due to cluster overlap. Note that the success of a particular clustering method depends on the input data and can be different for different networks [134]. Discussing the reasons for different performance of different clustering methods is beyond the scope of this thesis. Since the

[3] http://django.nubic.northwestern.edu/fundo/
[4] http://http://www.drugbank.ca/

choice of the best clustering method is data dependent, we try two methods described below (step 1. in Figure 2.2).

Hierarchical clustering (HIE). This method creates a dendrogram that represents a cluster tree, which is a multilevel hierarchy meaning that clusters at one level of the hierarchy are joined into a cluster at the next level. The process of creating clusters starts by assigning each node to its own cluster and follows by finding the "closest" pair of clusters to merge into a single cluster. Recall that we specify the closeness between a pair of nodes based on their GDV similarity. If there are many closest pairs, a single pair is chosen randomly. Then, we compute the "closeness" between the newly formed cluster and each of the old clusters as the average of GDV similarities between the nodes of the clusters. Again, the closest pair of clusters is merged into a single cluster. This process repeats until all nodes are clustered into one cluster. In order to create the desired number of disjoint clusters it is necessary to cut the hierarchical tree at some point. We denote the minimal number of clusters that are obtained with a cut by K_H.

K-medoids clustering (KM). A *medoid* is a node in a cluster whose average distance to all other nodes in the cluster is minimal. The algorithm randomly picks K_{KM} nodes as cluster medoids and assigns all remaining nodes to K_{KM} clusters. Each node is assigned to the cluster with the medoid minimally distant from the node in question. Ties are broken randomly. Then, in each cluster, a new medoid node is found with respect to the nodes of the cluster. All non-medoid nodes in the network are then reassigned to new K_{KM} clusters with these new medoids. These steps are repeated until the same set of nodes is chosen as cluster medoids.

Finding statistically significantly enriched clusters. For each cluster obtained by using each of the clustering methods described above, we compute the enrichment in CVD-related proteins. We compute statistical significance (p-value) of obtaining this or higher enrichment purely by chance. The p-value is computed using the hypergeometric cumulative distribution as follows. We denote the number of genes in the human PPI network with M, the number of genes that are involved in CVDs with K, and the size of the cluster in question with N. The p-value, or the probability that X or more disease genes will be found in the cluster by chance, is computed as follows:

$$p = 1 - \sum_{i=0}^{X-1} \frac{\binom{K}{i}\binom{M-K}{N-i}}{\binom{M}{N}}. \tag{2.1}$$

We apply *Benjamini-Hochberg false discovery rate* (FDR) correction [135] on the resulting p-values in order to take into account a possibility of obtaining significant p-values in a large number of experiments purely by chance. We report such corrected p-values. Sensible cut-offs for p-values are in the range from 10^{-2} to 10^{-8} [136]. We use the p-value of 0.01 as a cut-off to define clusters statistically significantly enriched in CVD-related genes.

First, we apply Hierarchical clustering to our PPI network. In different runs of the algorithm, we choose the minimum number of resulting clusters K_H to be: 50, 75, 100, 200, 500, 700, 1000 and 2000. These numbers are chosen to cover different sizes of clusters in order to identify the optimal size at which the enrichment in CVD genes would occur. Unfortunately, the obtained clusters were not statistically significantly enriched with CVD genes, indicating that HIE can not be used for obtaining clusters of CVD-enriched genes purely from the topology of the PPI network.

KM method produced clusters of proteins statistically significantly enriched in CVD genes. The number of medoids, and therefore clusters, K_{KM}, that we use are: 50, 75, 100, 200, 300, 500, 700 and 1000. K_{KM} larger than 1000 caused clusters to be too small for any statistical analyses. The obtained clusters depend on the initial random choice of medoids, as previously explained. Hence, for each value of K_{KM} mentioned above, we repeat the experiment five times. To increase coverage, we take a union of genes that are found in statistically significantly enriched clusters for all five experiments per choice of K_{KM}(step 2 in Figure 2.2). As a result, in CVD enriched clusters we identify following gene sets:

- For $K_{KM} = 50$: 86 CVD genes and 572 non-CVD genes;
- For $K_{KM} = 75$: 48 CVD genes and 282 non-CVD genes;
- For $K_{KM} = 100$: 54 CVD genes and 282 non-CVD genes;
- For $K_{KM} = 200$: 75 CVD genes and 277 non-CVD genes;
- For $K_{KM} = 300$: 13 CVD genes and 40 non-CVD genes;
- For $K_{KM} = 700$: 17 CVD genes and 23 non-CVD genes.

To find the "most important" CVD genes, we apply an additional filter: we seek CVD genes that are in the intersection of the above gene sets, obtained from statistically significantly enriched clusters for different values of K_{KM}(step 4 in Figure 2.2). We find 10 such genes (listed in Table 2.2) and analyse them further (see below).

Gene	GO term	CVD
ABL1	Intracellular signaling cascade (BP), Signal transducer activity (MF)	Viral myocarditis.
SHC1	Intracellular signaling cascade (BP), Signal transducer activity (MF)	Atherosclerosis.
SP1	Enzyme binding (MF)	Trombophlebitis.
AR	Intracellular signaling cascade (BP), Intracellular receptor-mediated signaling pathway (BP), Signal transducer activity (MF)	Atherosclerosis.
CTNNB1	Intracellular signaling cascade (BP), Intracellular receptor-mediated signaling pathway (BP), Enzyme binding (MF), Signal transducer activity (MF)	Arythmogenic right ventricular cardiomyopathy(ARVC).
FYN	Intracellular signaling cascade (BP)	Viral myocarditis.
ACTB	Enzyme binding (MF)	Arythmogenic right ventricular cardiomyopathy(ARVC), Hypertrophic cardiomyopathy (HCM), Viral myocarditis, Dilated Cardiomyopathy (DCM).
HDAC5		Heart failure.
EGFR	Intracellular signaling cascade (BP), Enzyme binding (MF), Signal transducer activity (MF)	Trombophlebitis, Stroke.
ESR1	Intracellular signaling cascade (BP), Intracellular receptor-mediated signaling pathway (BP), Signal transducer activity (MF)	Stroke, Atherosclerosis, Cerebrovascular disorder.

Table 2.2. Functional annotation of the ten key cardiovascular disease genes. First column: ten *key* CVD genes. Second column: GO terms that the genes are annotated with. We only take into consideration GO terms in which this set of 10 genes is statistically significantly enriched and that correspond to biological functions that the three drug mechanisms of interest rely on. BP denotes biological process, MF denotes molecular function of GO. Third column: CVDs that the genes are associated with.

Finding the core of the cardiovascular diseasome. We apply the k-core decomposition algorithm to the human PPI network [95, 96]. Recall that The Core Diseasome is obtained purely computationally by computing the k_{max}-core decomposition of the human PPI network, along with the k_{max}-core decomposition of its sub-network of only disease genes, described in [93]. Therefore, to investigate the importance of the 10

above-described CVD related genes, we find the core of the human PPI network and check if these 10 genes are in it. Also we find the core of the PPI sub-network consisting only of CVD related genes, and we check if this set of 10 genes appears in it (step 5 in Figure 2.2).

Gene name	GO term	PubMed ID
CREBBP	Receptor binding (MF), Signal transduction (BP).	14724353
MDM2	Enzyme binding (MF).	18375498
HDAC1	Enzyme binding (MF).	22226905
SMAD3	Enzyme binding (MF), Receptor binding (MF), Enzyme linked receptor protein signalling pathway (BP).	22167769, 22633655
SMAD2	Enzyme binding (MF), Receptor binding (MF), Signal transduction (BP), Intracellular signalling cascade (BP), Enzyme linked receptor protein signalling pathway (BP).	20829218, 22049534
JUN	Signal transduction (BP), Response to drug (BP), Enzyme linked receptor protein signalling pathway (BP).	22664133
BRCA1	Enzyme binding (MF), Receptor binding (MF), Signal transduction (BP), Intracellular signalling cascade (BP).	22186889
MYC		22402364
SRC	Signal transduction (BP), Intracellular signaling cascade (BP), Enzyme linked receptor protein signalling pathway (BP).	22287273
EP300	Receptor binding (MF), Signal transduction (BP), Response to drug (BP).	20375365
TP53	Enzyme binding (MF), Signal transduction (BP), Intracellular signalling cascade (BP), Response to drug (BP).	23074332, 22189267
GRB2	Receptor binding (MF), Signal transduction (BP), Intracellular signaling cascade (BP), Enzyme linked receptor protein signalling pathway (BP).	12639989
IKBKG	Signal transduction (BP), Intracellular signal. cascade (BP).	—
HSP90AA1/2	Signal transduction (BP).	—
PIK3R1	Enzyme binding (MF), Receptor binding (MF), Signal transduction (BP), Intracellular signalling cascade (BP), Enzyme linked receptor protein signalling pathway (BP).	—
YWHAZ	Signal transduction (BP), Response to drug (BP).	—
YWHAQ	Signal transduction (BP), Intracellular signalling cascade (BP).	—

Table 2.3. Predicted CVD genes. First column: predicted CVD genes. Second column: GO annotations of the gene. Third column: Validation that the predicted gene is associated with a CVD – PubMed ID of the reference; "—" means that we found no literature validation.

Since the core of the PPI network is known to contain driver genes and drug targets

[93], we examine if any of the 10 genes are among the 15 known driver genes, or are drug targets [94, 137–139] (step 5. in Figure 2.2). We obtain statistically significant findings (detailed in section 2.2.1.2), which allow us to postulate that these 10 genes are the *Key CVD Genes*. We further successfully validate this by checking the statistical significance of the overlap between Key CVD Genes and the Core Diseasome [93] (step 6 in Figure 2.2).

Predicting New CVD Genes. We use the above-described method (steps 1-4 in Figure 2.2) to predict novel CVD genes. We consider the 17 genes not currently known to be involved in CVDs, that are in clusters statistically significantly enriched in CVD genes, regardless of the value of the initial parameter K_{KM}. Table 2.3 lists this set of 17 genes, together with the GO terms in which the set is statistically significantly enriched in and which correspond to biological functions that the three drug mechanisms of interest rely on (which will be discussed later on in Section 2.2.1.2). Note that these 17 genes may have various GDV similarity to CVD genes, since all genes had to be assigned to clusters. Hence, we seek only genes that are statistically significantly similar in topology to CVD genes. To do that, we compute the distribution of GDV similarities of all pairs of proteins in the human PPI network (Figure 2.3). The top 1% of the most GDV-similar nodes have GDV similarity of at least 89% (corresponding to p-value of 0.01). Hence, amongst the 17 non-CVD genes, we look for those that are at least 89% GDV-similar to a CVD gene (step 7 in Figure 2.2).

2.2.1.2 Results and Discussion

Here, we first reason about the importance of the 10 CVD genes identified by our methodology (listed in Table 2.2). Then, we validate our predicted CVD genes (listed in Table 2.3). Next we explain the therapeutic potential of the identified genes and provide a comparison with other approaches. The results are summarised in Figure2.4.

The Key Cardiovascular Disease Genes. We examine the importance of the 10 key CVD genes as described in paragraph The Core of Cardiovascular Diseasome. We ask if they are in the k_{max}-core of the PPI network and the k_{max}-core of the PPI sub-network of CVD genes only (steps 5-6 in Figure 2.2), and if they are enriched in drug targets and driver genes.

We compute the k_{max}-core decomposition of the PPI network using the algorithm described in Section 1.3.1: it consists of 372 proteins (recall that the entire PPI network has 7,551 proteins). There are 44 genes in the intersection between these 372 proteins

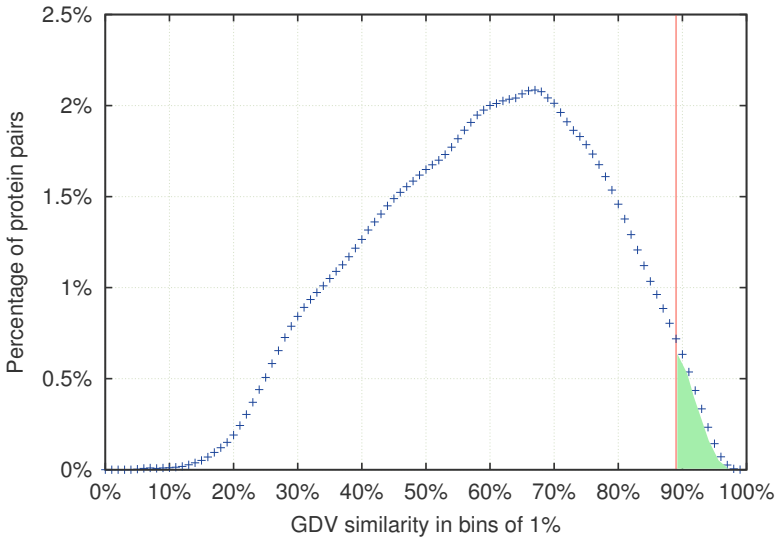


Figure 2.3. The distribution of GDV similarity of protein pairs in the human PPI network. Horizontal axis represents GDV-similarities of node pairs in the network in bins of 1%. Vertical axis represents percentages of protein pairs that have a particular GDV-similarity. The red line marks the threshold value of GDV similarity (89%): all pairs of nodes with GDV similarity above this threshold correspond to the top 1% of the most GDV-similar node-pairs (the area shaded in green represents 1% of the area under the distribution curve). The figure is taken from Sarajlić *et al.* [8].

and the entire set of 423 CVD proteins in the PPI network. Interestingly, all 10 key CVD genes, are among these 44 CVD -related genes that are in the core of the human PPI network. We calculate p-value for this to occur using the hypergeometric cumulative distribution with respect to entire human PPI network and with respect to 423 CVD-related genes. We find that both p-values are statistically significant, the first being $7.5 \cdot 10^{-14}$ and the second being $5.5 \cdot 10^{-11}$. Furthermore, the connected sub-network of the PPI network that consists only of CVD-related genes has 362 proteins, and its core consists of 43 genes. Again, all 10 key CVD genes are in this core (p-value $= 2 \cdot 10^{-10}$ with respect to the 362 CVD proteins).

Also, three of the key CVD genes: ABL1, CTNNB1, and EGFR, are among the 15 known driver genes (taken from [93]). The two p-values, computed as described above are $7.5 \cdot 10^{-7}$ (with respect to entire PPI network), and $1.85 \cdot 10^{-4}$(with respect to 423 CVD genes).

We find that six out of the 10 genes are among the 1245 known drug targets that are present in the human PPI network. Table 2.4 lists key CVD genes that are known drug

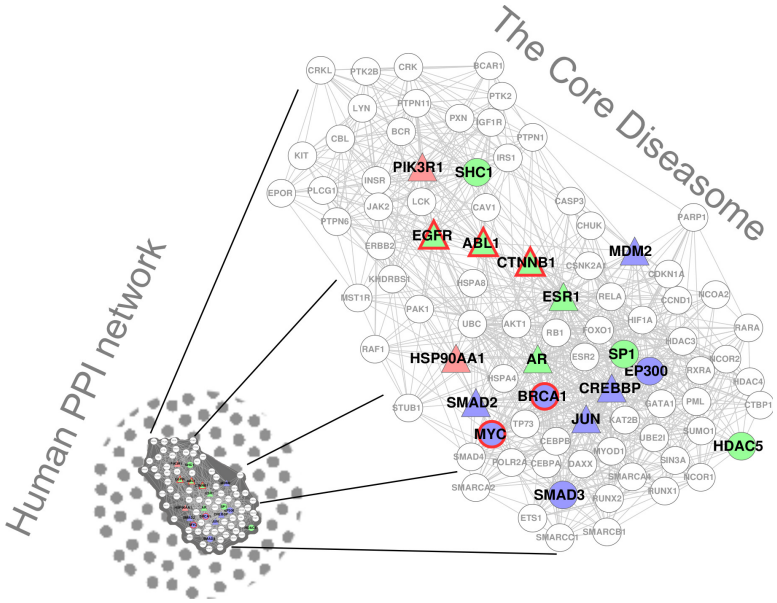


Figure 2.4. Summary of the results. The Core Diseasome of [93] is overlaid with the results of this study. Green nodes are the Key CVD Genes (from Table 2.2), which are in the Core Diseasome. Blue nodes are predicted CVD genes (from Table 2.3) that we validated in the literature and that are in the Core Diseasome. Red nodes are non-validated CVD gene predictions (from Table 2.3) that are in the Core Diseasome. Triangular nodes are drug targets. Driver genes are bordered in red. Figure is taken from Sarajlić *et al.* [8].

targets and number of drugs from Drugbank that target the corresponding gene. Since 199 out of 423 CVD genes in PPI network are known drug targets, the p-value of getting 6 to occur amongst 10 key CVD genes is not statistically significant. However, with respect to entire PPI network, this finding is statistically significant (p-value = 0.0023).

Entrez ID	Gene name	Number of drugs
367	AR	40
2099	ESR1	61
25	ABL1	11
1499	CTNNB1	1
2534	FYN	2
1956	EGFR	10

Table 2.4. The Key Cardiovascular Disease Genes that are known drug targets. First column: Entrez Gene ID. Second column: Official Gene Symbol. Third column: the number of drugs from Drugbank that target the corresponding gene.

We further validate the importance of our key CVD genes, by checking if they are a part of the Core Diseasome (step 6 in Figure 2.2). We find that the following 8 out of the 10 key CVD genes are in the Core Diseasome: SHC1, EGFR, ABL1, CTNNB1, ESR1, AR, SP1, HDAC5 (Figure2.4). We check the probability for this or higher enrichment to occur purely by chance. This overlap is statistically significant with p-values of $9.9 \cdot 10^{-15}$ and $1.3 \cdot 10^{-9}$ respectively (p-values computed as described in the beginning of this section). Note that GDV similarity measure is not necessary for the formation of the Core Diseasome, while the 10 key CVD genes are obtained solely by using GDV similarity. Hence, validating the importance of key CVD genes by checking their overlap with the Core Diseasome is not computationally biased.

Validation of CVD Gene Predictions. We predict 17 new CVD genes, listed in Table 2.3, as the result of the same methodology that we used to identify the key CVD genes. We confirm that all of the 17 predicted genes are statistically significantly similar to some of the CVD genes.

To validate our predictions, we perform literature curation for possible CVDs that these 17 genes may be involved in. Later on, we also examine the therapeutic potential of these predictions.

We perform the literature validations by text mining using CiteXplore[5]: for the 17 predicted genes, we search PubMed abstracts with CiteXplore using their official gene symbols. In Table 2.3, we list the PubMed IDs of the literature references for the 12 genes that we found a validation for their involvement in CVDs. For more details on the biological mechanisms for these findings please see Supplementary of the dissertation, Section A.1.

For genes IKBKG, HSP90AA1/2, PIK3R1, YWHAZ, and YWHAQ, we found no evidence in literature for their link to cardiovascular diseases. However, due to the high literature validation score of our CVD gene predictions (over 70% of our predictions are successfully validated in the literature), we predict that these genes are also involved in the processes related to cardiovascular diseases (step 8 in Figure 2.2). Two of these genes (PIK3R1 and HSP90AA1) are part of the Core Diseasome, as shown in Figure 2.4. PIK3R1 is associated with cancer and over-nutrition, while HSP90AA1 is associated with Alzheimer's disease, cancer, eating disorder, herpes, and Fanconi's anemia.

Therapeutic properties of key and predicted CVD genes. The five most common mechanisms by which drugs work are: (1) antibiotics, which disrupt bacterial

[5]http://www.ebi.ac.uk/citexplore/

cells causing them to die, or interfere with their essential reproduction machinery; (2) replacement drugs, which work by replacing substances missing from the body; (3) enzyme-acting drugs, which modify the enzymatic activity; (4) receptor-acting drugs, that either deliberately trigger cell surface receptors to activate the signalling machinery, or bind to those receptors to prevent ligands from performing their intended function; and (5) inter-cellular transport altering drugs, which modify the flow of molecules to and from a cell, thus changing their chemical composition and hijacking communication channels. Currently, therapeutic treatment of CVDs is achieved through drug mechanism types (3), (4) and (5) [140–142], while (1) is argued to have non-beneficial, or even harmful effects in treatment of CVDs [143]. This means that to be a CVD drug target, a protein would need to have a biological function that would facilitate the workings of the three above-mentioned drug mechanism types, (3), (4) and (5).

We use DAVID online tool[6] to calculate Gene Ontology (GO) terms enrichments for the set of 17 predicted CVD proteins and the set of 10 key CVD proteins. We upload each gene set separately to DAVID and use the entire set of human genes as a background set. We consider GO terms that correspond to enrichments that have p-values ≤ 0.05 after the *Benjamini-Hochberg false discovery rate* (FDR) correction is applied. We find that the 10 key CVD genes are statistically significantly enriched in the following GO terms which correspond to biological functions that the three drug mechanisms discussed above rely on: intracellular signalling cascade, intracellular receptor-mediated signalling pathway, signal transducer activity, and enzyme binding. We list these GO terms with their corresponding genes in Table 2.2. We find that the 17 predicted genes are statistically significantly enriched with the following GO terms which correspond to biological functions that the three drug mechanisms discussed above rely on: intracellular signaling cascade, signal transduction, enzyme linked receptor protein signalling pathway, response to drug, enzyme binding, and receptor binding. We list these GO terms with their corresponding genes in Table 2.3. We also check 199 known drug targets among CVD genes and find that they are statistically significantly enriched, with p-values ≤ 0.05, in biological functions that we list in Tables 2.2 and 2.3. This indicates that our methodology identifies important drug targets.

Comparison with other approaches. Our methodology is based solely on network topology. In particular, we rely on GDV similarity between proteins in the PPI network because, as discussed in the Introduction, this measure was shown to be a good indicator that proteins perform similar biological functions and are involved in the same

[6] http://david.abcc.ncifcrf.gov/

diseases [4, 144] and was used for predicting new melanogenesis related genes that were phenotypically validated [5]. GDV similarity was also shown to be robust to random addition, deletion and rewiring of up to 30% of edges in PPI networks [144]. In particular, none of these edge perturbations introduced a significant change in the distribution of GDV similarities of orthologous protein pairs in human PPI network - these proteins had higher signature similarities than randomly chosen protein pairs in the PPI network, regardless the noise. Robustness to noise in PPI networks for graphlet-based measures will be discussed in more detail in Section 2.2.2.1.

Here, we compare GDV similarity with baseline network-topology-based approaches to justify the use of GDV similarity for analysing this particular dataset. We examine clustering of proteins in the PPI network based only on the degrees (i.e. connectivity) of the nodes in the network. This method fails to identify any clusters statistically significantly enriched in CVD genes. Since the guilt-by-association approach, based on protein interactors (neighbours) has become a relatively standard approach [1–3, 145], we try to use it to identify "key" CVD genes. Hence, we look for statistically significant enrichment in CVD genes among the neighbours of each CVD gene in the network. There are 134 CVD genes that interact with sets of genes statistically significantly enriched in CVD genes. Therefore one may expect that these 134 CVD genes may be "key" for disease onset and therapy. Unfortunately this is not a case: this set of 134 genes is not statistically significantly enriched in the driver genes. Furthermore, it has no statistically significant overlap with the Core Diseasome and k_{max}-core of the PPI network. Hence, guilt-by-association can not be used to define key CVD genes.

To verify that our methodology did not produce statistically significantly enriched clusters purely by chance, we randomised the topology of the PPI network respecting the degree distribution (by relabelling the nodes in the network) and performed the above described analysis on randomised networks (step 3 in Figure 2.2). We repeated the randomisation 30 times both for KM and HIE clustering. This did not yield any clusters statistically significantly enriched in CVD genes, which shows that specific topology around genes in the PPI network is a major contributor to identifying key CVD genes and making predictions.

Note that one of the aims of this chapter was to show that new biological knowledge can be extracted solely from the topology of biological networks. The analysis of all CVD genes and prediction of new ones has not previously been done using only network topology, that is, our study is the first to use only topology to examine the importance of CVD genes and predict new ones.

2.2.2 Network Wiring of Pleiotropic Kinases Yields Insight into the Relationship between Diabetes and Aneurysm

Abdominal aortic aneurysm (AAA) is a permanent dilatation of the abdominal aorta and a leading cause of death amongst the older population [146]. Several studies suggest diabetes plays a protective role against the development of aneurysm [147,148]. De Rango *et al.* showed that the progression of small AAA is 60% lower in patients who suffer from diabetes [148]. Prakash *et al.* also confirmed that diabetes is associated with decreased rate of hospitalization due to thoracic aortic aneurysms (TAAD) [147]. This seems paradoxical, as diabetes is known to predispose cardiovascular diseases: peripheral, coronary, and cerebrovascular diseases [148, 149]. Also, vascular diseases are the principal cause of death and disability in people with diabetes and a common macro-vascular manifestation for this is atherosclerosis [150]. Note that atherosclerosis shares similar risk factors with aneurysm, such as male gender, higher age, hyperlipidaemia, and hypertension, and as such was considered as an underlying pathogenesis in AAA [146,151]. However, a decreased prevalence of AAA in patients with diabetes may suggest that atherosclerosis is an associated feature and not a cause of aneurysm [146]. Hence, we explore the possible mechanisms behind the protective role of diabetes on the development of aneurysm and why there is no similar diabetes-atherosclerosis association, as published work in this area is still inconclusive [148]. Therefore, to tackle the problem we use high-throughput molecular network data.

We use both the human PPI network and the genetic interaction network to find an explanation for the protective role of diabetes on aneurysm and why a similar relationship is not present in the case of diabetes and atherosclerosis. We hypothesise that a functional change of a protein on a pathway that is important for aneurysm could disrupt this pathway, thus preventing the onset of the disease. We suspect that a mutation of a gene on a pathway involved in diabetes is related to a functional change of a protein on an aneurysm-related pathway, explaining the protective role of diabetes on the development of aneurysm. To this end, we integrate PPI data with information from the human genetic interaction network. In a genetic interaction network nodes correspond to genes in the network and edges represent functional associations between them: an interaction between two genes occurs when the result of simultaneous mutations in the genes is not just a combination of phenotypes of single mutations [82]. It has been shown that genetic interactions are critical for understanding disease evolution [94] and a key to capturing disease-disease associations from molecular interaction data [152]. Although by definition a genetic interaction between two genes does not indicate a di-

rect interaction, it can indicate how strongly the function of one gene depends on the presence of the other, i.e., it can indicate how much the phenotype of one mutation is modified by the presence of the second mutation [153]. Even the order in which mutations occur in some cases is likely defined by the genetic interactions [94]. One such example in cancer progression is when P53 dysfunction usually precedes BRCA loss of function generating synthetic viability [94]. In the case of genetic interaction between genes whose protein products directly interact, a mutation in one protein that affects a physical interaction can be compensated by a mutation of its interacting partner, for example, proteins S12 and L19 in *Salmonella typhimurium* [154].

2.2.2.1 Methods

The complete methodological work-flow of this study is presented in Figure 2.5. We first identify pathways that play a role in formation of the three diseases. Then, we use information from the human genetic interaction network to single out pathways that contain genes (henceforth, we use terms protein and gene interchangeably), which take part in genetic interactions such that one interacting gene is part of a diabetes-related pathway while the other is part of an aneurysm- or an atherosclerosis-related pathway. We use selected pathways to create a disease-related sub-network of the human PPI network.

In search of genes whose change in functionality could disrupt a pathway, we rely on the network topology and look for genes in this disease PPI sub-network with a local topology that could explain a gene's high "destructiveness" for the related pathway—a set of "broker" genes. This set is statistically significantly enriched in biological functions that facilitate mechanisms that have already been suggested as possible causes of diabetes-aneurysm dissociation. We narrow down this set to 16 genes that are on aneurysm- or atherosclerosis-related pathways and participate in genetic interactions with genes from diabetes-related pathways. We find this set to be enriched in kinases and in biological function of phosphorylation. This confirms our hypothesis that identified proteins could disrupt the pathways, in particular, kinases can switch on and off proteins on an aneurysm-related pathway, which can lead to prevention of aneurysm formation. Importantly, two kinases from the set that are on both aneurysm- and atherosclerosis-related pathways are pleiotropic, explaining why a mutation of such genes could disrupt an aneurysm-related pathway but not affect the atherosclerosis-related pathway.

As discussed in Section 1.5, all currently available human PPI networks represent just a fraction of the complete networks [75] and this incompleteness is reflected in the over-

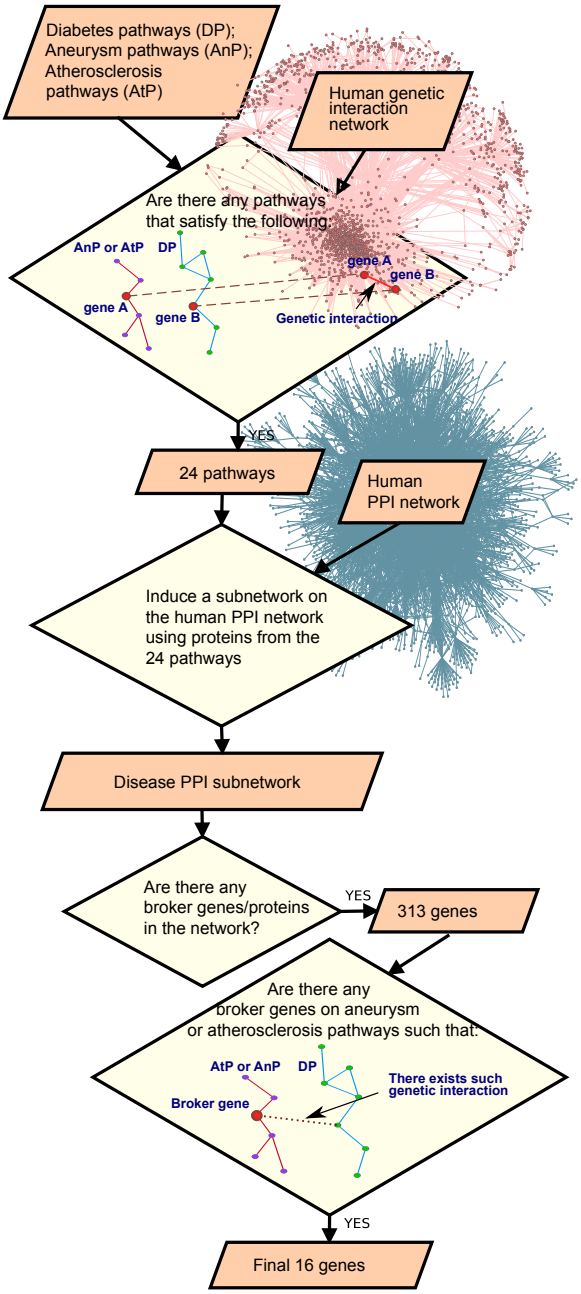


Figure 2.5. Work-flow of the study. Figure is taken from Sarajlić *et al.* [99].

all network topology [76]. It is therefore easy to question the validity of topology-based analyses and results obtained on such incomplete networks. Nevertheless, graphlet-based methods for network topology analysis were shown to be robust to different types

of noise in a network, including network incompleteness [13,144,155]. Namely, Yaverŏglu *et al.* [13] systematically compared robustness to noise of a number of topological similarity measures: GDDA, RGFD, GCD-73, GCD-11, degree distribution distance, network diameter, clustering coefficient and spectral distance. Noise robustness was evaluated through the clustering of synthetic networks based on topological similarity when up to 90% of edges in the networks are randomly removed or rewired. Also, to account for the fact that many real world networks are both noisy and incomplete, 40% of edges was first randomly rewired in the networks and than randomly removed in the increments of 10%. Graphlet-based measures showed high levels of noise robustness, with GCD-11 outperforming all other measures even for networks with 40 % of rewired and as much as 80% of missing edges [13]. Interestingly, it was also shown that the performance of GCD-11 only slightly decreases if as little as 30% of all graphlet degree vectors chosen at random were used to form GCM-11 [13]. Robustness of GDVs to noise in the human PPI network was also tested by comparing the distribution of GDV similarities between orthologous proteins in the network against the distribution of GDV similarities between all proteins in the network when up to 30% of edges were added, deleted or rewired at random [144]. None of these types of noise introduced any significant change in the distribution of GDV similarities and orthologous proteins remained having statistically significantly higher GDV similarities than all other protein pairs in the PPI network [144]. This is why we are confident that we can use topology based analysis on the available biological networks, although noisy and incomplete, to address open questions in biology and medicine. We are also motivated by a wide range of biologically relevant and validated findings that were obtained through topological analysis of such networks. Graphlet-based measures, for example, were used to predict new melanogenesis related genes which were phenotypically validated [5, 6]. Also, experimentally confirmed downstream signaling mechanism of nicastrin in breast cancer cells was identified using topological analysis of currently available human PPI network [156]. Below we discuss our methodology in more detail.

Datasets: Biological networks. We obtain the human PPI network from BioGRID [157], release 3.2.106, September 2013. We analyze the largest connected component of the network. To reduce noise we remove ubiquitin as the most connected protein in the network, since proteins with a large number of non-specific interaction partners might seriously bias the network topology leading to biased results. The resulting PPI network has 13,410 proteins (nodes) and 116,552 interactions (edges).

We download the human genetic interaction (GI) network from BioGRID in Septem-

ber 2013 (release 3.2.106). The network contains 986 genes and 1,295 genetic interactions. To increase coverage we also construct a *predicted* human GI network using new GI data on direct positive and negative genetic interactions in *S. cerevisiae* from Boone Lab[7], that they gave to us in September 2013. [8]). The yeast GI network contains 4,365 genes and 266,750 interactions. Then, we use information on homologous genes between *H. sapiens* and *S. cereviseiae* from Homologene database[9], version *build67*, downloaded in January 2014. There are 1,568 human genes that are yeast homologs. We create a *predicted* human GI network as follows: for each genetic interaction between yeast genes, we create a genetic interaction between their corresponding human homologs. This network of predicted human genetic interactions contains 1,088 genes and 34,160 genetic interactions between them. We merge the human GI network from BioGRID with the predicted human GI network, resulting in the final network of human genetic interactions containing 1,983 genes and 35,454 interactions. In this manuscript we refer to this network as the human genetic interaction (GI) network.

Datasets: Disease genes. We obtain a list of genes involved in aneurysm using several sources to increase coverage: KEGG DISEASE database [158], OMIM database [68] and Disease Ontology Lite[10]. We find in total 53 genes related to aneurysm, out of which 37 are present in the human PPI network. We find genes involved in atherosclerosis in the OMIM database and Disease Ontology (DO) Lite. We find in total 205 atherosclerosis related genes, out of which 184 are present in the human PPI network. We obtain genes involved in diabetes from KEGG DISEASE database, OMIM database and Disease Ontology Lite. To increase coverage, we also include genes from the following pathways in the KEGG PATHWAY database: Type I diabetes mellitus, Type II diabetes mellitus, and Maturity onset diabetes of the young. We find in total 503 diabetes genes, out of which 423 are present in the human PPI network. All data on disease genes are downloaded in November 2013. We then identify pathways that play a role in formation of the three diseases as follows.

[7] http://www.utoronto.ca/boonelab/

[8] We thank Charlie Boone for giving us his unpublished complete set of genetic interactions in baker's yeast.

[9] http://www.ncbi.nlm.nih.gov/homologene

[10] http://django.nubic.northwestern.edu/fundo

Pathway	KEGG ID	p-value
Pathways in cancer	hsa05200	2.1×10^{-3}
Cytokine-cytokine receptor interaction	hsa04060	4.5×10^{-3}
Vascular smooth muscle contraction	hsa04270	1.2×10^{-2}
Intestinal immune network for IgA production	hsa04672	1.9×10^{-2}
MAPK signaling pathway	hsa04010	2.6×10^{-2}
Viral myocarditis	hsa05416	3.7×10^{-2}
ECM-receptor interaction	hsa04512	5.0×10^{-2}
Colorectal cancer	hsa05210	5.0×10^{-2}

Table 2.5. Pathways related to aneurysm. First column: Pathways that are statistically significantly enriched in genes related to aneurysm. Second column: KEGG ID of the pathway. Third column: p-value of statistical significance of the enrichment.

Datasets: Pathways. We download all pathways relevant for diabetes mellitus from KEGG PATHWAY database in November 2013: Type I diabetes mellitus (hsa04940), Type II diabetes mellitus (hsa04930), and Maturity onset diabetes of the young (hsa04950). These pathways have 47, 48, and 25 genes in the human PPI network, respectively. KEGG PATHWAY database does not list a set of pathways directly related to aneurysm, so we identify pathways that may play a role in formation of this disease by checking the enrichment of all available KEGG pathways in genes known to be involved in this disease. Among all 282 pathways from KEGG, we find 8 pathways that are statistically significantly enriched in aneurysm genes (p-value threshold of 0.05). The obtained pathways and their KEGG IDs are listed in Table 2.5.

Henceforth, we refer to these pathways as "aneurysm pathways". Similarly, we identify 23 "atherosclerosis pathways," listed in Table 2.6.

Pathway	KEGG ID	p-value
Cytokine-cytokine receptor interaction	hsa04060	5.9×10^{-10}
Type I diabetes mellitus	hsa04940	5.9×10^{-7}
Toll-like receptor signalling pathway	hsa04620	9.2×10^{-7}
Hematopoietic cell lineage	hsa04640	4.4×10^{-5}
Allograft rejection	hsa05330	2.2×10^{-4}
Complement and coagulation cascades	hsa04610	2.7×10^{-4}
Graft-versus-host disease	hsa05332	3.4×10^{-4}
NOD-like receptor signalling pathway	hsa04621	7.7×10^{-4}
ECM-receptor interaction	hsa04512	1.0×10^{-3}
Focal adhesion	hsa04510	3.9×10^{-3}
Hypertrophic cardiomyopathy (HCM)	hsa05410	4.8×10^{-3}
Chemokine signalling pathway	hsa04062	6.3×10^{-3}
Intestinal immune network for IgA production	hsa04672	6.9×10^{-3}
PPAR signaling pathway	hsa03320	6.9×10^{-3}
Dilated cardiomyopathy	hsa05414	7.4×10^{-3}
Prion diseases	hsa05020	1.0×10^{-2}
Systemic lupus erythematosus	hsa05322	1.1×10^{-2}
Pathways in cancer	hsa05200	3.1×10^{-2}
Asthma	hsa05310	3.4×10^{-2}
Autoimmune thyroid disease	hsa05320	3.7×10^{-2}
Jak-STAT signalling pathway	hsa04630	3.8×10^{-2}
Arrhythmogenic right ventricular cardiomyopathy(ARVC)	hsa05412	3.9×10^{-2}
Cell adhesion molecules (CAMs)	hsa04514	4.5×10^{-2}

Table 2.6. Pathways related to atherosclerosis. First column: Pathways that are statistically significantly enriched in genes related to atherosclerosis. Second column: KEGG ID of the pathway. Third column: p-value of statistical significance of the enrichment.

Note that the same pathways can be involved in several diseases. For example, cytokine-cytokine receptor interaction pathway hsa04060 is enriched both in aneurysm and atherosclerosis genes (see Table 2.7). This is not specific to diseases that we study here as it is well known that some pathways are involved in many diseases, e.g. MAPK signalling pathway has been involved in many human diseases including Alzheimer's disease, Parkinson's disease, amyotrophic lateral sclerosis and various types of cancers [159].

Pathway name	KEGG ID	Disease
Colorectal cancer	hsa05210	An
MAPK signalling pathway	hsa04010	An
Viral myocarditis	hsa05416	An
Type I diabetes mellitus	hsa04940	D, At
Pathways in cancer	hsa05200	An, At
Vascular smooth muscle contraction	hsa04270	An
Type II diabetes mellitus	hsa04930	D
Maturity onset diabetes of the young	hsa04950	D
Cytokine-cytokine receptor interaction	hsa04060	An, At
Dilated cardiomyopathy	hsa05414	At
Graft-versus-host disease	hsa05332	At
Systemic lupus erythematosus	hsa05322	At
Arrhythmogenic right ventricular cardiomyopathy (ARVC)	hsa05412	At
Focal adhesion	hsa04510	At
Jak-STAT signalling pathway	hsa04630	At
Asthma	hsa05310	At
Hypertrophic cardiomyopathy (HCM)	hsa05410	At
Hematopoietic cell lineage	hsa04640	At
Toll-like receptor signalling pathway	hsa04620	At
PPAR signalling pathway	hsa03320	At
NOD-like receptor signalling pathway	hsa04621	At
Prion diseases	hsa05020	At
Allograft rejection	hsa05330	At
Chemokine signalling pathway	hsa04062	At

Table 2.7. The 24 pathways containing genes that participate in specific genetic interactions. First column: the 24 pathways that contain genes that are part of genetic interactions with one gene in a diabetes pathway and the other in an aneurysm or an atherosclerosis pathway. Second column: KEGG ID of the pathway. Third column: disease to which the pathway is related to (An denotes Aneurysm, At denotes Atherosclerosis, D denotes Diabetes).

Then, we use information from the human genetic interaction network to single out 24 pathways, of the pathways related to the three diseases. We choose pathways that contain genes, which take part in genetic interactions such that one interacting gene is part of a diabetes - related pathway while the other is part of an aneurysm - or an atherosclerosis -related pathway. The 24 pathways, together with their KEGG IDs, are listed in Table 2.7. We use selected pathways to create a disease-related sub-network of the human PPI network as follows.

Datasets: Disease PPI sub-network. We postulate that a mutation of a gene on a diabetes pathway is related to a functional change of a protein on an aneurysm pathway, such that it would disable the aneurysm pathway from causing the disease. A question remains why does diabetes not have a similar effect on atherosclerosis. As discussed in the Introduction, genetic interactions can point us to gene pairs such that a gene mutation on one gene can be indicative of a change in another gene's function. Hence, we identify pairs of genes involved in genetic interactions such that at least one gene is from a diabetes pathway while the other is from an atherosclerosis or an aneurysm pathway. We find 31 genes that take part in such genetic interactions. We find that 24 pathways involved in one or more of the 3 diseases contain these 31 genes. We induce a sub-network of the human PPI network on all proteins from these 24 pathways. This sub-network contains 958 proteins and 3,370 interactions. Henceforth, we refer to it as the "disease PPI sub-network."

In search of genes whose change in functionality could disrupt a pathway, we rely on the disease PPI sub-network topology and look for genes in this disease PPI sub-network with a local topology that could explain a gene's high "destructiveness" for the related pathway—the *broker genes*. We describe a broker gene property using the Simmelian brokerage measure [100] and find brokers in the disease PPI sub-network, as follows.

Brokerage measure. The Simmelian brokerage [100] is a measure that describes the significance of a node for the interconnectedness of its local neighbourhood in the network. To our knowledge, this measure is the only topological measure that quantifies the importance of a node for maintaining the connectivity between its neighbouring nodes. High brokerage of a node implies high topological importance for the connectivity between nodes in its neighbourhood. In other words, if the functionality of a protein that has a high brokerage score would be altered, this would influence the interconnectedness of the protein's neighbourhood, which in our disease PPI sub-network is a part of the pathway in which this protein plays a role.

Simmelian brokerage of a node i is calculated as follows: $B_i = k_i - (k_i - 1)\, E_i$, where k_i is the degree of node i, and E_i is the "local efficiency" of node i in the network, calculated as:

$$E_i = \frac{1}{k_i\,(k_i - 1)} \sum_{l \epsilon N_i} \sum_{m \epsilon N_i, m \neq l} \frac{1}{d_{lm}}, \tag{2.2}$$

where N_i denotes the neighbourhood of node i (the sub-network induced on the first neighbours of node i), and d_{lm} denotes distance between nodes l and m. The local efficiency is normalised to $0 \leq E_i \leq 1$, so that the "local brokerage" of a node, B_i, takes

values: $1 \leq B_i \leq k_i$. By definition, brokerage values for the nodes with degree 1 are equal to zero.

To be able to compare proteins based on their brokerage in the disease PPI sub-network, we normalise the described brokerage measure by scaling to the range $[0, 1]$, as follows: $B_{i,n} = \frac{B_i - 1}{k_i - 1}$, where $B_{i,n}$ is the normalised brokerage of node i.

Note that a high node degree does not implicate high brokerage (see first two rows of Table 2.8).

Calculating brokerage values. We calculate the brokerage values for all nodes of degree higher than 2 in the disease PPI sub-network. We assign nodes into bins in increments of 0.01 of brokerage values. We only take into account genes with degree higher than 2 for the following reasons. We are not interested in nodes with degree 1, as such local topology can not confirm or refute our hypothesis (we are looking for nodes whose removal will affect the interconnectedness of its first neighbours, and node with degree one has just one first neighbour). Also, there are 100 genes in the disease PPI sub-network with degree 2 whose neighbours are not directly connected. This means that their normalised brokerage equals 1. The number of such proteins is higher than the number of the remaining proteins in the disease PPI sub-network whose local wiring is non–trivial and yields brokerage scores of 1, so inclusion of degree 2 nodes would introduce noise to our analysis. The brokerage distribution is shown in Figure 2.6.

In the remainder of this section, we explain how we model the disease PPI sub-network and identify statistically significant brokerage values.

Modeling the disease PPI sub-network. We generate 60 random networks with the same number of nodes and edges as in the disease PPI sub-network for each of the six commonly used random network models (totaling $60 \times 6 = 360$ random networks): Erdos-Renyi random graphs (ER) [49], Erdos-Renyi random graphs with the same degree distribution as the data (ER-DD) [18], Geometric Random Graphs (GEO) [160], Geometric Random Graphs with Gene Duplications and Mutations (GEO-GD) [60], Scale free Barabasi–Albert type networks (SF–BA) [52], and stickiness-index-based networks (STICKY) [57].

To find the best fitting network model, we compare the disease PPI sub-network with these random networks using *graphlet degree distribution agreement (GDDA)* measure [44]: GDDA measures how similar the networks are in terms of distributions of small induced sub-graphs - *graphlets* [7]. The arithmetic average of scaled and normalised distributions of all 73 graphlets results in GDDA value in range [0,1]. We use GDDA

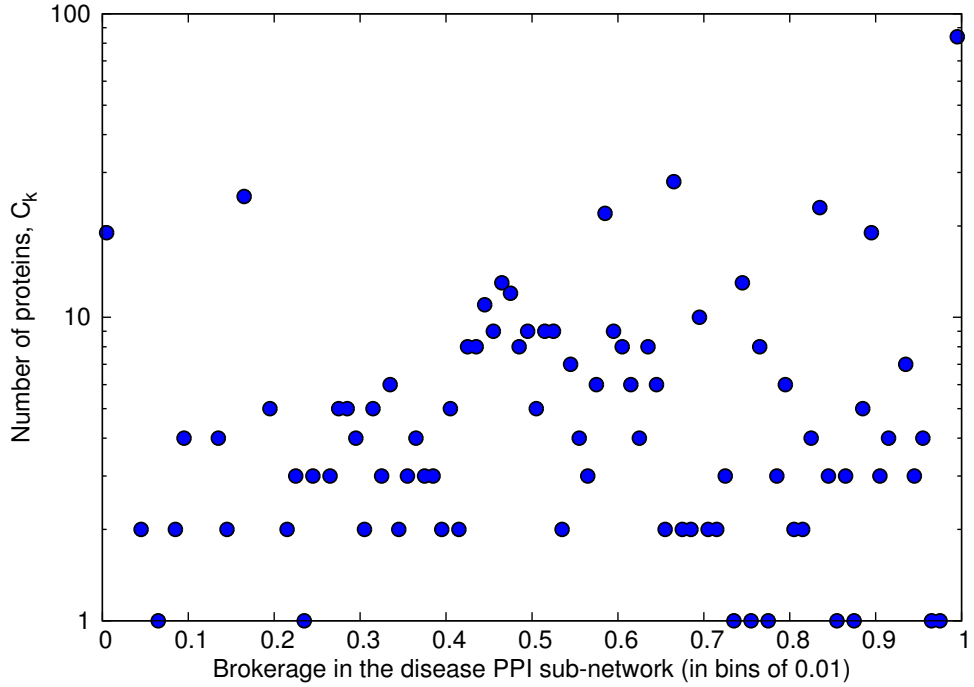


Figure 2.6. Distribution of brokerage values in the disease PPI sub-network. X–axis: brokerage values in bins of 0.01. Y–axis: numbers of proteins in the disease PPI sub-network that have a given brokerage. Figure is taken from Sarajlić *et al.* [99]

since it is a very sensitive measure for comparing network structure [44, 155]. The average GDDA values obtained for the GEO-GD, GEO, STICKY, SF-BA, ER-DD and ER network models are 0.85, 0.839, 0.825, 0.777, 0.755 and 0.673, with standard deviations of 0.01, 0.007, 0.007, 0.005, 0.006 and 0.008, respectively. Hence, GEO-GD and GEO models both provide a good fit to the disease PPI sub-network based on the best average GDDA value. Hence, we choose GEO-GD random network model for modeling the disease PPI sub-network.

Statistically significant brokerage values. We find statistically significant brokerage values by using GEO-GD as a well–fitting network model to the disease PPI sub-network. We generate 1,000 GEO-GD networks with the same number of nodes and edges as the disease PPI sub-network and calculate their brokerage distributions, again including only nodes with degree higher than 2. For each bin k and the corresponding node count, C_k, in the distribution shown in Figure 2.6 for the disease PPI sub-network, we calculate the p-value that corresponds to the probability of obtaining C_k or more nodes in this bin by chance. We do this by comparing C_k for the disease

PPI sub-network with the corresponding node counts in the 1,000 GEO-GD networks. We identify the statistically significant brokerage bins by using the threshold of 0.01 (p–value). We further examine the proteins with the brokerage scores in the statistically significant bins.

Note that when performing this statistical analysis, we have used different bin sizes. Comparing the results, the bin size of 0.01 resulted in the most natural barrier between statistical significance of low brokerage values and high brokerage values (see Figure 2.7). This bin size also resulted in the smallest number of bins whose statistical significance strongly deviates from the statistical significance of their neighbouring bins (scattered dots in Figure 2.7). Therefore we report the results obtained using the bin size of 0.01.

Bins with statistically significant p-values (< 0.01) are presented in Figure 2.7.

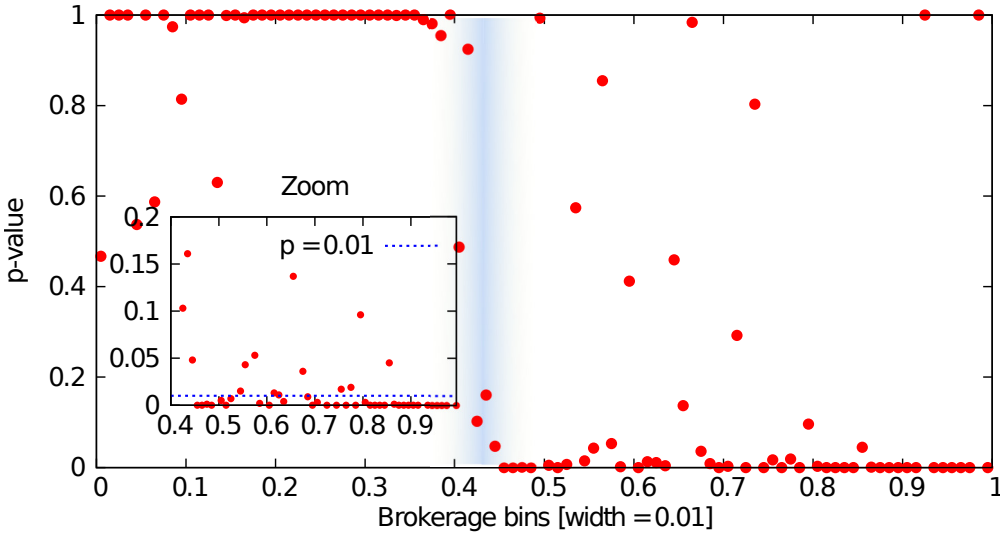


Figure 2.7. Statistically significant brokerage values. X–axis: brokerage values in bins of 0.01. Y–axis: p-value that corresponds to the probability of obtaining the same or higher numbers of proteins (as counted in the disease PPI sub-network) in the bin by chance. Inset in the bottom left: Red dots under the blue dotted line correspond to the statistically significant bins (p-values ≤ 0.01). Shaded blue line highlights the natural barrier reflecting the difference between statistical significance of low brokerage values and statistical significance of high brokerage values. Figure is taken from Sarajlić *et al.* [99].

2.2.2.2 Results

We hypothesize that identified broker genes, due to their importance for the interconnectedness of their neighbourhoods in the disease PPI sub-network, can lead to disabling signal transduction, or completion of chain reactions in the pathways.

In the disease PPI sub-network we find 313 proteins with statistically significant brokerage. Using the DAVID [161, 162] database we examine their functional enrichment and find this set to be enriched in a number of GO biological processes including phosphorylation (p-value$= 5.3 \times 10^{-31}$), as well as vascular development (p-value$= 5.1 \times 10^{-10}$) and regulation of cell-matrix adhesion (p-value$= 3.9 \times 10^{-10}$). Cell-matrix adhesion, i.e., binding of a cell to the extracellular matrix (ECM), plays important roles in regulation of many processes, such as cell adhesion, tissue homeostasis, and wound healing [163]. Matrix metalloproteinases (MMPs), proteolytic enzymes, exhibit increased activity in the human aneurysmal tissue [146]. MMP-2, which is among the 313 genes, takes part in the breakdown of the matrix proteins, including elastin, and therefore influences degradation of vessel wall in aneurysm. However, in diabetes, there is a reduced degradation of the matrix that results in an increased matrix volume [164]. Concentrations of MMP-2 and MMP-9 are reduced in coronary arteries of diabetic patients and it has been postulated that the reduction of MMPs activity can slow down the matrix loss, which is necessary for the pathogenesis of aneurysm [146]. This validates that presented methodology identifies genes enriched in biological processes that have already been proposed as causes of diabetes–aneurysm dissociation.

Out of the 313 genes we identify 16 genes that, in addition to taking part in aneurysm or atherosclerosis pathways, also take part in genetic interactions with genes from diabetes pathways. We postulate that among the 313 broker genes in the disease PPI sub-network, these 16 genes are the most likely to be responsible for the observed relationships between the diseases. Namely, as previously discussed, genetic interactions can point to pairs of genes such that mutation of one interacting partner can be indicative of a functional change of other interacting partner. In that sense, if there is a genetic interaction between a gene on a diabetes pathway and a broker gene on an aneurysm or an atherosclerosis pathway, then a mutation of a gene involved in a diabetes pathway can be related to a functional change of a broker gene on aneurysm or an atherosclerosis pathway. The 16 genes, their brokerage values, and KEGG IDs of the related pathways are presented in Table 2.8.

Gene name	Brok.	Degree	Pathways (KEGG ID)
MAPK7	1.0	7	hsa04010 (AN)
PPP3CA	1.0	4	hsa04010 (AN)
RPS6KA5	0.83	6	hsa04010 (AN)
MAPK8IP2	0.58	13	hsa04010 (AN)
GSK3A	0.83	4	hsa04062 (AT)
HSPA5	0.7	5	hsa05020 (AT)
PIK3CG	0.95	7	hsa05200 (AN,AT), hsa05210 (AN), hsa04630 (AT), hsa04062 (AT), hsa04620 (AT), hsa04510 (AT)
RAC1	0.84	29	hsa05200 (AN,AT), hsa04010 (AN), hsa05416 (AN), hsa05210(AN), hsa04510 (AT),hsa04620 (AT), hsa04062 (AT)
CDK2	0.60	36	hsa05200 (AN,AT)
ACTG1	0.58	4	hsa05416 (AN), hsa04510 (AT), hsa05410 (AT), hsa05412 (AT), hsa05414 (AT)
HDAC1	0.48	49	hsa05200 (AN,AT)
CCND1	0.48	16	hsa05200 (AN,AT), hsa05416 (AN), hsa05210 (AN), hsa04630 (AT), hsa04510 (AT)
MAP2K7	0.48	19	hsa04010 (AN),hsa04620 (AT)
MAP2K4	0.46	22	hsa04010 (AN), hsa04620 (AT)
BRAF	0.46	16	hsa04270 (AN), hsa05200 (AN,AT), hsa04010 (AN),hsa05210(AN), hsa04062 (AT), hsa04510 (AT)
CREBBP	0.46	49	hsa05200 (AN,AT), hsa04630 (AT)

Table 2.8. The 16 broker genes participating in specific genetic interactions. First column: the 16 genes that have statistically significant brokerage, that are on aneurysm or atherosclerosis pathways and that participate in genetic interactions such that one gene in the interaction is part of a diabetes pathway, while the other is part of an aneurysm or an atherosclerosis pathway. Second column: brokerage of the corresponding gene. Third column: the degree of the corresponding gene in the disease PPI sub-network. Fourth column: KEGG IDs of pathways in which the gene takes part. We additionally denote pathways with: (AN) for aneurysm-related pathway, and (AT) for atherosclerosis-related pathway.

Recall that the number of pathways related to atherosclerosis is much higher than the number of pathways related to aneurysm (as listed in Table 2.5 and Table 2.6). This is a consequence of a higher number of genes that are known to be related to the atherosclerosis in comparison to the number of genes that are known to be related

to the aneurysm, as detailed in Section 2.2.2.1. Therefore, the ratio of the number of identified broker genes on aneurysm pathways and the number of identified broker genes on atherosclerosis pathways might be influenced by this difference in size of available input data for the two diseases. With this in mind, note that the 16 genes that we further analyse are accurately identified. With additional data available in the future, possibly including biologically validated networks of pathways responsible for the two diseases, our methodology would be useful for identifying additional broker genes.

Using the DAVID database, we check functional enrichment of the 16 genes from Table 2.8. There are 8 kinases among the 16 genes: PIK3CG, MAP2K4, CDK2, GSK3A, RPS6KA5, BRAF, MAPK7, and MAP2K7. We use the hyper-geometric cumulative distribution to calculate the p-value that corresponds to the probability of finding 8 or more kinases among the 16 genes purely by chance. Since there are 151 kinases among 958 genes in the disease PPI sub-network, 8 out of 16 genes being kinases is statistically significant, p-value $= 0.0013$. To make sure that finding kinases is not just a consequence of possibly high number of kinases among the 313 broker genes, we also calculate the statistical significance of finding 8 or more kinases among the 16 genes when taking only 313 broker genes as the background set. There are 66 kinases among the 313 genes, so finding 8 or more kinases among the 16 genes is again statistically significant, p-value $= 0.008$. Out of the 16 genes, 9 are involved in phosphorylation: PIK3CG, BRAF, MAP2K4, CDK2, RPS6KA5, CCND1, GSK3A, MAPK7, MAP2K7 (p-value $= 1 \times 10^{-4}$). Clearly, all of the above listed 8 kinases are among them, as kinases are proteins that can be turned on or off by phosphorylation (adding phosphate groups). Phosphorylation usually results in a functional change of the target protein, cellular location, or association with other proteins. That can lead to rewiring of pathways that these kinases participate in, which in case of an aneurysm pathway could disrupt the onset of aneurysm.

The question remains why broker genes from our set that are kinases on an atherosclerosis pathway would not disrupt the onset of atherosclerosis. To answer this we check if any of the 16 genes have pleiotropic traits. Pleiotropy occurs when a gene influences multiple traits, for example, because the gene encodes a protein that is used for two or more functions, or has different functions in different tissues [165]. We find that PIK3CG phosphorylates phosphatidylinositol 4,5-bisphosphate to generate PIP3, which plays a pleiotropic role in regulating membrane signaling[11]. Pleiotropic activities of GSK3 have made it a therapeutical target for treatment of various human diseases, including type 2 diabetes [166]. It is also known that mutations that result from the pleiotropic ef-

[11] http://www.phosphosite.org/proteinAction.do?id=3655&showAllSites=true

fects of BRAF can lead to different transcriptional changes [167]. Also, MAPK7 has pleiotropic functions [168]. A mutation in a pleiotropic gene can have an effect on just one of its traits, or on all of them [165]. Two of these genes, BRAF and PIK3CG, are present both in aneurysm and atherosclerosis pathways (see Table 2.8), and since they genetically interact with diabetes-related genes this may explain why a mutation on such genes would influence the development of aneurysm and not atherosclerosis in diabetic patients.

The identified 16 genes should further be explored in search for exact mechanisms behind the protective role of diabetes on the development of aneurysm. The most likely candidate genes are MAPK7 and PPP3CA, as their brokerage values equal 1 (see Table 2.8), suggesting the high destructive potential on the pathways that they take part in. In fact, brokerage value 1 means that inactivity of MAPK7 or PPP3CA would completely destroy connectivity in their neighbourhoods. Note that MAPK7 and PPP3CA are on MAPK signaling pathway, which is related to aneurysm, therefore their functional change can disable signaling process that plays role in formation of this disease. Although both genes have been already linked to aneurysm, [169, 170] we here uncover that they may also play important role in the diabetes–aneurysm relationship.

2.3 Conclusions

In this chapter we presented two studies which illustrate the variety of applications of graph theory in current open problems in medicine or biology.

The first study, Sarajlić *et al.* [8], combines multiple methods in a novel way to extract the key CVD genes that are enriched in drug targets and driver genes and that have a large overlap with the Core Diseasome. We used our method to predict new CVD genes and validated a substantial portion of our predictions in the literature. Hence, it is likely that the remaining genes, for which we did not find validation in the literature, could be new genes involved in CVDs. Moreover, we found that the function of known CVD drug targets coincides with the function of many of our predicted CVD genes. This indicates that our method produces predictions that may be therapeutically exploited. The second study, Sarajlić *et al.* [99], addresses the important question of why patients with diabetes do not develop aneurysm, but do develop atherosclerosis, when the two diseases have similar risk factors. We applied the topological measure of Simmelian brokerage to find genes that have a high potential for disrupting their neighbourhoods' connectivity, meaning that functional changes on such genes would result in disabling the pathways that they are part of. Using this approach, we identified a set of 313

genes enriched in GO biological processes that facilitate the mechanisms behind these particular diseases relationships. Since genetic interactions involve pairs of genes such that a mutation on one gene is related to a functional change of the other [153], out of the 313 genes we identified 16 on aneurysm and atherosclerosis pathways that take part in genetic interactions with genes from diabetes pathways. We suggest these genes hold the answer for the relationships between the three diseases. We find that 8 out of the identified 16 genes are kinases (a statistically significant enrichment) that may act as switches on the related pathways. Also, two of the kinases, that are found on both aneurysm and atherosclerosis pathways, are pleiotropic, explaining why these genes could disable onset, formation and progression of aneurysm, but enable atherosclerosis.

In both of the studies we used biological networks that are undirected, but the pool of available data on the interaction between biomolecules is becoming richer with more directed interactions available, such as transcriptional regulatory networks, metabolic networks, etc. Thus, we are motivated to broaden the set of known local network properties of directed networks with the concept of directed graphlets and generalisation of all existing graphlet-based measures to a directed case. In the remainder of this dissertation, we present these new measures, validate them using synthetic model networks and apply them to directed metabolic networks.

2.4 Author's Contributions

Section 2.1 Anida Sarajlić independently performed the literature survey of network-based approaches in research of cardiovascular diseases, and wrote the paper in collaboration with Nataša Pržulj, resulting in a peer-reviewed scientific publication [98]: Anida Sarajlić and Nataša Pržulj, "Survey of Network-based Approaches to Research of Cardiovascular Diseases," BioMed Research International, 2014.

Section 2.2 Anida Sarajlić collaborated with Vuk Janjić, Neda Stojković, Djordje Radak and Nataša Pržulj on the work presented in this section. Anida Sarajlić collected the data, implemented and performed computational experiments (clustering, calculating enrichments, identifying key CVD genes and predicted CVD genes) and analysed all results (examining importance of key CVD genes and performing literature validation), except for identifying therapeutic properties of key and predicted CVD genes. Anida Sarajlić also took part in designing all experiments and writing the paper. This work resulted in a peer-reviewed scientific publication [8]: Anida Sarajlić, Vuk Janjić, Neda Stojković, Djordje Radak and Nataša Pržulj, "Network Topology Reveals Key Cardiovascular Disease Genes," PLoS ONE, 2013.

Section 2.2.2 Anida Sarajlić collaborated with Vladimir Gligorijević, Djordje Radak and Nataša Pržulj on the work presented in this section. Anida Sarajlić took part in designing the methodology, collected all data and constructed the disease PPI sub-network, took part in implementation of the methodology and performed all computational experiments (finding statistically significant brokerage values, modeling the disease PPI sub-network, identifying significant sets of genes) and analysed all results (examining importance of significant genes through statistical analysis, examining their role through the literature survey, identifying kineases and pleiotropic kineases and their significance for the study). Anida Sarajlic also wrote the paper in collaboration with Nataša Pržulj, resulting in a peer-reviewed scientific publication [99]: Anida Sarajlić, Vladimir Gligorijević, Djordje Radak and Nataša Pržulj, "Network Wiring of Pleiotropic Kinases Yields Insight into Protective Role of Diabetes on Aneurysm," Integrative Biology, 2014.

3 Directed Graphlet-based Methods

In this chapter, we introduce directed graphlets and graphlet-based measures for topological analysis and the comparison of directed networks. We define 40 up to four node directed graphlets and 129 orbits and implement the directed graphlet and orbits counting algorithm. Furthermore, we generalise the following undirected graphlet-based heuristics to the directed case: relative graphlet frequency distance, graphlet degree distribution similarity, graphlet degree vector similarity, and graphlet correlation distance (see Section 1.3.3 for details on these graphlet heuristics). We then compare our new directed graphlet-based measures with common existing measures for network comparison by evaluating their performance on model network clustering. For this, we use the existing directed network models and propose generalisations to the directed case for SF-GD, GEO and GEO-GD models. In addition, we evaluate the tolerance of our new distance measures to noise.

3.1 Methods

3.1.1 Directed Graphlets and Graphlet Orbits

Recall that a directed network is denoted as a pair $G = \{V, E\}$, where V is a set of vertices (nodes) and E, is a set of ordered pairs of vertices, often called arcs, directed edges, or arrows. A directed edge or arc $e = (x, y)$ is directed from x to y, where y is called the head and x is called the tail of the arc, y is a direct successor of x, and x is a direct predecessor of y. Multiple arcs (also called parallel arcs or a multi-arc) are two or more arcs that begin and end at the same two vertices. An *anti-parallel* pair of arcs is a pair of arcs such that one's head/tail is the other's tail/head. We consider simple directed graphs, that is, graphs that do not contain multiple edges or self-loops (self-loops are edges that start and end in the same node). We allow anti-parallel pairs of arcs in the network, which will be discussed in more detail below.

Undirected graphlets are small connected non-isomorphic induced sub-graphs of an undirected network (see Section 1.3.2). We generalise graphlets to directed ones and identify all directed graphlet orbits.

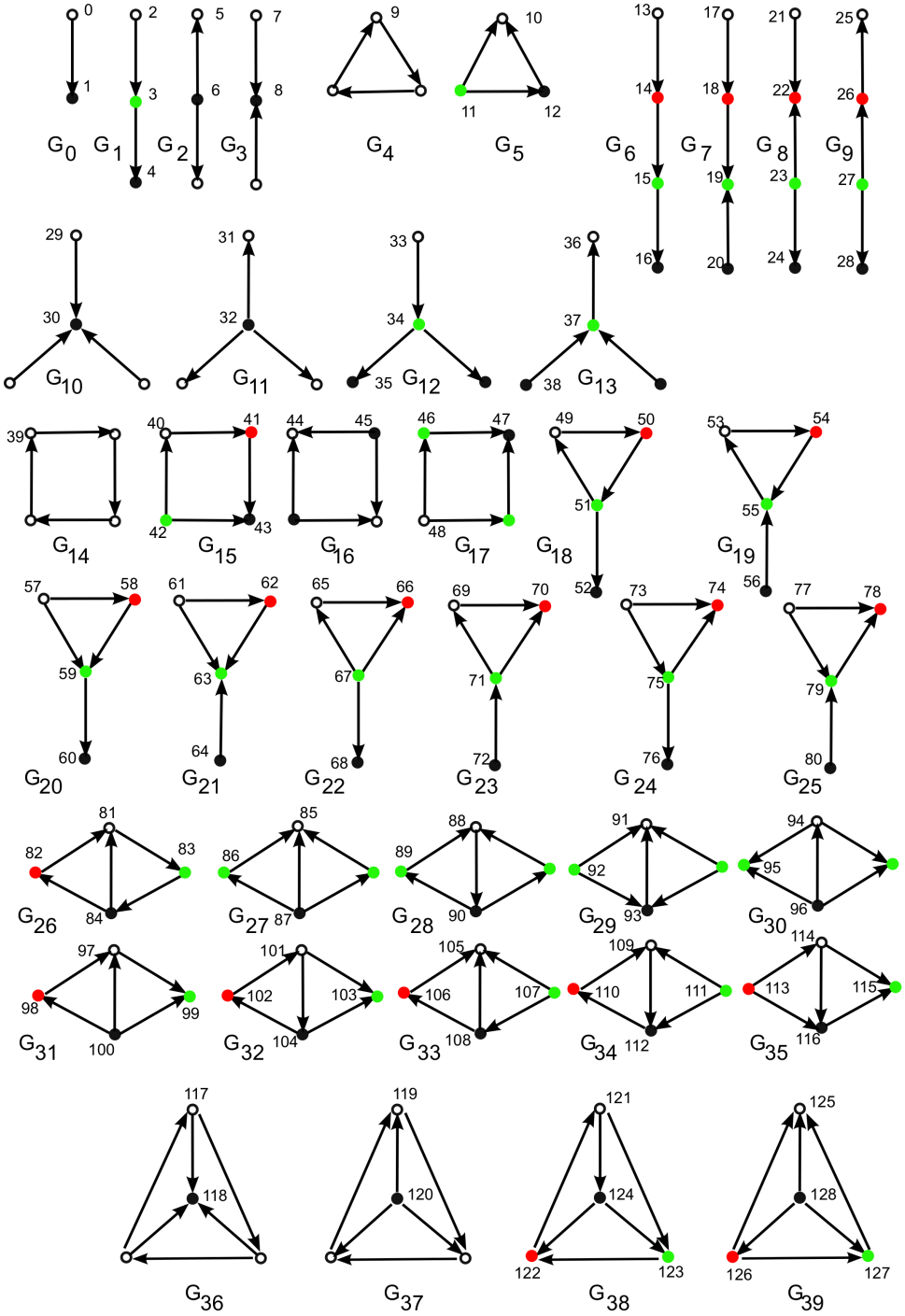
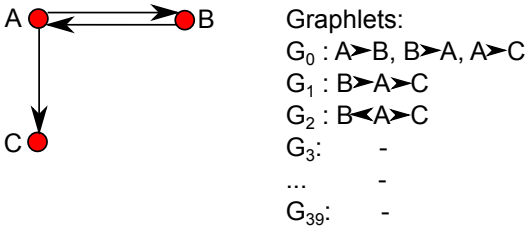


Figure 3.1. 40 Directed Graphlets and 129 orbits. For a given graphlet, nodes belonging to the same orbit are of the same color.

There are 40 up to four node directed graphlets, which are formally defined as small connected directed non-isomorphic induced sub-graphs of a simple directed network without anti-parallel pairs of arcs, with 129 orbits. This is shown in Figure 3.1. Here, we discuss the reasons why we consider only up to four node directed graphlets induced on simple directed networks that do not contain anti-parallel pairs of arcs. Recently, Yaveroglu *et.al* [13] pointed out that the computation of 5-node graphlet statistics increases the computational complexity and reduces the applicability of graphlet-based techniques to very large networks. They evaluated the contribution and necessity of 5-node graphlet statistics for network comparison as follows. The statistics of larger graphlets are bound by the statistics of smaller graphlets, resulting in redundancies and dependencies in the graphlet degrees of nodes. Yaveroglu *et.al* [13] eliminated the redundant statistics, and quantified the level of dependencies among the non-redundant orbits using Spearman's correlation coefficient to define a new network topology statistic, called the Graphlet Correlation Matrix, which we described in Section 1.3.3. They showed that Graphlet Correlation Distance measure, based on non-redundant up to four node graphlet orbits (GCD-11) outperforms the GCD-56 based on non-redundant up to five node orbits [13]. The up to 4-node graphlets introduce less noise in the corresponding new network statistic because there are fewer dependencies between up to 4-node graphlet orbits than between up to 5-node graphlet orbits. Yaveroglu *et.al* [13] also demonstrated that 5-node graphlets do not carry any significant information which is not already captured by the up to 4-node graphlets. Hence, we generalize up to 4-node graphlets to a directed case. Defining directed graphlets as non-isomorphic induced sub-graphs of a network with anti-parallel pairs of arcs would result in over 600 orbits in up to 4-node graphlets containing anti-parallel pairs of arcs. Such high number of orbits increases the computational complexity of the orbit counting process and computation of graphlet-based statistics; hence, we decide to take into account up to 4-node directed graphlets without anti-parallel pairs of arcs.

We account for anti-parallel pairs of arcs in a network as follows. Let an anti-parallel pair of arcs exist between nodes A and B in the network, as shown in Figure 3.2. Recall that an anti-parallel pair of arcs between nodes A and B corresponds to two arcs (two directed edges): one from A to B and one from B to A. This type of connection then contributes to two different graphlets on nodes A and B (each containing one directed edge). So, an anti-parallel pair of arcs between nodes A and B contributes to graphlet counts in a way that it accounts for two graphlets of type G_0 and it contributes to both orbits 0 and 1 for both node A and node B. A similar approach is used for 3- or 4-node graphlets induced on the set of nodes that contain nodes A and B. Note that all

the nodes in a graphlet are different. Hence, the path ABA, shown in Figure 3.2, does not correspond to graphlet G_1 and does not contribute to counts of orbits 2, 3 or 4 for nodes A and B.



Orbits:	0	1	2	3	4	5	6	7,...,129
Node A:	A➢B, A➢C	A➢B	-	B➢A➢C	-	-	B➢A➢C	-
Node B:	B➢A	B➢A	B➢A➢C	-	-	B➢A➢C	-	-
Node C:	-	C➢A	-	-	C➢A➢B	C➢A➢B	-	-

Figure 3.2. Inducing directed graphlets from the network with anti-parallel pairs of arcs.

In Section 3.1.2 below, we generalise existing graphlet-based network properties to the directed case.

3.1.2 Directed Graphlet-based Measures

Directed graphlet degree vector (DGDV), analogous to GDV [4] for undirected networks, counts the number of all two to four node directed graphlets that a node touches, taking into account different "symmetry groups" within each directed graphlet (numbered from 0 to 128 on Figure 3.1). These symmetry groups are called automorphism orbits (detailed in [44]). For example, it is topologically relevant whether a node touches graphlet G_3 at the middle node, or at one of the end nodes. These counts are coordinates in the 129-dimensional *Directed Graphlet Degree Vector (DGDV)* of a node.

The similarity between DGDVs of nodes u and v in graph G is computed in the same way as in the case of undirected networks [4]. If u_i is the i^{th} coordinate in the DGDV of node u, and v_i is the i^{th} coordinate in the DGDV of node v, then the distance between these two coordinates is computed as:

$$D_i(u,v) = w_i \times \frac{|log(u_i+1) - log(v_i+1)|}{log(max(u_i,v_i)+2)}, \tag{3.1}$$

where w_i denotes weight that corresponds to orbit i. Different weights are assigned

to different orbits because of the dependency that exists between the number of orbits in the network. For example, the number of orbits 0 that a node touches will also influence the number of orbits 2 that this node touches. Similarly as in the undirected case, to compute weights w_i we assign a value o_i to each orbit i. This value is obtained by counting the number of orbits (among the same or lower number of nodes than the orbit i) that affects orbit i. It is also considered that each orbit affects itself. For example, whether the node touches orbit 9 is conditional upon whether that node touches orbits 0 and 1. Therefore, $o_9 = 3$, denoting orbits 0, 1 and 9. Similarly, orbit 39 is influenced by orbits 0, 1, 3, 2 and 4 resulting in $o_{39} = 6$. We find all dependencies between orbits on directed graphlets and give a complete list in Table 3.1.

Orbit	Dependant orbits	Orbit	Dependant orbits	Orbit	Dependant orbits
0	0	43	43, 1, 4, 5, 8	86	86, 0, 1, 5, 7, 12
1	1	44	44, 1, 5, 8	87	87, 0, 6, 11
2	2, 0	45	45, 0, 6, 7	88	88, 0, 1, 8, 9
3	3, 0, 1	46	46, 0, 1, 3, 5, 7	89	89, 0, 1, 5, 7, 9
4	4, 1	47	47, 1, 4, 8	90	90, 0, 1, 6, 9
5	5, 1	48	48, 0, 2, 6	91	91, 1, 8, 10
6	6, 0	49	49, 0, 1, 5, 9	92	92, 0, 7, 11
7	7, 0	50	50, 0, 1, 2, 9	93	93, 0, 1, 8, 12
8	8, 1	51	51, 0, 1, 3, 6, 9	94	94, 0, 1, 6, 12
9	9, 0,1	52	52, 1, 4, 5	95	95, 1, 5, 10
10	10, 1	53	53, 0, 1, 4, 9	96	96, 0, 6, 11
11	11, 0	54	54, 0, 1, 7, 9	97	97, 0, 1, 3, 10, 12
12	12, 0, 1	55	55, 0, 1, 3, 8, 9	98	98, 0, 1, 2, 5, 12
13	13, 0, 2	56	56, 0, 2, 7	99	99, 1, 4, 5, 10
14	14, 0, 1, 2, 3	57	57, 0, 2, 11	100	100, 0, 6, 11
15	15, 0, 1, 3, 4	58	58, 0, 1, 2, 12	101	101, 0, 1, 3, 9, 11
16	16, 1, 4	59	59, 0, 1, 3, 10	102	102, 0, 1, 2, 5, 9
17	17, 0, 2	60	60, 1, 4	103	103, 1, 4, 5, 10
18	18, 0, 1, 3, 7	61	61, 0, 7, 11	104	104, 0, 1, 6, 9, 12
19	19, 1, 4, 8	62	62, 0, 1, 7, 12	105	105, 1, 8, 10
20	20, 0, 7	63	63, 1, 8, 10	106	106, 0, 1, 4, 7, 12
21	21, 0, 7	64	64, 0, 7	107	107, 0, 2, 7, 12
22	22, 1, 5, 8	65	65, 0, 1, 5, 12	108	108, 0, 1, 3, 11, 12

23	23, 0, 6, 7	66	66, 1, 5, 10	109	109, 0, 1, 8, 9, 12
24	24, 1, 5	67	67, 0, 6, 11	110	110, 0, 1, 4, 7, 9
25	25, 1, 4	68	68, 1, 5	111	111, 0, 2, 7, 11
26	26, 0, 1, 3, 5	69	69, 0, 1, 4, 12	112	112, 0, 1, 3, 9, 10
27	27, 0, 2, 6	70	70, 1, 4, 10	113	113, 0, 2, 11
28	28, 1, 5	71	71, 0, 1, 3, 11	114	114, 0, 1, 3, 11, 12
29	29, 0, 7	72	72, 0, 2	115	115, 1, 4, 10
30	30, 1, 8	73	73, 0, 2, 11	116	116, 0, 1, 3, 10, 12
31	31, 1, 5	74	74, 1, 5, 10	117	117, 0, 1, 9, 11, 12
32	32, 0, 6	75	75, 0, 1, 3, 6, 12	118	118, 1, 10
33	33, 0, 2	76	76, 1, 4, 5	119	119, 0, 1, 9, 10, 12
34	34, 0, 1, 3,6	77	77, 0, 7, 11	120	120, 0, 11
35	35, 1, 4, 5	78	78, 1, 4, 10	121	121, 0, 1, 9, 11
36	36, 1, 4	79	79, 0, 1, 3, 8, 12	122	122, 0, 1, 9, 10
37	37, 0, 1, 3, 8	80	80, 0, 2, 7	123	123, 0, 1, 9, 10, 12
38	38, 0, 2, 7	81	81, 0, 1, 3, 9, 10	124	124, 0, 1, 9, 11, 12
39	39, 0, 1, 2, 3, 4	82	82, 0, 1, 2, 4, 12	125	125, 1, 10
40	40, 0, 1, 2, 3, 5	83	83, 0, 1, 2, 4, 9	126	126, 0, 1, 11, 12
41	41, 0, 1, 3, 4, 7	84	84, 0, 1, 3, 9, 11	127	127, 0, 1, 10, 12
42	42, 0, 2, 6, 7	85	85, 1, 8, 10	128	128, 0, 11

Table 3.1. The complete list of orbit dependencies for all directed 2 to 4 node graphlets.

Hence, the values o_i are:

- 1 for $i \in \{0,1\}$,
- 2 for $i \in \{2, 4, 5, 6, 7, 8, 10, 11\}$,
- 3 for $i \in \{3, 9, 12, 13, 16, 17, 20, 21, 24, 25, 28, 29, 30, 31, 32, 33, 36, 60, 64, 68, 72, 118, 120, 125, 128\}$,
- 4 for $i \in \{19, 22, 23, 27, 35, 38, 44, 45, 47, 48, 52, 56, 57, 61, 63, 66, 67, 70, 73, 74, 76, 77, 78, 80, 85, 87, 91, 92, 95, 96, 100, 105, 113, 115\}$,
- 5 for $i \in \{14, 15, 18, 26, 34, 37, 42, 43, 49, 50, 53, 54, 58, 59, 62, 65, 69, 71, 88, 90, 93, 94, 99, 103, 107, 111, 121, 122, 126, 127\}$,

- 6 for $i \in \{39, 40, 41, 46, 51, 55, 75, 79, 81, 82, 83, 84, 86, 89, 97, 98, 101, 102, 104, 106, 108, 109, 110, 112, 114, 116, 117, 119, 123, 124\}$.

The value of w_i is calculated as:

$$w_i = 1 - \frac{log(o_i)}{log(129)}. \tag{3.2}$$

The total distance between DGDVs of nodes u and v, normalized in $[0, 1]$ range, is calculated as:

$$D(u, v) = \frac{\sum_{i=0}^{128} D_i}{\sum_{i=0}^{128} w_i}. \tag{3.3}$$

Finally, **DGDV similarity** of the two nodes is computed as:

$$S(u, v) = 1 - D(u, v). \tag{3.4}$$

Directed graphlet degree distribution (DGDD) is defined as follows: for each of the 129 automorphism orbits (Figure 3.1), the distribution of nodes touching a particular graphlet at the node belonging to a particular orbit is calculated. In other words, for a particular orbit we count the number of nodes touching a graphlet at that orbit. This results in a spectrum of 129 directed graphlet degree distributions, where the out-degree and in-degree distributions are the first two. Networks can be compared based on a DGDD agreement measure, which we define similarly as in the case of GDD (see section 1.3.3): Let d_G^j be DGDD for the j^{th} automorphism orbit in network G. Normalised distribution for the network G is defined as

$$N_G^j(k) = \frac{S_G^j(k))}{T_G^j}, \tag{3.5}$$

where d_G^j is scaled as $S_G^j(k) = \frac{d_G^j(k))}{k}$ to decrease the contribution of larger degrees in DGDD, and then the distribution is normalised with respect to its total area:

$$T_G^j = \sum_{k=1}^{\infty} S_G^j(k). \tag{3.6}$$

The distance between normalised j^{th} distributions for the two networks G and H is:

$$D^j(G, H) = \frac{1}{\sqrt{2}} \cdot \left(\sum_{k=1}^{\infty} [N_G^j(k) - N_H^j(k)]^2 \right)^{\frac{1}{2}}, \tag{3.7}$$

where the resulting value is between 0 and 1, 0 meaning that the j^{th} DGDDs are identical, 1 that they are dissimilar. The j^{th} DGDD agreement (DGDDA) is obtained as:

$$A^j(G,H) = 1 - D^j(G,H). \tag{3.8}$$

Finally, the **DGDD agreement** between two networks is defined either as the arithmetic or geometric mean of all DGDD agreements over 129 automorphism orbits. Note that in this dissertation we use the arithmetic mean.

Relative directed graphlet frequency (RDGF), following the same approach as for the RGF (see Section 1.3.3), is defined as $\frac{N_i(G)}{T(G)}$, where $N_i(G)$ is the number of graphlets of type $i, i \in \{1, ..., 39\}$ in the network G, and $T(G) = \sum_{i=1}^{39} N_i(G)$ is the total number of graphlets in G [7]. The relative directed graphlet frequency distance $D(G,H)$ for the two graphs G and H is defined as:

$$D(G,H) = \sum_{i=1}^{39} |F_i(G) - F_i(H)|\,, \tag{3.9}$$

where $F_i(G) = -\frac{log(N_i(G))}{log(T(G))}$. To avoid dominance of the most frequent graphlets in the networks, we use relative directed graphlet frequency in the *log* form.

We denote this measure as RDGF-3 distance, as we are considering the 3 and 4 node directed graphlets, omitting two node graphlets, similarly as it was suggested for undirected graphlets [7].

We also consider the RDGF-2 distance, where we take into account all directed graphlets on 2, 3 and 4 nodes, i.e. all graphlets labelled G_0 to G_{39}.

Directed graphlet correlation matrix. Analogous to the case of undirected graphlets, the statistics of different orbits on directed graphlets are not independent of each other. The reason behind this is that smaller graphlets are induced sub-graphs of larger graphlets. We have already pointed out such dependencies in Table 3.1. This motivates us to explore whether these correlation patterns can be used to measure topological similarity between networks. Hence, we generalise the concept of GCM to the directed case as follows. For each node in the networks, its DGDV is constructed. Then, we construct a matrix containing DGDVs as the rows, the number of rows corresponding to the number of nodes in the network. We calculate a Spearman's correlation between each two pairs of columns in the resulting matrix, i.e. correlation between the orbits in the network. We present these correlations as a 129×129 dimensional directed graphlet correlation matrix (DGCM-129), which is symmetric and contains Spearman's correlation values in [-1,1] range. Note that some graphlets, and hence orbits, may not

appear in the network, which would result in an entire column of zeros. Spearman's correlation coefficient is not defined when all values in one of the input vectors are the same (zero is the only possibility in our case), so, as proposed by Yaverŏglu *et al.* [13], we address this issue by introducing a dummy node in the network with a DGDV vector with all values set to 1. This way the correlation between non-existing orbits will be 1, and the correlation between a non existing orbit and any other orbit, whose column has non-zero values, will be close to 0. *Directed graphlet correlation distance* (DGCD-129) between two networks is defined as the Euclidian distance of the upper triangle values of their DGCMs. We also look at the performance of a DGCM matrix for up to three node directed graphlets, which takes into account only the first 6 graphlets (G_0-G_5) and 13 automorphism orbits (0-12). We denote this 13×13 dimensional matrix with DGCM-13. DGCM-13 and DGCD-13 are defined in the same way as DGCM-129 and DGCD-129, by taking into account orbits from orbit 0 to orbit 12.

3.1.3 Redundancies between Directed Graphlet Orbits

An orbit is redundant if its degree can be derived from the degrees of other orbits. Recall that in the case of up to five node undirected graphlets there are 17 independent equations that define redundancies between orbits [13]. Thus, 17 redundant orbits can be removed from graphlet degree vectors (GDVs), resulting in 56 dimensional GDVs containing the counts of non-redundant orbits only. In the case of up to four node undirected graphlets there are 11 non-redundant orbits in a GDV, resulting in 11×11 dimensional GCM. Here, we explore the possibility of finding redundant orbits for the case of up to four node directed graphlets.

In the case of networks without anti-parallel pairs of arcs, directed graphlets are induced sub-graphs and we can perform the following reasoning. The left panel of Figure 3.3 represents the case when the two orbits 0 that node A touches (orbit AB and orbit AC) are combined. Possible outcomes are: (1) node A also touches orbit 6 in case nodes B and C are not connected, (2) node A touches orbit 11 in case nodes B and C are connected (regardless of the direction of the edge). If we denote the number of orbits 0 that node A touches with C_0, then the number of pairs of two orbits 0, which equals $\binom{C_0}{2}$, equals the sum of the counts of orbits 6 and 11. We derive redundancy equations by performing a similar analysis on all other possible combinations of orbits.

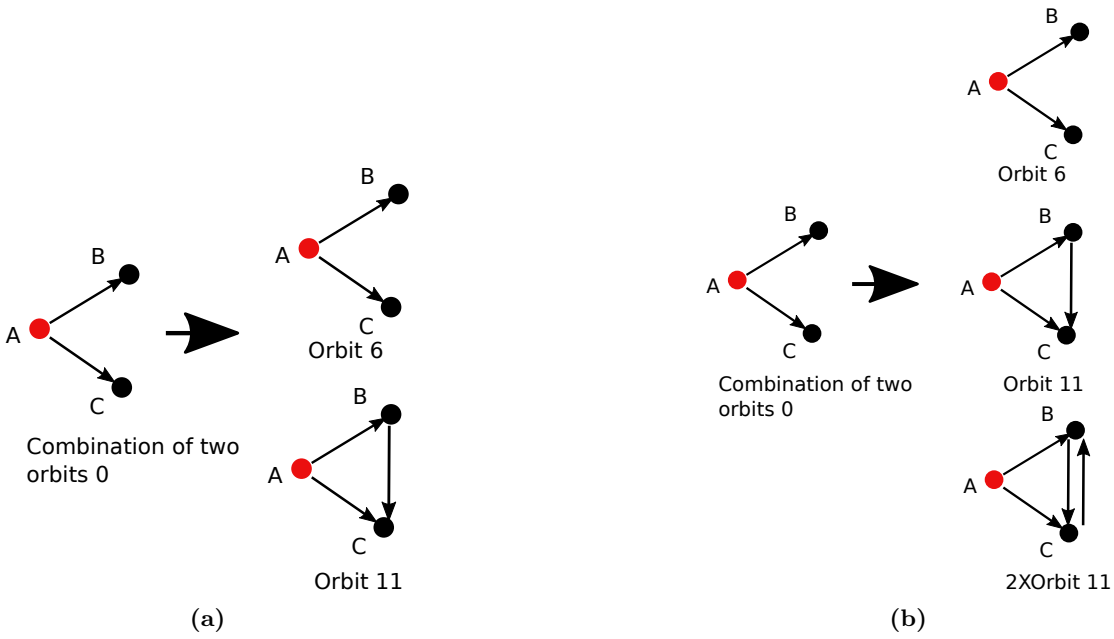


Figure 3.3. Illustration of the redundancies between directed graphlet orbits 0, 6 and 11. (a) Possible orbits that node A can touch, in the case when A touches two orbits 0 in a network without anti-parallel pairs of arcs. (b) Possible orbits that node A can touch, in case when A touches two orbits 0 in a network with anti-parallel pairs of arcs.

Below, we list the 23 independent equations which describe relationships between orbits on up to four node directed graphlets in directed network without anti-parallel pairs of arcs:

1. $\binom{C_0}{2} = C_6 + C_{11}$
2. $\binom{C_0}{1} \cdot \binom{C_1}{1} = C_3 + C_9 + C_{12}$
3. $\binom{C_1}{2} = C_8 + C_{10}$
4. $\binom{C_2}{1} \cdot \binom{C_0-1}{1} = C_{27} + C_{57} + C_{73} + 2 \cdot C_{113} + C_{107} + C_{111} + 2 \cdot C_{48} + C_{42}$
5. $\binom{C_1}{1} \cdot \binom{C_2}{1} = C_{58} + C_{50} + C_{40} + C_{39} + C_{82} + C_{98} + C_{102} + C_{14} + C_{83}$
6. $\binom{C_0}{1} \cdot \binom{C_5}{1} = C_{26} + C_{65} + C_{49} + C_{40} + C_{46} + C_{102} + C_{86} + C_{89} + C_{98}$
7. $\binom{C_5}{1} \cdot \binom{C_1-1}{1} = C_{22} + C_{74} + C_{66} + 2 \cdot C_{44} + C_{43} + 2 \cdot C_{95} + C_{99} + C_{103}$
8. $\binom{C_7}{1} \cdot \binom{C_0-1}{1} = C_{23} + C_{61} + C_{77} + C_{42} + 2 \cdot C_{45} + C_{107} + 2 \cdot C_{92} + C_{111}$

9. $\binom{C_7}{1} \cdot \binom{C_1}{1} = C_{18} + C_{62} + C_{54} + C_{46} + C_{41} + C_{86} + C_{110} + C_{106} + C_{89}$
10. $\binom{C_4}{1} \cdot \binom{C_0}{1} = C_{15} + C_{53} + C_{69} + C_{39} + C_{41} + C_{83} + C_{106} + C_{110} + C_{82}$
11. $\binom{C_4}{1} \cdot \binom{C_1-1}{1} = C_{19} + C_{78} + C_{70} + C_{43} + 2 \cdot C_{47} + C_{99} + 2 \cdot C_{115} + C_{103}$
12. $\binom{C_3}{1} \cdot \binom{C_1-1}{1} = 2 \cdot C_{37} + 2 \cdot C_{59} + C_{79} + C_{55} + C_{97} + C_{112} + C_{116} + C_{81}$
13. $\binom{C_3}{1} \cdot \binom{C_0-1}{1} = 2 \cdot C_{34} + C_{51} + C_{75} + 2 \cdot C_{71} + C_{101} + C_{108} + C_{84} + C_{114}$
14. $\binom{C_6}{1} \cdot \binom{C_0-2}{1} = 3 \cdot C_{32} + 2 \cdot C_{67} + C_{96} + C_{87} + C_{100}$
15. $\binom{C_6}{1} \cdot \binom{C_1}{1} = C_{34} + C_{75} + C_{51} + C_{94} + C_{90} + C_{104}$
16. $\binom{C_8}{1} \cdot \binom{C_0}{1} = C_{37} + C_{55} + C_{79} + C_{88} + C_{93} + C_{109}$
17. $\binom{C_8}{1} \cdot \binom{C_1-2}{1} = 3 \cdot C_{30} + 2 \cdot C_{63} + C_{85} + C_{91} + C_{105}$
18. $\binom{C_9}{1} \cdot \binom{C_0-1}{1} = C_{51} + C_{104} + 2 \cdot C_{90} + C_{84} + C_{101} + 2 \cdot C_{121} + C_{117} + C_{124}$
19. $\binom{C_9}{1} \cdot \binom{C_1-1}{1} = C_{55} + C_{81} + C_{112} + C_{109} + 2 \cdot C_{88} + C_{119} + 2 \cdot C_{122} + C_{123}$
20. $\binom{C_{10}}{1} \cdot \binom{C_1-2}{1} = C_{63} + 2 \cdot C_{105} + 2 \cdot C_{85} + 2 \cdot C_{91} + 3 \cdot C_{125} + 3 \cdot C_{118}$
21. $\binom{C_{10}}{1} \cdot \binom{C_0}{1} = C_{59} + C_{97} + C_{81} + C_{116} + C_{112} + C_{127} + C_{122} + C_{119} + C_{123}$
22. $\binom{C_{11}}{1} \cdot \binom{C_0-2}{1} = C_{67} + 2 \cdot C_{87} + 2 \cdot C_{100} + 2 \cdot C_{96} + 3 \cdot C_{128} + 3 \cdot C_{120}$
23. $\binom{C_{11}}{1} \cdot \binom{C_1}{1} = C_{71} + C_{84} + C_{108} + C_{114} + C_{101} + C_{121} + C_{126} + C_{124} + C_{117}$

Note that it is possible to form additional equations using the approach from the left panel of Figure 3.3, however all other possible equations can be derived from the above-listed 23 main equations. Here, we give a few examples. By observing orbits that can be formed by combining orbits 0 and 12 we can see that the following equation holds:

$$\binom{C_{12}}{1} \cdot \binom{C_0 - 1}{1} = C_{75} + 2 \cdot C_{94} + C_{104} + C_{108} + C_{114} + 2 \cdot C_{126} + C_{117} + C_{124} \quad (3.10)$$

This equation can be derived using equations 1, 2, 13, 15, 18 and 23. Similarly, the equation:

$$\binom{C_{12}}{1} \cdot \binom{C_1 - 1}{1} = C_{79} + C_{97} + C_{116} + C_{109} + 2 \cdot C_{93} + C_{119} + 2 \cdot C_{127} + C_{123} \quad (3.11)$$

can easily be derived from equations 2, 3, 12, 16, 19 and 21. The 23 redundancy equations allow 23 of the 129 orbits to be removed when formulating directed graphlet-based measures. However, this only holds for the networks without anti-parallel pairs of arcs. Let us examine the case of networks with anti-parallel pairs of arcs, as presented in Figure 3.3-b. Again, let the number of orbits 0 that the node A touches be C_0. We can choose $\binom{C_0}{2}$ pairs of orbits 0 that node A touches. As shown in Figure 3.3-b, each pair can result in (1) node A touching an orbit 6 in case nodes B and C are not connected, (2) node A touching orbit 11 in case nodes B and C are connected with an edge regardless of its direction, or (3) the nodes B and C are connected with a bidirectional edge which does not correspond to any of the 129 orbits, but, as explained in Section 3.1.1, contributes to two orbits 11. The orbit counts that are the consequence of scenarios (1), (2) and (3) cannot be simply added because scenarios (2) and (3) are not mutually independent, thus the correct equation cannot be derived.

Since the majority of real world directed networks contain anti-parallel pairs of arcs, we perform all performance evaluation experiments for directed graphlet-based measures based on all 129 orbits on up to four node graphlets. The performance of modified measures, which would take into account only the non–redundant orbits, in the networks without anti-parallel pairs of arcs, is a matter of future research.

3.1.4 Implementation of Directed Graphlets and Orbits Counting Algorithm

We implemented a counting algorithm that counts all up to four node graphlets in a directed network, as defined in Section 3.1.1, and all the orbits that each node in the network touches. As discussed in Section 3.1.1 we count graphlets and orbits in directed networks which can contain anti-parallel pairs of arcs but no multiple edges or self-loops. So, when the network is loaded into a data structure in memory, we pre-process the data to remove all selfloops (and any nodes that were solely involved in that type of interactions) and remove all, if any, multiple edges in the network.

There are different approaches for implementing an algorithm to count the sub-graphs of a network. Some of the approaches which focus on speed performance include sampling [171,172], are based on pattern similarities [173], or rely on reconfigurable hardware accelerators based on Field-Programmable Gate Array (FPGA) chips, where hardware design was implemented using Verilog hardware description language [174]. The first counting algorithm for undirected graphlets was based on direct enumeration, with corrections for the over-counting graphlets and orbits [175]. One of the more recent

undirected graphlet counter implementations is a combinatorial method [176] that uses a system of equations to link counts of orbits from up to five nodes graphlets, which allows it to compute all orbit counts by enumerating just a single one. However, for the 40 directed graphlet orbits from Figure 3.1, a similar set of equations can be constructed only if we were to implement a counter for networks without anti-parallel arcs where graphlets are induced sub-graphs. Our implementation counts graphlets and orbits in the networks with anti-parallel pairs of arcs, where graphlets are not strictly induced (recall Figure 3.2) and we cannot establish the system of equations, as discussed before. Thus, our method of choice is the direct enumeration approach.

For each node in the network we construct the list of node's successors and predecessors. We visit each node in the network and update counts of all up to four node orbits as follows. Graphlet G_0 contains orbits 0 and 1 (Figure 3.1). When the counting algorithm visits a node in the network, it counts the node's successors to determine the count of orbits 0 for the visited node. In the same iteration, the counter increments the count of orbits 1 for the node's successors and updates the count of graphlets G_0 in the network. Similarly, for each set of orbits that belong to the same graphlet, we update counts of these orbits and the count of the corresponding graphlet in the same iteration, when visiting a node in the network. For counting three-node orbits of the visited node, the algorithm iterates through the lists of the node's successors and/or predecessors and checks their relationships, or through the lists of the node's successors or predecessors and then their successors or predecessors (depending on the orbit counted). Similarly, in order to update counts for all four node orbits of a visited node, the algorithm needs to examine up to three-level-deep neighbourhood of a node. To improve the time efficiency we aim to group the graphlets that are subgraphs of one another, and update their counts and orbits in the same iteration. For example, we update counts of orbits 5 and 6 (graphlet G_2) and orbits 25, 26, 27 and 28 (graphlet G_9) in the same iteration because graphlet G_2 is induced on graphlet G_9 (see Figure 3.1).

Following the approach described above, the graphlets containing automorphism orbits are over-counted during the counting process. An example of this is shown in Algorithm 1 (based on the original counter implementation, source code is available in Appendices in Section B.1). On graphlet G_2 there are two non-interacting nodes touching orbit 5 and one middle node touching orbit 6 (see graphlet G_2 in Figure 3.1). Algorithm 1 updates orbits 5 and 6 and the counts of graphlet G_2 by visiting each node in the network and, among the node's successors, it looks for pairs of successors that do not interact. When found, the count of orbit 6 is updated for the visited node i, and the count of orbit 5 is updated for both nodes in the identified non-interacting pair

Algorithm 1 Updating counts of orbits 5,6,25,26,27,28 and graphlets 2 and 9.

input: G, directed graph as an edge list
V = list of nodes from G
$pred$ = container with vector of predecessors $pred(n)$ for each node $n \in V$
$succ$ = container with vector of successors $succ(n)$ for each node $n \in V$
//Note: Multiple edges and selfloops are discarded when creating pred and succ.//
output: $graphlets$, vector of graphlet counts $graphlets(i)$ for each graphlet $i \in [0, 39]$
output: $orbits$, matrix of orbits counts $orbit(n)(j)$ for each node $n \in V$ and orbit $j \in [0, 128]$
output: $dictionary$, list of nodes in the order that corresponds to node indexes from the $orbits$ matrix.
for $n \in V$ **do**
    Updating counts of orbits 5,6,25,26,27,28 and graphlets 2 and 9:
    for s1∈succ(n) **do**
        for s2∈succ(n) **do**
            if s1≠s2 and s1∉succ(s2) and s1∉pred(s2) **then**
                orbit(s1)(5)++
                orbit(s2)(5)++
                orbit(n)(6)++
                graphlets(2)++
                for s3 in succ(s1) **do**
                    if s3≠s2 and s3≠n and s3∉succ(s2) and s3∉pred(s2) and s3∉succ(n) and s3∉pred(n) **then**
                        orbit(s1)(26)++
                        orbit(s2)(28)++
                        orbit(n)(27)++
                        orbit(s3)(25)++
                        graphlets(9)++
                    end if
                    //Note: In the complete counter we use the last for loop to explore relationships between nodes n, s1, s2 and s3, and update their counts of orbits (49,50,51,52), (65,66,67,68), (46,47,48), (85,86,87) and (88,89,90) and counts of graphlets 18,22,17,27,28 respectively.//
                end for
            end if
        end for
    end for
end for
Correcting for overcounted graphlets and orbits:
//Note: Here, we correct overcounts only for orbits 5,6,25,26,27,28 and corresponding graphlets 2 and 9.//
for $n \in V$ **do**:
    $orbit(n)(5) = \frac{orbit(n)(5)}{2}$;
    $orbit(n)(6) = \frac{orbit(n)(6)}{2}$;
end for
$graphlets(2) = \frac{graphlets(2)}{2}$;

of successors. However, we need to distinguish the nodes in the non-interacting pair of successors of i, because, as described above, in the same iteration we are updating counts of orbits 25, 26, 27 and 28 and the corresponding graphlet G_9. This means that if the neighbourhood of the node i also corresponds to orbit 27 on G_9, then the count of orbit 26 is updated for one of the nodes in the non-interacting pair of successors of i, while orbit 28 is updated for the other node in the pair. As a result, each non-interacting pair of successors (j, k) of the node i needs to be examined twice: once to check if the node j corresponds to orbit 26 (consequently the nodes i and k correspond to orbits 27 and 28 respectively) and a second time to check if the node k corresponds to orbit 26 (consequently the nodes i and j correspond to orbits 27 and 28 respectively). If both scenarios are true, the count of orbit 27 for the node i will be updated twice, which corresponds to two different graphlets G_9 in the network (on each of them the nodes j and k touch different orbits). However, examining the nodes i, j and k twice results in updating the count of orbit 6 for the node i twice, updating the count of orbit 5 twice for each of the nodes j and k, and counting graphlet G_2 twice, although the nodes i, j and k account for only one graphlet G_2. Hence, when the counting process is completed for all nodes in a network, we divide the following counts by two: the counts of orbit 5 for all nodes, the counts of orbit 6 for all nodes and the count of graphlets G_2 in the network. Similarly, the other graphlets containing automorphism orbits are over-counted, depending on how the algorithm iterates over the nodes. We solve this by correcting for all such orbit and graphlet over-counts after the counting process is finalised.

The complexity of our algorithm is $O(N \times d^3)$, where N is the number of nodes in the network and d is the maximum degree over all nodes in the network. However, since the counting algorithm is implemented so that each node in the network is visited separately, the code can easily be parallelised by dividing nodes in the network into sets and assigning each set of nodes to a separate job. Each job should separately maintain the temporary DGDVs for all nodes in the network, so when the job is counting orbits that a particular node touches, it can still update the orbits for other nodes, even if they are not within its set. This approach gives jobs the flexibility to be either separate threads/processes on a single CPU or distributed over a cluster resource, adding scalability to our approach. After all the jobs are completed, values from all temporary DGDVs for each node are added together. All the corrections for the over-counts discussed above, should be performed after the merging of the temporary vectors.

3.2 Evaluation of Directed Graphlet-based Methods for Network Comparison using Synthetic Data

To evaluate directed graphlet-based network comparison measures, we assess their ability to classify directed network models. Since real world networks are noisy and incomplete, we also validate the robustness of the measures to noise by evaluating the performance of clustering model networks to which we introduced three types of noise: (1) random addition of edges, (2) random removal of edges, and (3) random rewiring of edges. We contrast the performance of graphlet-based measures with other common directed network comparison measures (degree distribution and spectral distance).

3.2.1 Standard Methods for Evaluation of Clustering Performance

To formally assess and compare the clustering quality, we quantify the performance of different measures by using the standard Receiver Operator Characteristic (ROC) curve [177]. We use the following approach. For small increments of the value ϵ, in the range between minimum and maximum value of the distance between any two networks from the set that we perform the clustering on, we count:

- *True Positives* (TP) – number of pairs that are of the same model and have a pairwise distance smaller than ϵ,
- *False Positives* (FP) – number of pairs that are not of the same model but have a pairwise distance smaller than ϵ,
- *True Negatives* (TN) – number of pairs that are not of the same model and have a pairwise distance greater than ϵ,
- *False Negatives* (FN) – number of pairs that are of the same model but have a pairwise distance greater than ϵ,

We then compute the *True Positive Rate* (TPR) and *False Positive Rate* (FPR) as follows:

$$TPR = \frac{TP}{TP + FN}; \tag{3.12}$$

$$FPR = \frac{FP}{FP + TN}; \tag{3.13}$$

ROC curve is a plot of TPR against FPR for all increments of ϵ. The Area Under the ROC Curve (AUC) measures the quality of the grouping obtained using a given distance measure. For any two random pairs of items being grouped, such that one is

a pair of *true* items and the other is a pair of *false* items, AUC value corresponds to probability that the distance between the items from the *true* pair is smaller than the distance between the items from the *false* pair. In our case, the true pair corresponds to two networks of the same model, while the false pair corresponds to two networks of different models.

We also measure the area under the "truncated" ROC curve (AUC_n) [178], which evaluates early identification performance and measures TPR against FPR up to a given false positive threshold n, which means that n false positives are allowed. The average number of incorrectly clustered networks per query network is called Errors Per Query (EPQ), so we have n=EPQ x N, where N is number of networks in comparison. We use EPQ=10. We are interested in truncated ROC curve analysis because it captures the performance on early retrieval, i.e. how well the most similar pairs are clustered, which is often the focus of clustering algorithms.

Finally, we observe the Precision-Recall curve which plots precision (the fraction of retrieved items that are relevant) against recall (also known as sensitivity – the fraction of relevant items that are retrieved):

$$Precission = \frac{TP}{TP + FP}; \tag{3.14}$$

$$Recall = \frac{TP}{TP + FN}; \tag{3.15}$$

The Area Under Precision and Recall curve (AUPR), also called *average precision*, measures the quality of the grouping obtained using a given distance measure. It is more robust to negatives than ROC curve analysis [179], and model fitting experiments deal with significant portions of negatives, as those are pairs of networks from different models that are grouped together. The Precision-Recall curve evaluates the classifications of positives, whether they are true or false, and shows how many true positives can be retrieved before making an error (false positive).

3.2.2 Clustering Directed Model Networks

We use existing directed network models for SF directed graphs and ER directed graphs and propose SF gene-duplication (SF-GD), GEO, and GEO-GD directed network models to generate the random networks as follows:

- Scale free directed graph. Scale free networks can be created using the Albert-Barabasi model [52], based on the preferential attachment described in Section

1.4. Starting with a small number of vertices (m_0), we build directed scale free networks also using preferential attachment model [180] where a new node v is added to the network with probability α and a directed edge (v, w) is added to the existing node w - which is chosen based on the existing nodes' in-degrees. Similarly, a new node w is added to the network with probability γ and a directed edge (v, w) is added to the existing node v - which is chosen based on the existing nodes' out-degrees. Also, there is a probability β of adding an edge between two existing nodes in the network (v, w), where v is chosen based on the nodes' out-degrees and w is chosen based on the nodes' in-degrees. The sum of α, β and γ equals 1. To achieve the desired network densities we allow the node being added to the network to be connected to $m \leq m_0$ existing nodes, analogous to Albert-Barabasi model [52]. We do not allow duplicated edges.

- The scale free gene-duplication (SF-GD) model is a biologically motivated model that imitates gene duplication and mutation processes [181]. We implement the directed model as follows. In the duplication step a node in the network is selected at random and a new node is created together with the connections to/from nodes that the "parent" node had. The edge between the new node and his parent node is added with probability p, and edge direction is decided randomly with probabilities of 0.5 for both directions. The mutation step is imitated so that each edge that the new node "inherits" from a parent node is deleted with the probability q. This procedure is repeated until the desired number of nodes is obtained. In our implementation we start with probabilities $p = q = 0.5$. When the desired number of nodes is reached, we check the density of the obtained network. If the density is lower or greater than desired value, we decrease or increase the value q by value q_{step} respectively, starting with the value $q_{step} = 0.1$. We then repeat the process of generating the network. In case we "skipped" the desired network density by decreasing or increasing the value q by q_{step} in the previous iteration, we adjust the value of q_{step} to $q_{step}/2$, and repeat the process until the desired network density is obtained.

- ER directed graph is generated so that edges, with the direction decided at random, are randomly placed between nodes with the same probability, so the desired network density is obtained.

- GEO directed graph with random edge direction. We propose this model for directed networks, based on the GEO model for the undirected networks [7]. We

use the algorithm for generating the GEO random graph described in Section 1.4, while choosing the edge direction randomly with the probability of 0.5 for each direction.

- GEO-GD directed graph with random edge direction. We propose this model for directed networks, based on the GEO-GD model for the undirected networks described in Section 1.4. We use the same algorithm, while choosing the edge direction randomly with the probability of 0.5 for each direction.

We generate 30 random networks for each of the 5 network models and each of the following sizes and densities:

- 500 nodes, densities 1% and 0.5%
- 1000 nodes, densities 1% and 0.5%
- 2000 nodes, densities 1% and 0.5%

We choose these network sizes and densities as they cover the range of sizes and densities of directed metabolic networks which are the focus of our case study, see Chapter 4.

All together, we consider $5 \times 3 \times 2 \times 30 = 900$ networks, where 5 is the number of models, 3 is the number of network sizes, 2 is the number of network densities and 30 is the number of random networks generated per each network model, size and density. We assess the ability to cluster networks of the same model. We consider two different cases when performing the clustering:

- Comparing all-to-all networks. This means we are using distances between all pairs of the 900 model networks; this gives us $\binom{900}{2} = 404,550$ pairs of networks to examine.
- Comparing only the networks of same density and size; This gives us $3 \cdot 2 \cdot \binom{5 \cdot 30}{2} = 67,050$ pairs of networks to examine.

The first case assesses the ability to cluster the same model networks, regardless of their sizes and densities. The second case considers the easier case of clustering the same model networks among the networks of same size and densities.

We evaluate the clustering using the following network distance measures: RDGF-2 distance, RGDF-3 distance, DGDDA (note that we use the value of DGDD distance, obtained as $DGDD_{dist} = 1 - DGDDA$), DGCD-13, DGCD-129, in-degree distribution distance, out-degree distribution distance, sum of in and out degrees distribution distance, and spectral distance.

We first compare the performances of these network measures on the clustering when all networks are compared. Figure 3.4 shows ROC curves, truncated ROC curves and precision-recall curves respectively. Table 3.2 shows scores for $AUPR$, AUC and $AUC_{EPQ=10}$.

Similarity measure	$AUPR$	AUC	$AUC_{EPQ=10}$
DGCD-129	0.8392	0.9345	0.01288
DGCD-13	**0.9004**	**0.9714**	**0.0159**
DGDDA	0.6874	0.8105	0.01127
RDGF-3	0.7222	0.8808	0.0066
RDGF-2	0.7232	0.8804	0.0067
In deg. dis. distance	0.4817	0.7174	0.0045
Out deg. dis. distance	0.4822	0.7175	0.0046
In/Out deg. dis. distance	0.4978	0.7104	0.0053
Spectral distance	0.2957	0.5579	0.0022

Table 3.2. AUC, AUPR and $AUC_{EPQ=10}$ scores for clustering model networks when comparing all-to-all networks. First column: Similarity measure. Second column: $AUPR$ score. Third column: AUC score. Fourth column: $AUC_{EPQ=10}$ score.

For the clustering evaluation where all networks are being compared, the ROC and Precision-Recall curves in Figure 3.4 show that directed graphlet-based measures outperform other tested distance measures. DCGD measures perform the best in model clustering. Table 3.2 indicates that DGCD-13 (up to three node graphlets) slightly outperforms DGCD-129 where all up to four node graphlets are taken into account, however the truncated ROC curve in Figure 3.4 shows that DGCD-129 does slightly better when it comes to the number of correctly clustered pairs that are at a shorter distance: these are retrieved earlier by the distance measure.

We then contrast the performances of these network measures on the clustering when the same size networks are compared. Figure 3.5 shows the resulting ROC curves, truncated ROC curves and precision-recall curves respectively. Table 3.3 shows scores for $AUPR$, AUC and $AUC_{EPQ=10}$, as defined in Section 3.2.

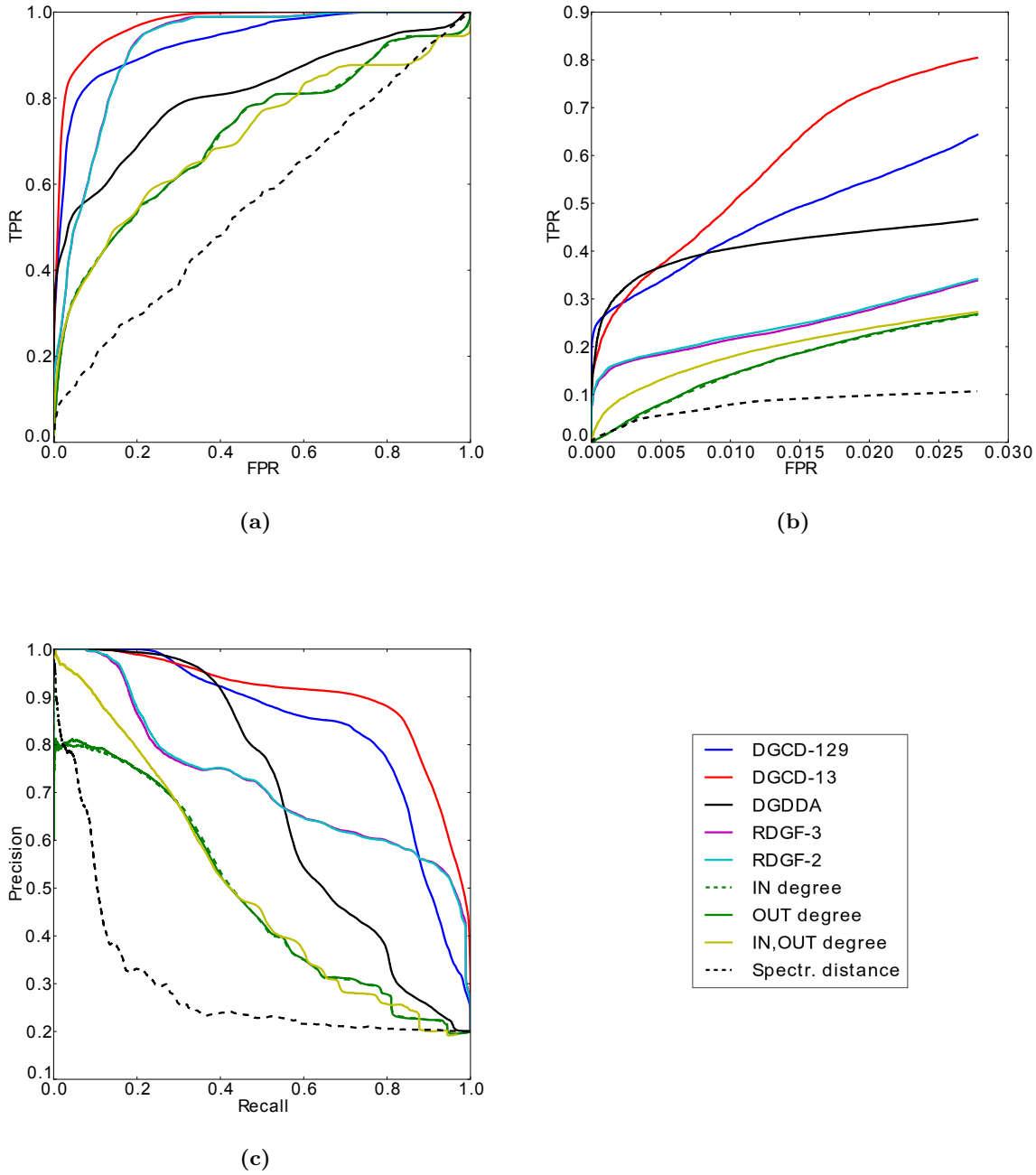


Figure 3.4. Model clustering performance when comparing all-to-all networks. (a) ROC curves. (b) Truncated ROC curves for for EPQ=10. (c) Precision-recall curves.

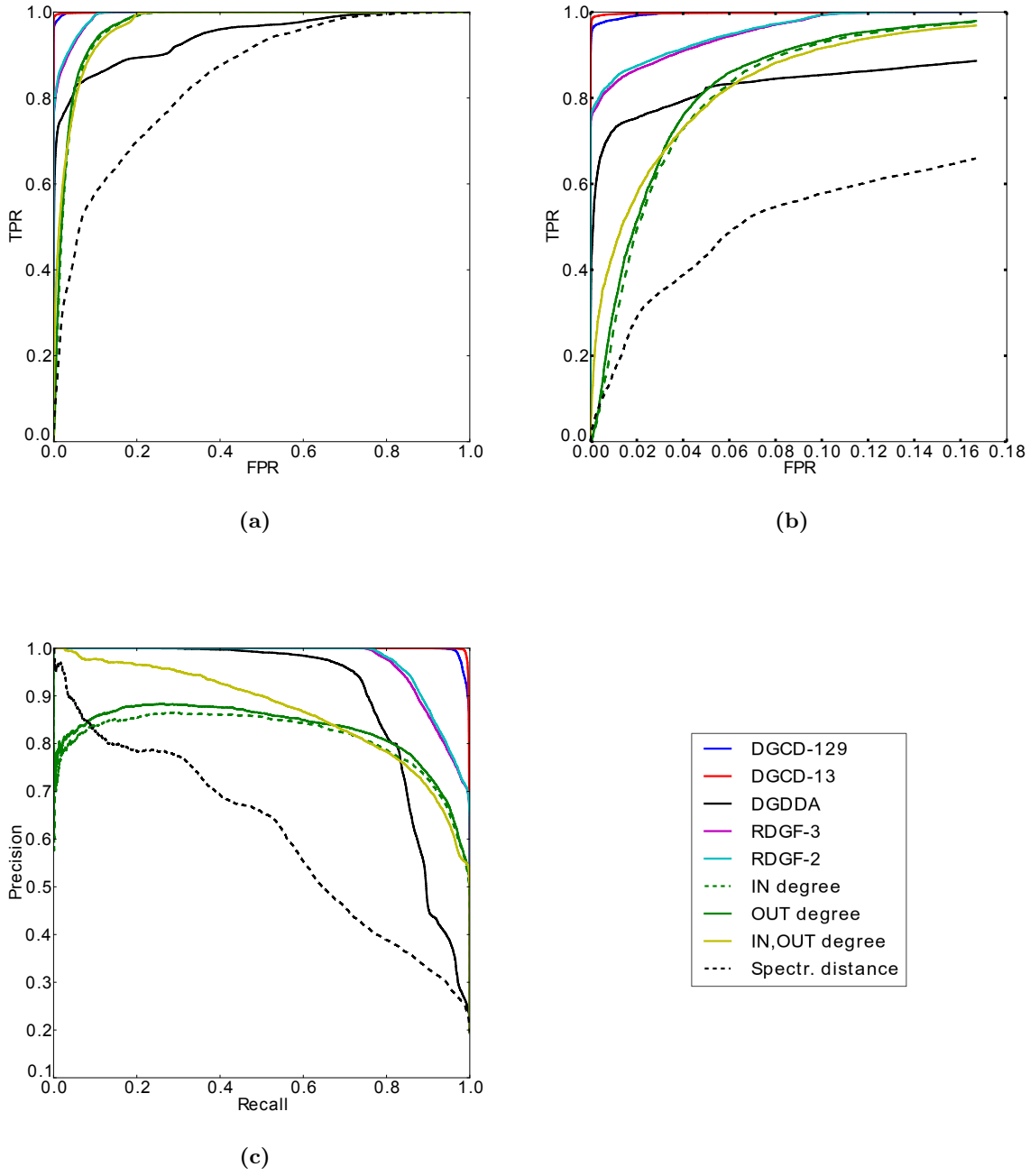
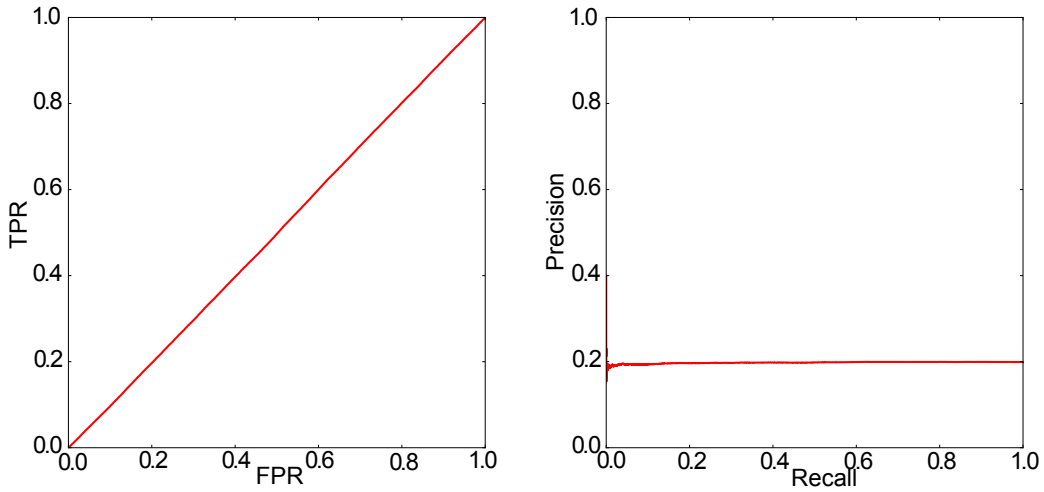


Figure 3.5. Model clustering performance when comparing networks of the same size and density. (a) ROC curves. (b)Truncated ROC curves for for EPQ=10. (c) Precision-recall curves.

Similarity measure	$AUPR$	AUC	$AUC_{EPQ=10}$
DGCD-129	0.9981	**0.9995**	0.1662
DGCD-13	**0.9988**	**0.9995**	**0.1663**
DGDDA	0.8906	0.9436	0.1374
RDGF-3	0.9701	0.9915	0.1582
RDGF-2	0.9722	0.9920	0.1587
In deg. dis. distance	0.8119	0.9652	0.1327
Out deg. dis. distance	0.8270	0.9678	0.1351
In/Out deg. dis. distance	0.8684	0.9679	0.1354
Spectral distance	0.6066	0.8443	0.0818

Table 3.3. AUC, AUPR and $AUC_{EPQ=10}$ scores for clustering model networks when comparing the networks of same size and density. First column: Similarity measure. Second column: $AUPR$ score. Third column: AUC score. Fourth column: $AUC_{EPQ=10}$.

The ROC and Precision-Recall curves in Figure 3.5 show that directed graphlet-based measures outperform other distance measures for directed networks comparison, with directed graphlet correlation distance measures showing the best performance. Again, DCGD measures perform the best in model clustering. The AUC and AUPR scores in Table 3.3 indicate that DGCD-13 (up to three node graphlets) slightly outperforms DGCD-129 (up to four node graphlets).



(a) **(b)**

Figure 3.6. Model clustering performance for random values of similarity scores. (a) ROC curve. (b)Precision-recall curve.

In addition, Figure 3.6 shows ROC and Precision-Recall curves for random network model clustering - when similarity scores between the networks are obtained at random (we take uniformly distributed random values in the range between 0 and 1). The ROC and Precision-Recall curves in Figures 3.4, 3.5 and 3.6 show that using any of the examined similarity measures yields better clustering performance than expected at random.

3.2.3 Noise Tolerance

Real world network data are noisy. This is especially true for biological networks which are still incomplete and contain false edges. Hence, we evaluate the robustness to noise for all directed graphlet-based measures and contrast them to spectral distance measure and total degree distribution distance (sum of in- and out-degree). We evaluate clustering performance for the following types of noise in the networks:

- *Networks with missing edges, which correspond to real world scenarios of incomplete networks.* We repeat the model clustering evaluation for different percentages of missing edges in the networks described in Section 3.2.2 as follows. For each of the 5 network models, 3 different network sizes (500, 1000, and 2000 nodes) and 2 network densities (1% and 0.5%) we generate 10 networks, resulting in $5 \times 3 \times 2 \times 10 = 300$ networks to cluster. We remove a random 10% of edges from each network and evaluate the clustering performance by measuring AUC. To account for the variability of the results obtained from the randomisation, we remove 10% of edges from the original networks and measure AUC 20 times to calculate the mean, maximum and minimum value of AUC for clustering networks with 10% missing edges. Following the same approach, we evaluate model clustering in cases when 20%, 30%, 40%, 50%, 60% and 70% of edges are removed, by calculating mean, maximum and minimum value of AUC. We consider removing up to 70% of edges, as the quality of clustering significantly drops for percentage of noise higher than 70%. Same as in Section 3.2.2, we evaluate the model clustering performance of different measures for the two cases: (1) we compare networks of the same density and size, and (2) we compare all networks. Note that we used 10 instead of 30 instances of each network model, size and density, as was the case for the original settings, to reduce the time complexity required for computing DGDV for a large number of networks caused by repeating the clustering 20 times for each level of noise.

- *Rewired networks which correspond to noisy real world networks.* Following the approach presented above, we calculate the mean, maximum and minimum value of AUC for different percentages of rewired edges: from 10% to 70% in increments of 10%. Similarly as in Section 3.2.2, we evaluate model clustering performance of different measures for the two cases: (1) we compare networks of the same density and size, and (2) we compare all networks.
- *Networks with added edges which correspond to networks with falsely identified edges.* Following the approach presented above, we calculate the mean, maximum and minimum value of AUC for different percentages of added edges: from 10% to 70% in increments of 10%. Similarly as in Section 3.2.2, we evaluate model clustering performance of different measures for the two cases: (1) we compare networks of the same density and size, and (2) we compare all networks.

Figure 3.7-a presents the minimum, mean and maximum AUC values for network clustering, when comparing all-to-all networks, against growing percentages of missing edges. Figure 3.7-b presents the minimum, mean and maximum AUC values for network clustering, when comparing same size networks, against growing percentages of missing edges.

Figure 3.8-a presents the minimum, mean and maximum AUC values for network clustering, when comparing all-to-all networks, against growing percentages of rewired edges. Figure 3.8-b presents the minimum, mean and maximum AUC values for network clustering, when comparing same size networks, against growing percentages of rewired edges.

Figure 3.9-a presents the minimum, mean and maximum AUC values for network clustering, when comparing all-to-all networks, against growing percentages of added edges. Figure 3.9-b presents the minimum, mean and maximum AUC values for network clustering, when comparing same size networks, against growing percentages of added edges.

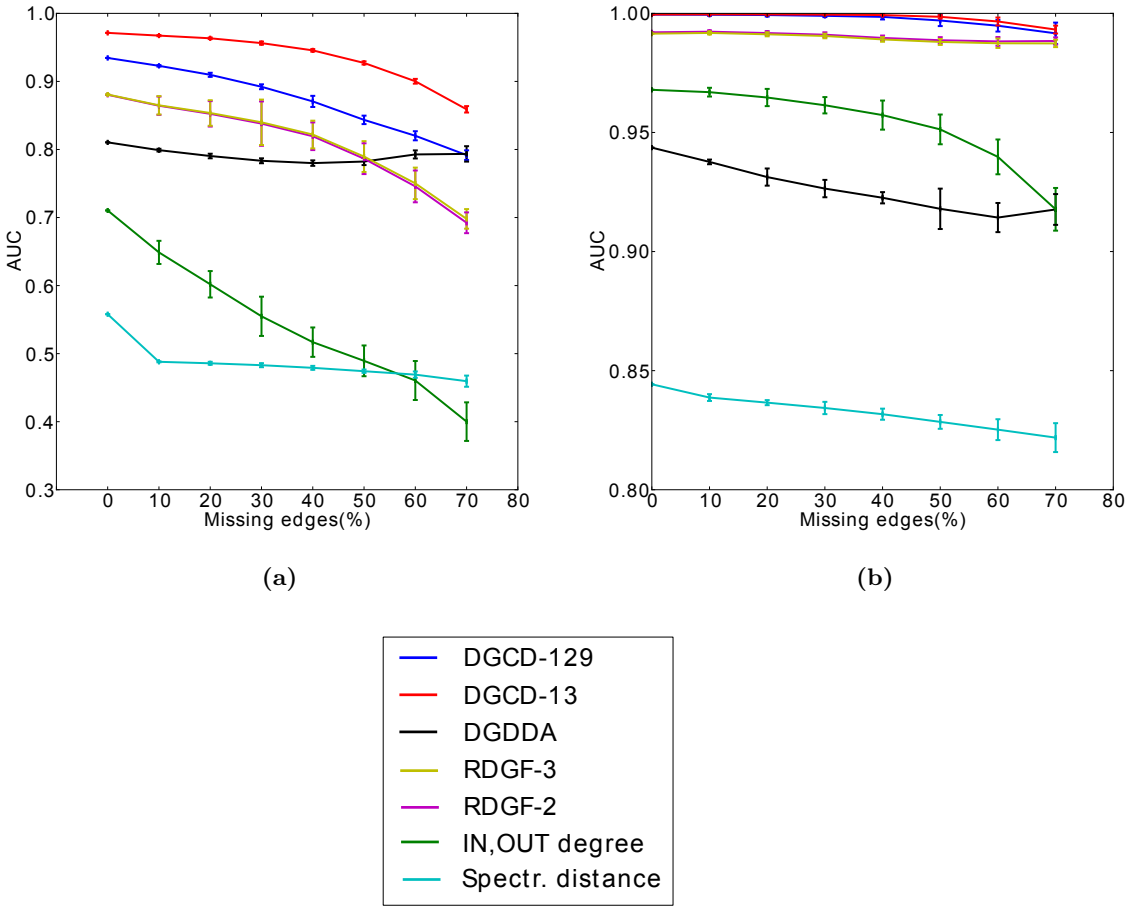


Figure 3.7. Effects of missing network edges on model clustering performance of different network distance measures. The vertical axis represents the mean, maximum and minimum value of AUC scores for the 20 randomised experiments that are performed at each of the noise levels that are presented by the horizontal axis independently. (a) AUC scores obtained by comparing all pairs of the 300 networks. (b) AUC scores obtained by comparing only the same size and density networks.

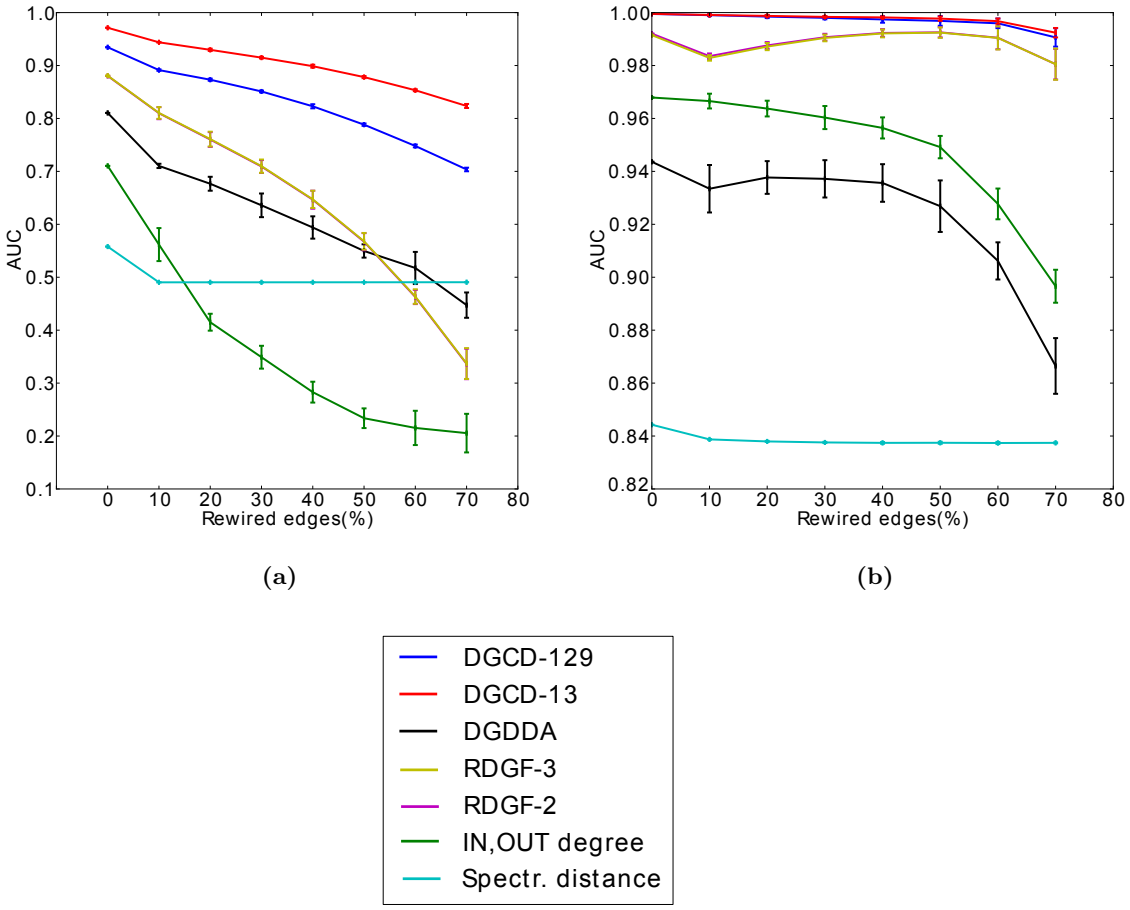


Figure 3.8. Effects of rewiring networks on model clustering performance of different network distance measures. The vertical axis represents the mean, maximum and minimum value of AUC scores for the 20 randomised experiments that are performed at each of the noise levels that are presented by the horizontal axis independently.(a) AUC scores obtained by comparing all pairs of the 300 networks. (b) AUC scores obtained by comparing only the same size and density networks.

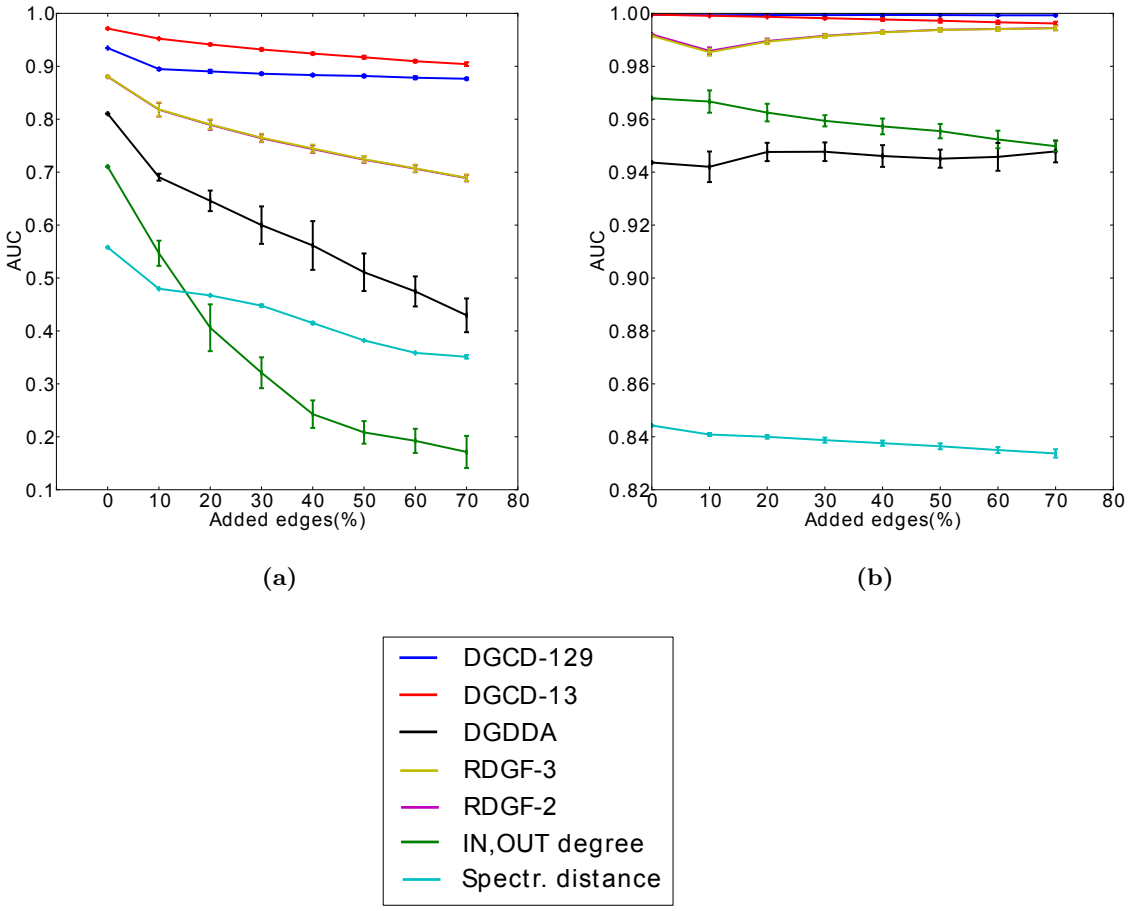


Figure 3.9. Effects of adding network edges on model clustering performance of different network distance measures. The vertical axis represents the mean, maximum and minimum value of AUC scores for the 20 randomised experiments that are performed at each of the noise levels that are presented by the horizontal axis independently. (a) AUC scores obtained by comparing all pairs of the 300 networks. (b) AUC scores obtained by comparing only the same size and density networks.

The evaluation of clustering of noisy networks show that DCGD-13 measure outperforms all other measures, except in case of noise modelled by random addition of edges in the network and clustering of networks when the same size and density network are

compared (Figure 3.9-b) when it is outperformed by DCGD-129. Also, in all observed cases, the two DCGD and the two RDGF measures outperform other network comparison measures (Figures 3.7, 3.8, 3.9). Degree distribution distance measure and DGDDA compete for the fifth position: DGDDA performs better in clustering model networks when all networks are compared regardless their size and density (Figures 3.7-a, 3.8-a, 3.9-a), while degree distribution distance better clusters the model networks when only same size and density networks are compared (Figures 3.7-b, 3.8-b, 3.9-b).

Overall results emphasise the significance of graphlet-based measures for directed networks comparison. The best results are obtained by using DCGD measures, followed by RDGF distance. RDGF-2 distance, which also takes into account the two-node orbits, slightly outperforms the RDGF-3 where these orbits are omitted.

3.3 Conclusions

In this chapter we introduced up to 4 node directed graphlets and orbits and defined and implemented directed graphlet-based heuristics. We identified orbit dependencies and accounted for them when defining directed graphlet degree vector similarity between two nodes in a network. We also derived 23 equations that describe relationships between directed graphlet orbits in networks without anti-parallel pairs of arcs. We implemented directed graphlet and orbit counting algorithm and used it on synthetic model networks when evaluating our new measures for network comparison: relative directed graphlet frequency distance, directed graphlet degree distribution similarity and directed graphlet correlation distance. We compared these measures to other common directed network comparison measures, by evaluating their performance on model network clustering and found that directed graphlet-based measures outperform others. The directed graphlet correlation distance performed the best in model clustering and showed the highest tolerance to noise, regardless of the type of noise in networks: random addition of edges, random removal of edges or random edge rewiring.

3.4 Author's Contributions

Section 3.1 Anida Sarajlić defined 40 directed graphlets and 129 orbits, implemented the graphlet and orbits counting algorithm, generalised the existing graphlet measures to directed case, identified the redundancies between directed graphlet orbits and derived all orbit redundancy equations.

Section 3.2 Anida Sarajlić generated the directed random model networks, implemented and performed experiments for evaluation of model clustering for all analysed distance measures, generated noisy networks, implemented and performed experiments for evaluation of model clustering in the presence of noise and analysed results.

Anida Sarajlić was supervised on the work presented in this chapter by Dr. Noël Malod-Dognin and Dr. Nataša Pržulj who defined the research topic and assigned it to Anida Sarajlić.

Anida Sarajlić wrote the first draft for the paper: Anida Sarajlić, Noël Malod-Dognin, Ömer Nebil Yaverŏglu and Nataša Pržulj: "Directed Graphlets Uncover Topology–Function Relationships in Directed Metabolic Networks of Eukaryotes" in August 2015. This paper draft contained the work presented in Chapters 3 and 4. Currently (December 2015), the results presented in that paper draft are being merged with the results of application of directed graphlets to directed world trade networks, aiming for a publication with wider range of applications (Note: Anida Sarajlić provided the directed orbit and graphlet counts for the directed world trade networks, while further experiments and analyses on world trade networks were performed by Noël Malod-Dognin and Ömer Nebil Yaverŏglu).

4 Application of Directed Graphlet-based Methods to Metabolic Networks

In Chapter 3 we used synthetic data to show that our new graphlet-based measures for directed network comparison outperform non-graphlet-based measures. In this chapter we apply our methodology to biological data, in particular to directed metabolic networks.

First, we contrast our new directed graphlet-based measures with other similarity measures, exploring if the topology-based clustering of directed metabolic networks of eukaryotic species agrees with their taxonomic classification. We then use directed graphlet degree vector (DGDV) similarity to show that similar local topology around enzymes in metabolic networks is an indicator of their shared biological functions. To further explore this, we use a canonical correlation analysis [182] to quantify the relationships between the local topology around the enzymes (described using DGDV) and their biological functions. We then use these relationships to predict novel functional annotations based solely on the network topology. Finally, we look for conserved topology–function relationships across metabolic networks of different eukaryotic species.

4.1 Topology-based Clustering of Metabolic Networks of Eukaryotes Agrees with Taxonomic Classification

The principal goal of evolutionary biology is to understand the evolutionary relationships between different species, which can be quantified by constructing phylogenetic trees [183]. Traditionally, phylogenetic similarities among species have been studied based on phenotypical similarities or sequence similarities [184, 185]. The phylogenetic trees can be reconstructed from the sequence alignments using maximum likelihood methods [186, 187] or Bayesian methods [188, 189]. More extensive review of phylogeny reconstruction methods is beyond the scope of this dissertation and can be found in [190].

Since the topology around molecules in biological networks is shown to be related to similar biological functions, as discussed in Section 1.1, it is expected that phylogenetically similar species have similar biological network topologies. Hence, the topological properties of networks have already contributed to constructing phylogenetic trees. Examples are the similarities between metabolic pathways, obtained by combining global network properties (diameter, clustering coefficient) and similarities of neighbourhoods around nodes [191, 192]. Another approach relies on topological properties such as network size and connectivities of common metabolites, to quantify the similarities between undirected metabolic networks [193] and to use them for phylogenetic reconstruction. Also, a graphlet-based alignment algorithm GRAAL [194] was applied to PPI networks to demonstrate that species phylogeny can be extracted from purely topological alignments. This foregrounds network topology as a new source of phylogenetic information, complementing the sequence information.

Here, we explore the correspondence between the topological similarity between directed metabolic networks of eukaryotic species and their known phylogenetic classification. For network comparison we use our directed graphlet-based measures and contrast their performance with other similarity measures to directed networks. We use the taxonomic classification of species, which is based on the evolutionary relationships between organisms and is directly related to phylogenetic trees.

4.1.1 Methods

4.1.1.1 Data Sets

Metabolic networks. We parsed the organism-specific pathway data of all eukaryotes (299 species) which were available from the KEGG/PATHWAY database [67] in December 2014 and reconstructed the metabolic networks as follows. The KEGG/-PATHWAY database maintains the molecular interaction and reaction relations for each organism specific pathway and provides information such as: (1) pathways with the reactions (links) between enzyme-coding genes and metabolites (enzymes are given in Entrez gene notation) and (2) hierarchical classification that groups enzymes into families which catalyse similar reactions. Links can be directed or undirected, depending on the chemical reversibility of a specific reaction. We consider only the directed links. A directed link between two enzymes in a metabolic network denotes that one enzyme catalyses a reaction whose product is a substrate for a reaction catalysed by the other enzyme. We construct the directed metabolic network where nodes correspond to enzyme-coding genes. Note that we use the terms gene and enzyme interchangeably

when we refer to nodes, as the nodes in the network correspond to genes coding for enzymes. The sizes of our metabolic networks vary, with the number of nodes being mainly between 500 to 2000, and edge densities in the 0.5%–1% range.

Taxonomic classification. We downloaded the taxonomic classification of eukaryotes from the NCBI database in February 2015. This database provides a classification of species according to: domain, kingdom, phylum, class, order, family and genus. We evaluate the clustering of eukaryotic species from KEGG that were identified in NCBI database files (297 out of 299 of them) according to six levels of taxonomic classifications: kingdom, phylum, class, order, family and genus, where kingdom corresponds to the most general and genus corresponds to the most specific level of classification. Note that not all 297 species have every taxonomy level specified. Specifically: (1) 297 species are related to a specific genus levels, but 181 of them are related to a genus that has only one member (species) in its cluster, leaving only 116 species to cluster based on genus, (2) 274 species are related to specific families, but 112 of them are the only member of their cluster (these are family clusters with just one member), leaving 162 species to cluster based on family, (3) 273 species are related to specific order, 60 of them are the only member of their cluster (order clusters with just one member), leaving 213 species to cluster based on order, (4) 237 species are related to a specific class, 20 of them are the only member of their cluster (class clusters with just one member), leaving 217 species to cluster based on class, (5) 271 species are related to a specific phylum, 7 of them are the only member of their cluster (phylum clusters with just one member), leaving 264 species to cluster based on phylum, (6) 251 species are related to specific kingdom.

4.1.1.2 Clustering Evaluation

We assess the ability of directed network distance measures to cluster directed metabolic networks according to the six levels of the NCBI taxonomic classification. We evaluate the following measures for comparison of directed networks: RDGF-2, RDGF-3, DGDDA, DGCD-13, DGCD-129, in-degree distribution distance, out-degree distribution distance, sum of in and out degrees distribution distance, and spectral distance (all described in detail in Sections 1.3.1 and 3.1.2). We evaluate the quality of clustering using ROC and Precision-Recall curves and compare $AUPR$ and AUC scores (described in section 3.2) for all analysed distance measures.

Notice that there is an observational bias in the data, because the interactions in metabolic networks of less explored species are inferred from experimentally more ex-

plored species, based on their phylogenetic similarity. Hence, it is expected that the similarity of the topologies of metabolic networks will agree with the taxonomic classification of the corresponding species. We are not aiming to confirm or refute this, but to show that directed graphlet-based measures can be used to correctly group the species according to their taxonomic classification and that they outperform other commonly used measures for directed networks comparison.

4.1.2 Results

Figures 4.1 and 4.2 show ROC and Precision-Recall curves, respectively, for topology -based clustering of eukaryotic species metabolic networks, for six levels of taxonomic classification (kingdom, phylum, class, order, family and genus) and each of the evaluated network distance measures. Tables 4.1 and 4.2 present the corresponding AUC and $AUPR$ scores.

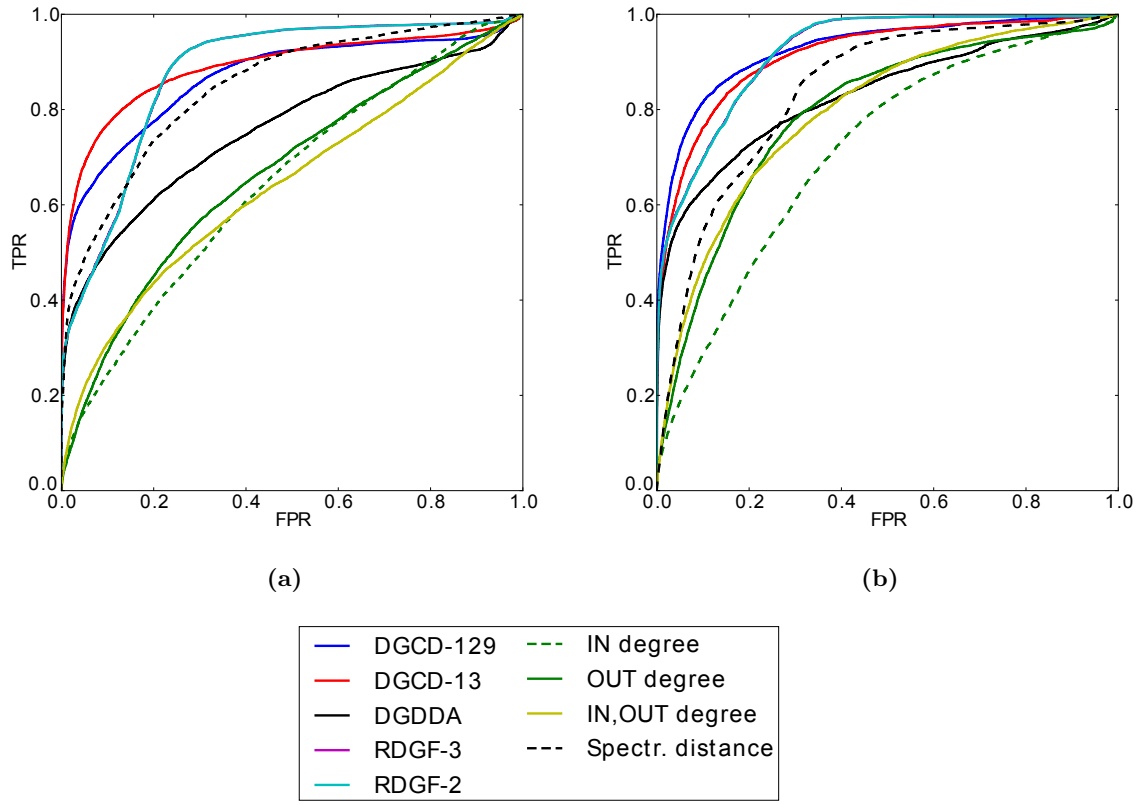


Figure 4.1. ROC curves for clustering of metabolic networks. We evaluated clustering according to: (a) kingdom, (b) phylum (continues on the next page).

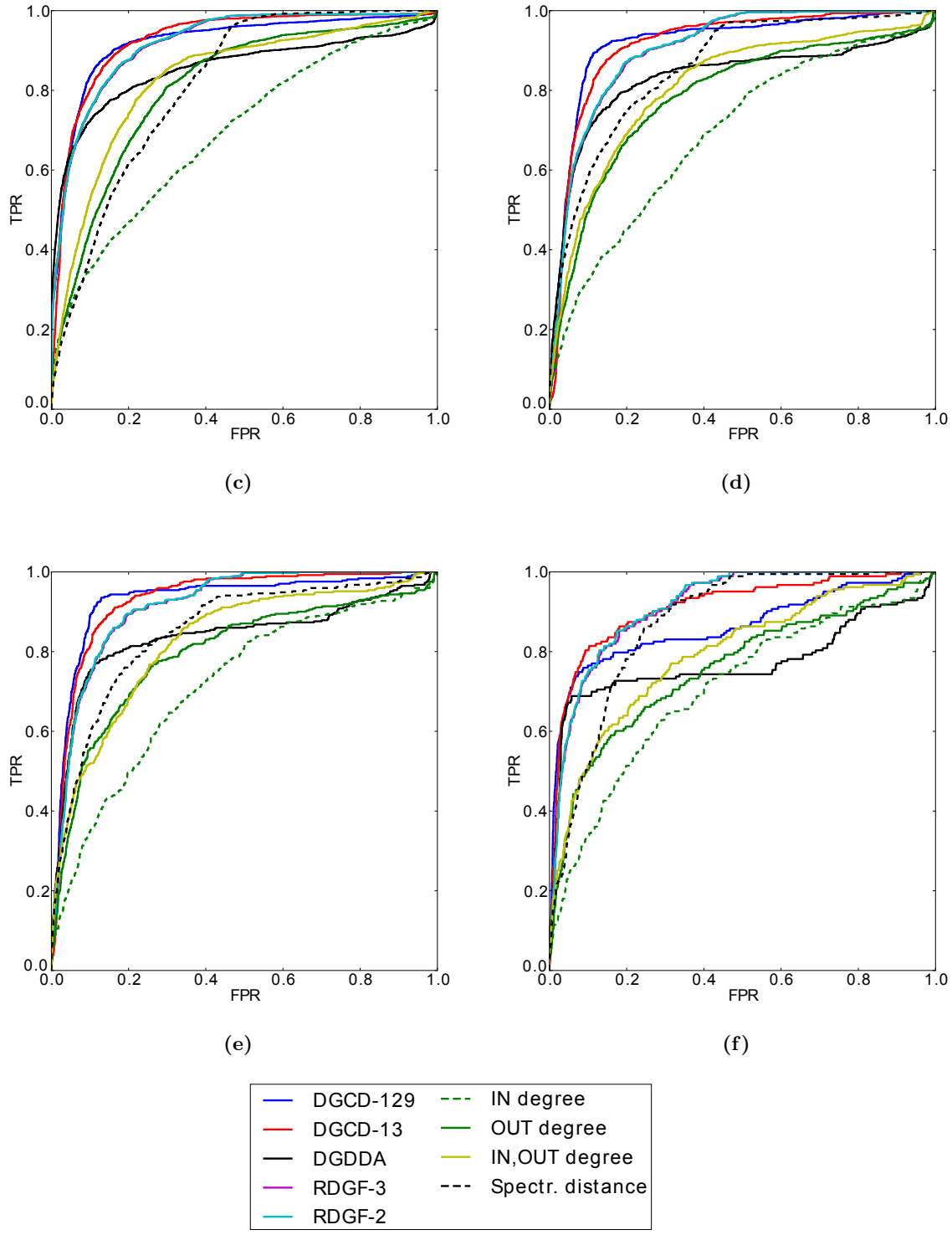


Figure 4.1. ROC curves for clustering of metabolic networks. We evaluated clustering according to: (c) class, (d) order, (e) family and (f) genus.

Sim. measure	Kingdom	Phylum	Class	Order	Family	Genus
DGCD-129	0.8670	**0.9277**	0.9213	**0.9175**	0.9292	0.8631
DGCD-13	**0.8890**	0.9150	**0.9275**	0.9150	**0.9304**	0.9112
DGDDA	0.7574	0.8317	0.8549	0.8307	0.8382	0.7722
RDGF-3	0.8778	0.9218	0.9189	0.9044	0.9152	0.9150
RDGF-2	0.8775	0.9219	0.9198	0.9054	0.9161	**0.9168**
In deg.	0.6457	0.7144	0.6947	0.6920	0.7152	0.7058
Out deg.	0.6667	0.7919	0.8081	0.7856	0.7963	0.7578
In/Out deg.	0.6407	0.7974	0.8302	0.8103	0.8263	0.7934
Spect.dist.	0.8479	0.8393	0.8129	0.8582	0.8522	0.8713

Table 4.1. AUC scores for clustering metabolic networks according to taxonomic classification. First column: Similarity measure. Second to seventh column: AUC scores for clustering based on the kingdom, phylum, class, order, family and genus respectively. The best score for each level is printed in bold.

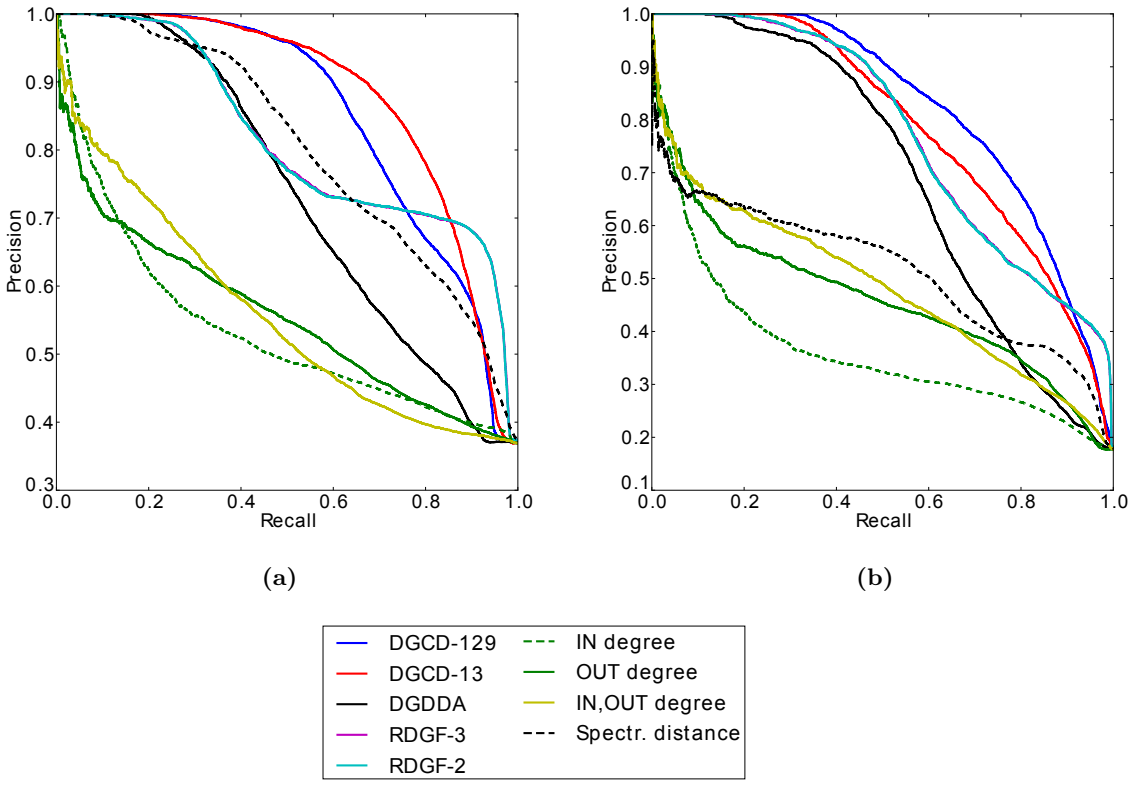


Figure 4.2. Precision-recall curves for clustering of metabolic networks. We evaluated clustering according to: (a) kingdom, (b) phylum (continues on the next page).

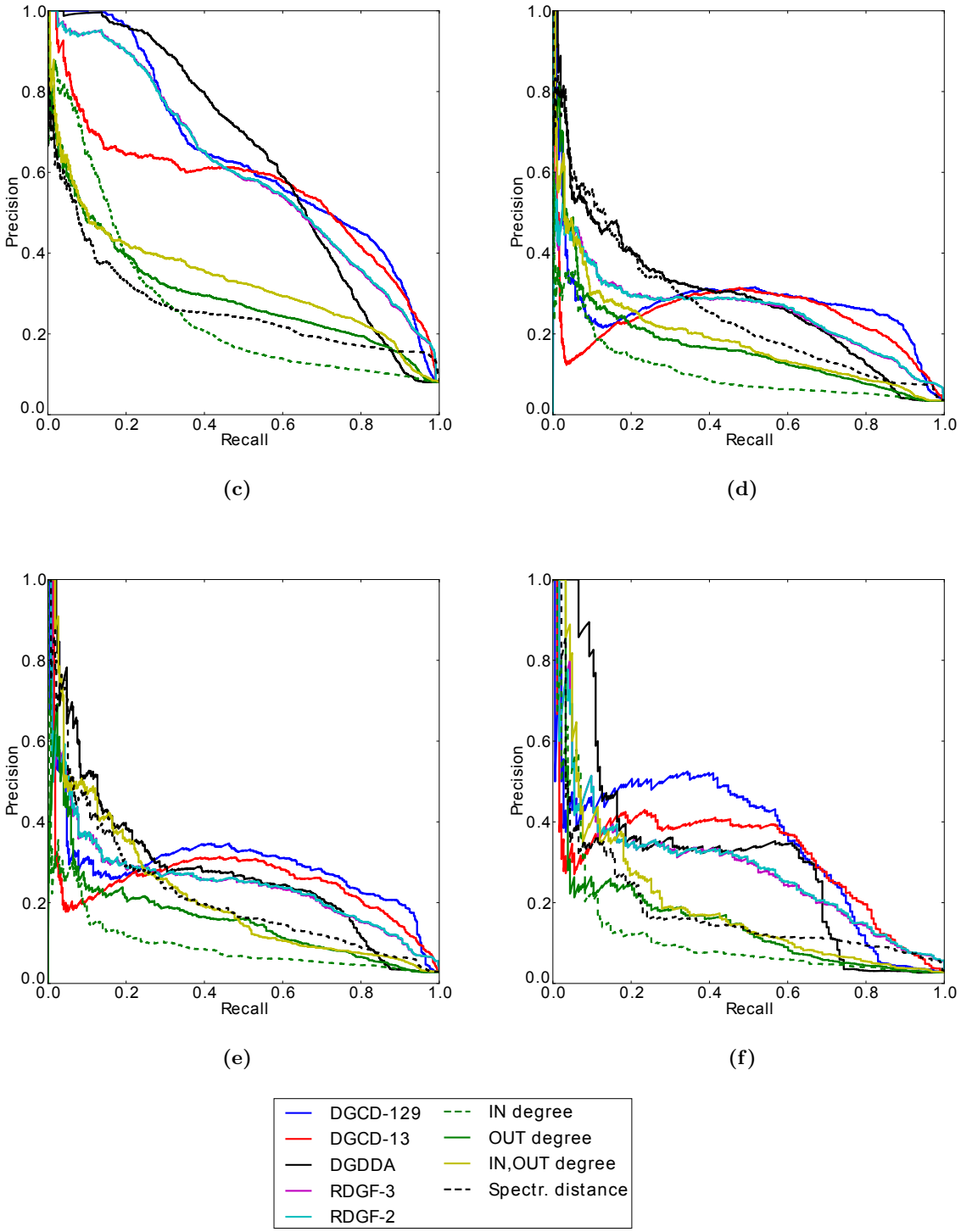


Figure 4.2. Precision-recall curves for clustering of metabolic networks. We evaluated clustering according to: (c) class, (d) order, (e) family and (f) genus.

Sim. measure	Kingdom	Phylum	Class	Order	Family	Genus
DGCD-129	0.8521	**0.8240**	**0.6388**	0.2747	**0.2891**	**0.349**
DGCD-13	**0.8787**	0.7870	0.5596	0.2414	0.2531	0.3125
DGDDA	0.7345	0.6936	0.6291	**0.2889**	0.2810	0.3192
RDGF-3	0.8137	0.7739	0.6028	0.2553	0.2499	0.2841
RDGF-2	0.8135	0.7739	0.6040	0.2557	0.2511	0.2860
In deg.	0.5346	0.3631	0.2638	0.1054	0.0935	0.1173
Out deg.	0.5552	0.4623	0.2992	0.1723	0.1576	0.1566
In/Out deg.	0.5583	0.4830	0.3357	0.1879	0.2162	0.1974
Spect. dist.	0.8016	0.5167	0.2674	0.2589	0.2197	0.1834

Table 4.2. AUPR scores for clustering metabolic networks according to taxonomic classification. First column: Similarity measure. Second to seventh column: AUPR for clustering based on the kingdom, phylum, class, order, family and genus respectively. The best scores are printed in bold.

These results indicate that the topology of directed metabolic networks can be used for the taxonomic classification of species, with the best AUC scores being between 0.89 and 0.93 for all taxonomic classification levels. The scores in Tables 4.1 and 4.2 reveal that graphlet-based measures outperform all other tested measures for directed network comparison. Like in the case of synthetic networks (Section 3.2.2), the best results are obtained for DGCD-129 and DGCD-13 measures.

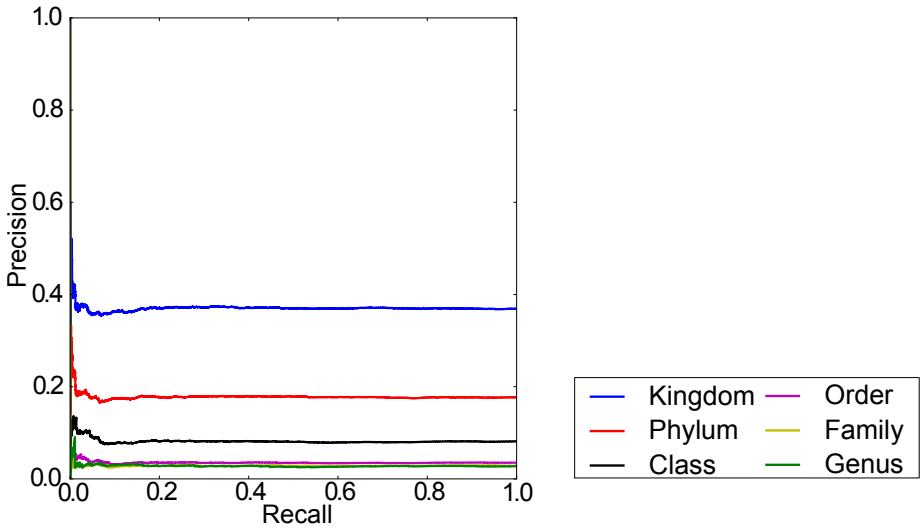


Figure 4.3. Precision-recall curves for clustering of directed metabolic networks for random values of similarity scores. We evaluated clustering according to all six levels of taxonomic classification.

In addition, Figure 4.3 displays the Precision-Recall curves for the clustering of the metabolic networks according to all six levels of taxonomic classification, when the similarity scores between the networks are obtained at random (we take uniformly distributed random values in the range between 0 and 1). The Precision-Recall curves in Figures 4.2 and 4.3 show that using any of the examined similarity measures yields better clustering performance than expected at random. Also, regardless the similarity measure, the average precision of clustering decreases for the more specific levels of classification (genus, family, order), suggesting that the metabolic networks of species that have diverged more recently in time differ less than networks of species that diverged earlier in evolutionary history. Figure 4.3 conveys that even for random similarity scores between the networks, the clustering precision decreases as the levels of taxonomic classification become more specific (*i.e.*, the highest and the lowest precision is obtained for kingdom-based and genus-based clustering, respectively). This is because more specific levels of taxonomic classification have a larger number of clusters with fewer elements, compared to the number and size of clusters at more general levels of taxonomic classification. In particular, at genus level there are 116 species divided into 42 genus clusters, while at kingdom level there are 251 species divided into just 3 different kingdoms. Hence, a higher number of false positive species pairs is expected when evaluating clustering based on genus, resulting in lower precision values.

Figure 4.4 shows a phylogenetic tree for eukaryotes constructed using the distance matrix containing the values of DGCD-13 between the metabolic networks. We use T-Rex (Tree and Reticulogram Reconstruction) web server [195] to visualise the phylogenetic tree. Since the obtained phylogenetic tree includes all 299 eukaryotic species, whose names we cannot capture in the figure, Figure 4.4 focuses on parts of the tree around *H.sapiens* (from the kingdom of *Animalia*) and two model organisms, *S. cerevisiae* and *A. thaliana* from the kingdoms of *Fungi* and *Plantae* respectively. The clusters of species grouped according to three different kingdoms from the NCBI database, *Animalia*, *Plantae*, and *Fungi*, are captured well in Figure 4.4 (recall that out of the 299 species for which we obtained the tree, we have the information about the kingdoms for 251 of them). However, when exploring the tree at more specific levels of taxonomic classification, we can find inconsistencies, such as human (*H. sapiens*) being closer to domestic cow (*B. taurus*) than to chimpanzee (*Pan troglodytes*). Notice that this corresponds to low Precision-Recall scores obtained for the clustering based on the more specific levels of taxonomic classification (order, family and genus), as shown in Table 4.2.

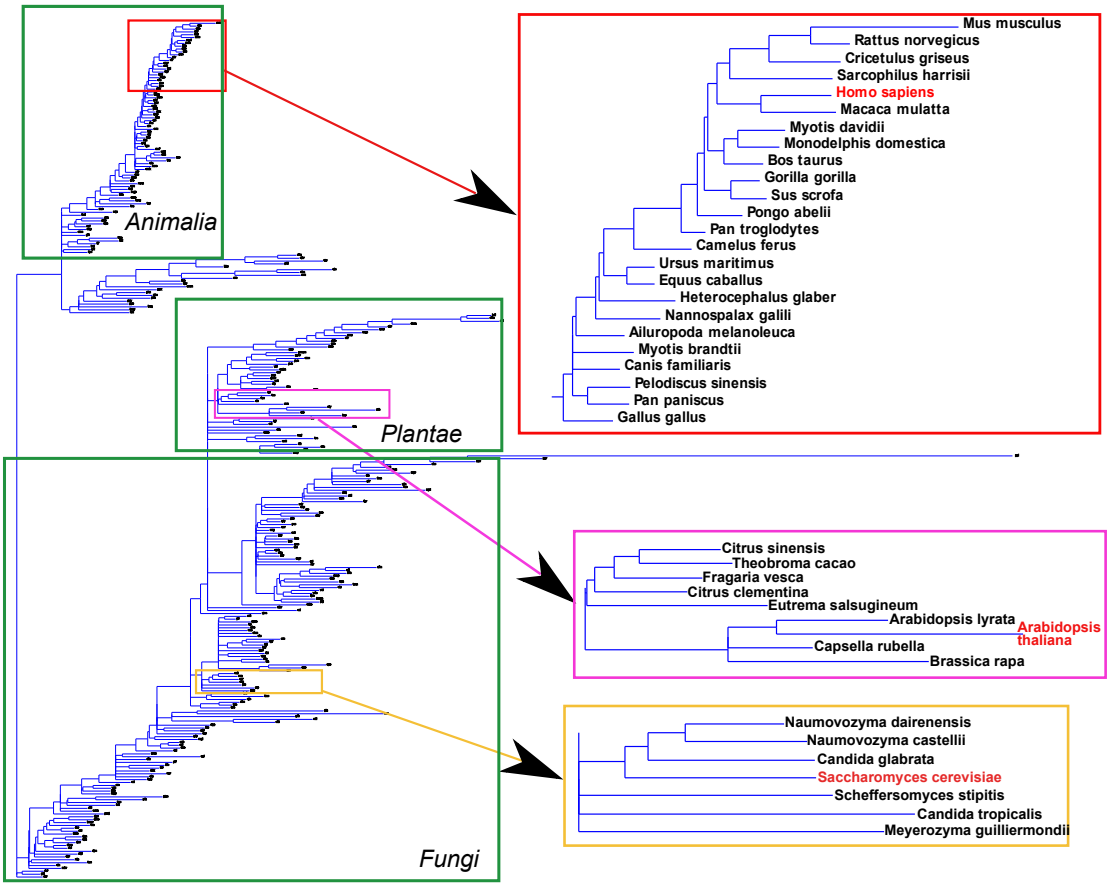


Figure 4.4. Phylogenetic tree of eukaryotes. Tree is obtained using DGCD-13 measure.

4.1.3 Comparison with Undirected Metabolic Networks

To evaluate our directed graphlet-based measures against the undirected graphlet-based ones, in this section, we explore if the topological similarities between undirected metabolic networks of eukaryotic species agree with their known phylogenetic classification. Here, we use undirected network comparison measures and evaluate the clustering of eukaryotic species according to all six levels of taxonomic classification. We compare the obtained scores with the above-presented ones. The goal is to examine whether the direction of the edges in the metabolic networks contributes to the quality of topology-based taxonomic classification. We perform the clustering evaluation in the same way as for the directed metabolic networks, with the following differences:

- We consider all edges in the metabolic networks from Section 4.1.1.1 to be undirected.

- We use undirected graphlet-based measures for network similarity: GCD-73, GCD-11, GDDA and RGFD, as defined in Section 1.3.3.
- As in the case of directed networks, we also include the following commonly used measures for the similarity between undirected networks: degree distribution distance and spectral distance.

Figures 4.5 and 4.6 show ROC and Precision-Recall curves, respectively, for topology-based clustering of eukaryotic species' metabolic networks, for six levels of taxonomic classification (kingdom, phylum, class, order, family and genus) and each of the evaluated network distance measures. The performance of undirected measures is shown in dotted lines, against the performance of comparable directed measures. Tables 4.3 and 4.4 show the corresponding AUC and $AUPR$ scores for directed and undirected measures.

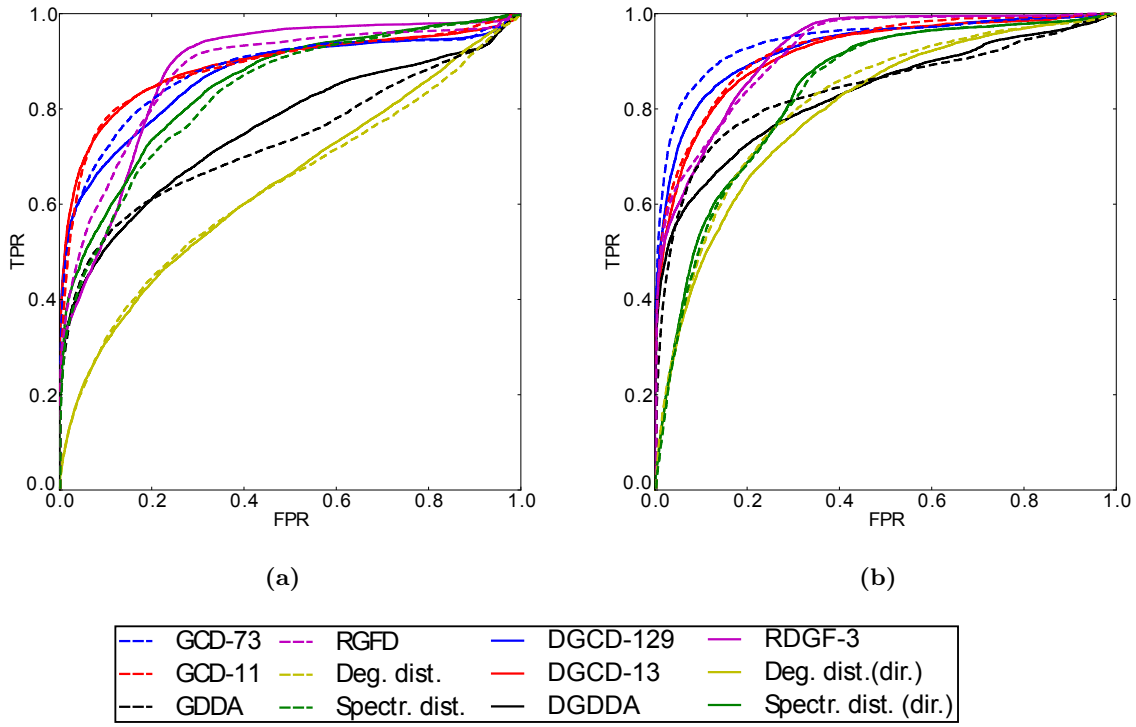


Figure 4.5. Comparison of ROC curves for clustering of undirected and directed metabolic networks. We evaluated clustering according to: (a) kingdom, (b) phylum (continues on the next page).

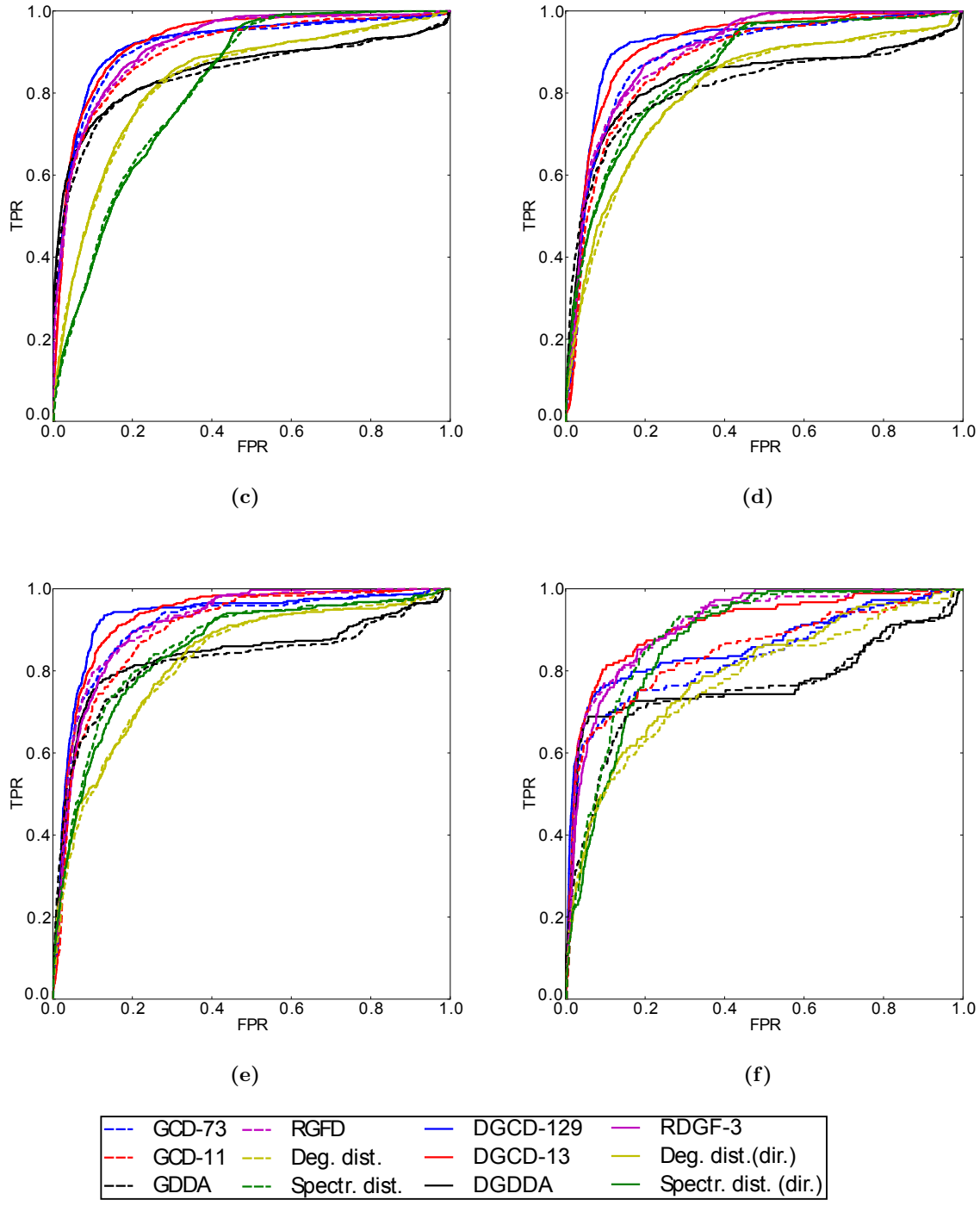


Figure 4.5. Comparison of ROC curves for clustering of undirected and directed metabolic networks. We evaluated clustering according to: (c) class, (d) order, (e) family and (f) genus.

Sim. measure	Kingdom	Phylum	Class	Order	Family	Genus
GCD-73	0.8799	**0.9447**	0.9138	0.8925	0.9083	0.8347
GCD-11	**0.8891**	0.9256	0.9028	0.8785	0.8984	0.8502
GDDA	0.7334	0.8422	0.8465	0.8156	0.8236	0.7550
RGFD	0.8785	0.922	0.9168	0.9014	0.9179	**0.9179**
Deg. dist.	0.635	0.8174	0.8271	0.8061	0.8203	0.7747
Spect. dist.	0.8342	0.8361	0.8149	0.8628	0.8606	0.8924
DGCD-129	0.8670	0.9277	0.9213	**0.9175**	0.9292	0.8631
DGCD-13	0.8890	0.9150	**0.9275**	0.9150	**0.9304**	0.9112
DGDDA	0.7574	0.8317	0.8549	0.8307	0.8382	0.7722
RDGF-3	0.8778	0.9218	0.9189	0.9044	0.9152	0.9150
Deg. dist. (dir.)	0.6407	0.7974	0.8302	0.8103	0.8263	0.7934
Spect.dist. (dir.)	0.8479	0.8393	0.8129	0.8582	0.8522	0.8713

Table 4.3. Comparison of AUC scores for clustering undirected and directed metabolic networks according to taxonomic classification. Top part of the table: Undirected network measures. Bottom part of the table: Directed network measures. First column: Similarity measure. Second to seventh column: AUC scores for clustering based on the kingdom, phylum, class, order, family and genus respectively. The best score for each level is printed in bold.

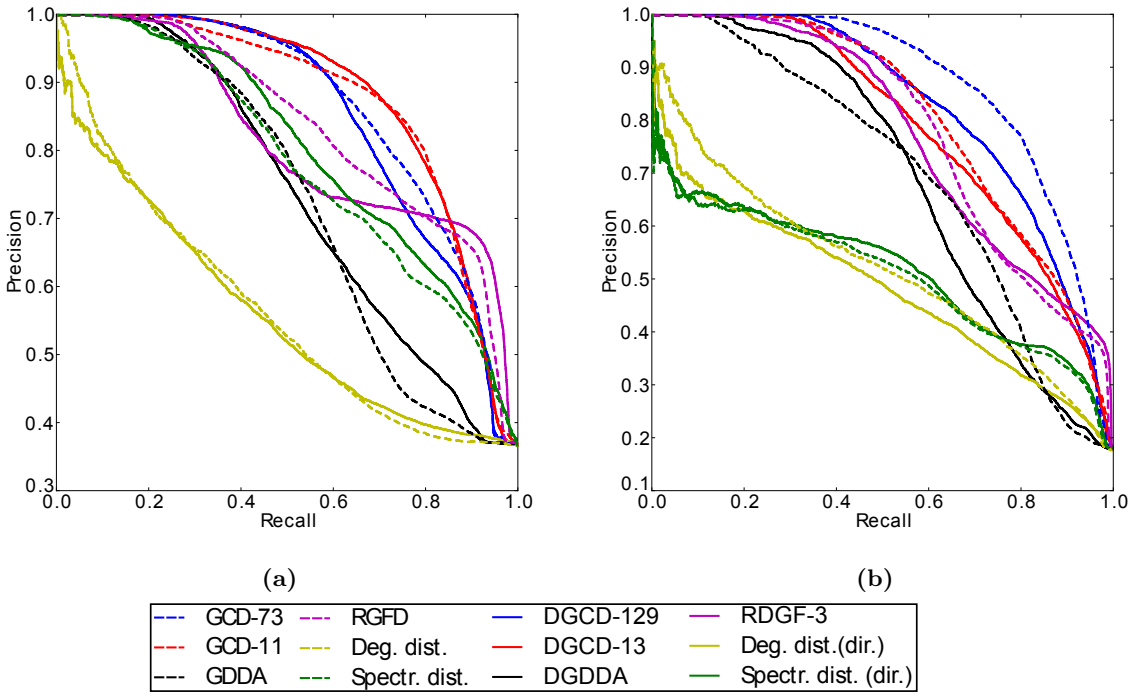


Figure 4.6. Comparison of precision-recall curves for clustering of undirected and directed metabolic networks. We evaluated clustering according to: (a) kingdom, (b) phylum (continues on the next page).

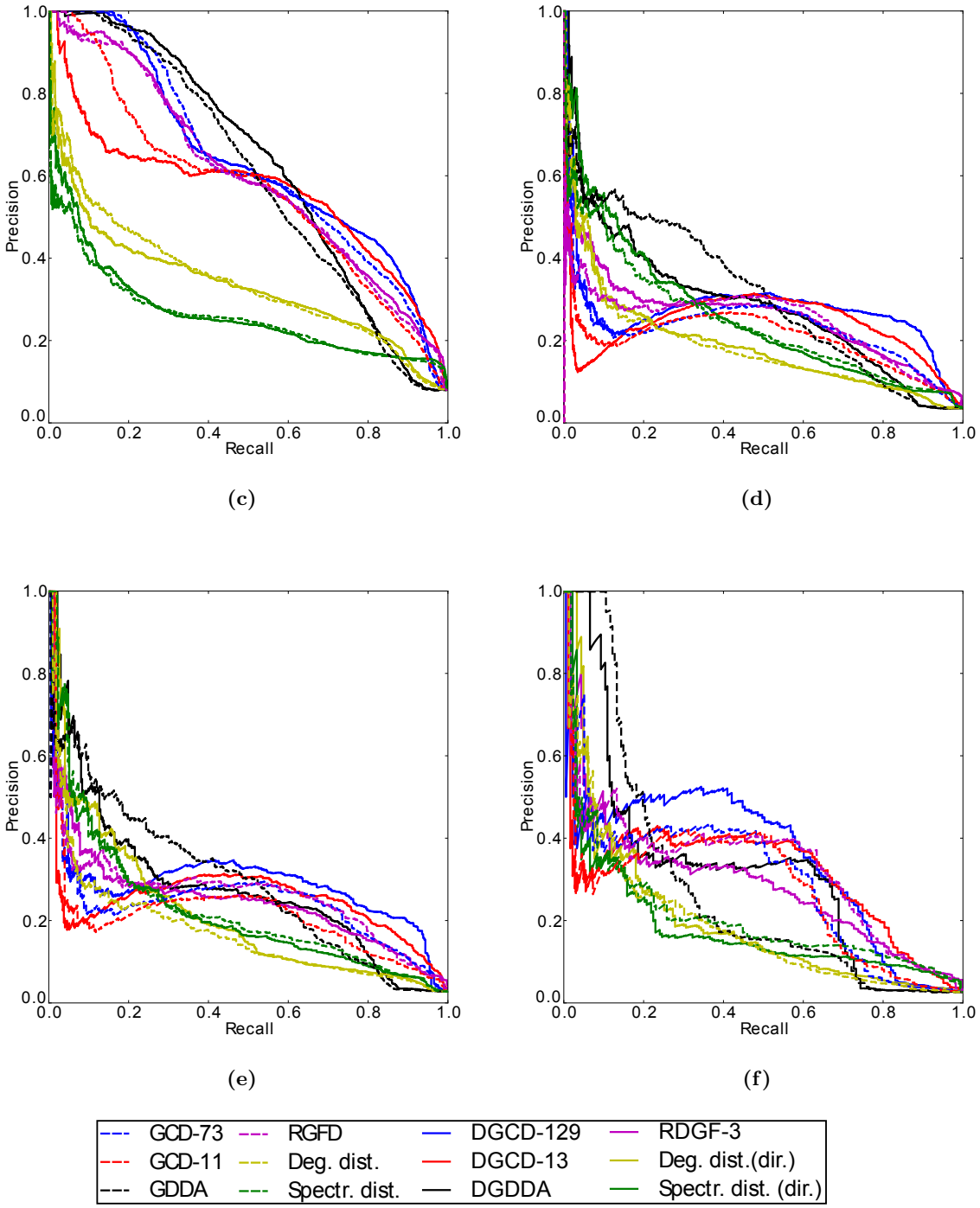


Figure 4.6. Comparison of precision-recall curves for clustering of undirected and directed metabolic networks. We evaluated clustering according to: (c) class, (d) order, (e) family and (f) genus.

Sim. measure	Kingdom	Phylum	Class	Order	Family	Genus
GCD-73	0.8647	**0.8717**	0.6342	0.2301	0.2384	0.3005
GCD-11	0.8732	0.8104	0.5678	0.2023	0.2104	0.2741
GDDA	0.7256	0.6983	0.6013	**0.3099**	**0.29180**	0.2803
RGFD	0.8377	0.7904	0.5998	0.2455	0.2629	0.3280
Deg. dist.	0.5636	0.5216	0.3466	0.1837	0.1714	0.1876
Spect. dist.	0.7823	0.5078	0.262	0.2426	0.2278	0.2077
DGCD-129	0.8521	0.8240	**0.6388**	0.2747	0.2891	**0.3490**
DGCD-13	**0.8787**	0.7870	0.5596	0.2414	0.2531	0.3125
DGDDA	0.7345	0.6936	0.6291	0.2889	0.2810	0.3192
RDGF-3	0.8137	0.7739	0.6028	0.2553	0.2499	0.2841
Deg. dist. (dir.)	0.5583	0.4830	0.3357	0.1879	0.2162	0.1974
Spect. dist. (dir.)	0.8016	0.5167	0.2674	0.2589	0.2197	0.1834

Table 4.4. Comparison of AUPR scores for clustering undirected and directed metabolic networks according to taxonomic classification. Top part of the table: Undirected network measures. Bottom part of the table: Directed network measures. First column: Similarity measure. Second to seventh column: AUPR for clustering based on the kingdom, phylum, class, order, family and genus respectively. The best scores are printed in bold.

These results indicate that the topology of undirected metabolic networks can be used for taxonomic classification of species, with best AUC scores between 0.88 and 0.945 for all taxonomic classification levels. The scores in the top parts of the Tables 4.3 and 4.4 show that undirected graphlet-based measures outperform other tested measures for undirected network comparison.

Tables 4.3 and 4.4 also show that by taking into account directionality of the edges and applying our new directed graphlet-based methods, the clustering performance is better only for several taxonomic classification levels: AUPR scores are higher for clustering performed according to kingdom, class and genus; AUC scores are higher for clustering performed according to class, order and family. For other cases, the scores are slightly better when using undirected metabolic networks and the corresponding undirected measures. This is also visible from the comparison of ROC and Precision-Recall curves shown in Figures 4.5 and 4.6. These results indicate that, at the global level, the topology of metabolic networks is not mainly characterised by the directionality of the edges. This can be explained by the high correlations between in-degree and out-degree based graphlet orbits in the DGCMs of the metabolic networks. This means that, for example, the nodes with high in-degrees in the directed metabolic networks will also have high out-degrees and these nodes correspond to high-degree nodes in the undirected metabolic networks.

In the next section, we use the metabolic network of *H.sapiens* to study the topology–function relationships of enzymes.

4.2 Similar Wirings around Enzymes in Metabolic Network of *H. Sapiens* Correspond to Similar Biological Functions

As previously discussed in Section 1.1, undirected biological networks, such as PPI networks were used to show that similarly wired proteins carry out similar functions. This was exploited to transfer functional annotations and roles between proteins [3,4,56]; graphlets have been particularly useful for this purpose in the domain of disease research, as discussed in Sections 2.1.1 and 2.2.1.

Having defined directed graphlets, we now turn to directed biological networks: in this section, we explore whether similar wirings around enzymes in directed metabolic networks correspond to similar biological functions. In particular, in *H. Sapiens* metabolic network, we analyse the GO enrichment of enzyme clusters that were obtained based on the DGDV similarity between the enzymes. If similar wirings around enzymes do correspond to similar biological functions in the *H. Sapiens* metabolic network, we are then motivated to further look for conserved topology–function relationships across different species (presented in Section 4.3).

4.2.1 Methods

4.2.1.1 Data Sets

Metabolic networks. We utilise the metabolic network of *H. sapiens*, constructed as described in section 4.1.1.1. The metabolic network of *H. sapiens* contains 1,455 enzymes as nodes and 19,194 interactions between them.

GO term annotations. To functionally annotate enzyme-coding genes, we use Gene Ontology (GO). A GO term represents the biological process (BP), molecular function (MF) or cellular component (CC) that is characteristic for a gene. A single gene can be annotated with more than one GO term, while GO term dependencies are described as a GO hierarchy. We downloaded the information on gene-to-GO-term mapping from NCBI[1] in March 2015. We only used experimentally confirmed GO annotations, *i.e.*,

[1] ftp://ftp.ncbi.nlm.nih.gov/gene/DATA/gene2go.gz

those with experimental GO evidence codes. We downloaded the GO hierarchy[2] in March 2015 and cut GO tree at level 5 to standardise GO annotation [196–198]. 781 enzyme-coding genes from the *H. sapiens* metabolic network have at least one GO annotation.

4.2.1.2 Clustering method and cluster enrichment

We cluster the nodes/enzymes of the *H. sapiens* metabolic network using Chavl [199], a publicly available tool for clustering which proposes a hierarchical ascendant classification of nodes, given the similarity scores (we use DGDV similarity described in Section 3.1.2). We use Chavl to cluster the nodes, because, unlike other common clustering methods, it proposes optimal cuts of the classification tree based on likelihood linkage analysis [199].

For the proposed cuts we analyse the GO term enrichment of the resulting clusters to check if the enzymes with similar wiring patterns in the network are annotated with similar GO terms. We use a standard model of sampling without replacement, as used in Kuchaiev *et al.* [194], to calculate the p-value for the enrichment of each cluster with each GO term. The p-value corresponds to the probability of obtaining the same or higher enrichment purely by chance. To calculate the p-value of the enrichment for a particular GO term and cluster, we use the cumulative hypergeometric function:

$$p = 1 - \sum_{i=0}^{X-1} \frac{\binom{K}{i}\binom{M-K}{N-i}}{\binom{M}{N}}, \tag{4.1}$$

where: (1) N, corresponds to cluster size (only annotated genes from the cluster are taken into account) (2) X, corresponds to the number of genes in the cluster annotated with a specific GO term in question, (3) t M, corresponds to the number of all genes in the network that are annotated with any GO term (at the level 5 of the GO tree), (4) K, corresponds to the number of genes in the network that are annotated with the GO term in question. We take into account all GO terms with p-value $p \leq 0.01$.

4.2.2 Results

Here, we analyse the GO term enrichment of the obtained enzyme clusters in the *H. sapiens* metabolic network. The optimal cuts of the classification tree, as proposed by the Chavl algorithm, are at the levels resulting in the formation of either 4 or 19

[2]http://www.geneontology.org/ontology/obo format 1 2/

clusters. Tables 4.5 and 4.6 list the number of GO terms that each cluster is enriched in, for the experiments that resulted in 4 and 19 clusters, respectively.

Cluster ID	1	2	3	4
Cluster size	190	225	220	146
Number of GO terms	5	14	23	13

Table 4.5. Number of enriched GO terms in clusters in *H. sapiens* metabolic network; number of clusters: 4. First row: Number of genes annotated with GO terms in each of the four clusters. Second row: Number of GO terms that each of the four clusters is enriched in.

Cluster ID	1	2	3	4	5	6	7	8	9	
Cluster size	25	19	18	35	38	30	50	78	24	
No. of GO terms	11	16	11	9	34	23	16	28	5	
Cluster ID	10	11	12	13	14	15	16	17	18	19
Cluster size	57	54	57	57	23	52	22	27	49	66
No. of GO terms	15	18	9	27	13	9	5	11	7	20

Table 4.6. Number of enriched GO terms in clusters in *H. sapiens* metabolic network; number of clusters: 19. First row: Number of genes annotated with GO terms in each of the 19 clusters. Second row: Number of GO terms that each of the 19 clusters is enriched in.

Tables 4.5 and 4.6 convey that each cluster of enzymes, constructed based on the topological similarity between the nodes in the metabolic network, is significantly enriched in GO terms. In Appendices, Section C.1, we provide the list of all the GO terms that the clusters are enriched in.

By examining the GO terms, we observe that each cluster is enriched in GO terms which correspond to similar functions. Here we give more details on clusters from Table 4.5. Cluster 1 is enriched in several cellular lipid catabolic processes (long-chain fatty acid-CoA ligase activity, prostanoid metabolic process, membrane lipid catabolic process), cluster 2 is enriched in processes supporting methylation (protein methyltransferase activity, N-methyltransferase activity, peptidyl-lysine methylation, S-adenosylmethionine-dependent methyltransferase activity), cluster 3 is enriched in several acid binding processes (retinoic acid binding, monocarboxylic acid binding) and regulation of defense response (positive regulation of defense response, regulation of innate immune response), and cluster 4 is enriched in several processes related to carbohydrate metabolic processes (poly-N-acetyllactosamine metabolic process, hexose

metabolic process, protein O-linked glycosylation via serine, polysaccharide biosynthetic process).

The enrichment of each cluster with GO terms that are related to similar biological processes indicates that specific wiring around genes in *H. sapiens* metabolic network corresponds to similar biological functions and motivates us to explore topology–function relationships across the metabolic networks of different species (see Section 4.3).

4.3 Topology–Function Relationships in Metabolic Networks are Conserved across Different Species

Recently, several methods have focused on finding evolutionary conserved topologies across different species and on relating them to functional annotations. For example, alignment algorithms applied to the PPI networks of various species have been used to find evolutionary conserved parts of the networks [200] and aligned parts of the networks were then used to transfer functional annotations across species. Then, a novel framework [201] based on Canonical Correlation Analysis (CCA) [202], that uses GDVs to describe local topology around proteins in PPI networks, successfully characterised statistically significant topology–function relationships in human (*H. sapiens*) and yeast (*S. cerevisiae*). This method uncovered the functions termed *topologically orthologous functions* which have conserved topology in PPI networks across the two species [201]. In particular, 15 biological processes and 9 cellular components are found to be topologically orthologous between *H. sapiens* and *S. cerevisiae*. To increase the coverage, Davis *et al.* used the full GO hierarchy for obtaining these results. Recall that there are redundancies among GO terms in the full GO hierarchy: GO terms have parent-child relationships, meaning that a more specific GO term is a part (a child) of a more generic GO term. After taking this into consideration, Davis *et al.* identified 7 non-redundant topologically orthologous biological processes and 2 non-redundant topologically orthologous cellular components [201].

In section 4.2, we used our new DGDV similarity measure and showed that genes with similar wiring patterns in a directed metabolic network share similar GO term annotations. Hence, we are motivated to explore whether topology–function relationships are conserved in directed metabolic networks of *H. sapiens* (human) and four well-annotated model organism networks: *M. musculus* (mouse), *D. melanogaster* (fruitfly), *S. cerevisiae* (yeast) and *A. thaliana* (a flowering plant). First, we use DGDV and CCA to identify which local topologies are significantly correlated to particular GO terms. We use the results to predict new GO term annotations for each of the analysed species.

We then follow the framework of Davis *et al.* [201] to identify topologically orthologous functions across the metabolic networks of the five species.

4.3.1 Methods

4.3.1.1 Data Sets

Metabolic networks. We use directed metabolic networks of the following five well-annotated species: *H. sapiens*, *M. musculus*, *D. melanogaster*, *S. cerevisiae* and *A. thaliana* obtained as described in Section 4.1.1.1.

Gene Ontology Annotations. We downloaded GO annotation data from NCBI[3] in March 2015. We use only experimental GO evidence codes and the full GO hierarchy[4]. Table 4.7 lists the five species, the sizes of their metabolic networks and the number of annotated genes per species.

Species	Common name	Abbr.	Number of genes	Number of annot. genes	Number of edges
H. sapiens	Human	hsa	1,455	1,143	19,194
M. musculus	Mouse	mmu	1,513	1,069	23,801
D. melanogaster	Fruitfly	dme	993	503	9,012
S. cerevisiae	Yeast	sce	758	745	3,226
A. thaliana	Flowering plant	ath	2,119	1,626	28,261

Table 4.7. Metabolic networks of *H. sapiens* and four model species. First column: The name of the species. Second column: The species abbreviation. Third column: The number of genes in directed metabolic network. Fourth column: The number of genes from the network that are annotated with GO terms. Fifth column: The number of edges in directed metabolic network.

4.3.1.2 Finding topology–function relationships using CCA

We quantify relationships between topological patterns around enzymes and their GO annotations using a multivariate analysis of variance, *i.e.* Canonical Correlation Analysis (CCA) [202], following the approach of Davis *et al.* [201]. CCA uncovers linear relationships between two sets of variables, in our case topological descriptors and functional annotations, as follows.

[3] ftp://ftp.ncbi.nlm.nih.gov/gene/DATA/gene2go.gz
[4] http://www.geneontology.org/ontology/obo format 1 2/

For each species, we construct the variable set R^t which captures the topology through the directed graphlet degree vectors (DGDV) of the enzymes (genes) in the metabolic network. We only consider GO annotated genes. The second variable set R^f represents the functional information about genes, *i.e.* for each gene in the network, we encode its GO annotations as binary variables: 1 if the gene is annotated with the GO term, and 0 otherwise. We only include the GO terms that have at least 5 annotated genes because reliable patterns are unlikely to be found with fewer than 5 example cases and we also eliminate the GO terms that annotate more than 5% of the genes in the metabolic network in question [201]. Given n pairs of variable vectors from $R^t \times R^f$ for n genes as an input, the CCA outputs weight vectors so that the Pearson's correlation between the weighted sums of R^t and R^f (i.e. between canonical variates) is maximised.

After finding the first set of weights, CCA iterates $min\{t, f\}$ times to find more weight vectors, such that the resulting canonical variates are not correlated with any of the previous canonical variates. The weight matrices W_1 and W_2, for R^t and R^f respectively, are constructed by combining all of the identified weight vectors. *The association matrix* which represents pairwise relations between topology and function is then constructed as $W_1 \times S \times W_+^2$, where S is a diagonal matrix of canonical correlations (i.e., Pearson's Correlations among canonical variates) that weights the variates according to their correlation strength (W_+^2 is the Moore-Penrose pseudoinverse [203] of W_2, as detailed in [201]).

4.3.1.3 Predicting GO term annotations

We use the above-described CCA relationships between topological patterns around enzyme-coding genes and their GO annotations to propose new GO annotations. For each of the analysed species (*H. sapiens* and four model organisms) we compute the *association matrix*, which combines all topology–function relationships as described in Section 4.3.1.2. We then use this matrix to transform graphlet degree vectors to vectors of real-value topology-based annotations by multiplying the matrix of DGDVs and the association matrix. This results in a *matrix of predicted GO term annotations*.

The values in the predicted GO term matrix represent the association scores between genes and GO terms (higher scores indicate higher probability that the gene is annotated with corresponding GO term in the matrix). We create the predicted GO term matrix for each of the five species and use the values in the matrices to predict new GO annotations as follows. We use the original GO term annotation matrix (1 and 0 denoting whether the gene is annotated to a GO term or not, respectively) as the gold

standard to calculate precision, recall and F_1 score for the association scores. The F_1 score is a measure of a test's accuracy and is calculated as:

$$F_1 = 2 \cdot \frac{precision \cdot recall}{precision + recall}. \tag{4.2}$$

The score reaches its best value at 1 and worst at 0. We then use the value of the association score that corresponds to maximum F_1 value as a *predictive threshold value*, *i.e.*, if the association score between a gene and GO term in the predicted GO term matrix is equal to or higher than the threshold, we predict this gene to be annotated with the given GO term.

Recall that only the experimentally confirmed GO annotations were used as the functional information about genes and were input for the CCA. Hence, we validate our predictions by comparing them to GO annotation data from NCBI[5] with non-experimental evidence codes (independent of the data used for obtaining the predictions). The non-experimental GO annotations have the following evidence codes in NCBI (*i.e* are inferred from):

- Sequence or structural Similarity (ISS),
- Sequence Orthology (ISO),
- Sequence Alignment (ISA),
- Sequence Model (ISM),
- Genomic Context (IGC),
- Biological Aspect of Ancestor (IBA),
- Biological Aspect of Descendant (IBD),
- Key Residues (IKR),
- Rapid Divergence(IRD),
- Reviewed Computational Analysis (RCA).

[5] ftp://ftp.ncbi.nlm.nih.gov/gene/DATA/gene2go.gz

4.3.1.4 Identifying conserved topology–function relationships

Following the approach of Davis *et.al.* [201], we look for topologically orthologous functions between all pairs of species listed in Table 4.7. For each pair of species, we perform the CCA analysis as described in section 4.3.1.2. In the input data, for a given species pair, we only include the GO terms common to both (the number of common GO terms per species pair is given in Table 4.8). Note that, unlike in the original framework [201], we did not filter the nodes according to their degrees. This was for two reasons. First, there exist two types of degrees in our networks: in-degree and out-degree. Considering the relatively small sizes of metabolic networks and both types of degree, excluding the nodes with degrees lower than 4, would significantly reduce the number of nodes to analyse. Another reason to keep all the nodes, even if the node degree is small, is because they can contribute to a wide range of graphlets.

BP	ATH	HSA	SCE	DME	MMU
ATH		331	264	232	340
HSA			288	289	615
SCE				155	281
DME					359
MF	ATH	HSA	SCE	DME	MMU
ATH		125	84	54	109
HSA			80	56	134
SCE				39	86
DME					64
CC	ATH	HSA	SCE	DME	MMU
ATH		42	29	18	37
HSA			39	21	55
SCE				7	26
DME					18

Table 4.8. Number of Common GO terms per analysed species pair. We separately report biological processes (BP), molecular functions (MF) and cellular components (CC).

The methodology of Davis *et al.* [201] introduces the following measures:

- *The structure association strength*, which identifies the GO terms that are strongly linked with a specific topological pattern by quantifying the linear dependence between the topology-based GO annotations and the original GO annotations using the Pearson's correlation.
- *Orbit contribution strength*, which identifies the most important orbits for the topological pattern of a GO term by quantifying the linear dependencies between

graphlet degrees of each orbit and topology-based GO annotations using Pearson's correlations.

Species	GO Type	Number of GO terms in species 1	Number of GO terms in species 2
ATH-DME	BP	142	103
ATH-MMU	BP	184	194
ATH-SCE	BP	127	167
DME-MMU	BP	100	172
HSA-ATH	BP	193	186
HSA-DME	BP	156	99
HSA-MMU	BP	270	277
HSA-SCE	BP	151	166
SCE-DME	BP	75	65
SCE-MMU	BP	159	150
ATH-DME	MF	49	32
ATH-MMU	MF	81	71
ATH-SCE	MF	65	64
DME-MMU	MF	36	52
HSA-ATH	MF	89	91
HSA-DME	MF	52	34
HSA-MMU	MF	101	87
HSA-SCE	MF	59	57
SCE-DME	MF	31	23
SCE-MMU	MF	65	64
ATH-DME	CC	16	6
ATH-MMU	CC	24	8
ATH-SCE	CC	16	11
DME-MMU	CC	10	10
HSA-ATH	CC	27	29
HSA-DME	CC	15	10
HSA-MMU	CC	19	8
HSA-SCE	CC	18	10
SCE-DME	CC	4	3
SCE-MMU	CC	7	3

Table 4.9. Number of GO terms with statistically significant topology–function relationships (statistically significant structure association strengths). First column: Species pair. Second column: GO term type. Third column: The number of GO terms that have significant topology–function relationship for the first species from the experiment, i.e. for a given GO term there is a significant p–value (≤ 0.05) of structure association strength. Fourth column: The number of GO terms that have significant topology–function relationship for the second species from the experiment, i.e. for a given GO term there is a significant p–value (≤ 0.05) of structure association strength.

In order to identify orthologous topological patterns in a pair of species, firstly the structure association strengths and orbit contribution strengths are computed separately for each of the two species. The GO terms with statistically significant topology–function relationships per species are all GO terms with statistically significant structure association strengths (p-value$\leq$ 0.05 after the Benjamini Hochberg correction). The number of such GO terms is given in Table 4.9. We evaluate the statistical significance of a structure association strength value as follows. We shuffle the DGDVs between the genes and repeat the experiment to calculate a randomly obtained structure association strength. We repeat the experiment 10,000 times, each time shuffling the DGDVs between the genes, in order to calculate the p-value that corresponds to the probability of obtaining the same or higher value of structure association strength by chance.

Then, for a species pair, two values are calculated:

- The *multi-species structure association strength* is computed by taking the minimum of the two structure association strengths from both species. This value shows that there is a strong topology function relationship for a given GO term in both species.
- The *orbit contribution similarity* for a GO term in two species is quantified using the Spearman's Correlation of the per-species orbit contribution strengths of the species. This value shows that, for a given GO term, the topological profiles described for the two organisms correlate and, therefore, the same set of orbits is statistically significantly associated with the topology (*i.e.* the topology is conserved between the species).

GO term is topologically orthologous for a species pair if both multi-species structure association strength and orbit contribution similarity are statistically significant (p-value$\leq$ 0.05 after the Benjamini Hochberg correction).

4.3.2 Results

4.3.2.1 GO term predictions

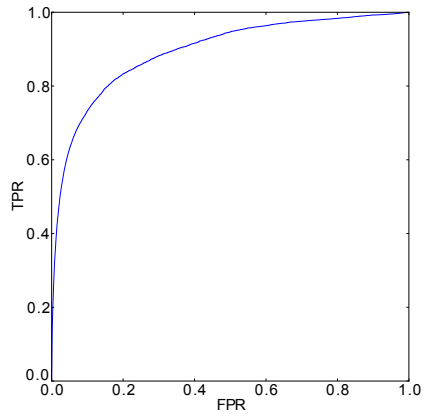
Here we present the predicted GO annotations obtained based on the topology of metabolic networks. Using CCA, as described in Section 4.3.1.3, we calculated the matrix of predicted GO annotations, and using the maximum F_1 score, we identified the threshold value for predicting GO annotations. Figure 4.7 shows the ROC curves for our GO annotations predictions, taking the original GO term annotation matrix (from Section 4.3.1.3) as a golden standard. For all five species the AUC scores are higher than 0.87, suggesting that this method can be used for predicting GO annotations.

Species	Analysed genes	Genes with predicted GO annotations	Genes possible to validate	Supported predictions
H. sapiens	1,143	817	295	73
D. melanog.	503	299	246	33
S. cerevisiae	745	368	130	2
M. musculus	1,069	751	678	182
A. thaliana	1,626	1,154	1,017	148

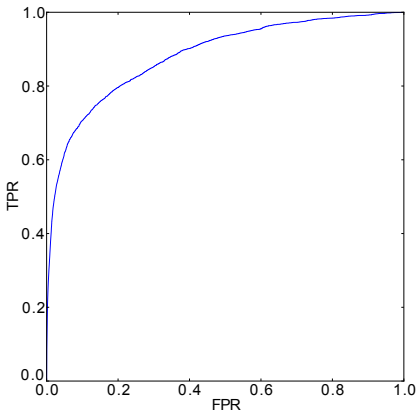
Table 4.10. Number of genes with predicted GO terms. First column: Species in the experiment. Second column: The number of analysed genes (annotated genes in the metabolic network, based on experimental methods). Third column: The number of genes with new predicted GO annotations (BP), *i.e.* genes for which our method provided GO annotations that are new to the experimentally confirmed GO annotations. Fourth column: The portion of genes from the fourth column that also have GO annotations from the sources other than experimental. Fifth column: The number of genes for which some of our predicted GO annotations are supported by other prediction methods.

Molecular function and cellular component are the properties of a biomolecule, while, in order to perform a biological process, molecules interact, giving a specific topology to biological networks. Hence, it is more interesting to explore biological processes in the context of the wiring patterns of biological networks; therefore, we only consider the predicted biological processes (BP) GO annotations. Table 4.10, column 3, shows the number of genes for which our method proposes novel annotations with BP, *i.e.*, the annotations which are not in the set of the experimentally validated. Our method predicts new annotations for more than half of the genes that are already experimentally annotated.

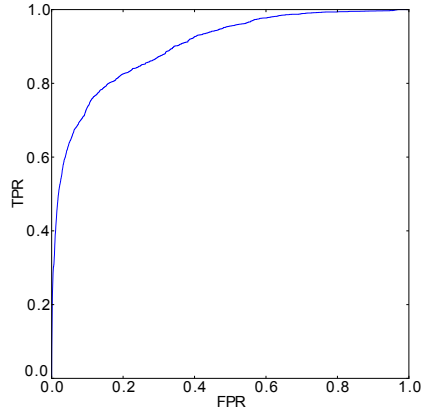
To assess the relevance of our predicted annotations we compare them to the GO annotations that are supported by non-experimental methods (annotation data from NCBI with evidence codes different from experimental). In column 5 of Table 4.10, we report the number of genes with predictions for which we find support. We take into account that there is only a portion of genes for which non-experimentally obtained GO annotations are available in the NCBI (column 4 of Table 4.10). For example, in *H. sapiens*, of 817 genes with new predictions, only 295 have additional non-experimental annotations. Out of those 295 genes, we find 73 genes for which our predicted GO annotations are supported with annotations obtained from non-experimental methods, which corresponds to 24.7%. Similarly, for *D. melanogaster* we have supported the predictions for 13.4% of genes, for *S. cereviseae* 1.5% of genes, for *M. musculus* 26.8% of genes, and for *A. thaliana* we have supported the predictions for 14.6% of genes.



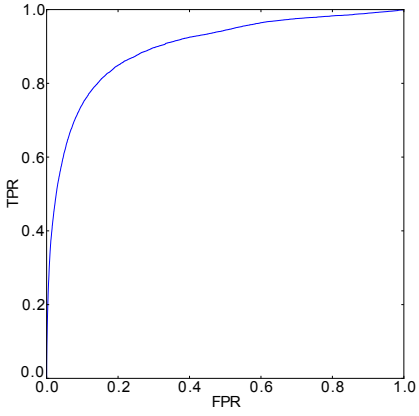
(a) H. sapiens, AUC=0.89



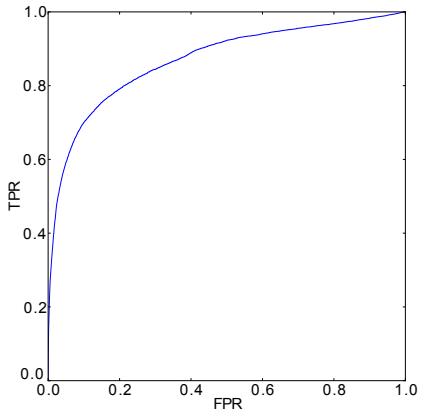
(b) D. melanogaster, AUC=0.88



(c) S. cerevisiae, AUC=0.90

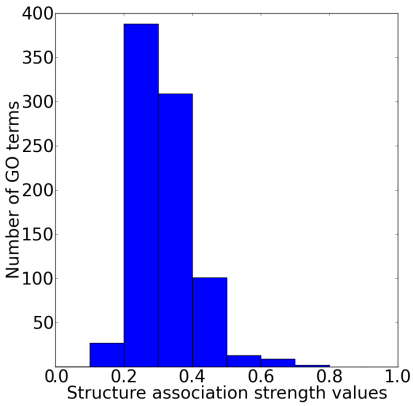


(d) M. musculus, AUC=0.90

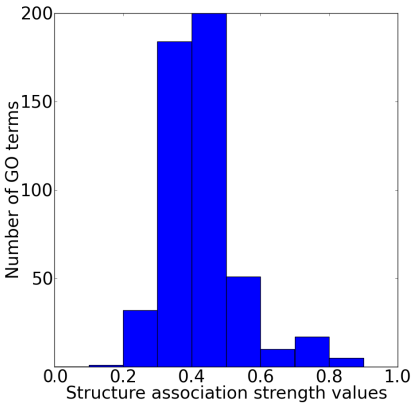


(e) A. thaliana, AUC=0.87

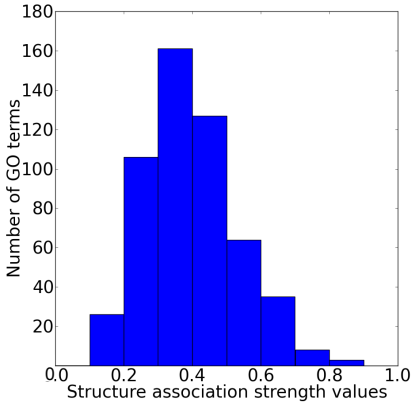
Figure 4.7. ROC curves for predicting GO annotations.



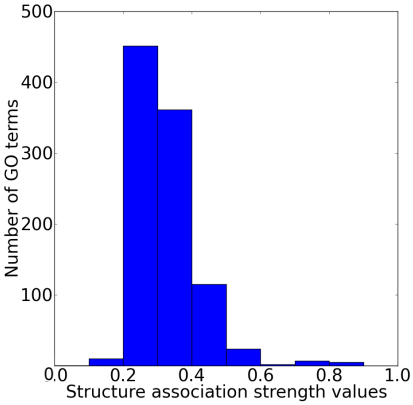
(a) H.sapiens



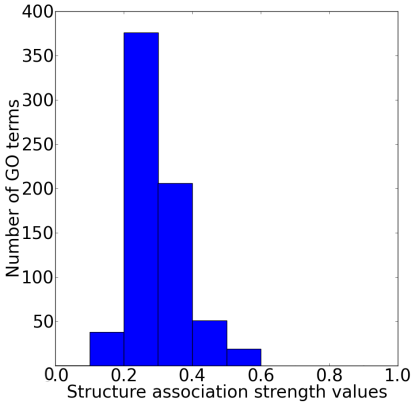
(b) D. melanogaster



(c) S. cerevisiae



(d) M. musculus



(e) A. thaliana

Figure 4.8. Number of GO terms per structure association strength value. X-axis: structure association strength values in bins of 0.1. Y-axis: number of GO terms with structure association strengths corresponding to the bins.

Note that these results reflect the predictive power of the network topology for annotating genes (enzymes) with all experimentally validated GO terms. Successful predictions can be expected only in the case of GO terms with high values of structure association strengths (defined in Section 4.3.1.4); the reason being that such GO terms are strongly linked with a specific topological pattern. Figure 4.8 shows that, in the case of all five species, most of the GO terms have structure association strength values lower than 0.5.

4.3.2.2 Topologically orthologous GO terms

Here, we present the conserved topology–function relationships across the networks of five eukaryotic species, obtained using the methodology described in Section 4.3.1.4. Table 4.11 lists how many GO terms are topologically orthologous for each species pair among *H. sapiens*, *M. musculus*, *D. melanogaster*, *S. cerevisiae* and *A. thaliana*. We compare the number of topologically orthologous GO terms with the number of GO terms that were analysed per each species pair (Table 4.8), and observe that the method identifies a larger set of topological orthologs in the case of species pairs with a larger set of common GO terms.

BP	ATH	HSA	SCE	DME	MMU
ATH		1	16	0	13
HSA			2	29	92
SCE				6	12
DME					32
MF	ATH	HSA	SCE	DME	MMU
ATH		4	19	1	13
HSA			9	24	57
SCE				0	32
DME					11
CC	ATH	HSA	SCE	DME	MMU
ATH		9	5	3	7
HSA			11	6	10
SCE				3	3
DME					1

Table 4.11. The number of topologically orthologous GO terms per species pair. Number of GO terms for the pairs of species that have statistically significant Multi-species structure association strength and Orbit contribution similarities. Results are reported separately for biological processes, molecular functions and cellular components.

Topologically orthologous GO term	Species pairs
DNA metabolic process	ath-mmu, ath-sce, dme-mmu, hsa-ath, hsa-mmu, hsa-sce, sce-dme, sce-mmu
Phosphatidylinositol metabolic process	dme-mmu, hsa-dme, hsa-mmu
Protein alkylation	dme-mmu, hsa-dme, hsa-mmu
Hydrogen transport	ath-mmu, ath-sce, sce-mmu
Histone lysine methylation	dme-mmu, hsa-dme, hsa-mmu
Cyclic nucleotide metabolic process	dme-mmu, hsa-dme, hsa-mmu
Ribose phosphate biosynthetic process	dme-mmu, hsa-dme, hsa-mmu
Macromolecule methylation	dme-mmu, hsa-dme, hsa-mmu
Inorganic ion transmembrane transport	ath-mmu, ath-sce, sce-mmu
Alpha-amino acid metabolic process	dme-mmu, hsa-dme, hsa-mmu
Proton transport	ath-mmu, ath-sce, sce-mmu
Methylation	dme-mmu, hsa-dme, hsa-mmu
Purine nucleotide biosynthetic process	dme-mmu, hsa-dme, hsa-mmu
Ribose phosphate metabolic process	dme-mmu, hsa-dme, hsa-mmu
Protein methylation	dme-mmu, hsa-dme, hsa-mmu
Histone methylation	dme-mmu, hsa-dme, hsa-mmu
Purine ribonucleotide biosynthetic process	dme-mmu, hsa-dme, hsa-mmu
Hydrogen ion transmembrane transport	ath-mmu, ath-sce, sce-mmu
Purine ribonucleotide metabolic process	dme-mmu, hsa-dme, hsa-mmu
Monovalent inorganic cation transport	ath-mmu, ath-sce, sce-mmu
Peptidyl-lysine methylation	dme-mmu, hsa-dme, hsa-mmu
Ribonucleotide metabolic process	dme-mmu, hsa-dme, hsa-mmu
Purine nucleotide metabolic process	dme-mmu, hsa-dme, hsa-mmu
Lipid phosphorylation	dme-mmu, hsa-mmu, sce-dme
Ribonucleotide biosynthetic process	dme-mmu, hsa-dme, hsa-mmu
cGMP metabolic process	dme-mmu, hsa-dme, hsa-mmu
Inorganic cation transmembrane transport	ath-mmu, ath-sce, sce-mmu
Cation transmembrane transport	ath-mmu, ath-sce, sce-mmu
Phospholipid metabolic process	ath-mmu, hsa-mmu
Covalent chromatin modification	hsa-dme, hsa-mmu
Chromatin organization	hsa-dme, hsa-mmu
Chromatin modification	hsa-dme, hsa-mmu
Generator of precursor metabolites and energy	hsa-dme, hsa-mmu
Terpenoid metabolic process	ath-sce, hsa-mmu
Cellular modified amino acid metabolic process	hsa-mmu, hsa-sce
Ion transmembrane transport	ath-sce, sce-mmu
Peptidyl-lysine modification	hsa-dme, hsa-mmu
Transmembrane transport	ath-sce, sce-mmu
Pyridine nucleotide metabolic process	dme-mmu, hsa-dme
Nicotinamide nucleotide metabolic process	dme-mmu, hsa-dme
Histone modification	hsa-dme, hsa-mmu
Lipid modification	hsa-mmu, sce-dme

Table 4.12. Topologically orthologous biological processes across species pairs.

Again, we explore the BP GO annotations in further detail, because the topology of a biological network comes as a result of biomolecules interacting to perform biological functions. For all topologically orthologous biological processes, we check how many species pairs they appear in. We hypothesise that if a given GO term is topologically orthologous for several species pairs, then its underlying topology is more constrained through the evolution process. We list such biological processes and the corresponding species pairs in Table 4.12. Some of the biological processes that are topologically orthologous across several pairs of species are essential biological processes. Examples are DNA metabolic process (identified as a topological ortholog in 8 pairs of species: ath-mmu, ath-sce, dme-mmu, hsa-ath, hsa-mmu, hsa-sce, sce-dme, sce-mmu) and ribonucleotide and ribose phosphate metabolic processes (topological orthologs in 3 pairs of species dme-mmu, hsa-dme, hsa-mmu).

In regards to the topologically orthologous biological processes, we explore which particular orbits contribute the most to their topological profiles, *i.e* for a species pair we examine the *orbit contribution strength profiles* represented using heat-maps. An example is Figure 4.9 (corresponding to orthologous biological processes between *H. sapiens* and *D. melanogaster*). The orbit contribution strength profile of a GO term in a species pair is obtained by averaging the orbit contribution strength vectors of the two species. We examine such heat-maps for all species pairs and, for a particular GO term and all 129 orbits, we look for orbits that are linked to the GO term (see Figure 4.9). Using this approach, we identify the following topological patterns of interest:

- 4-node star graphlets and the two-degree orbits on graphlets $G_{26} - G_{35}$ characterise the DNA metabolic process, a topologically orthologous process across 8 species pairs listed in Table 4.12. The DNA metabolic process is one of the essential processes for all living organisms and with this finding we have confirmed that a specific local wiring in the metabolic network plays important role in the DNA metabolic process and as such is conserved across different species.

- Four-node cliques (orbits 117-129) characterise the following biological processes: (1) the phospholipid metabolic process—a topologically orthologous GO term between *H. sapiens* and *M. musculus* and between *A. thaliana* and *M. musculus*, (2) lipid modification – a topologically orthologous GO term between *H. sapiens* and *M. musculus*, and (3) lipid phosphorylation—a topologically orthologous GO term between *D. melanogaster* and *M. musculus* and between *H. sapiens* and *M. musculus*.

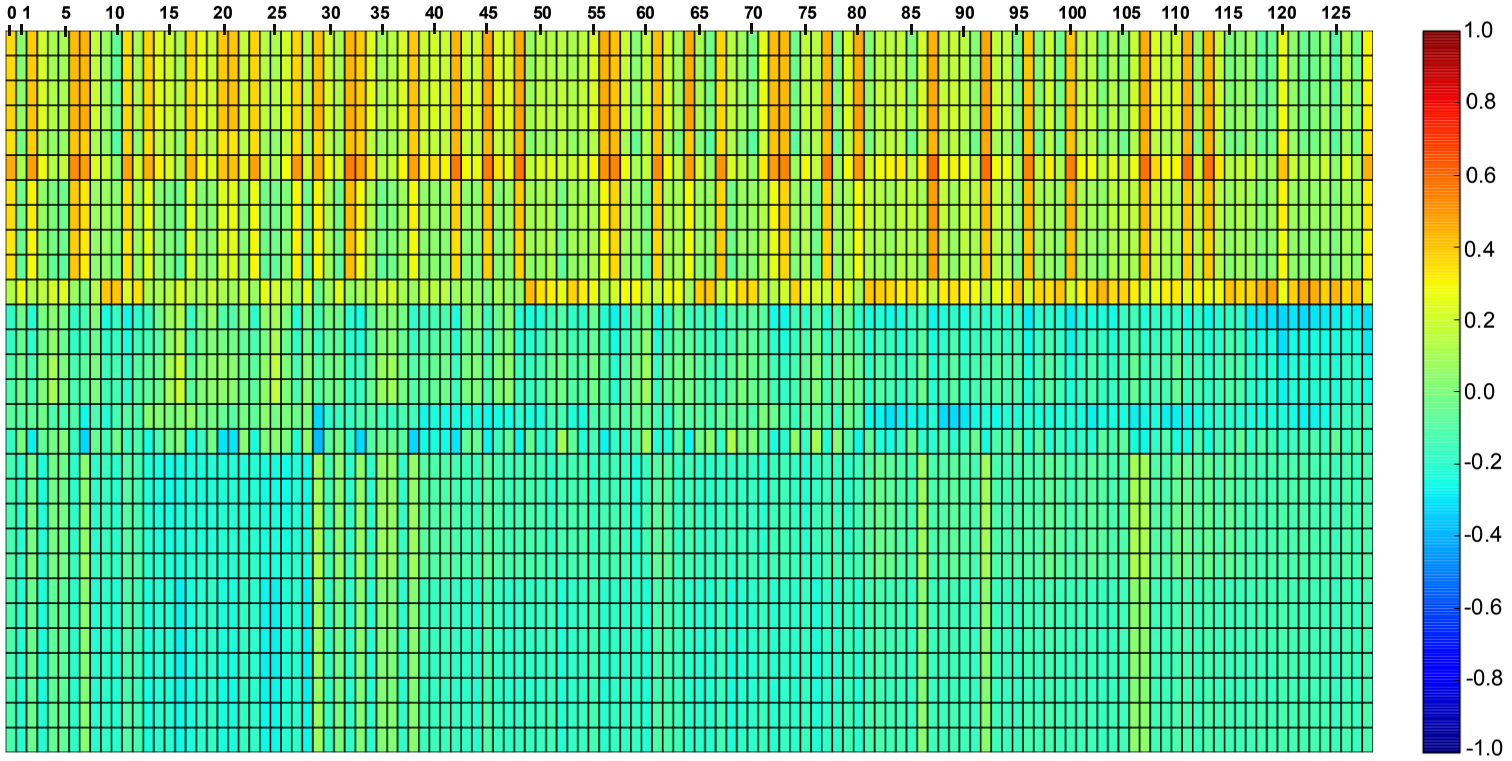


Figure 4.9. Orbit contribution strength profiles of the topologically orthologous biological processes between *H. sapiens* and *D. melanogaster*. Each row corresponds to the average orbit contribution strength profile of the GO term. Each cell represents the average of orbit contribution strengths in both species for given GO term and orbit. The first 10 rows corresponds to: Purine nucleotide metabolic process, Purine ribonucleotide metabolic process, Ribose phosphate metabolic process, Ribonucleotide metabolic process, cGMP metabolic process, Cyclic nucleotide metabolic process, Ribose phosphate biosynthetic process, Ribonucleotide biosynthetic process, Purine nucleotide biosynthetic process, Purine ribonucleotide biosynthetic process, respectively. We only list the first ten BPs, because for them there are orbits with distinctive positive values of average orbit contribution strengths.

- Red nodes in Figure 4.10 show topological patterns that characterise 10 biological processes that are topologically orthologous between *D. melanogaster* and *M. musculus*, *H. sapiens* and *D. melanogaster* and *H. sapiens* and *M. musculus*. The 10 biological processes are:
  - Ribonucleotide metabolic process
  - Cyclic nucleotide metabolic process
  - Ribose phosphate metabolic process
  - Purine nucleotide metabolic process
  - Purine ribonucleotide metabolic process
  - Ribonucleotide biosynthetic process
  - Purine ribonucleotide biosynthetic process
  - Ribose phosphate biosynthetic process
  - Purine nucleotide biosynthetic process
  - cGMP metabolic process

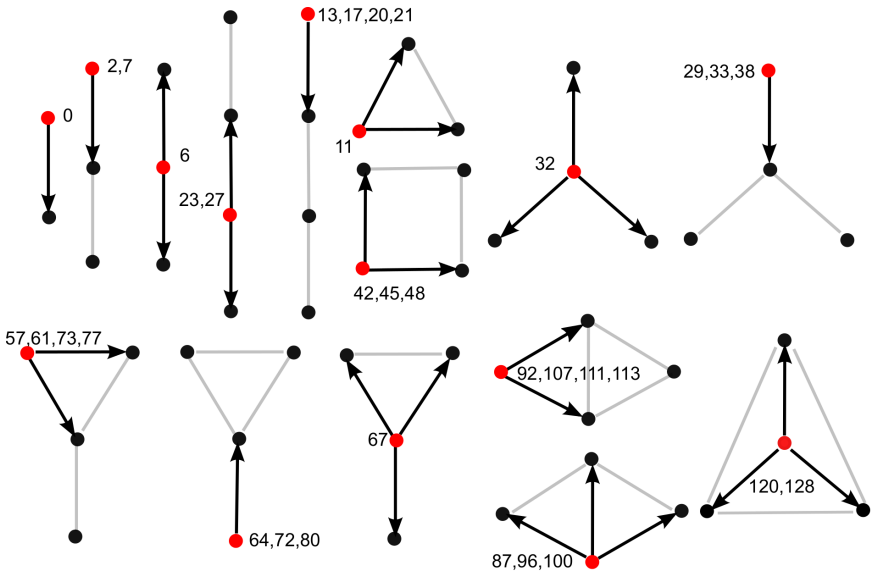


Figure 4.10. Illustration of topological patterns linked to some of the topologically orthologous GO terms. Red nodes correspond to topological patterns that characterise 10 biological processes that are topologically orthologous between *D. melanogaster* and *M. musculus*, *H. sapiens* and *D. melanogaster* and *H. sapiens* and *M. musculus*. Edges coloured in grey can have any direction.

The red nodes in Figure 4.10 correspond to orbits with outgoing edges. In the next section, we investigate the biological mechanisms behind these topological patterns. Specifically, our goal is to find biological confirmation that such patterns, involving outgoing edges, characterise the 10 topologically orthologous biological processes listed above. We find that these 10 biological processes can be grouped according to their relationships in the GO hierarchy. In particular, as shown in Figure 4.11, some of the topologically orthologous BP that we identify are part of the remaining, more generic, topologically orthologous BPs. Hence, we decide to analyse the parent processes: purine nucleotide metabolic process, ribose phosphate metabolic process and cyclic nucleotide metabolic process as they capture the functions of the more specific (children) biological processes.

4.3.2.3 Biological confirmation of the topologically orthologous biological processes

We explore why the purine nucleotide metabolic process, ribose phosphate metabolic process and cyclic nucleotide metabolic process are correlated with topological patterns involving outgoing directed edges (shown in Figure 4.10) in the metabolic networks of human, mouse and fruit fly.

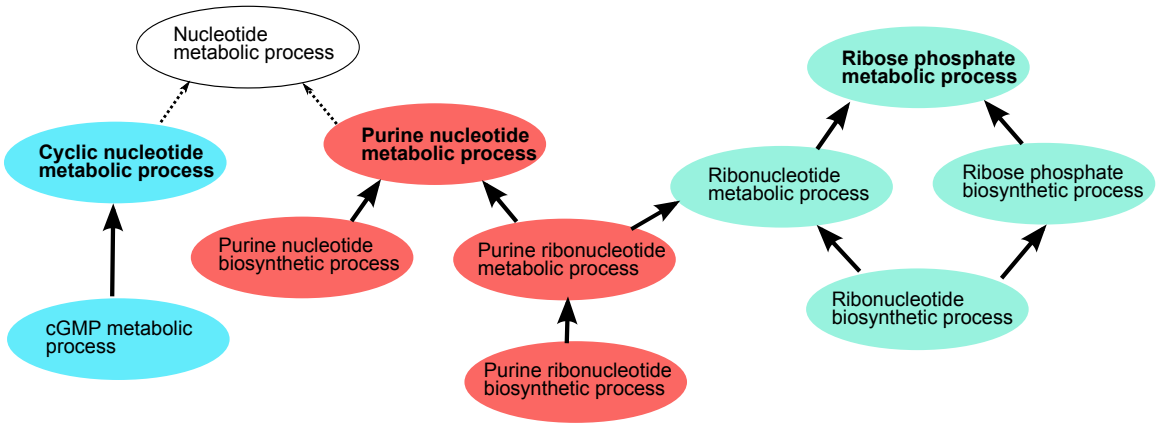


Figure 4.11. Parent-child relationships between topologically orthologous biological processes. An arrow denotes that one biological process is a child of another. For example, a ribonucleotide biosynthetic process is also a ribonucleotide metabolic process. We group the biological processes according to common parents: blue colour denotes cyclic nucleotide metabolic processes, red colour denotes purine nucleotide metabolic processes and green colour denotes ribose phosphate metabolic process. Cyclic nucleotide metabolic process and purine nucleotide metabolic process are part of nucleotide metabolic processes.

We conduct a detailed analysis on the metabolic network of *H. sapiens* because it is the most annotated one among the three. We also provide explanation for the specific wirings in the *M. musculus* and *D. melanogaster* networks.

We perform the analysis as follows. For a topologically orthologous biological process of interest and its characterising orbit O_C, we list the enzymes annotated with that particular biological process that appear in the metabolic network. For each of these enzymes, we look into the metabolic network to find all graphlets G that the enzyme touches at orbit O_C. Then, on each of the identified graphlets G and for each of the remaining orbits O_R on graphlet G (orbits that are not orbit O_C), we find a set of enzymes in the metabolic network that touch the graphlet G at orbit O_R. Finally, for this enzyme set, we calculate the GO term enrichment using the same approach as in Section 4.2.1.2. This way we can observe which biological processes are characteristic for orbits O_R on all graphlets G in the metabolic network, such that the topologically orthologous biological process of interest (i.e. enzyme annotated with it) touches the graphlet G at the characteristic orbit O_C.

In the presented case studies, we explore orbit 6 (graphlet G_2) and orbit 11 (graphlet G_5) in more detail, because these graphlets can be induced on most of the topological patterns shown in Figure 4.10. In addition, orbit 6 and orbit 11 capture the outgoing edges that the topological patterns shown in Figure 4.10 are characterised with.

Case study: Purine nucleotide metabolic process. Enzymes that are annotated with this GO term participate in the metabolism (synthetisation or degradation) of purine nucleotides, which are one of the constituting blocks of DNA and RNA.

For the enzymes annotated with purine nucleotide metabolic process and that touch graphlets G_2 at orbit 6 in *H. sapiens* metabolic network, we find that the set of enzymes touching these graphlets at orbit 5 is statistically significantly enriched, among others, in GO terms shown in Figure 4.12. Note that we list GO terms for which we found the biological meaning behind the topological pattern. A full list of GO terms is given in Appendices in Section C.2, Table C.3. Recall that a directed edge from enzyme A to enzyme B in a metabolic network, denotes that the enzyme A catalyses process α which results in a product that is used in process β catalysed by enzyme B. This also means that in order for the process β to take place, the process α needs to first take place.

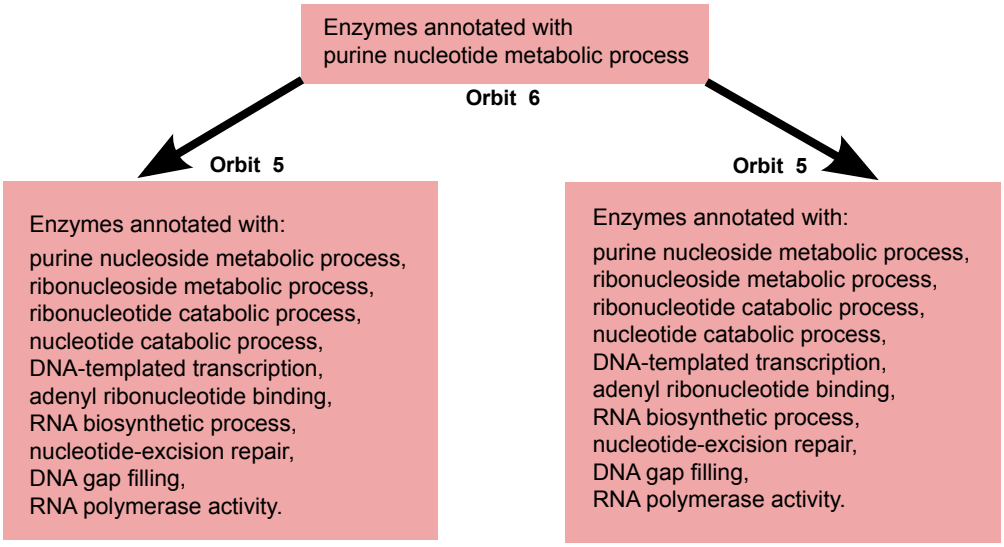


Figure 4.12. GO enrichment around enzymes touching orbit 6, that are annotated with purine nucleotide metabolic process.

Figure 4.12 shows the directed links from enzymes at orbit 6, annotated with the purine nucleotide metabolic process, towards enzymes at orbit 5 that are annotated with the ribonucleoside metabolic process, purine nucleoside metabolic processes and ribonucleotide catabolic process. These directed links are all relevant and correspond to the process of degrading purine nucleotides into purine nucleosides, e.g. when nucleases and nucleotidases degrade nucleic acid chains down to free nucleotides (nucleotide metabolic process) and then to nucleosides (through purine nucleoside metabolic process, ribo/nucleotide catabolic process or ribonucleoside metabolic processes). Similarly, the directed links towards enzymes at orbit 5 that are annotated with DNA-templated transcription, adenyl ribonucleotide binding, RNA biosynthetic process, nucleotide-excision repair, DNA gap filling and RNA polymerase activity are also relevant. Nucleic acids (DNAs and RNAs) are built from nucleotides, and the processes involving DNA or RNA all require purine nucleotides to be synthesised first (through purine nucleotide metabolic process). Finally, the absence of links between the two nodes at orbits 5 is also relevant: there is no direct reaction degrading DNAs/RNAs directly into purine nucleosides or synthesising/transforming DNAs/RNAs by directly using nucleosides.

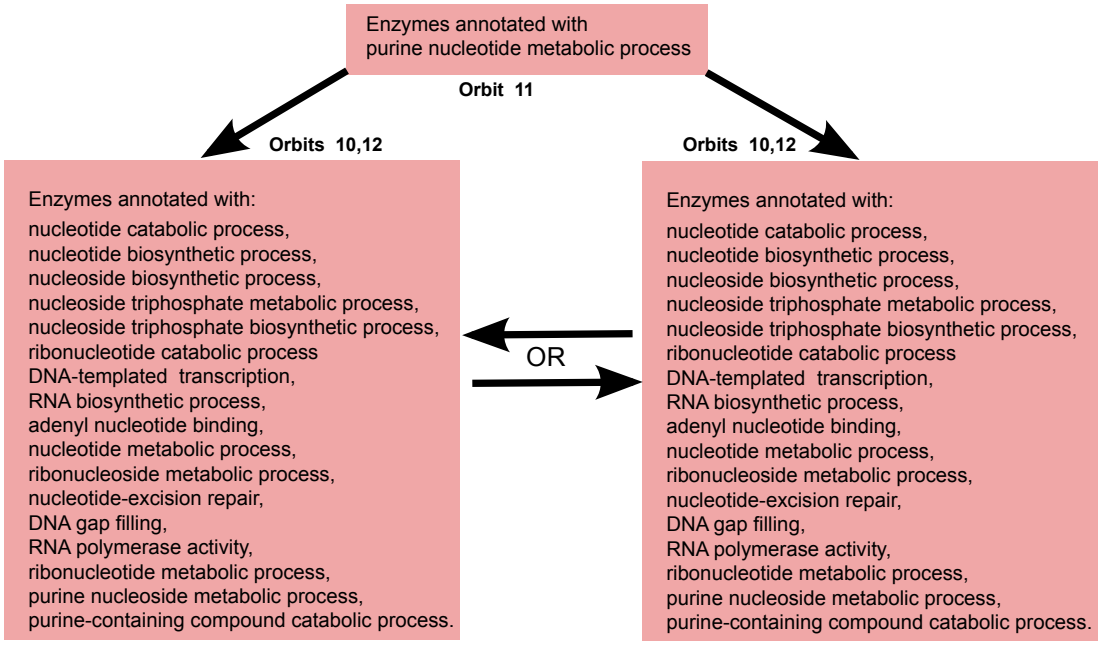


Figure 4.13. GO enrichment around enzymes touching orbit 11, that are annotated with purine nucleotide metabolic process.

For the enzymes annotated with the purine nucleotide metabolic process and that touch graphlets G_5 at orbit 11 in the *H. sapiens* metabolic network, we find that the set of enzymes touching graphlets G_5 at the remaining orbits (weather it is orbit 10 or 12 is irrelevant as indicated in Figure 4.10), are among other GO terms, also statistically significantly enriched in GO terms shown in Figure 4.13 (a full list is available in Appendices in Section C.2, Table C.4). First, we observe that most of the BPs that are found to be enriched at orbits 10 and 12 are the same that we previously found to be enriched at orbit 5. Since we have already described the biological mechanisms behind the directed edges from the ribonucleotide metabolic process toward these processes, we look for a biological confirmation of the existing edges between these processes themselves, as shown in Figure 4.13.

A nucleotide is composed of a nucleoside and one or more phosphate groups. Nucleosides can be phosphorylated by specific kinases in the cell to produce nucleotides. Hence, a directed edge from nucleoside biosynthetic process to nucleotide biosynthetic process is relevant: a nucleoside biosynthetic process produces nucleosides which are then used, in a nucleotide biosynthetic process, to produce nucleotides. Also, directed edges from nucleotide catabolic process to nucleoside biosynthetic process, nucleoside triphosphate

metabolic process and nucleoside triphosphate biosynthetic process are all relevant as they correspond to the production of nucleosides through the break down of nucleotides: *i.e.* a nucleotide catabolic process (a breakdown of nucleotides) results in nucleosides, which are then broken down into nucleobases and ribose or deoxyribose through any of the listed processes. Directed edges between enzymes annotated with the nucleotide biosynthetic process and enzymes annotated with DNA-templated transcription, RNA biosynthetic process, nucleotide-excision repair, DNA gap filling and RNA polymerase activity can be explained as follows. Nucleic acids are built from nucleotides. In order that processes involving DNA or RNA take place, a nucleotide biosynthetic process that produces nucleotides-nucleic acid building blocks, needs to be completed first.

For the enzymes annotated with the purine nucleotide metabolic process in the *M. musculus* metabolic network, that touch graphlets G_2 at orbit 6, we find that the set of enzymes touching these graphlets at orbit 5 is statistically significantly enriched, among others, in the following GO terms (full list is available in Appendices in Section C.2, Table C.5): nucleotide catabolic process, nucleotide biosynthetic process, nucleoside biosynthetic process, nucleoside triphosphate metabolic process, nucleoside triphosphate catabolic process, deoxyribonucleotide biosynthetic process, deoxyribonucleotide catabolic process, DNA-directed RNA polymerase I complex, nucleoside diphosphate metabolic process, ribonucleoside metabolic process, ribonucleotide metabolic process, nucleoside monophosphate biosynthetic process, nucleoside monophosphate catabolic process, purine-containing compound biosynthetic process, deoxyribonucleotide metabolic process. We have described the biological mechanisms behind the directed edges between the enzymes annotated with the purine nucleotide metabolic process and enzymes annotated with several of the listed processes for the case of *H. sapiens*. Another example is the process of synthesis of deoxyribonucleotides from ribonucleotides: a purine nucleotide metabolic process is required to produce ribonucleotides and is followed by a deoxyribonucleotide metabolic process which synthesises the deoxyribonucleotides.

For the enzymes annotated with the purine nucleotide metabolic process in the *M. musculus* metabolic network, that touch graphlets G_5 at orbit 11, we find that the set of enzymes touching these graphlets at the remaining orbits (10 or 12) is statistically significantly enriched in the same processes as listed above for the case of graphlets G_2 (a full list is available in Appendices in Section C.2, Table C.6). The links between the enzymes annotated with these processes, as described in the case of *H. sapiens*, can correspond to the breakdown of the nucleotides, synthesis of the nucleotides, processes involving DNA or RNA etc. In the case of *D. melanogaster* we find enrichments in the ribonucleotide metabolic process, purine nucleotide metabolic process, DNA-directed

RNA polymerase II, core complex, nucleotide metabolic process, hence the similar reasoning as in the case of the species above can be applied (a full list of GO terms is available in Appendices in Section C.2, Tables C.7 and C.8).

Case study: Ribose phosphate metabolic process. In the case of the ribose phosphate metabolic process, the analyses of topological patterns in metabolic networks of human, mouse and fruit fly result in similar lists of GO terms that were identified around the purine nucleotide metabolic process (Figures 4.12 and 4.13). This is due to the fact that several purine nucleotide metabolic processes (purine ribonucleotide metabolic process and purine ribonucleotide biosynthetic process) are child GO terms of the ribose phosphate metabolic process, as shown in Figure 4.11. The ribose phosphate metabolic process denotes chemical reactions and pathways resulting in the formation of ribose phosphate, any phosphorylated ribose sugar. A ribonucleotide or ribotide is a nucleotide containing ribose as its pentose component, hence, in order to be synthesised, a ribose phosphate metabolic process needs to first take place. This is facilitated through existing directed links in metabolic networks between enzymes annotated with the ribose phosphate metabolic process and enzymes annotated with the RNA biosynthetic process. Directed links between enzymes annotated with the nucleotide catabolic process and enzymes annotated with the nucleoside production processes, as well as directed links between enzymes annotated with the nucleotide biosynthetic process and enzymes annotated with DNA-templated transcription, RNA biosynthetic process or RNA polymerase activity are already discussed above within the analysis of topological patterns around the purine nucleotide metabolic process.

A full list of GO terms is given in Appendices in Section C.2, Tables C.9, C.10, C.11, C.12, C.13, C.14.

Case study: Cyclic nucleotide metabolic process. In eukaryotic cells, cyclic nucleotides, such as cAMP and cGMP, are secondary messengers. They relay the signals of many first messengers, such as hormones and neurotransmitters, to their physiological destinations, in both hormone and ion-channel signalling. cGMP, which is a cyclic nucleotide, is involved in the regulation of some protein-dependent kinases. cGMP binds to sites on the regulatory units of protein kinases G and activates the catalytic units, enabling them to phosphorylate their substrates.

Again, we present the more detailed analysis for the case of the human metabolic network and give a brief discussion for the case of the mouse and fruit fly. For the enzymes annotated with the cyclic nucleotide metabolic process and that touch graphlets

G_2 at orbit 6 in *H. sapiens* metabolic network, we find that the set of enzymes touching these graphlets at orbit 5 is statistically significantly enriched, among others, in GO terms shown in Figure 4.14. Again, note that we list GO terms for which we found the biological meaning behind the topological pattern. A full list of GO terms is given in Appendices in Section C.2, Table C.15.

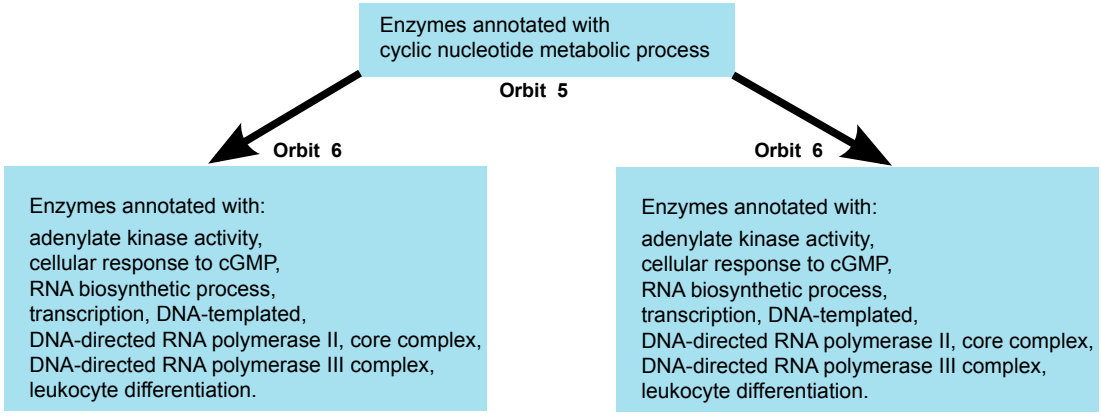


Figure 4.14. GO enrichment around enzymes touching orbit 6, that are annotated with cyclic nucleotide metabolic process.

The directed edge between the enzymes annotated with the cyclic nucleotide metabolic process at orbit 6 and enzymes annotated with adenylate kinase activity at orbit 5, is relevant as it supports the regulation of protein-dependent kinases: a cyclic nucleotide metabolic process that synthetizes cGMP is required so that cGMP can enable the adenylate kinase activity. Also, the directed edges between enzymes annotated with the cyclic nucleotide metabolic process at orbit 5 and enzymes annotated with GO terms related to RNA or DNA synthesis at orbit 6 (see Figure 4.14) are supported by findings that cyclic nucleotides regulate RNA by modulating the nucleotide precursors pool [204] and that cyclic AMPs stimulate RNA and DNA synthesis [205]. Directed edges between the enzymes annotated with the cyclic nucleotide metabolic process at orbit 6 and enzymes annotated with leukocyte differentiation at orbit 5 are supported by the fact that cyclic AMP plays a role in regulation, at a transcriptional level, of the expression of the CD7 leukocyte differentiation antigen [206]. Also, the directed edges between the enzymes annotated with a cyclic nucleotide metabolic process that synthetizes cGMP and enzymes annotated with cellular response to cGMP reflect the order of occurrence of these processes in the cell. Finally, the absence of links between the two nodes at orbit 5 is also relevant: for example, there are no direct connections between leukocyte

differentiation and RNA/DNA biosynthetic or transcriptional processes.

In the case of the *M. musculus* metabolic network we also find biological confirmation for the discussed topological pattern around cyclic nucleotide metabolic process. In particular, for the enzymes annotated with cyclic nucleotide metabolic process that touch graphlets G_2 at orbit 6, we find that the set of enzymes touching these graphlets at orbit 5 is statistically significantly enriched, among others, in the following GO terms: pyruvate kinase activity, DNA-directed RNA polymerase I complex and regulation of ERBB signaling pathway. A full list of GO terms is given in Appendices in Section C.2, Table C.17. Directed edges between enzymes annotated with the cyclic nucleotide metabolic process at orbit 6 and the enzymes annotated with processes related to kinase activity or DNA and RNA complex at orbit 5, are already supported in the discussion for the case of human metabolic network above. Directed edges between the enzymes annotated with the cyclic nucleotide metabolic process and enzymes annotated with the regulation of ERBB signaling pathway correspond to regulation of ERBB expression by the cAMP-dependent protein kinase [207]. In the case of the *D. melanogaster* metabolic network we find directed edges between enzymes annotated with the cyclic nucleotide metabolic process and processes related to DNA and RNA complexes (a full list of GO terms is given in Appendices in Section C.2, Table C.19.), which has already been discussed in the case of the human network.

For the enzymes annotated with the cyclic nucleotide metabolic process and that touch graphlets G_5 at orbit 11 in *H. sapiens* metabolic network, we find that the set of enzymes touching these graphlets at the remaining orbits (10 or 12) is statistically significantly enriched in similar processes as listed in Figure 4.15. An example of the biological functions that correspond to a local topology that is characteristic for graphlets G_5 where enzymes enriched in the cyclic nucleotide metabolic process touch orbit 11, is shown in Figure 4.15. In the case of *M. musculus* and *D. melanogaster*, we also find similar enrichments as for graphlets G_2. A full list of GO terms for graphlets G_5 and all three species is given in Appendices in Section C.2, Tables C.16, C.18 and C.20.

As discussed above, cyclic AMPs stimulate RNA and DNA synthesis, hence a cyclic nucleotide metabolic process which synthetises cyclic AMPs is required in order for the RNA biosynthetic process or DNA-templated transcription process to take place, thus explaining the directed edges from enzymes annotated with the cyclic nucleotide metabolic process (see Figure 4.15). The transcription process in which a particular segment of DNA is copied into RNA (mRNA) corresponds to edges between enzymes annotated with the DNA-templated transcription process and enzymes annotated with the RNA biosynthetic process.

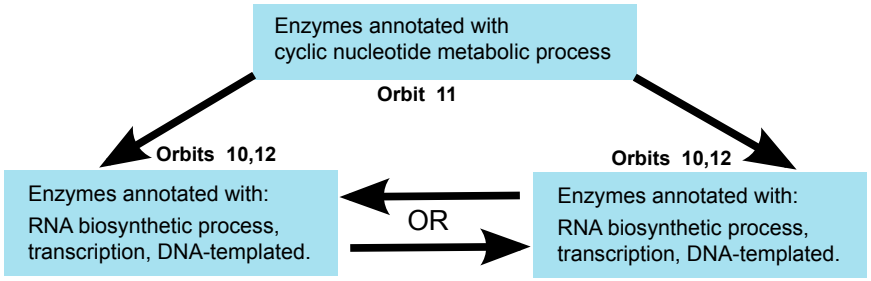


Figure 4.15. GO enrichment around enzymes touching orbit 11, that are annotated with cyclic nucleotide metabolic process.

4.4 Conclusions

In this chapter we applied directed graphlet-based heuristics to directed metabolic networks of eukaryotic species. We first confirmed that graphlet-based measures for directed network comparison outperform other commonly used measures, by evaluating their performance on clustering metabolic networks of different species, according to their taxonomic classification. We also showed that the quality of clustering decreases as the clustering is performed according to more specific levels of taxonomic classification. This indicates that topologies of metabolic networks of species with more recent divergence times differ less than those of the species that diverged further back in evolutionary history.

Then, motivated by the fact that the topology of PPI networks can be successfully used for the functional annotation of genes (proteins), and having defined directed graphlet-based heuristics, we explored whether similar local topology around enzymes in directed metabolic networks corresponds to the same GO term annotations. We discovered that each cluster of enzymes in the human metabolic network, constructed based on the similarity of local topology around enzymes measured using DGDV similarity, is statistically significantly enriched with similar GO terms. Based on this finding, we used known enzyme functional annotations and their local wiring patterns to show that the topology of metabolic networks can be a predictor of function: we utilised CCA to predict novel GO annotations in five eukaryotic species (*H. sapiens*, *M. musculus*, *D. melanogaster*, *S. cerevisiae* and *A. thaliana*).

We then searched for conserved topology–function relationships across different species following the framework of Davis *et. al* [201] and performed 10 pairwise experiments on metabolic networks of the five eukaryotic species. We found that the DNA metabolic

process, an essential process for all living organisms, is topologically orthologous across 8 out of 10 species pairs, revealing how evolution has conserved a specific local wiring in the metabolic network required for the DNA metabolic process. We also found 27 biological processes that are topologically orthologous across 3 species pairs, as well as 14 biological processes that are topologically orthologous across 2 species pairs. Additionally, we identified distinctive wirings in metabolic networks which correspond to various biological functions. For example, lipid-related biological processes are characterised with the four-node cliques; the purine nucleotide metabolic process, the ribose phosphate metabolic process and the cyclic nucleotide metabolic process are correlated to the number of orbits with outgoing edges. Finally, we offered a biological explanation as to how the specific wirings with outgoing edges facilitate topologically orthologous biological functions.

4.5 Author's Contributions

Section 4.1 Anida Sarajlić collected and preprocessed the directed metabolic networks of eukaryotes and the taxonomic classification of species, designed, implemented and performed experiments for evaluation of clustering of eukaryotes and analysed the results.

Section 4.2 Anida Sarajlić collaborated with Noël Malod-Dognin on the work presented in this section. Anida Sarajlić designed, implemented and performed experiments for calculating the cluster enrichments in GO terms and analysed the results.

Section 4.3 Anida Sarajlić collaborated with Noël Malod-Dognin, Ömer Nebil Yaveroğlu and Nataša Pržulj on the work presented in this section. Anida Sarajlić preprocessed all input data (DGDVs of genes for directed metabolic network of the 5 analysed species, GO annotations for the genes of the 5 analysed species) for the existing framework for identification of topologically orthologous GO terms (experiments conducted by Ömer Nebil Yaveroğlu). Anida Sarajlić implemented and performed precision-recall analysis for the GO term predictions, analysed results for the GO term predictions and analysed results for the topologically orthologous GO terms (identifying biological topologically orthologous processes across different species pairs from the output data provided by Ö.N.Y, identifying specific wiring patterns characteristic for topologically orthologous processes and performing the biological analysis).

Anida Sarajlić was supervised by Dr. Noël Malod-Dognin and Dr. Nataša Pržulj for the work presented in this chapter.

Anida Sarajlić wrote the first draft for the paper: Anida Sarajlić, Noël Malod-Dognin,

Ömer Nebil Yaverŏglu and Nataša Pržulj: "Directed Graphlets Uncover Topology–Function Relationships in Directed Metabolic Networks of Eukaryotes" in August 2015. This paper draft contained the work presented in Chapters 3 and 4. Currently (December 2015), the results presented in that paper draft are being merged with the results of application of directed graphlets to directed world trade networks, aiming for a publication with wider range of applications (Note: Anida Sarajlić provided the directed orbit and graphlet counts for the directed world trade networks, while further experiments and analyses on world trade networks were performed by Noël Malod-Dognin and Ömer Nebil Yaverŏglu).

5 Conclusions and Future Directions

In this chapter we summarise the contributions made by this dissertation. We also discuss future applications of our new methodology for directed network analysis.

5.1 Conclusions

The topology of undirected biological networks has already been linked to biological functions [1–6, 13, 145]. We confirmed this in Chapter 2 where we first reviewed the existing approaches that utilise the topological properties of biological networks in the research of complex diseases, in particular cardiovascular disease (CVD) [98], and then contributed with our CVD case studies [8,99]. We identified the key CVD genes that are statistically significantly enriched in drug targets and driver genes, using the topology of the human PPI network and we predicted novel CVD genes (70% validated) which are functionally similar to currently known CVD drug targets, confirming the potential for improving the therapy of CVDs [8]. We also tackled the reasons behind the protective role of diabetes in cases of aneurysm patients and, using the topologies of the human PPI and genetic networks, identified pleiotropic kinases potentially responsible for this relationship between the diseases [99]. These findings confirm the value of the information that is encoded in the topology of biological networks and its usability for a broad spectrum of open questions in biology and medicine.

Graphlet based properties of undirected networks have particularly contributed to new findings in computational biology [4–6, 8, 9, 13, 45]. Some of the biological networks are undirected by definition, while many, such as metabolic networks or transcriptional regulatory networks, are complete only if the directionality of interactions is taken into account, given that directionality adds an additional level of information to the network data.

Hence, in Chapter 3, as the main contribution of this dissertation, we defined directed graphlets and orbits and implemented an algorithm for counting all graphlets in a directed network as well as all graphlet orbits for each of the network nodes. We then generalised existing graphlet-based measures: we defined directed graphlet-based

measures for topological similarity between nodes in a network (directed graphlet degree vector distance) and network comparison (relative directed graphlet frequency distance, directed graphlet degree distribution similarity and directed graphlet correlation distance). We compared the new measures against degree distribution based and spectral distance measures, by evaluating their performance on model network clustering. We used existing directed random network models for SF directed graphs and ER directed graphs and proposed SF-GD, GEO, and GEO-GD directed random network models to generate sets of model networks. We separately evaluated clustering for two distinctive cases: when only the same size and density networks are compared and when all-to-all networks are compared. We also tested the noise robustness for all the measures, by repeating the model clustering evaluation when up to 70% of noise is introduced to the model networks (in increments of 10%). We took into account three types of noise separately: random addition of edges to the network, random removal of edges from the network and random rewiring of the edges in the network. The results showed that our proposed graphlet-based measures for network comparison outperform those based on degree distribution and network spectrum. Among graphlet-based measures, directed graphlet correlation distance performed the best in network model identification and was the most resilient to the noise in the networks.

In Chapter 4, we applied directed graphlets to metabolic networks. We first used our directed graphlet-based measures for network comparison to show that the topology of metabolic networks can be used to reconstruct phylogenetic relationships between eukaryotic species. In particular, we explored whether the grouping of all eukaryotic species based on the similarity of the topologies of their metabolic networks corresponds to the taxonomic classification of the species. We evaluated this by assessing the quality of the clustering of 299 eukaryotes based on topological similarity of their metabolic networks, according to six levels of taxonomic classification, yielding AUC scores as high as 0.93. The best results were obtained for the directed graphlet correlation distance measure. We also found that the quality of the clustering decreases for more specific levels of taxonomic classification, suggesting that the metabolic networks of species that have diverged more recently in time do not differ as much as for those species that diverged earlier in evolutionary history.

We then used DGDV similarity, a measure that quantifies the topological similarity between the nodes in a network, to cluster the nodes in the metabolic network of *H. sapiens* and found that each of the obtained clusters is statistically significantly enriched in the GO terms that correspond to related functions. This indicates that similar wirings in directed metabolic networks correspond to similar biological functions. For

example, we found clusters enriched in catabolic processes, processes supporting methylation, acid binding processes, organism defence response and processes related to sugar metabolism. Motivated by the findings, we used the metabolic networks to successfully predict new GO annotations based solely on the DGDVs of the genes (enzymes) in the networks. Finally, we explored if there exist topology–function relationships in metabolic networks that are conserved across different species. For this, we followed the existing framework of Davis *et.al* [201], where several topologically orthologous GO terms were uncovered using undirected PPI networks of human and yeast. We extended this approach to directed metabolic networks and explored the metabolic networks of human, yeast, mouse, fruit fly and arabidopsis. We performed 10 pairwise analyses across these 5 species and found that evolution has conserved a specific local wiring in the metabolic network that is required for the DNA metabolic process. In addition, we found 27 biological processes that are topologically orthologous for 3 pairs of species and 14 biological processes topologically orthologous for 2 pairs of species. We also identified distinctive wirings in the metabolic networks that correspond to particular biological functions: *e.g.* lipid-related biological processes correspond to enzymes with the local topology characterised with four-node cliques, while purine nucleotide metabolic processes, ribose phosphate metabolic processes and cyclic nucleotide metabolic processes correspond to enzymes with the local topology characterised with the outgoing edges. We also offer a biological explanation as to why these particular processes are related to the outgoing edges in the metabolic networks.

5.2 Future directions

Parallelising the graphlet counting algorithm As discussed in Section 3.1.4, the directed graphlet and orbit counting algorithm was implemented in C++ and has a time complexity of $O(N \times d^3)$. We have used the software on the networks with the maximum size of 2000 nodes and edge density of 1%. However, for future applications on denser graphs (when $d \to N$) computations of DGDVs are bound to be time consuming. Hence, we are planning to parallelise the code to improve its time efficiency. This is a straight forward process as the counting algorithm was implemented in the way that each node in the network is visited separately, and the number of orbits that the node touches (counted by examining it's three node deep neighbourhood) is added to the node's directed graphlet degree vector (DGDV). We will parallelise the counter by dividing network nodes to sets and assigning each set of nodes to a separate job. Each job will then separately maintain the temporary DGDV for all nodes in the network, so when

the job is counting orbits that a particular node touches, it can still update the orbits for the neighbouring nodes, even if they are not within its designated set. After all the jobs are completed, values from all temporary DGDVs for each node should be added together. All the corrections for the over-counts discussed in Section 3.1.4, should be performed after the merging the temporary vectors.

Evaluation of graphlet based properties when the set of redundant orbits is removed. As discussed in Section 3.1.3, in the case of networks without anti-parallel pairs of arcs, 23 out of the 129 orbits are redundant and can be derived using the counts of the remaining 106 orbits. The redundant orbits can be omitted when calculating the directed graphlet based measures: graphlet degree distribution similarity, graphlet degree vector similarity and graphlet correlation matrix distance [13]. We plan to define these measures without including redundant orbits and evaluate their performance.

Graphlets for the analysis of directed weighted networks. Many networks, both undirected and directed, can have weights assigned to their edges, providing additional information about the data. For example, in the world trade networks, the nodes correspond to countries, edges correspond to the trade between the countries, and the amounts of the trade can be accounted for as the weights of the edges. In biological networks, for example, it is possible to assign weight to the edges based on confidence levels (for example the statistical significance of the interactions between nodes). The additional step in graphlet generalisation is to take into account the edge weights. The concept of weighted graphlets and how to apply graphlet based measures to a weighted network is an open research problem that would be addressed in the future research.

Integrating directed graphlet counter and directed model network generators with Graph Crunch software. GraphCrunch [208] is a software tool created for network analysis. The software enables the comparison of networks against random graph models. It generates random networks for user-specified random graph models and evaluates the fit of a variety of network models to real-world networks with respect to a series of global and local network properties. The software was further upgraded to version GraphCrunch 2 [209] which also provides the best fitting model for the network data. Graph Crunch 2 supports the following network models: Erdös-Rényi random graphs (ER), Erdös-Rényi random graphs with the same degree distribution as the data (ER-DD), scale-free Barabási-Albert preferential attachment models (SF), geometric random graphs (GEO), stickiness-index based models (STICKY), scale-free gene

duplication models (SF-GD), and geometric gene duplication models (GEO-GD) and provides pairwise network comparison using graphlet-based heuristics, the alignment of two networks using the GRAAL algorithm [194], and clustering network nodes based on the similarity of their neighbourhood topology. Graph Crunch 2 is user friendly and has an intuitive drag and drop interface. The upgraded version of software, Graph Crunch 3, is in preparation and is currently implemented in Python as a web server. We plan to include in the Graph Crunch 3 the directed graphlet and orbit counter, directed graphlet-based heuristics (relative graphlet frequency distance graphlet degree distribution similarity, graphlet degree vector similarity, and graphlet correlation matrix distance) and directed network models.

Model fitting of directed real world networks. Finding a well-fitting network model for a real world network can help explain the evolution of the network and uncover additional information from the network data such as missing links in the network. For example, biological data are incomplete and noisy due to sampling, biases in data collection and interpretation, and limitations in technology [46, 47]. If it is possible to find an adequate theoretical network model that fits a network - that precisely reproduces the networks structure and laws - then that model can be used to predict missing interactions or to filter out the false ones. Also, a well-fitting model can provide easier computational manipulation of the network data and aid understanding of the mechanisms of biological processes within the cell [48].

The fitting of a network model to a given network is performed as follows: (1) for an input network G, number of networks from the evaluated network model are generated, with the same size and density of the input network (30 networks per model have been suggested as a standard [59, 60, 210]), (2) the topologies of the generated model networks are compared to the topology of the input network G using global or local network properties.

Przulj *et al.* showed that a random geometric model fits the PPI data using RGF distance [7]. Yaveroğlu *et al.* [13] used GCD-11 network distance measure to find the best fitting network models for autonomous system networks, Facebook networks, metabolic networks, protein structure networks, and world trade networks. It was shown that for autonomous networks ER-DD is the best fitting model, while Facebook, metabolic, and protein structure networks are best modelled by GEO, GEO-GD and SF-GD models. All these networks were analysed as undirected.

Since there is available information on the directionality of the edges in metabolic, Facebook and world-trade networks, we suggest finding directed model networks that

are the best fit for these directed real world networks by using the directed graphlet correlation distance measure that performed best in model identification (Section 3.2.2). In addition, we plan to perform model fitting on directed transcriptional regulatory networks and effective connectivity brain networks.

Alignment of directed networks based on directed graphlets. Exact comparisons of large networks are computationally infeasible due to NP-completeness of the underlying subgraph isomorphism problem [211]. Subgraph isomorphism problem asks whether a graph G exists as an exact subgraph of another graph H. Network alignment is a more general problem which asks what the best way to fit network G into the network H is. There exist local and global network alignments. In *local network alignment* (LNA) methods regions of network are mapped independently, and as a result one node in the network can be mapped to several nodes in another network. *Global network alignments* (GNA) align each node in a smaller network to exactly one node in a larger network, maximising the overall match between the two networks.

Because the network alignment is NP complete problem, heuristic approaches have been used to address it. In the case of global network aligners for the alignment of the biological networks, the mappings of the nodes in the alignment are guided by similarity scores, which take into account the topological similarity between nodes (*e.g* GRAAL family of aligners, discussed below), the biological information (*e.g.* sequence similarity in PISwap [212]), or both ((*e.g* IsoRank [213], Natalie [214], GHOST [215]).

As mentioned above, the GRAAL family of network aligners can align networks by using only the topological information around the nodes in the network (*e.g.* GDV similarity), so it can be applied for the alignment of networks other than biological. However, GRAAL aligners still allow the inclusion of other information, such as biological, to the cost function of matching the nodes. GRAAL family includes: (1) GRAAL [194] - uses the greedy "seed and extend" approach to align network nodes where the cost is decreased as the degrees of the nodes involved increase, making sure that the densest parts of the networks are aligned first, (2) H-GRAAL [216] - uses the Hungarian algorithm to produce optimal global alignment between two networks using any cost function, (3) MI-GRAAL [217] - designed so it can integrate any number or type of similarity measures between nodes; it follows the seed and extend approach and builds the matrix of confidence scores, simultaneously constructing a priority queue of node pairs in decreasing order with respect to their confidence scores, (4) C-GRAAL [218] - finds a seed alignment in the networks and expends around it, by finding the alignment between the neighbours of already aligned proteins and (5) L-GRAAL [219] - which is based on

integer programming and Lagrangian relaxation [220] and uses GDV similarity as the topological information for the cost function.

We plan to extend the GRAAL family with a network aligner for directed networks. Since L-GRAAL, the most recent aligner in the GRAAL family, outperforms the others - it uncovers the largest common sub-graphs between the networks [219] - we will focus on generalising L-GRAAL to a directed network aligner. The main algorithm would remain similar, but we plan to use the DGDV similarity instead of GDV similarity between the nodes.

Bibliography

[1] B. Schwikowski and P. Uetz. A network of protein-protein interactions in yeast. *Nature Biotechnology*, 18:1257–1261, 2000.

[2] H. N. Chua, W.-K. Sung, and L. Wong. Exploiting indirect neighbours and topological weight to predict protein function from protein-protein interactions. *Bioinformatics*, 22(13):1623–1630, 2006.

[3] M. P. Samanta and S. Liang. Predicting protein functions from redundancies in large-scale protein interaction networks. *Proceedings of the National Academy of Sciences USA*, 100:12579–12583, 2003.

[4] T. Milenković and N. Pržulj. Uncovering biological network function via graphlet degree signatures. *Cancer Informatics*, 4:257–273, 2008.

[5] T. Milenković, V. Memišević, A.K. Ganesan, and N. Pržulj. Systems-level cancer gene identification from protein interaction network topology applied to melanogenesis-related functional genomics data. *Journal of the Royal Society Interface*, 44(7):353–350, 2010.

[6] H. Ho, T. Milenković, V. Memišević, J. Aruri, N. Pržulj, and A.K. Ganesan. Protein interaction network topology uncovers melanogenesis regulatory network components within functional genomics datasets. *BMC Systems Biology*, 4(1):84, 2010.

[7] N. Pržulj, D. G. Corneil, and I. Jurisica. Modeling interactome: Scale-free or geometric? *Bioinformatics*, 20(18):3508–3515, 2004.

[8] A. Sarajlić, V. Janjić, N. Stojković, Dj. Radak, and N. Pržulj. Network topology reveals key cardiovascular disease genes. *PLoS ONE*, 8(8):e71537+, 2013.

[9] V. Gligorijević, V. Janjić, and N. Pržulj. Integration of molecular network data reconstructs gene ontology. *Bioinformatics*, 30:I594–I600, 2014.

[10] C. F. Schaefer, K. Anthony, S. Krupa, J. Buchoff, M. Day, T. Hannay, and K. H. Buetow. Pid: the pathway interaction database. *Nucleic Acids Research*, 37(Database issue):674–679, January 2009.

[11] L. R. Varshney, B. L. Chen, E. Paniagua, D. H. Hall, and D. B. Chklovskii. Structural properties of the caenorhabditis elegans neuronal network. *PLoS Computational Biology*, 7(2), 2011.

[12] Y. Rosen and Y. Louzoun. Directionality of real world networks as predicted by path length in directed and undirected graphs. *Physica A: Statistical Mechanics and its Applications*, 401(C):118–129, 2014.

[13] Ö. N. Yaverŏglu, N. Malod-Dognin, D. Davis, Z. Levnajić, R. Karapandža V. Janjić, A. Stojmirović, and N. Pržulj. Revealing the hidden language of complex networks. *Scientific Reports*, 4:4547, 2014.

[14] A. S. Go, D. Mozaffarian, V. L. Roger, E. J. Benjamin, J. D. Berry, W. B. Borden, D. M. Bravata, S. Dai, E. S. Ford, C. S. Fox, S. Franco, H. J. Fullerton, C. Gillespie, S. M. Hailpern, J. A. Heit, V. J. Howard, M. D. Huffman, B. M. Kissela, S. J. Kittner, D. T. Lackland, J. H. Lichtman, L. D. Lisabeth, D. Magid, G. M. Marcus, A. Marelli, D. B. Matchar, D. K. McGuire, E. R. Mohler, C. S. Moy, M. E. Mussolino, G. Nichol, N. P. Paynter, P. J. Schreiner, P. D. Sorlie, J. Stein, T. N. Turan, S. S. Virani, N. D. Wong, D. Woo, M. B. Turner, American Heart Association Statistics Committee, and Stroke Statistics Subcommittee. Executive summary: heart disease and stroke statistics–2013 update: a report from the american heart association. *Circulation*, 127(1):143–152, 2013.

[15] B. Bollobas. Paul erdos and probability theory. *Random Structures and Algorithms*, 13(3-4):521–533, 1998.

[16] R. Albert and A.-L. Barabási. Statistical mechanics of complex networks. *Reviews of Modern Physics*, 74:47–97, 2002.

[17] D. B. West. *Introduction to Graph Theory.* Prentice Hall inc., 2nd edition, 2001.

[18] M. Newman. *Networks: An Introduction.* Oxford University Press, Inc., 2010.

[19] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness (Series of Books in the Mathematical Sciences).* W. H. Freeman, 1979.

[20] J. van Leeuwen, editor. *Handbook of Theoretical Computer Science (Vol. A): Algorithms and Complexity*. MIT Press, Cambridge, MA, USA, 1990.

[21] S. Cook. The p versus np problem. In *Clay Mathematical Institute; The Millennium Prize Problem*, 2000.

[22] L. Fortnow. The status of the p versus np problem. *Communications of the ACM*, 52(9):78–86, September 2009.

[23] R. Milo, S. Shen-Orr, S. Itzkovitz, N. Kashtan, D. Chklovskii, and U. Alon. Network motifs: Simple building blocks of complex networks. *Science*, 298(5594):824–827, 2002.

[24] M. E. J. Newman. The structure and function of complex networks. *Computer Physics Communications*, 147(1):40–45, 2001.

[25] O. Sporns, C. J Honey, and R. Kötter. Identification and classification of hubs in brain networks. *PloS one*, 2(10):e1049–e1049, 2007.

[26] J. Kleinberg and S. Lawrence. The structure of the web. *Science*, 294(5548):1849–1850, 2001.

[27] J. M. Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM (JACM)*, 46(5):604–632, 1999.

[28] E. A. Bender, E. R. Canfield, and B. D. McKay. The asymptotic number of labeled connected graphs with a given number of vertices and edges. *Random Structures and Algorithms*, 1(2):127–169, 1990.

[29] D. J. Watts and S. H. Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393(6684):440–442, 1998.

[30] M. E. J. Newman. Properties of highly clustered networks. *Physical Review E*, 68:026121, 2003.

[31] S. N. Soffer and A. Vazquez. Network clustering coefficient without degree-correlation biases. *Physical Review E*, 71(057101), 2005.

[32] G. Valiente. *Algorithms on Trees and Graphs*. Springer, 2002.

[33] D. J. Watts. *Small Worlds – The Dynamics of Networks between Order and Randomneess*. Princeton University Press, 1999.

[34] P. Bonacich. Power and Centrality: A family of measures. *American Journal of Sociology*, 92:1170–1182, 1987.

[35] L. Katz. A new status index derived from sociometric analysis. *Psychometrika*, 18(1):39–43, 1953.

[36] S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. *Computer Networks and ISDN Systems*, 30(1-7):107 – 117, 1998.

[37] P. Zhu and R. C. Wilson. A study of graph spectra for comparing graphs. In *BMVC*. British Machine Vision Association, 2005.

[38] G. Sabidussi. The centrality index of a graph. *Psychometrika*, 31(4):581–603, 1966.

[39] L. C. Freeman. A set of measures of centrality based upon betweenness. *Sociometry*, 40:35–41, 1977.

[40] S. Carmi, S. Havlin, S. Kirkpatrick, Y. Shavitt, and E. Shir. From the Cover: A model of Internet topology using k-shell decomposition. *Proceedings of the National Academy of Sciences USA*, 104(27):11150–11154, 2007.

[41] R. Milo, S. Itzkovitz, N. Kashtan, R. Levitt, S. Shen-Orr, I. Ayzenshtat, M. Sheffer, and U. Alon. Superfamilies of evolved and designed networks. *Science*, 303:1538–1542, 2004.

[42] U. Alon. Network motifs: theory and experimental approaches. *Nat. Rev. Genet.*, 8(6):450–461, 2007.

[43] Y. Artzy-Randrup, S. J. Fleishman, N. Ben-Tal, and L. Stone. Comment on "network motifs: Simple building blocks of complex networks" and "superfamilies of evolved and designed networks". *Science*, 305(5687):1107, 2004.

[44] N. Pržulj. Biological network comparison using graphlet degree distribution. *Bioinformatics*, 23(2):e177–e183, 2007.

[45] C. Guerrero, T. Milenković, N. Pržulj, J. J. Jones, P. Kaiser, and L. Huang. Characterization of the yeast proteasome interaction network by qtax-based tag-team mass spectrometry and protein interaction network analysis. *Proceedings of the National Academy of Sciences USA*, 105(36):13333–13338, 2008.

[46] J.-D. J. Han, D. Dupuy, N. Bertin, M. E. Cusick, and M. Vidal. Effect of sampling on topology predictions of protein-protein interaction networks. *Nature Biotechnology*, 23(7):839–844, 2005.

[47] E. de Silva, T. Thorne, P. Ingram, I. Agrafioti, J. Swire, C. Wiuf, and M.P.H. Stumpf. The effects of incomplete protein interaction data on structural and evolutionary inferences. *BMC Biology*, 4(1):1–13, 2006.

[48] V. Memisevic, T. Milenkovic, and N. Pržulj. An integrative approach to modeling biological networks. *J. Integrative Bioinformatics*, 7(3), 2010.

[49] P. Erdös and A. Rényi. On random graphs. *Publicationes Mathematicae Debrecen*, 6:290, 1959.

[50] E. de Silva and M.P. Stumpf. Complex networks and simple models in biology. *Journal of the Royal Society, Interface*, 2(5):419–30, 2005.

[51] M. E. J. Newman, S. H. Strogatz, and D. J. Watts. Random graphs with arbitrary degree distributions and their applications. *Physical Review E*, 64(2):026118, 2001.

[52] A.-L. Barabási and R. Albert. Emergence of scaling in random networks. *Science*, 286(5439):509–512, 1999.

[53] W. Aiello, F. Chung, and L. Lu. A random graph model for power law graphs. *Experimental Mathematics*, 10:53–66, 2000.

[54] A.-L. Barabási and Z. N. Oltvai. Network biology: understanding the cell's functional organization. *Nature Reviews Genetics*, 5(2):101–113, 2004.

[55] M. P. H. Stumpf, C. Wiuf, and R. M. May. Subnets of scale-free networks are not scale-free: Sampling properties of networks. *Proceedings of the National Academy of Sciences USA*, 102(12):4221–4224, 2005.

[56] A. Vazquez, A. Flammini, A. Maritan, and A. Vespignani. Modeling of protein interaction networks. *Complexus*, 1(1):38–44, 2003.

[57] N. Przulj and Higham D.J. Modelling protein-protein interaction networks via a stickiness index. *Journal of the Royal Society Interface*, 3(10):711–6, 2006.

[58] O. Kuchaiev and N. Przulj. Learning the structure of protein-protein interaction networks. In *Pacific Symposium on Biocomputing*, pages 39–50, 2009.

[59] D. J. Higham, M. Rasajski, and N. Przulj. Fitting a geometric graph to a protein-protein interaction network. *Bioinformatics*, 24(8):1093–1099, 2008.

[60] N. Przulj, O. Kuchaiev, A. Stevanovic, and W. Hayes. Geometric evolutionary dynamics of protein interaction networks. In *Pacific Symposium on Biocomputing*, pages 178–189. World Scientific Publishing, 2010.

[61] D. K. Arrell and A. Terzic. Network systems biology for drug discovery. *Clin Pharmacol Ther*, 88(1):120–125, 2010.

[62] A. Chatr-aryamontri, B.-J. Breitkreutz, S. Heinicke, L. Boucher, A. Winter, C. Stark, J. Nixon, L. Ramage, N. Kolas, L. O'Donnell, T. Reguly, A. Breitkreutz, A. Sellam, D. Chen, C. Chang, J. Rust, M. Livstone, R. Oughtred, K. Dolinski, and M. Tyers. The biogrid interaction database. *Nucleic Acids Research*, 41(D1):D816–D823, 2013.

[63] S. Peri, J. D. Navarro, R. Amanchy, T. Z. Kristiansen, C. K. Jonnalagadda, V. Surendranath, V. Niranjan, B. Muthusamy, T.K.B. Gandhi, M. Gronborg, N. Ibarrola, N. Deshpande, K. Shanker, H.N. Shivashankar, B.P. Rashmi, M.A. Ramya, Z. Zhao, K.N. Chandrika, N. Padma, H.C. Harsha, A.J. Yatish, M.P. Kavitha, M. Menezes, D. R. Choudhury, S. Suresh, N. Ghosh, R. Saravana, S. Chandran, S. Krishna, M. Joy, S. K. Anand, V. Madavan, A. Joseph, G. W. Wong, W. P. Schiemann, S. N. Constantinescu, L. Huang, R. Khosravi-Far, H. Steen, M. Tewari, S. Ghaffari, G. C. Blobe, C. V. Dang, J.G.N. Garcia, J. Pevsner, O. N. Jensen, P. Roepstorff, K. S. Deshpande, A. M. Chinnaiyan, A. Hamosh, A. Chakravarti, and A. Pandey. Development of human protein reference database as an initial platform for approaching systems biology in humans. *Genome Research*, 13(10):2363–2371, 2003.

[64] I. Xenarios, L. Salwinski, X. J. Duan, P. Higney, S.-M. Kim, and D. Eisenberg. DIP, the database of interacting proteins: A research tool for studying cellular networks of protein interactions. *Nucleic Acids Research*, 30(1):303–305, 2002.

[65] M. Persico, A. Ceol, C. Gavrila, R. Hoffmann, A. Florio, and G. Cesareni. Homomint: an inferred human network based on orthology mapping of protein interactions discovered in model organisms. *BMC Bioinformatics*, 6(S-4), 2005.

[66] K. R. Brown and I. Jurisica. Online predicted human interaction database. *Bioinformatics*, 21(9):2076–2082, 2005.

[67] M. Kanehisa, S. Goto, Y. Sato, M. Furumichi, and M. Tanabe. Kegg for integration and interpretation of large-scale molecular data sets. *Nucleic Acids Research*, 40(Database issue):D109–14, 2012.

[68] A. Hamosh, A. F. Scott, J. S. Amberger, C. A. Bocchini, and V. A. McKusick. Online mendelian inheritance in man (omim), a knowledgebase of human genes and genetic disorders. *Nucleic Acids Research*, 33(suppl 1):D514–D517, 2005.

[69] J. L. Y. Koh, H. Ding, M. Costanzo, A. Baryshnikova, K. Toufighi, G. D. Bader, C. L. Myers, B. J. Andrews, and C. Boone. Drygin: a database of quantitative genetic interaction networks in yeast. *Nucleic Acids Research*, 38(Database-Issue):502–507, 2010.

[70] H. Salgado, M. Peralta-Gil, S. Gama-Castro, A. Santos-Zavaleta, L. Muñiz Rascado, J. S. García-Sotelo, V. Weiss, H. Solano-Lira, I. Martínez-Flores, A. Medina-Rivera, G. Salgado-Osorio, S. Alquicira-Hernández, K. Alquicira-Hernández, A. López-Fuentes, L. Porrón-Sotelo, A. M. Huerta, C. Bonavides-Martínez, Y. I. Balderas-Martínez, L. Pannier, M. Olvera, A. Labastida, V. Jiménez-Jacinto, L. Vega-Alvarado, V. Del Moral-Chávez, A. Hernández-Alvarez, E. Morett, and J. Collado-Vides. RegulonDB v8.0: omics data sets, evolutionary conservation, regulatory phrases, cross-validated gold standards and more. *Nucleic acids research*, 41(Database issue), 2013.

[71] G. Joshi-Tope, M. Gillespie, I. Vastrik, P. D'Eustachio, E. Schmidt, B. de Bono, B. Jassal, G. R. Gopinath, G. R. Wu, L. Matthews, S. Lewis, E. Birney, and L. Stein. Reactome: a knowledgebase of biological pathways. *Nucleic Acids Research*, 33(Database-Issue):428–432, 2005.

[72] A. G. Murzin, S. E. Brenner, T. Hubbard, and C. Chothia. SCOP: a structural classification of proteins database for the investigation of sequences and structures. *J Mol Biol.*, 247(4):536–540, 1995.

[73] U. Stelzl, U. Worm, M. Lalowski, C. Haenig, F.H. Brembeck, H. Goehler, M. Stroedicke, M. Zenkner, A. Schoenherr, S. Koeppen, S. Timm, J.and Mintzlaff, C. Abraham, N. Bock, S. Kietzmann, A. Goedde, E. Toksoz, A. Droege, S. Krobitsch, B. Korn, W. Birchmeier, H. Lehrach, and E.E. Wanker. A human protein-protein interaction network: a resource for annotating the proteome. *Cell*, 122(6):957–968, 2009.

[74] G. Rigaut, A. Shevchenko, B. Rutz, M. Wilm, M. Mann, and B. Séraphin. A generic protein purification method for protein complex characterization and proteome exploration. *Nature Biotechnology*, 17(10):1030–1032, 1999.

[75] L. Hakes, J.W. Pinney, D.L. Robertson, and S.C. Lovell. Protein-protein interaction networks and biology–what's the connection?. *Nature Biotechnology*, 26(1):69–72, 2008.

[76] M.P.H. Stumpf, C. Wiuf, and R.M. May. Subnets of scale-free networks are not scale-free: Sampling properties of networks. *Proceedings of the National Academy of Sciences USA*, 102(12):4221–4224, 2005.

[77] N. Pržulj. Protein-protein interactions: making sense of networks via graph-theoretic modeling. *BioEssays : News and Reviews in Molecular, Cellular and Developmental Biology*, 33(2):115–123, 2011.

[78] D. Szklarczyk, A. Franceschini, M. Kuhn, M. Simonovic, A. Roth, P. Minguez, T. Doerks, M. Stark, J. Muller, P. Bork, L. J. Jensen, and C. von Mering. The string database in 2011: functional interaction networks of proteins, globally integrated and scored. *Nucleic Acids Research*, 39(Database-Issue):561–568, 2011.

[79] M. Costanzo, A. Baryshnikova, J. Bellay, Y. Kim, E. D. Spear, C. S. Sevier, H. Ding, J. L. Y. Koh, K. Toufighi, S. Mostafavi, J. Prinz, R. P. St. Onge, B. VanderSluis, T. Makhnevych, F. J. Vizeacoumar, S. Alizadeh, S. Bahr, R.e L. Brost, Y. Chen, M. Cokol, R. Deshpande, Z. Li, Z.-Y. Lin, W. Liang, M. Marback, J. Paw, B.-J. San Luis, E. Shuteriqi, A. H. Y. Tong, N. van Dyk, I. M. Wallace, J. A. Whitney, M. T. Weirauch, G. Zhong, H. Zhu, W. A. Houry, M. Brudno, S. Ragibizadeh, B. Papp, C. Pal, F. P. Roth, G. Giaever, C. Nislow, O. G. Troyanskaya, H. Bussey, G.y D. Bader, A.-C.e Gingras, Q. D. Morris, P.p M. Kim, C. A. Kaiser, C. L. Myers, B. J. Andrews, and C. Boone. The Genetic Landscape of a Cell. *Science*, 327, 2010.

[80] R. Kelley and T. Ideker. Systematic interpretation of genetic interactions using protein networks. *Nature Biotechnology*, 23(5):561, 2005.

[81] S.L. Ooi, D.D. Shoemaker, and Boeke J.D. Dna helicase gene interaction network defined using synthetic lethality analyzed by microarray. *Nature Genetics*, 35(3):277–86, 2003.

[82] R. Mani, R. P. St. Onge, J. L. Hartman, G. Giaever, and F. P. Roth. Defining genetic interaction. *Proceedings of the National Academy of Sciences USA*, 105:3461–3466, 2008.

[83] P. Beltrao, G. Cagney, and N. J. Krogan. Quantitative genetic interactions reveal biological modularity. *Cell*, 141:739–745, 2010.

[84] A. Hin Yan Tong, M. Evangelista, A. B. Parsons, H. Xu, G. D. Bader, N. Pagé, M. Robinson, S. Raghibizadeh, C. W. V. Hogue, H. Bussey, B. Andrews, M. Tyers, and C. Boone. Systematic genetic analysis with ordered arrays of yeast deletion mutants. *Science*, 294(5550):2364–2368, 2001.

[85] L. H. Hartwell, J. J. Hopfield, S. Leibler, and A. W. Murray. From molecular to modular cell biology. *Nature*, 402(6761 Suppl):C47–C52, 1999.

[86] H. Jeong, B. Tombor, R. Albert, Z. N. Oltvai, and A.-L. Barabasi. The large-scale organization of metabolic networks. *Nature*, 407(6804):651–654, 2000.

[87] D.-S. Lee, K. A. Kay, N. A. Christakis, Z. N. Oltvai, and A.-L. Barabasi. The implications of human metabolic network topology for disease comorbidity. *Proceedings of The National Academy of Sciences*, 2008.

[88] M. Koyutrk, S. Subramaniam, and A. Grama. Introduction to network biology. In *Functional Coherence of Molecular Networks in Bioinformatics*, pages 1–13. Springer New York, 2012.

[89] T. I. Lee, N. J. Rinaldi, F. Robert, D. T. Odom, Z. Bar-Joseph, G. K. Gerber, N. M. Hannett, C. T. Harbison, C. M. Thompson, I. Simon, J. Zeitlinger, E. G. Jennings, H. L. Murray, D. B. Gordon, B. Ren, J. J. Wyrick, J.-B. Tagne, T. L. Volkert, E. Fraenkel, D. K. Gifford, and R. A. Young. Transcriptional regulatory networks in saccharomyces cerevisiae. *Science*, 298:799–804, 2002.

[90] F. Schacherer, C. Choi, U.e Götze, M.s Krull, S. Pistor, and E. Wingender. The transpath signal transduction database: a knowledge base on signal transduction networks. *Bioinformatics*, 17(11):1053–1057, 2001.

[91] T. Milenković, I. Filippis, M. Lappe, and N. Pržulj. Optimized null model for protein structure networks. *PLoS ONE*, 4(6):e5967, 2009.

[92] K. Goh, M. E. Cusick, D. Valle, B. Childs, M. Vidal, and A.-L. Barabási. The human disease network. *Proceedings of the National Academy of Sciences USA*, 104(21):8685–8690, 2007.

[93] V. Janjić and N. Pržulj. The core diseasome. *Molecular Biosystems*, 8(10):2614–2625, 2012.

[94] A. Ashworth, C. J Lord, and J. S. Reis-Filho. Genetic interactions in cancer progression and treatment. *Cell*, 145(1):30–38, 2011.

[95] S. B. Seidman. Network structure and minimum degree. *Social Networks*, 5(3):269–287, 1983.

[96] V. Batagelj and M. Zaversnik. An o(m) algorithm for cores decomposition of networks. *Symposium A Quarterly Journal In Modern Foreign Literatures*, cs.DS/0310(m):1–10, 2003.

[97] V. L. Roger, A. S. Go, D. M. Lloyd-Jones, E. J. Benjamin, J. D. Berry, W. B. Borden, D. M. Bravata, S. Dai, E. S. Ford, C. S. Fox, H. J. Fullerton, C. Gillespie, S. M. Hailpern, J. A. Heit, V. J. Howard, B. M. Kissela, S. J. Kittner, D. T. Lackland, J. H. Lichtman, L. D. Lisabeth, D. M. Makuc, G. M. Marcus, A. Marelli, D. B. Matchar, C. S. Moy, D. Mozaffarian, M. E. Mussolino, G. Nichol, N. P. Paynter, E. Z. Soliman, P. D. Sorlie, N. Sotoodehnia, T. N. Turan, S. S. Virani, N. D. Wong, D. Woo, and M. B. Turne. Heart disease and stroke statistics–2012 update: a report from the american heart association. *Circulation*, 125:e3–e5, 2012.

[98] A. Sarajlić and N. Pržulj. Survey of network-based approaches to research of cardiovascular diseases. *BioMed Research International*, 2014, 2014.

[99] A. Sarajlić, V. Gligorijević, D. Radak, and N. Pržulj. Network wiring of pleiotropic kinases yields insight into protective role of diabetes on aneurysm. *Integrative Biology*, 6(11):1049–1057, 2014.

[100] V. Latora, V. Nicosia, and P. Panzarasa. Social cohesion, structural holes, and a tale of two measures. *Journal of Statistical Physics*, 151(3-4):745–764, 2013.

[101] D. B. Mark, F. J. Van de Werf, R. J. Simes, H. D. White, L. Wallentin, R. M. Califf, and P. W. Armstrong. Cardiovascular disease on a global scale : defining the path forward for research and practice. *European Heart Journal*, 28(21):2678–2684, 2007.

[102] A. J. Lusis and J. N. Weiss. Cardiovascular networks systems-based approaches to cardiovascular disease. *Circulation*, 121(1):157–170, 2010.

[103] W. R. MacLellan, Y. Wang, and A. J. Lusis. Systems-based approaches to cardiovascular disease. *Nature Reviews Cardiology*, 9(3):172–184, 2012.

[104] F. J. Azuaje, F. E. Dewey, D. L. Brutsaert, Y. Devaux, E. A. Ashley, and D. R. Wagner. Systems-based approaches to cardiovascular biomarker discovery. *Circulation: Cardiovascular Genetics*, 5(3):360–367, 2012.

[105] S. Y. Chan, K. White, and J. Loscalzo. Deciphering the molecular basis of human cardiovascular disease through network biology. *Current opinion in cardiology*, 27(3):202–209, 2012.

[106] A. L. Barabasi, N. Gulbahce, and J. Loscalzo. Network medicine: a network-based approach to human disease. *Nature Reviews Genetics*, 12(1):56–68, 2011.

[107] T. Ideker and R. Sharan. Protein networks in disease. *Genome Research*, 18:644–652, 2008.

[108] R. Aragues, C. Sander, and B. Oliva. Predicting cancer involvement of genes from heterogeneous data. *BMC Bioinformatics*, 9(172):172, 2008.

[109] M. Krauthammer, C. A. Kaufmann, T. C. Gilliam, and A. Rzhetsky. Molecular triangulation: bridging linkage and molecular-network information for identifying candidate genes in alzheimer's disease. *Proceedings of the National Academy of Sciences of the United States of America*, 101(42):15148–15153, 2004.

[110] S. Fortunato. Community detection in graphs. *Physics Reports*, 486(3-5):75 – 174, 2010.

[111] S. E. Schaeffer. Graph clustering. *Computer Science Review*, 1(1):27 – 64, 2007.

[112] K. Mitra, A.R. Carvunis, S. K. Ramesh, and T. Ideker. Integrative approaches for finding modular structure in biological networks. *Nature Reviews Genetics*, 14(10):719–732, 2013.

[113] N. Bonifaci, A. Berenguer, J. Díez, O. Reina, I. Medina, J. Dopazo, V. Moreno, and M. A Pujana. Biological processes, properties and molecular wiring diagrams of candidate low-penetrance breast cancer susceptibility genes. *BMC medical genomics*, 1(1):62, 2008.

[114] L. M. Heiser, N. J. Wang, C. L. Talcott, K. R. Laderoute, M.l Knapp, Y. Guan, Z. Hu, S. Ziyad, B. L. Weber, S. Laquerre, J. R. Jackson, R. F. Wooster, W. L. Kuo, J. W. Gray, and P. T. Spellman. Integrated analysis of breast cancer cell lines reveals unique signaling pathways. *Genome Biology*, 10(3):R31, 2009.

[115] H.-Y. Chuang, E. Lee, Y.-T. Liu, D. Lee, and T. Ideker. Network-based classification of breast cancer metastasis. *Molecular Systems Biology*, 3(1):–, 2007.

[116] J. Ruan and W. Zhang. Identifying network communities with a high resolution. *Physical Review E*, 77(016104):1–12, 2008.

[117] M. Ray, J. Ruan, and W. Zhang. Variations in the transcriptome of alzheimer's disease reveal molecular networks involved in cardiovascular diseases. *Genome biology*, 9(10):R148+, 2008.

[118] S. Navlakha and C. Kingsford. The power of protein interaction networks for associating genes with diseases. *Bioinformatics*, 26(8):1057–1063, 2010.

[119] S. Köhler, S. Bauer, D. Horn, and P. N. Robinson. Walking the interactome for prioritization of candidate disease genes. *American journal of human genetics*, 82(4):949–958, 2008.

[120] J. Skogsberg, J. Lundstrom, A. Kovacs, R. Nilsson, P. Noori, S. Maleki, M. Kohler, A. Hamsten, J. Tegner, and J. Bjorkegren. Transcriptional profiling uncovers a network of cholesterol-responsive atherosclerosis target genes. *PLoS Genetics*, 4(3), 2008.

[121] E.A. Ashley, R. Ferrara, JY. King, A. Vailaya, A. Kuchinsky, X. He, B. Byers, U. Gerckens, S. Oblin, A. Tsalenko, A. Soito, JM. Spin, R. Tabibiazar, AJ. Connolly, JB. Simpson, E. Grube, and T. Quertermous. Network analysis of human in-stent restenosis. *Circulation*, 114(24):2644–2654, 2006.

[122] G. Jin, X. Zhou, H. Wang, H. Zhao, K. Cui, X. S. Zhang, L. Chen, S. L. Hazen, K. Li, and S. T. C. Wong. The knowledge-integrated network biomarkers discovery for major adverse cardiac events. *Journal of Proteome Research*, 7(9):4013–4021, 2008.

[123] J. Y. King, R. Ferrara, R. Tabibiazar, J. M. Spin, M. M. Chen, A. Kuchinsky, A. Vailaya, R. Kincaid, A. Tsalenko, D. X.-F. X. Deng, A. Connolly, P. Zhang,

E. Yang, C. Watt, Z. Yakhini, A. Ben-Dor, A. Adler, L. Bruhn, P. Tsao, T. Quertermous, and E. A. Ashley. Pathway analysis of coronary atherosclerosis. *Physiological genomics*, 23(1):103–118, 2005.

[124] S. A. Ramsey, E. S. Gold, and A. Aderem. A systems biology approach to understanding atherosclerosis. *EMBO Molecular Medicine*, 2(3):79–89, 2010.

[125] L. Zhang, X. Li, J. Tai, W. Li, and Chen L. Predicting candidate genes based on combined network topological features: A case study in coronary artery disease. *PLoS ONE*, 7(6), 2012.

[126] A. Camargo and F. Azuaje. Linking gene expression and functional network data in human heart failure. *PLoS ONE*, 2(12):e1347, 2007.

[127] A. Camargo and F. Azuaje. Identification of dilated cardiomyopathy signature genes through gene expression and network data integration. *Genomics*, 92:404–413, 2008.

[128] D. Diez, A. M. Wheelock, S. Goto, J. Z. Haeggstrom, G. Paulsson-Berne, G. K. Hansson, U. Hedin, A. Gabrielsen, and C. E. Wheelock. The use of network analyses for elucidating mechanisms in cardiovascular disease. *Mol. BioSyst.*, 6(2):289–304, 2010.

[129] W. Zhu, L. Yang, and Z. Du. Layered functional network analysis of gene expression in human heart failure. *PLoS One*, 4(7):e6288, 2009.

[130] D. Rende, N. Baysal, and B. Kirdar. A novel integrative network approach to understand the interplay between cardiovascular disease and other complex disorders. *Mol. BioSyst.*, 7:2205–2219, 2011.

[131] D. He, Z. P. Liu, and L. Chen. Identification of dysfunctional modules and disease genes in congenital heart disease by a network-based approach. *BMC Genomics*, 12(1):592+, 2011.

[132] F. E. Dewey, M. V. Perez, M. T. Wheeler, C. Watt, J. Spin, P. Langfelder, S. Horvath, S. Hannenhalli, T. P. Cappola, and E. A. Ashley. Gene coexpression network topology of cardiac development, hypertrophy, and failure. *Circulation: Cardiovascular Genetics*, 4(1):26–35, 2011.

[133] P. Du, G. Feng, J. Flatow, J. Song, M.e Holko, W. A. Kibbe, and S. M. Lin. From disease ontology to disease-ontology lite: statistical methods to adapt a

general-purpose ontology for the test of gene-ontology associations. *Bioinformatics*, 25(12):i63–i68, 2009.

[134] M. Maier, U. von Luxburg, and M. Hein. How the result of graph clustering methods depends on the construction of the graph. *ESAIM: Probability and Statistics*, 17:370–418, 2013.

[135] Y. Benjamini and Y. Hochberg. Controlling the false discovery rate: A practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society. Series B (Methodological)*, 57(1):pp. 289–300, 1995.

[136] A. D. King, N. Pržulj, and I. Jurisica. Protein complex prediction via cost-based clustering. *Bioinformatics*, 20(17):3013–3020, 2004.

[137] X. Ji, J. Tang, R. Halberg, D. Busam, S. Ferriera, M. M. O. Peña, C. Venkataramu, T. J. Yeatman, and S. Zhao. Distinguishing between cancer driver and passenger gene alteration candidates via cross-species comparison: a pilot study. *BMC Cancer*, 10:426, 2010.

[138] A. Youn and R. Simon. Identifying cancer driver genes in tumor genome sequencing studies. *Bioinformatics*, 27(2):175–181, 2011.

[139] U. D. Akavia, O. Litvin, J. Kim, F. Sanchez-Garcia, D. Kotliar, H. C. Causton, P. Pochanard, E. Mozes, L. A. Garraway, and D. Pe'er. An integrated approach to uncover drivers of cancer. *Cell*, 143(6):1005–1017, 2010.

[140] I. Ahrens, G.Y.H. Lip, K. Peter, et al. New oral anticoagulant drugs in cardiovascular disease. *Thrombosis & Haemostasis*, 104(1):49, 2010.

[141] M. Burnier, H.R. Brunner, et al. Angiotensin ii receptor antagonists. *Lancet*, 355(9204):637, 2000.

[142] H. Ju, T. Scammel-La Fleur, I.M. Dixon, et al. Altered mrna abundance of calcium transport genes in cardiac myocytes induced by angiotensin ii. *Journal of molecular and cellular cardiology*, 28(5):1119, 1996.

[143] M.P. Gabay and R. Jain. Role of antibiotics for the prevention of cardiovascular disease. *The Annals of Pharmacotherapy*, 36(10):1629–1636, 2002.

[144] V. Memišević, T. Milenković, and N. Pržulj. Complementarity of network and sequence structure in homologous proteins. *Journal of Integrative Bioinformatics*, 7(3):135, 2010.

[145] R. Sharan, I. Ulitsky, and R. Shamir. Network-based prediction of protein function. *Molecular Systems Biology*, 3(1):88, 2007.

[146] S. Shantikumar, R. Ajjan, K.E. Porter, and D.J.A. Scott. Diabetes and the abdominal aortic aneurysm. *European Journal of Vascular and Endovascular Surgery*, 39(2):200 – 207, 2010.

[147] S. K. Prakash, C. Pedroza, Y. A. Khalil, and D. M. Milewicz. Diabetes and reduced risk for thoracic aortic aneurysms and dissections: A nationwide case-control study. *Journal of the American Heart Association*, 1(2), 2012.

[148] P. De Rango, P. Cao, E. Cieri, G. Parlani, M. Lenti, G. Simonte, and F. Verzini. Effects of diabetes on small aortic aneurysms under surveillance according to a subgroup analysis from a randomized trial. *Journal of Vascular Surgery*, 56(6):1555 – 1563, 2012.

[149] S. R. Preis, S.-J. Hwang, S. Coady, M. J. Pencina, R. B. D'Agostino, P. J. Savage, D. Levy, and Ca. S. Fox. Trends in all-cause and cardiovascular disease mortality among women and men with and without diabetes mellitus in the framingham heart study, 1950 to 2005. *Circulation*, 119(13):1728–1735, 2009.

[150] M.A. Creager, T.F. Lüscher, F. Cosentino, and J.A. Beckman. Diabetes and vascular disease: pathophysiology, clinical consequences, and medical therapy: Part I. *Circulation*, 108(12):1527–1532, 2003.

[151] M.I. Patel, D.T. Hardman, C.M. Fisher, and M. Appleberg. Current views on the pathogenesis of abdominal aortic aneurysms. *J Am Coll Surg.*, 181(4):371–82, 1995.

[152] M. Žitnik, V. Janjić, C. Larminie, B. Zupan, and N. Pržulj. Discovering disease-disease associations by fusing systems-level molecular data. *Scientific Report*, 3:3202, 2013.

[153] M. C. Bassik, M. Kampmann, R. J. J. Lebbink, S.i Wang, M. Y. Hein, I. Poser, J. Weibezahn, M. A. Horlbeck, S. Chen, M. Mann, A. A. Hyman, E. M. Leproust, M. T. McManus, and J. S. Weissman. A systematic mammalian genetic interaction map reveals pathways underlying ricin susceptibility. *Cell*, 152(4):909–922, 2013.

[154] Ben Lehner. Molecular mechanisms of epistasis within and between genes. *Trends in Genetics*, 27(8):323–331, 2011.

[155] W. Hayes, K. Sun, and N. Przulj. Graphlet-based measures are suitable for biological network comparison. *Bioinformatics*, 29(4):483–491, 2013.

[156] A. Sarajlić, V. Filipović, A.and Janjić, R.C. Coombes, and N. Pržulj. The role of genes co-amplified with nicastrin in breast invasive carcinoma. *Breast Cancer Res Treat*, 143(2):393–401, 2014.

[157] A. Chatr-Aryamontri, B.-J. Breitkreutz, S. Heinicke, L. Boucher, A. G. Winter, C. Stark, J. Nixon, L. Ramage, N. Kolas, L. O'Donnell, T. Reguly, A. Breitkreutz, A. Sellam, D. Chen, C. Chang, J. M. Rust, M. S. Livstone, R. Oughtred, K. Dolinski, and M. Tyers. The biogrid interaction database: 2013 update. *Nucleic Acids Research*, 41(Database-Issue):816–823, 2013.

[158] M. Kanehisa, S. Goto, Y. Sato, M. Furumichi, and M. Tanabe. Kegg for integration and interpretation of large-scale molecular data sets. *Nucleic Acids Research*, 40(D1):D109–D114, 2012.

[159] E. K. Kim and E.-J. Choi. Pathological roles of {MAPK} signaling pathways in human diseases. *Biochimica et Biophysica Acta (BBA) - Molecular Basis of Disease*, 1802(4):396 – 405, 2010.

[160] M. Penrose. *Random Geometric Graphs (Oxford Studies in Probability)*. Oxford University Press, USA, 2003.

[161] D. W. Huang, B. T. Sherman, and R. A. Lempicki. Systematic and integrative analysis of large gene lists using DAVID bioinformatics resources. *Nature Protocols*, 4(1):44–57, 2008.

[162] D. W. Huang, B. T. Sherman, and R. A. Lempicki. Bioinformatics enrichment tools: paths toward the comprehensive functional analysis of large gene lists. *Nucleic Acids Research*, 37(1):1–13, 2009.

[163] A.L. Berrier and K.M. Yamada. Cell-matrix adhesion. *Journal of Cellular Physiology*, 213(3):565–73, 2007.

[164] P.E. Norman, T.M.E. Davis, M.T.Q. Le, and J. Golledge. Matrix biology of abdominal aortic aneurysms in diabetes: mechanisms underlying the negative association. *Connective Tissue Research*, 48(3):125–31, 2007.

[165] F. W. Stearns. One hundred years of pleiotropy: a retrospective. *Genetics*, 186(3):767–773, 2010.

[166] J. Van Wauwe and B. Haefner. Glycogen synthase kinase-3 as drug target: from wallflower to center of attention. *Drug News and Perspectives*, 16(9):557–65, 2003.

[167] S. Pavey, P. Johansson, L. Packer, J. Taylor, M. Stark, P. M. Pollock, G. J. Walker, G. M. Boyle, U. Harper, S.-J. Cozzi, K. Hansen, L. Yudt, C. Schmidt, P. Hersey, K. A. O. Ellem, M. G. E. O Rourke, P. G. Parsons, P. Meltzer, M. Ringner, and N. K. Hayward. Microarray expression profiling in melanoma reveals a braf mutation signature. *Oncogene*, 23(23):4060–7, 2004.

[168] D. Bond and E. Foley. A quantitative RNAi screen for JNK modifiers identifies Pvr as a novel regulator of Drosophila immune signaling. *PLoS Pathogens*, 5(11), 2009.

[169] Y. Wang, C.C. Barbacioru, D. Shiffman, S. Balasubramanian, O. Iakoubova, M. Tranquilli, G. Albornoz, J. Blake, N.N. Mehmet, D. Ngadimo, K. Poulter, F. Chan, R.R. Samaha, and J. A. Elefteriades. Gene expression signature in peripheral blood detects thoracic aortic aneurysm. *PLoS ONE*, 2(10), 2007.

[170] B. Bakir-Gungor and O. U.r Sezerman. The identification of pathway markers in intracranial aneurysm using genome-wide association data from two different populations. *PLoS One*, 8(3):e57022, 2013.

[171] N. Kashtan, S. Itzkovitz, R. Milo, and U. Alon. Efficient sampling algorithm for estimating subgraph concentrations and detecting network motifs. *Bioinformatics*, 20:2004, 2004.

[172] S. Wernicke. Efficient detection of network motifs. *Computational Biology and Bioinformatics, IEEE/ACM Transactions on*, 3(4):347–359, 2006.

[173] A. Stoica and Christophe P. 0002. Structure of neighborhoods in a large social network. In *CSE (4)*, pages 26–33. IEEE Computer Society, 2009.

[174] B. Betkaoui, D. B. Thomas, W. Luk, and N. Przulj. A framework for fpga acceleration of large graph problems: graphlet counting case study. In *Field-Programmable Technology (FPT), 2011 International Conference*, pages 1–8. IEEE, 2011.

[175] N. Przulj, D.G. Corneil, and I. Jurisica. Efficient estimation of graphlet frequency distributions in protein-protein interaction networks. *Bioinformatics*, 22:974–980, 2006.

[176] T. Hočevar and J. Demšar. A combinatorial approach to graphlet counting. *Bioinformatics*, 30(4):559–565, 2014.

[177] A. P. Bradley. The use of the area under the ROC curve in the evaluation of machine learning algorithms. *Pattern Recognition*, 30(7):1145–1159, 1997.

[178] Y.-K. Yu, E. M. Gertz, R. Agarwala, A. A. Schffer, and S. F. Altschul. Retrieval accuracy, statistical significance and compositional similarity in protein sequence database searches. *Nucleic Acids Research*, 34(20):5966–5973, 2006.

[179] J. Davis and M. Goadrich. The relationship between precision-recall and roc curves. In *Proceedings of the 23rd international conference on Machine learning*, ICML '06, pages 233–240. ACM, 2006.

[180] B. Bollobas, C. Borgs, J. T. Chayes, and O. Riordan. Directed scale-free graphs. In *SODA*, pages 132–139. ACM/SIAM, 2003.

[181] A. Vázquez, A. Flammini, A. Maritan, and A. Vespignani. Modeling of protein interaction networks. *Complexus*, 1:38–44, 2003.

[182] J. F. Hair, R. L. Tatham, R. E. Anderson, and W. Black. *Multivariate Data Analysis (5th Edition)*. Prentice Hall, 5th edition, 1998.

[183] M. Nei and S. Kumar. *Molecular Evolution and Phylogenetics*. Oxford University Press, USA, 1 edition, 2000.

[184] C. Luo, S. T. Walk, D. M. Gordon, M. Feldgarden, J. M. Tiedje, and K. T. Konstantinidis. Genome sequencing of environmental Escherichia coli expands understanding of the ecology and speciation of the model bacterial species. *Proceedings of the National Academy of Sciences USA*, 108:7200–7205, 2011.

[185] L.M. Rodriguez-R, A. Grajales, M.L. Arrieta-Ortiz, C. Salazar, S. Restrepo, and A. Bernal. Genomes-based phylogeny of the genus xanthomonas. *BMC Microbiology*, 12(1), 2012.

[186] A. Stamatakis. Raxml-vi-hpc: maximum likelihood-based phylogenetic analyses with thousands of taxa and mixed models. *Bioinformatics*, 22(21):2688–2690, 2006.

[187] S. Guindon, JF Dufayard, V. Lefort, M. Anisimova, W. Hordijk, and O. Gascuel. New algorithms and methods to estimate maximum-likelihood phylogenies: assessing the performance of phyml 3.0. *Systematic Biology*, 59(3):307–321, 2010.

[188] N. Lartillot, T. Lepage, and S. Blanquart. Phylobayes 3: a bayesian software package for phylogenetic reconstruction and molecular dating. *Bioinformatics*, 25(17):2286–2288, 2009.

[189] J. P. Huelsenbeck and F. Ronquist. Mrbayes: Bayesian inference of phylogenetic trees. *Bioinformatics*, 17(8):754–755, 2001.

[190] Z. Yang and B. Rannala. Molecular phylogenetics: principles and practice. *Nature Reviews Genetics*, 13(5):303–314, 2012.

[191] Y. Zhang, S. Li, G. Skogerbo, Z. Zhang, X. Zhu, Z. Zhang, S. Sun, H. Lu, B. Shi, and R. Chen. Phylophenetic properties of metabolic pathway topologies as revealed by global analysis. *BMC Bioinformatics*, 7(1):252+, 2006.

[192] M. Heymans and A. K. Singh. Deriving phylogenetic trees from the similarity analysis of metabolic pathways. In *ISMB (Supplement of Bioinformatics)*, pages 138–146, 2003.

[193] D. Gamermann, A. Montagud, J.A. Conejero, J.F. Urchueguia, and P.F. de Cordoba. New approach for phylogenetic tree recovery based on genome-scale metabolic networks. *Journal of Computational Biology*, 21(7):508–519, 2014.

[194] O. Kuchaiev, T. Milenković, V. Memišević, W. Hayes, and N. Pržulj. Topological network alignment uncovers biological function and phylogeny. *Journal of the Royal Society Interface*, 7:1341–1354, 2010.

[195] V. Makarenkov. T-rex: reconstructing and visualizing phylogenetic trees and reticulation networks. *Bioinformatics*, 17(7):664–668, 2001.

[196] F. Alkan and C. Erten. Beams: Backbone extraction and merge strategy for the global many-to-many alignment of multiple ppi networks. *Bioinformatics*, 2013.

[197] C.-S. Liao, K. Lu, M. Baym, Rohit Singh, and B. Berger. Isorankn: spectral methods for global alignment of multiple protein networks. *Bioinformatics*, 25(12), 2009.

[198] R. Singh, J. Xu, and B. Berger. Global alignment of multiple protein interaction networks with application to functional orthology detection. *Proceedings of the National Academy of Sciences USA*, 105(35):12763–12768, 2008.

[199] I. C. Lerman. Foundations of the likelihood linkage analysis (lla) classification method. *Applied Stochastic Models and Data Analysis*, 7(1):63–76, 1991.

[200] C. Clark and J. Kalita. A comparison of algorithms for the pairwise alignment of biological networks. *Bioinformatics*, 30(16):2351–2359, 2014.

[201] D. Davis, Ö.N. Yaverŏglu, N. Malod-Dognin, A. Stojmirović, and N. Pržulj. Topology-function conservation in protein-protein interaction networks. *Bioinformatics*, 2015.

[202] H. Hotelling. Relations between two sets of variates. *Biometrika*, 28(3/4):321–377, 1936.

[203] A. Albert. *Regression and the Moore-Penrose Pseudoinverse*. Elsevier, 1972.

[204] M.A. Pisarev and D.L. Pisarev. Action of cyclic nucleotides on protein and rna synthesis in the thyroid. *Acta Endocrinologica*, 84(2):297–302, 1977.

[205] C.W. Abell and T.M. Monahan. The role of adenosine 3',5'-cyclic monophosphate in the regulation of mammalian cell division. *The Journal of Cell Biology*, 59(3):549–58, 1973.

[206] M. Rincón, A. Tugores, and M. López-Botet. Cyclic amp and calcium regulate at a transcriptional level the expression of the cd7 leukocyte differentiation antigen. *The Journal of Cell Biology*, 267(25):18026–31, 1992.

[207] J.W. Yang, M.R. Kim, H.G. Kim, S.K. Kim, H.G. Jeong, and K.W. Kang. Differential regulation of erbb2 expression by camp-dependent protein kinase in tamoxifen-resistant breast cancer cells. *Archives of Pharmacal Research*, 31(3):350–6, 2008.

[208] T. Milenković, J. Lai, and N. Pržulj. Graphcrunch: a tool for large network analyses. *BMC Bioinformatics*, 9(70), 2008.

[209] O. Kuchaiev, A. Stevanovic, W. Hayes, and N. Pržulj. GraphCrunch 2: Software tool for network modeling, alignment and clustering. *BMC Bioinformatics*, 12, 2011.

[210] T Milenkovic, I Filippis, M Lappe, and N Przulj. Optimized null model for protein structure networks. *PLOS ONE*, 4, 2009.

[211] S. A. Cook. The complexity of theorem-proving procedures. In *STOC*, pages 151–158. ACM, 1971.

[212] L. Chindelevitch, C.-S. Liao, and B. Berger. Local optimization for global alignment of protein interaction networks. In *Pacific Symposium on Biocomputing*, pages 123–132, 2010.

[213] R. Singh, J. Xu, and B. Berger. Pairwise global alignment of protein interaction networks by matching neighborhood topology. In *RECOMB*, volume 4453 of *Lecture Notes in Computer Science*, pages 16–31. Springer, 2007.

[214] G. Klau. A new graph-based method for pairwise global network alignment. *BMC Bioinformatics*, 10(Suppl 1):S59, 2009.

[215] R. Patro and C. Kingsford. Global network alignment using multiscale spectral signatures. *Bioinformatics*, 28(23):3105–3114, 2012.

[216] T. Milenković, W. Leong Ng, W. Hayes, and N. Pržulj. Optimal network alignment with graphlet degree vectors. *Cancer Informatics*, 9:121–137, 2010.

[217] O. Kuchaiev and N. Przulj. Integrative network alignment reveals large regions of global network similarity in yeast and human. *Bioinformatics*, 27(10):1390–1396, 2011.

[218] V. Memišević and N. Pržulj. C-graal: Common-neighbors-based global graph alignment of biological networks. *Integrative Biology*, page 10, 2012.

[219] N Malod-Dognin and N Prulj. L-graal: Lagrangian graphlet-based network aligner. *Bioinformatics*, 31:2182–2189, 2015.

[220] M. Held and R. M. Karp. The traveling-salesman problem and minimum spanning trees: Part II. *Mathematical Programming*, 1(1):6–25, 1971.

[221] V. Gerzanich, S. Ivanova, and J. M. Simard. Early pathophysiological changes in cerebral vessels predisposing to stroke. *Clinical Hemorheology and Microcirculation*, 29(3-4):291–294, 2003.

[222] E. J. Birks, N. Latif, K. Enesa, T. Folkvang, L. A. Luong, P. Sarathchandra, M. Khan, H. Ovaa, C. M. Terracciano, P. J. R. Barton, M. H. Yacoub, and P. C. Evans. Elevated p53 expression is associated with dysregulation of the ubiquitin-proteasome system in dilated cardiomyopathy. *Cardiovasc Research*, 79(3):472–480, 2008.

[223] Q. Zhang, X. He, L. Chen, C. Zhang, X. Gao, Z. Yang, and G. Liu. Synergistic regulation of p53 by mdm2 and mdm4 is critical in cardiac endocardial cushion morphogenesis during heart development. *The Journal of Pathology*, 228(3):416–428, 2012.

[224] K.-C. Chen, Y.-C. Liao, I-C. Hsieh, Y.-S. Wang, C.-Y. Hu, and S.-H. H. Juo. Oxldl causes both epigenetic modification and signaling regulation on the microrna-29b gene: novel mechanisms for cardiovascular diseases. *Journal of Molecular and Cellular Cardiology*, 52(3):587–595, 2012.

[225] I. M. B. H. van de Laar, D. van der Linde, E. H. G. Oei, P. K. Bos, J. H. Bessems, S. M. Bierma-Zeinstra, B. L. van Meer, G. Pals, R. A. Oldenburg, J. A. Bekkers, A. Moelker, B. M. de Graaf, G. Matyas, I. M. E. Frohn-Mulder, J. Timmermans, Y. Hilhorst-Hofstee, J. M. Cobben, H. T. Bruggenwirth, L. van Laer, B. Loeys, J. De Backer, P. J. Coucke, H. C. Dietz, P. J. Willems, B. A. Oostra, A. De Paepe, J. W. Roos-Hesselink, A. M. Bertoli-Avella, and M. W. Wessels. Phenotypic spectrum of the smad3-related aneurysms-osteoarthritis syndrome. *Journal of Medical Genetics*, 49(1):47–57, 2012.

[226] D. van der Linde, I. M. B. H. van de Laar, A. M. Bertoli-Avella, R. A. Oldenburg, J. A. Bekkers, F. U. S. Mattace-Raso, A. H. van den Meiracker, A. Moelker, F. van Kooten, I. M. E. Frohn-Mulder, J. Timmermans, E. Moltzer, J. M. Cobben, L. van Laer, B. Loeys, J. De Backer, P. J. Coucke, A. De Paepe, Y. Hilhorst-Hofstee, M. W. Wessels, and J. W. Roos-Hesselink. Aggressive cardiovascular phenotype of aneurysms-osteoarthritis syndrome caused by pathogenic smad3 variants. *Journal of the American College of Cardiology*, 60(5):397–403, 2012.

[227] D. Gomez, A. Coyet, V. Ollivier, X. Jeunemaitre, G. Jondeau, J.-B. Michel, and R. Vranckx. Epigenetic control of vascular smooth muscle cells in marfan and non-marfan thoracic aortic aneurysms. *Cardiovascular Research*, 89(2):446–456, 2011.

[228] Johannes L. Bjørnstad, B. Skrbic, H. S. Marstein, A. Hasic, I. Sjaastad, W. E. Louch, G. Florholmen, G. Christensen, and T. Tønnessen. Inhibition of smad2 phosphorylation preserves cardiac function during pressure overload. *Cardiovascular Research*, 93(1):100–110, 2012.

[229] C. Chang, C. Zhang, X. Zhao, X. Kuang, H. Tang, and X. Xiao. Differential regulation of mitogen-activated protein kinase signaling pathways in human with different types of mitral valvular disease. *Journal of Surgical Research*, 2012.

[230] C. A. Souders, S. L. K. Bowers, I. Banerjee, J. W. Fuseler, J. L. Demieville, and T. A. Baudino. c-myc is required for proper coronary vascular formation via cell- and gene-specific signaling. *Arteriosclerosis, Thrombosis, and Vascular Biology*, 32(5):1308–1319, 2012.

[231] J. Min, M. Reznichenko, R. H. Poythress, C. M. Gallant, S. Vetterkind, Y. Li, and K. G. Morgan. Src modulates contractile vascular smooth muscle function via regulation of focal adhesions. *Journal of Cellular Physiology*, 227(11):3585–3592, 2012.

[232] P. C. Shukla, K. K. Singh, A. Quan, M. Al-Omran, H. Teoh, F. Lovren, L. Cao, I. I. Rovira, Y. Pan, C. Brezden-Masley, B. Yanagawa, A. Gupta, C.-X. Deng, J. G. Coles, H. Leong-Poi, W. L. Stanford, T. G. Parker, M. D. Schneider, T. Finkel, and S. Verma. Brca1 is an essential regulator of heart function and survival following myocardial infarction. *Nature Communications*, 2:593, 2011.

[233] S. M. Haldar, Y. Lu, D. Jeyaraj, D. Kawanami, Y. Cui, S. J. Eapen, C. Hao, Y. Li, Y.-Q. Doughman, M. Watanabe, K. Shimizu, H. Kuivaniemi, J. Sadoshima, K. B. Margulies, T. P. Cappola, and M. K. Jain. Klf15 deficiency is a molecular link between heart failure and aortic aneurysm formation. *Science Translational Medicine*, 2(26):26ra26, 2010.

[234] E. Reiling, V. Lyssenko, J. M. A. Boer, S. Imholz, W. M. M. Verschuren, B. Isomaa, T. Tuomi, L. Groop, and M. E. T. Dollé. Codon 72 polymorphism (rs1042522) of tp53 is associated with changes in diastolic blood pressure over time. *European Journal of Human Genetics*, 20(6):696–700, 2012.

[235] A. M. Zawada, K. S. Rogacev, B. Hummel, O. S. Grün, A. Friedrich, B. Rotter, P. Winter, J. Geisel, D. Fliser, and G. H. Heine. Supertag methylation-specific digital karyotyping reveals uremia-induced epigenetic dysregulation of atherosclerosis-related genes. *Circulation Cardiovascular Genetics*, 5(6):611–620, 2012.

[236] S. Zhang, C. Weinheimer, M. Courtois, A. Kovacs, C. E. Zhang, A. M. Cheng, Y. Wang, and A. J. Muslin. The role of the grb2-p38 mapk signaling pathway in

cardiac hypertrophy and fibrosis. *Journal of Clinical Investigation*, 111(6):833–841, 2003.

Appendices

A Appendix to Chapter 2

A.1 Literature Validations of Predicted CVD Genes

CREBBP gene is mentioned in connection with pathophysiological changes in cerebral vessels predisposing to stroke [221]. Gerzanich et al. [221] study three models of human conditions associated with stroke: chronic angiotensin II-hypertension, chronic nicotine administration and oxidative endothelial injury. All three models show significant up-regulation of expression of proliferative cell nuclear antigen (PCNA) in arterioles in situ, which is associated with increased activation of the nuclear transcription factor, phospho-cAMP response element binding protein (phospho-CREB).

It is shown that dilated cardiomyopathy tissues contain elevated levels of p53 and its regulators MDM2 and HAUSP (p-value$\leq$ 0.01) compared to non-failing hearts [222]. Also, regulation of MDM2 is critical in cardiac endocardial cushion morphogenesis during heart development [223]. Chen et al. [224] show that down-regulation of HDAC1 gene and the modifications on histone 3 lysine 4 (H3K4) and H3K9 significantly affect microRNA-29b expression in the context of signaling regulation of microRNA-29b, which is connected to novel mechanisms for cardiovascular diseases.

Aneurysms-osteoarthritis syndrome (AOS) is a newly discovered autosomal dominant syndromic form of thoracic aortic aneurysms and dissections, that is characterised by the presence of arterial aneurysms and tortuosity, mild craniofacial, skeletal and cutaneous anomalies, and early-onset osteoarthritis. AOS is caused by mutations in the SMAD3 gene [225]. It is known that aggressive cardiovascular phenotype of aneurysms-osteoarthritis syndrome is caused by pathogenic SMAD3 variants [226]. Also, SMAD2 dysregulation is associated with thoracic aortic aneurysms [227]. Inhibition of SMAD2 phosphorylation preserves cardiac function during pressure overload [228].

JUN gene is linked to different types of mitral valvular disease (MVD), including mitral regurgitation (MR) and mitral stenosis (MS) [229]. It is shown that c-Jun mRNA are significantly upregulated in patients with MS compared with those with MR (with p-value ≤ 0.05) and that phosphorylated c-Jun N-terminal kinase in the MR group of patients is significantly greater than that in the MS group (with p-value ≤ 0.001).

It is demonstrated that proper expression of MYC in cardiac fibroblasts and myocytes is essential to cardiac angiogenesis, therefore MYC is required for proper coronary vascular formation [230]. It is shown that SRC protein regulates focal adhesion protein function, which influences contractility of vascular smooth muscle [231]. This also points to novel therapeutic approaches to CVDs, in terms of targeting SRC protein [231]. BRCA1 is an essential regulator of heart fuction [232]. BRCA1 and MYC are also driver genes [94](see Figure 2.4).

Inhibition of EP300 can neutralize deficiency of KLF15 which is shown to be a molecular link between heart failure and aortic aneurysm formation [233].

It is known that TP53 is involved in cardiovascular functioning [234]. TP53 is also mentioned as one of the candidate genes associated with proatherogenic and inflammatory processes in chronic kidney disease (CKD) [235]. Zawada et al. aimed to point to new therapeutic strategies in CKD-associated atherosclerotic disease [235].

It is shown that GRB2 plays a role in the signaling pathway for cardiac hypertrophy and fibrosis [236]

B Appendix to Chapter 3

B.1 The Source Code for the Directed Graphlets and Orbits Counter

The source code for directed graphlet and orbit counter is listed below (implemented in C++ by Anida Sarajlić in June 2014). Input is a text file *input_file_name* representing a network edgelist (entry "A B" in a row represents a directed edge from node A to node B). Output files are:

- *input_file_name.signatures.txt* - each line represents a DGDV of a node (for the node names refer to the second column of *input_file_name.dictionary.txt*). Number of lines corresponds to number of nodes in the network minus the nodes that were only involved in selfloops.
- *input_file_name.dictionary.txt* - the first column corresponds to the line number from the *input_file_name.signatures.txt* (numbered from 0) and the second column is the name of the node.
- *input_file_name.graphletcounts.txt* - the first column denotes the graphlet (numbered from 0 to 39 for the 40 directed graphlets) and the second column represents the number of such graphlets in the network.

```
#include <fstream>
#include <iostream>
#include <string>
#include <sstream>
#include <map>
#include <vector>
#include <set>
#include <algorithm>
```

```
using namespace std;
typedef long long int64;

int64 position_in_vector(string a, vector<string> vektor)
{
  int64 pozicija(0);
  for (vector<string>::iterator it = vektor.begin(); it !=
     ↪ vektor.end(); ++it) { if ((*it)==a) {pozicija =
     ↪ distance(vektor.begin(), it);}}
  return pozicija;
}

template <typename T>
int64 is_it_in_vector(T a, vector<T> vektor)
{
        int64 response(0);
if(find(vektor.begin(), vektor.end(), a) != vektor.end()) {
    response = 1;
}
    return response;
}

//Main program

int main ( int argc, char *argv[] ){

    vector<pair<int64, int64> > edgelista;
    vector<string> dictionary;
    string x1, x2, line;
    vector<vector<int64> > pred, succ, signatures;
    vector<int64> graphlets;

//Building the edgelist (ignoring self-loops). Nodes are
   ↪ encoded with numbers from 0 to (number of nodes)-1.
   ↪ Building a vector of nodenames.
```

```
    if (argc != 2)
        cout<<"Usage: "<< argv[0] <<" <in_filename> "<<endl;
    else{
        ifstream data_file(argv[1]);

        if (!data_file.is_open())
            cout<<"Could not open file\n"<<endl;
        else{
            while(getline(data_file,line)){
                stringstream string_linije(line);
                string_linije >> x1 >> x2;
                if (x1 != x2){
                        if (is_it_in_vector(string(x1),
                            ↪ dictionary)==0) {dictionary.
                            ↪ push_back(x1);}
                        if (is_it_in_vector(string(x2),
                            ↪ dictionary)==0) {dictionary.
                            ↪ push_back(x2);}
                        edgelista.push_back(make_pair(
                            ↪ position_in_vector(x1,dictionary)
                            ↪ ,position_in_vector(x2,dictionary
                            ↪ )));
                }
            }
        }
        data_file.close();
    }

//Building a container (vector of vectors) of nodes'
   ↪ predecessors and a container of nodes' succesors - pred
   ↪ and succ. Multiple edges are ignored.
//Building a container of signature vectors - signatures.
   ↪ Building a vector of graphlet counts - graphlets.

    pred.resize(dictionary.size());
    succ.resize(dictionary.size());
```

```
    signatures.resize(dictionary.size());
    graphlets.resize(40);
    for (vector<vector<int64> >::iterator it = signatures.begin
       ↪ (); it !=signatures.end(); ++it) {
        (*it).resize(129);
    }
    for (vector<pair<int64,int64> >::iterator it = edgelista.
       ↪ begin(); it != edgelista.end(); ++it){
        if (is_it_in_vector((*it).second,succ[(*it).first]) ==
           ↪ 0) {succ[(*it).first].push_back((*it).second);}
        if (is_it_in_vector((*it).first,pred[(*it).second]) ==
           ↪ 0) {pred[(*it).second].push_back((*it).first);}
    }

cout<<"pred and succ matrixes over"<<endl;

//UPDATING NUMBER OF ORBITS AND GRAPHLETS

for (int64 i(0); i<dictionary.size(); ++i){

//orb 0,1
signatures[i][0]=succ[i].size();
for (vector<int64>::iterator it = succ[i].begin(); it !=succ[i
   ↪ ].end(); ++it) {
        ++signatures[*it][1];
        ++graphlets[0];
    }

//orb 2,3,4
for (vector<int64>::iterator it1 = pred[i].begin(); it1 !=pred
   ↪ [i].end(); ++it1) {
        for (vector<int64>::iterator it2 = succ[i].begin();
           ↪ it2 !=succ[i].end(); ++it2) {
            if ((*it1)!=(*it2) and (is_it_in_vector(*it1,
               ↪ succ[*it2]) == 0) and (is_it_in_vector(*
               ↪ it1,pred[*it2]) == 0)) {
```

```
++signatures[*it1][2];
++signatures[*it2][4];
++signatures[i][3];
++graphlets[1];

//orb 13,14,15,16 and orb 53,54,55,56
   ↪ and orb 39 and orb 81,82,83,84
   ↪ adn orb 40,41,42,43 and orb
   ↪ 105,106,107,108 and orb
   ↪ 109,110,111,112
for (vector<int64>::iterator it3 =
   ↪ succ[*it2].begin(); it3 !=succ[*
   ↪ it2].end(); ++it3) {
        if ((*it3)!=(*it1) and (*it3)
           ↪ !=(i) and (
           ↪ is_it_in_vector(*it3,succ
           ↪ [*it1]) == 0) and (
           ↪ is_it_in_vector(*it3,pred
           ↪ [*it1]) == 0) and (
           ↪ is_it_in_vector(*it3,succ
           ↪ [i]) == 0) and (
           ↪ is_it_in_vector(*it3,pred
           ↪ [i]) == 0)) {
                ++signatures[*it1][13];
                ++signatures[*it2][15];
                ++signatures[i][14];
                ++signatures[*it3][16];
                ++graphlets[6];
        }
        if ((*it3)!=(*it1) and (*it3)
           ↪ !=(i) and (
           ↪ is_it_in_vector(*it3,succ
           ↪ [*it1]) == 0) and (
           ↪ is_it_in_vector(*it3,pred
           ↪ [*it1]) == 0)  and (
           ↪ is_it_in_vector(*it3,pred
```

```
    ↪ [i]) == 1)) {
          ++signatures[*it1][56];
          ++signatures[*it2][53];
          ++signatures[i][55];
          ++signatures[*it3][54];
          ++graphlets[19];
    }
    if ((*it3)!=(*it1) and (*it3)
    ↪ !=(i) and (
    ↪ is_it_in_vector(*it3,pred
    ↪ [*it1]) == 1) and (
    ↪ is_it_in_vector(*it3,succ
    ↪ [i]) == 0) and (
    ↪ is_it_in_vector(*it3,pred
    ↪ [i]) == 0)) {
          ++signatures[*it1][39];
          ++signatures[*it2][39];
          ++signatures[i][39];
          ++signatures[*it3][39];
          ++graphlets[14];
    }
    if ((*it3)!=(*it1) and (*it3)
    ↪ !=(i) and (
    ↪ is_it_in_vector(*it3,pred
    ↪ [*it1]) == 1) and (
    ↪ is_it_in_vector(*it3,pred
    ↪ [i]) == 1)) {
          ++signatures[*it1][82];
          ++signatures[*it2][83];
          ++signatures[i][81];
          ++signatures[*it3][84];
          ++graphlets[26];
    }
    if ((*it3)!=(*it1) and (*it3)
    ↪ !=(i) and (
    ↪ is_it_in_vector(*it3,succ
```

```
    ↪ [*it1]) == 1) and (
    ↪ is_it_in_vector(*it3,succ
    ↪ [i]) == 0) and (
    ↪ is_it_in_vector(*it3,pred
    ↪ [i]) == 0)) {
        ++signatures[*it1][42];
        ++signatures[*it2][41];
        ++signatures[i][40];
        ++signatures[*it3][43];
        ++graphlets[15];
}
if ((*it3)!=(*it1) and (*it3)
    ↪ !=(i) and (
    ↪ is_it_in_vector(*it3,succ
    ↪ [*it1]) == 1) and (
    ↪ is_it_in_vector(*it3,succ
    ↪ [i]) == 1)) {
        ++signatures[*it1
            ↪ ][107];
        ++signatures[*it2
            ↪ ][106];
        ++signatures[i][108];
        ++signatures[*it3
            ↪ ][105];
        ++graphlets[33];
}
if ((*it3)!=(*it1) and (*it3)
    ↪ !=(i) and (
    ↪ is_it_in_vector(*it3,succ
    ↪ [*it1]) == 1)  and (
    ↪ is_it_in_vector(*it3,pred
    ↪ [i]) == 1)) {
        ++signatures[*it1
            ↪ ][111];
        ++signatures[*it2
            ↪ ][110];
```

```
                    ++signatures[i][112];
                    ++signatures[*it3
                        ↪ ][109];
                    ++graphlets[34];
                }
            }

            //orb 17,18,19,20 and orb 77,78,79,80
                ↪ and orb 69,70,71,72 and
                ↪ 113,114,115,116
            for (vector<int64> ::iterator it4 =
                ↪ pred[*it2].begin(); it4 !=pred[*
                ↪ it2].end(); ++it4) {
                    if ((*it4)!=(*it1) and (*it4)
                        ↪ !=(i) and (
                        ↪ is_it_in_vector(*it4,succ
                        ↪ [*it1]) == 0) and (
                        ↪ is_it_in_vector(*it4,pred
                        ↪ [*it1]) == 0) and (
                        ↪ is_it_in_vector(*it4,succ
                        ↪ [i]) == 0) and (
                        ↪ is_it_in_vector(*it4,pred
                        ↪ [i]) == 0)) {
                            ++signatures[*it1][17];
                            ++signatures[*it2][19];
                            ++signatures[i][18];
                            ++signatures[*it4][20];
                            ++graphlets[7];
                    }
                    if ((*it4)!=(*it1) and (*it4)
                        ↪ !=(i) and (
                        ↪ is_it_in_vector(*it4,succ
                        ↪ [*it1]) == 0) and (
                        ↪ is_it_in_vector(*it4,pred
                        ↪ [*it1]) == 0) and  (
                        ↪ is_it_in_vector(*it4,pred
```

```
    ↪ [i]) == 1)) {
        ++signatures[*it1][80];
        ++signatures[*it2][78];
        ++signatures[i][79];
        ++signatures[*it4][77];
        ++graphlets[25];
}
if ((*it4)!=(*it1) and (*it4)
    ↪ !=(i) and (
    ↪ is_it_in_vector(*it4,succ
    ↪ [*it1]) == 0) and (
    ↪ is_it_in_vector(*it4,pred
    ↪ [*it1]) == 0) and  (
    ↪ is_it_in_vector(*it4,succ
    ↪ [i]) == 1)) {
        ++signatures[*it1][72];
        ++signatures[*it2][70];
        ++signatures[i][71];
        ++signatures[*it4][69];
        ++graphlets[23];
}
if ((*it4)!=(*it1) and (*it4)
    ↪ !=(i) and (
    ↪ is_it_in_vector(*it4,succ
    ↪ [*it1]) == 1)  and  (
    ↪ is_it_in_vector(*it4,succ
    ↪ [i]) == 1)) {
        ++signatures[*it1
            ↪ ][113];
        ++signatures[*it2
            ↪ ][115];
        ++signatures[i][114];
        ++signatures[*it4
            ↪ ][116];
        ++graphlets[35];
}
```

```
                        }
                    }
                }
            }
        }

        //orb 5,6
        for (vector<int64>::iterator it1 = succ[i].begin(); it1 !=succ
            ↪ [i].end(); ++it1) {
                for (vector<int64>::iterator it2 = succ[i].begin();
                    ↪ it2 !=succ[i].end(); ++it2) {
                        if ((*it1)!=(*it2) and (is_it_in_vector(*it1,
                            ↪ succ[*it2]) == 0) and (is_it_in_vector(*
                            ↪ it1,pred[*it2]) == 0)) {
                                ++signatures[*it1][5];
                                ++signatures[*it2][5];
                                ++signatures[i][6];
                                ++graphlets[2];

                                //orb 21,22,23,24 and orb 73,74,75,76
                                    ↪ and orb 97,98,99,100 and orb
                                    ↪ 101,102,103,104 and orb 44,45 and
                                    ↪  orb 94,95,96
                                for (vector<int64>::iterator it3 =
                                    ↪ pred[*it1].begin(); it3 !=pred[*
                                    ↪ it1].end(); ++it3) {
                                        if ((*it3)!=(*it2) and (*it3)
                                            ↪ !=(i) and (
                                            ↪ is_it_in_vector(*it3,succ
                                            ↪ [*it2]) == 0) and (
                                            ↪ is_it_in_vector(*it3,pred
                                            ↪ [*it2]) == 0) and (
                                            ↪ is_it_in_vector(*it3,succ
                                            ↪ [i]) == 0) and (
                                            ↪ is_it_in_vector(*it3,pred
                                            ↪ [i]) == 0)) {
                                                ++signatures[*it1][22];
```

```
                ++signatures[*it2][24];
                ++signatures[i][23];
                ++signatures[*it3][21];
                ++graphlets[8];
        }
        if ((*it3)!=(*it2) and (*it3)
            ↪ !=(i) and (
            ↪ is_it_in_vector(*it3,succ
            ↪ [*it2]) == 0) and (
            ↪ is_it_in_vector(*it3,pred
            ↪ [*it2]) == 0) and (
            ↪ is_it_in_vector(*it3,pred
            ↪ [i]) == 1)) {
                ++signatures[*it1][74];
                ++signatures[*it2][76];
                ++signatures[i][75];
                ++signatures[*it3][73];
                ++graphlets[24];
        }
        if ((*it3)!=(*it2) and (*it3)
            ↪ !=(i) and (
            ↪ is_it_in_vector(*it3,succ
            ↪ [*it2]) == 1)  and (
            ↪ is_it_in_vector(*it3,succ
            ↪ [i]) == 1)) {
                ++signatures[*it1][99];
                ++signatures[*it2][98];
                ++signatures[i][100];
                ++signatures[*it3][97];
                ++graphlets[31];
        }
        if ((*it3)!=(*it2) and (*it3)
            ↪ !=(i) and (
            ↪ is_it_in_vector(*it3,succ
            ↪ [*it2]) == 1)  and (
            ↪ is_it_in_vector(*it3,pred
```

```
    ↪ [i]) == 1)) {
        ++signatures[*it1
            ↪ ][103];
        ++signatures[*it2
            ↪ ][102];
        ++signatures[i][104];
        ++signatures[*it3
            ↪ ][101];
        ++graphlets[32];
}
if ((*it3)!=(*it2) and (*it3)
    ↪ !=(i) and (
    ↪ is_it_in_vector(*it3,pred
    ↪ [*it2]) == 1)  and (
    ↪ is_it_in_vector(*it3,succ
    ↪ [i]) == 0) and (
    ↪ is_it_in_vector(*it3,pred
    ↪ [i]) == 0)) {
        ++signatures[*it1][44];
        ++signatures[*it2][44];
        ++signatures[i][45];
        ++signatures[*it3][45];
        ++graphlets[16];
}
if ((*it3)!=(*it2) and (*it3)
    ↪ !=(i) and (
    ↪ is_it_in_vector(*it3,pred
    ↪ [*it2]) == 1)  and (
    ↪ is_it_in_vector(*it3,succ
    ↪ [i]) == 1)) {
        ++signatures[*it1][95];
        ++signatures[*it2][95];
        ++signatures[i][96];
        ++signatures[*it3][94];
        ++graphlets[30];
}
```

```
}

//orb 25,26,27,28 and orb 49,50,51,52
   ↪ and orb 65,66,67,68 and orb
   ↪ 46,47,48 and orb 85,86,87 and orb
   ↪  88,89,90
for (vector<int64> ::iterator it4 =
   ↪ succ[*it1].begin(); it4 !=succ[*
   ↪ it1].end(); ++it4) {
        if ((*it4)!=(*it2) and (*it4)
           ↪ !=(i) and (
           ↪ is_it_in_vector(*it4,succ
           ↪ [*it2]) == 0) and (
           ↪ is_it_in_vector(*it4,pred
           ↪ [*it2]) == 0) and (
           ↪ is_it_in_vector(*it4,succ
           ↪ [i]) == 0) and (
           ↪ is_it_in_vector(*it4,pred
           ↪ [i]) == 0)) {
              ++signatures[*it1][26];
              ++signatures[*it2][28];
              ++signatures[i][27];
              ++signatures[*it4][25];
              ++graphlets[9];
        }
        if ((*it4)!=(*it2) and (*it4)
           ↪ !=(i) and (
           ↪ is_it_in_vector(*it4,succ
           ↪ [*it2]) == 0) and (
           ↪ is_it_in_vector(*it4,pred
           ↪ [*it2]) == 0)  and (
           ↪ is_it_in_vector(*it4,pred
           ↪ [i]) == 1)) {
              ++signatures[*it1][49];
              ++signatures[*it2][52];
              ++signatures[i][51];
```

```
                ++signatures[*it4][50];
                ++graphlets[18];
        }
        if ((*it4)!=(*it2) and (*it4)
            ↪ !=(i) and (
            ↪ is_it_in_vector(*it4,succ
            ↪ [*it2]) == 0) and (
            ↪ is_it_in_vector(*it4,pred
            ↪ [*it2]) == 0)  and (
            ↪ is_it_in_vector(*it4,succ
            ↪ [i]) == 1)) {
                ++signatures[*it1][65];
                ++signatures[*it2][68];
                ++signatures[i][67];
                ++signatures[*it4][66];
                ++graphlets[22];
        }
        if ((*it4)!=(*it2) and (*it4)
            ↪ !=(i) and (
            ↪ is_it_in_vector(*it4,succ
            ↪ [*it2]) == 1)  and (
            ↪ is_it_in_vector(*it4,succ
            ↪ [i]) == 0) and (
            ↪ is_it_in_vector(*it4,pred
            ↪ [i]) == 0)) {
                ++signatures[*it1][46];
                ++signatures[*it2][46];
                ++signatures[i][48];
                ++signatures[*it4][47];
                ++graphlets[17];
        }
        if ((*it4)!=(*it2) and (*it4)
            ↪ !=(i) and (
            ↪ is_it_in_vector(*it4,succ
            ↪ [*it2]) == 1)  and (
            ↪ is_it_in_vector(*it4,succ
```

```
                    ↪ [i]) == 1)) {
                        ++signatures[*it1][86];
                        ++signatures[*it2][86];
                        ++signatures[i][87];
                        ++signatures[*it4][85];
                        ++graphlets[27];
                }
                if ((*it4)!=(*it2) and (*it4)
                    ↪ !=(i) and (
                    ↪ is_it_in_vector(*it4,succ
                    ↪ [*it2]) == 1)  and (
                    ↪ is_it_in_vector(*it4,pred
                    ↪ [i]) == 1)) {
                        ++signatures[*it1][89];
                        ++signatures[*it2][89];
                        ++signatures[i][90];
                        ++signatures[*it4][88];
                        ++graphlets[28];
                }
        }

        //orb 31,32
        for (vector<int64>::iterator it5 =
            ↪ succ[i].begin(); it5 !=succ[i].
            ↪ end(); ++it5) {
                if ((*it5)!=(*it2) and (*it5)
                    ↪ !=(*it1) and (
                    ↪ is_it_in_vector(*it5,succ
                    ↪ [*it2]) == 0) and (
                    ↪ is_it_in_vector(*it5,pred
                    ↪ [*it2]) == 0) and (
                    ↪ is_it_in_vector(*it5,succ
                    ↪ [*it1]) == 0) and (
                    ↪ is_it_in_vector(*it5,pred
                    ↪ [*it1]) == 0)) {
                        ++signatures[*it1][31];
```

```
                                        ++signatures[*it2][31];
                                        ++signatures[i][32];
                                        ++signatures[*it5][31];
                                        ++graphlets[11];
                                    }
                                }

                                //orb 33,34,35
                                for (vector<int64>::iterator it6 =
                                    ↪ pred[i].begin(); it6 !=pred[i].
                                    ↪ end(); ++it6) {
                                        if ((*it6)!=(*it2) and (*it6)
                                            ↪ !=(*it1) and (
                                            ↪ is_it_in_vector(*it6,succ
                                            ↪ [*it2]) == 0) and (
                                            ↪ is_it_in_vector(*it6,pred
                                            ↪ [*it2]) == 0) and (
                                            ↪ is_it_in_vector(*it6,succ
                                            ↪ [*it1]) == 0) and (
                                            ↪ is_it_in_vector(*it6,pred
                                            ↪ [*it1]) == 0)) {
                                                ++signatures[*it1][35];
                                                ++signatures[*it2][35];
                                                ++signatures[i][34];
                                                ++signatures[*it6][33];
                                                ++graphlets[12];
                                        }
                                }
                        }
                }
        }

        //orb 7,8
        for (vector<int64>::iterator it1 = pred[i].begin(); it1 !=pred
            ↪ [i].end(); ++it1) {
```

```
for (vector<int64> ::iterator it2 = pred[i].begin();
   ↪ it2 !=pred[i].end(); ++it2) {
        if ((*it1)!=(*it2) and (is_it_in_vector(*it1,
           ↪ succ[*it2]) == 0) and (is_it_in_vector(*
           ↪ it1,pred[*it2]) == 0)) {
                ++signatures[*it1][7];
                ++signatures[*it2][7];
                ++signatures[i][8];
                ++graphlets[3];
                //orb 29,30 and orb 61,62,63,64
                for (vector<int64> ::iterator it3 =
                   ↪ pred[i].begin(); it3 !=pred[i].
                   ↪ end(); ++it3) {
                        if ((*it3)!=(*it2) and (*it3)
                           ↪ !=(*it1) and (
                           ↪ is_it_in_vector(*it3,succ
                           ↪ [*it2]) == 0) and (
                           ↪ is_it_in_vector(*it3,pred
                           ↪ [*it2]) == 0) and (
                           ↪ is_it_in_vector(*it3,succ
                           ↪ [*it1]) == 0) and (
                           ↪ is_it_in_vector(*it3,pred
                           ↪ [*it1]) == 0)) {
                                ++signatures[*it1][29];
                                ++signatures[*it2][29];
                                ++signatures[i][30];
                                ++signatures[*it3][29];
                                ++graphlets[10];
                        }
                        if ((*it3)!=(*it2) and (*it3)
                           ↪ !=(*it1) and  (
                           ↪ is_it_in_vector(*it3,pred
                           ↪ [*it2]) == 1) and (
                           ↪ is_it_in_vector(*it3,succ
                           ↪ [*it1]) == 0) and (
                           ↪ is_it_in_vector(*it3,pred
```

```
                ↪ [*it1]) == 0)) {
                    ++signatures[*it1][64];
                    ++signatures[*it2][62];
                    ++signatures[i][63];
                    ++signatures[*it3][61];
                    ++graphlets[21];
            }
        }

        //orb 36,37,38 and orb 91,92,93
        for (vector<int64> ::iterator it4 =
            ↪ succ[i].begin(); it4 !=succ[i].
            ↪ end(); ++it4) {
                if ((*it4)!=(*it2) and (*it4)
                    ↪ !=(*it1) and (
                    ↪ is_it_in_vector(*it4,succ
                    ↪ [*it2]) == 0) and (
                    ↪ is_it_in_vector(*it4,pred
                    ↪ [*it2]) == 0) and (
                    ↪ is_it_in_vector(*it4,succ
                    ↪ [*it1]) == 0) and (
                    ↪ is_it_in_vector(*it4,pred
                    ↪ [*it1]) == 0)) {
                        ++signatures[*it1][38];
                        ++signatures[*it2][38];
                        ++signatures[i][37];
                        ++signatures[*it4][36];
                        ++graphlets[13];
                }
                if ((*it4)!=(*it2) and (*it4)
                    ↪ !=(*it1) and (
                    ↪ is_it_in_vector(*it4,succ
                    ↪ [*it2]) == 1)  and (
                    ↪ is_it_in_vector(*it4,succ
                    ↪ [*it1]) == 1)) {
                        ++signatures[*it1][92];
```

```
                                        ++signatures[*it2][92];
                                        ++signatures[i][93];
                                        ++signatures[*it4][91];
                                        ++graphlets[29];
                                    }
                                }

                            }
                        }

}

//orb9
for (vector<int64>::iterator it1 = pred[i].begin(); it1 !=pred
   ↪ [i].end(); ++it1) {
        for (vector<int64>::iterator it2 = succ[i].begin();
           ↪ it2 !=succ[i].end(); ++it2) {
                if ((*it1)!=(*it2) and (is_it_in_vector(*it1,
                   ↪ succ[*it2]) == 1)) {
                        ++signatures[*it1][9];
                        ++signatures[*it2][9];
                        ++signatures[i][9];
                        ++graphlets[4];

                }
        }
}

//orb10,11,12
for (vector<int64>::iterator it1 = pred[i].begin(); it1 !=pred
   ↪ [i].end(); ++it1) {
        for (vector<int64>::iterator it2 = pred[i].begin();
           ↪ it2 !=pred[i].end(); ++it2) {
                if ((*it1)!=(*it2) and (is_it_in_vector(*it1,
                   ↪ pred[*it2]) == 1)) {
                        ++signatures[*it1][11];
```

```
++signatures[*it2][12];
++signatures[i][10];
++graphlets[5];

//orbits 57,58,59,60 and orb
    ↪ 121,122,123,124 and orb 119,120
    ↪ and orb 125,126,127,128
for (vector<int64>::iterator it3 =
    ↪ succ[i].begin(); it3 !=succ[i].
    ↪ end(); ++it3) {
        if ((*it3)!=(*it2) and (*it3)
            ↪ !=(*it1) and (
            ↪ is_it_in_vector(*it3,succ
            ↪ [*it2]) == 0) and (
            ↪ is_it_in_vector(*it3,pred
            ↪ [*it2]) == 0) and (
            ↪ is_it_in_vector(*it3,succ
            ↪ [*it1]) == 0) and (
            ↪ is_it_in_vector(*it3,pred
            ↪ [*it1]) == 0)) {
                ++signatures[*it1][57];
                ++signatures[*it2][58];
                ++signatures[i][59];
                ++signatures[*it3][60];
                ++graphlets[20];
        }
        if ((*it3)!=(*it2) and (*it3)
            ↪ !=(*it1) and (
            ↪ is_it_in_vector(*it3,pred
            ↪ [*it2]) == 1) and (
            ↪ is_it_in_vector(*it3,pred
            ↪ [*it1]) == 1)) {
                ++signatures[*it1
                    ↪ ][124];
                ++signatures[*it2
                    ↪ ][123];
```

```
                    ++signatures[i][122];
                    ++signatures[*it3
                        ↪ ][121];
                    ++graphlets[38];
                }
                if ((*it3)!=(*it2) and (*it3)
                    ↪ !=(*it1) and (
                    ↪ is_it_in_vector(*it3,pred
                    ↪ [*it2]) == 1) and (
                    ↪ is_it_in_vector(*it3,succ
                    ↪ [*it1]) == 1)) {
                    ++signatures[*it1
                        ↪ ][120];
                    ++signatures[*it2
                        ↪ ][119];
                    ++signatures[i][119];
                    ++signatures[*it3
                        ↪ ][119];
                    ++graphlets[37];
                }
                if ((*it3)!=(*it2) and (*it3)
                    ↪ !=(*it1) and (
                    ↪ is_it_in_vector(*it3,succ
                    ↪ [*it2]) == 1) and (
                    ↪ is_it_in_vector(*it3,succ
                    ↪ [*it1]) == 1)) {
                    ++signatures[*it1
                        ↪ ][128];
                    ++signatures[*it2
                        ↪ ][126];
                    ++signatures[i][127];
                    ++signatures[*it3
                        ↪ ][125];
                    ++graphlets[39];
                }
            }
```

```
                        //orb 117,118
                        for (vector<int64>::iterator it4 =
                            ↪ pred[i].begin(); it4 !=pred[i].
                            ↪ end(); ++it4) {
                                if ((*it4)!=(*it2) and (*it4)
                                    ↪ !=(*it1) and  (
                                    ↪ is_it_in_vector(*it4,pred
                                    ↪ [*it1]) == 1) and (
                                    ↪ is_it_in_vector(*it4,succ
                                    ↪ [*it2]) == 1)) {
                                        ++signatures[*it1
                                            ↪ ][117];
                                        ++signatures[*it2
                                            ↪ ][117];
                                        ++signatures[i][118];
                                        ++signatures[*it4
                                            ↪ ][117];
                                        ++graphlets[36];
                                }
                        }

                }
        }
}
cout<<"node "<<i<<" over"<<endl;
}

//CORRECTING FOR OVERCOUNTS
for (int64 i(0); i<dictionary.size(); ++i){
        signatures[i][5]=signatures[i][5]/2;
        signatures[i][6]=signatures[i][6]/2;
        signatures[i][7]=signatures[i][7]/2;
        signatures[i][8]=signatures[i][8]/2;
        signatures[i][9]=signatures[i][9]/3;
        signatures[i][29]=signatures[i][29]/6;
        signatures[i][30]=signatures[i][30]/6;
```

```
        signatures[i][31]=signatures[i][31]/6;
        signatures[i][32]=signatures[i][32]/6;
        signatures[i][33]=signatures[i][33]/2;
        signatures[i][34]=signatures[i][34]/2;
        signatures[i][35]=signatures[i][35]/2;
        signatures[i][36]=signatures[i][36]/2;
        signatures[i][37]=signatures[i][37]/2;
        signatures[i][38]=signatures[i][38]/2;
        signatures[i][39]=signatures[i][39]/4;
        signatures[i][44]=signatures[i][44]/4;
        signatures[i][45]=signatures[i][45]/4;
        signatures[i][46]=signatures[i][46]/2;
        signatures[i][47]=signatures[i][47]/2;
        signatures[i][48]=signatures[i][48]/2;
        signatures[i][94]=signatures[i][94]/2;
        signatures[i][95]=signatures[i][95]/2;
        signatures[i][96]=signatures[i][96]/2;
        signatures[i][85]=signatures[i][85]/2;
        signatures[i][86]=signatures[i][86]/2;
        signatures[i][87]=signatures[i][87]/2;
        signatures[i][88]=signatures[i][88]/2;
        signatures[i][89]=signatures[i][89]/2;
        signatures[i][90]=signatures[i][90]/2;
        signatures[i][91]=signatures[i][91]/2;
        signatures[i][92]=signatures[i][92]/2;
        signatures[i][93]=signatures[i][93]/2;
        signatures[i][119]=signatures[i][119]/3;
        signatures[i][120]=signatures[i][120]/3;
        signatures[i][117]=signatures[i][117]/3;
        signatures[i][118]=signatures[i][118]/3;

}
graphlets[2]=graphlets[2]/2;
graphlets[3]=graphlets[3]/2;
graphlets[4]=graphlets[4]/3;
graphlets[10]=graphlets[10]/6;
```

```
graphlets[11]=graphlets[11]/6;
graphlets[12]=graphlets[12]/2;
graphlets[13]=graphlets[13]/2;
graphlets[14]=graphlets[14]/4;
graphlets[16]=graphlets[16]/4;
graphlets[17]=graphlets[17]/2;
graphlets[30]=graphlets[30]/2;
graphlets[27]=graphlets[27]/2;
graphlets[28]=graphlets[28]/2;
graphlets[29]=graphlets[29]/2;
graphlets[37]=graphlets[37]/3;
graphlets[36]=graphlets[36]/3;

//Outputs

    //Output signatures

    ofstream out_file1;
    out_file1.open(string(string(argv[1])+".signatures.txt").
       ↪ c_str());
    for (vector<vector<int64> >::iterator it = signatures.begin
       ↪ (); it != signatures.end(); ++it){
        for (int64 i(0); i<129; ++i){
                out_file1<<(*it)[i]<<" ";
        }
        out_file1<<endl;
    }

    //Output dictionary

    ofstream out_file2;
    out_file2.open(string(string(argv[1])+".dictionary.txt").
       ↪ c_str());
    for (int64 i(0); i<dictionary.size(); ++i)
       out_file2<<i<<" "<<dictionary[i]<<endl;
```

```
    //Output graphlet counts
    ofstream out_file3;
    out_file3.open(string(string(argv[1])+".graphletcounts.txt
        ↪ ").c_str()) ;
    for (int64 i(0); i<graphlets.size();++i)
        out_file3<<i<<" "<<graphlets[i]<<endl;
    out_file1.close();
    out_file2.close();
    out_file3.close();

}
```

C Appendix to Chapter 4

C.1 GO Enrichment of Enzyme Clusters in the Metabolic Network of *H. sapiens*

The 4 enzyme clusters in the metabolic network of *H. sapiens* from Table 4.5, Section 4.2.2 are statistically significantly enriched in GO terms listed in Table C.1.

GO term	p-value
Cluster	**1**
long-chain fatty acid-CoA ligase activity	$4.63308452459\times10^{-5}$
prostanoid metabolic process	0.00831050325419
oligosaccharide catabolic process	0.00341941781287
coenzyme A metabolic process	0.00393556983194
membrane lipid catabolic process	0.00831050325419
Cluster	**2**
pyrophosphatase activity	0.00901649693998
aminoglycan catabolic process	0.00289070431908
monovalent inorganic cation transport	0.000544951554996
very long-chain fatty acid metabolic process	0.000374493455158
N-methyltransferase activity	$6.37802077991\times10^{-5}$
covalent chromatin modification	0.00869777477011
peptidyl-lysine methylation	0.0025556364929
ncRNA metabolic process	0.00880839878982
osteoblast differentiation	0.00880839878982
protein methyltransferase activity	$6.37802077991\times10^{-5}$
palmitoyltransferase activity	0.00675811663364
histone lysine methylation	0.00202845881394
S-adenosylmethionine-dependent methyltransferase activity	0.000843521737388
tRNA aminoacylation	0.00192219276182

Cluster	**3**
alcohol dehydrogenase (NAD) activity	0.00777004841489
phospholipase activity	0.00576798981364
nuclear body	0.00617334192272
retinoic acid binding	$1.45975079802\times10^{-8}$
monocarboxylic acid binding	$1.71229944224\times10^{-7}$
steroid catabolic process	0.00791946172056
arachidonic acid metabolic process	0.00588356084718
uronic acid metabolic process	0.00160940371616
positive regulation of defense response	0.00791946172056
regulation of innate immune response	0.00171614138423
diacylglycerol kinase activity	0.00617334192272
caffeine oxidase activity	0.00617334192272
nucleoside diphosphate kinase activity	$2.0022774061\times10^{-7}$
nucleoside triphosphate biosynthetic process	0.00728068929927
RNA biosynthetic process	$4.17762963012\times10^{-6}$
terpenoid metabolic process	0.00808893002574
regulation of type I interferon production	0.000475477315632
glucuronosyltransferase activity	0.000100350407124
DNA-directed RNA polymerase III complex	$2.70833243265\times10^{-6}$
DNA-directed RNA polymerase II, core complex	$2.0022774061\times10^{-7}$
transcription, DNA-templated	$3.19686654415\times10^{-6}$
exogenous drug catabolic process	0.000131293091226
glycerophospholipid metabolic process	0.00501167864473
Cluster	**4**
ceramide metabolic process	0.00245509594718
poly-N-acetyllactosamine biosynthetic process	0.0050405873955
protein O-linked glycosylation via threonine	0.00118074658032
fructose-bisphosphate aldolase activity	0.00642392195327
polysaccharide biosynthetic process	0.00740025407149
protein O-linked glycosylation via serine	0.00642392195327
acetylglucosaminyltransferase activity	0.000784576091033
acetylgalactosaminyltransferase activity	$8.81983077949\times10^{-6}$
regulation of peptide hormone secretion	0.0050405873955
sphingolipid biosynthetic process	0.000775055279451

hexose metabolic process	0.00239493428124
poly-N-acetyllactosamine metabolic process	0.0050405873955
alpha-amino acid catabolic process	0.00131819398653

Table C.1. GO term enrichment of 4 enzyme clusters in *H. sapiens* metabolic network. First column: GO term. Second column: p-value of the enrichment.

The 19 enzyme clusters in the metabolic network of *H. sapiens* from Table 4.6, Section 4.2.2 are statistically significantly enriched in GO terms listed in Table C.2.

GO term	p-value
Cluster	**1**
regulation of peptide transport	0.000545392244054
acetylglucosaminyltransferase activity	0.00146310828643
regulation of peptide hormone secretion	0.000278599665101
glutamine family amino acid metabolic process	0.00229566962877
poly-N-acetyllactosamine metabolic process	0.00927994934645
regulation of secretion by cell	0.00300397923731
poly-N-acetyllactosamine biosynthetic process	0.00927994934645
positive regulation of hormone secretion	0.000984930562568
glutamate dehydrogenase (NAD+) activity	0.000984930562568
glutamate dehydrogenase [NAD(P)+] activity	0.00289663147545
carbohydrate homeostasis	0.000545392244054
Cluster	**2**
N,N-dimethylaniline monooxygenase activity	0.00056141042073
ceramidase activity	0.00537283265039
6-phosphofructokinase complex	0.00056141042073
negative regulation of signal transduction	0.00730648426577
adenyl nucleotide binding	0.00308649892024
negative regulation of cell communication	0.00730648426577
zymogen activation	0.00165972811201
ATP binding	0.0056058757954
sphingolipid biosynthetic process	0.00925860695639
regulation of peptidase activity	$4.8250421169\times10^{-5}$
phosphofructokinase activity	0.00537283265039
adenyl ribonucleotide binding	0.00308649892024
sphingoid metabolic process	0.000937495623343

dihydroceramidase activity	0.00056141042073
sphingosine biosynthetic process	$4.8250421169\times10^{-5}$
purine ribonucleoside binding	0.00354482336319
Cluster	**3**
arginase activity	0.000502314586797
acetylgalactosaminyltransferase activity	$7.90145726626\times10^{-13}$
nitric-oxide synthase activity	0.000502314586797
protein O-linked glycosylation via threonine	$1.98917102789\times10^{-7}$
protein O-linked glycosylation via serine	$1.03171173853\times10^{-5}$
chondroitin sulfate biosynthetic process	0.000502314586797
glycosaminoglycan metabolic process	0.00319938099115
O-glycan processing	0.000502314586797
cellular protein modification process	0.00495061943503
macromolecule glycosylation	$2.09150420165\times10^{-5}$
glycosaminoglycan biosynthetic process	0.00053717366259
Cluster	**4**
acetylglucosaminyltransferase activity	0.00396351381255
ceramide metabolic process	$1.95846324655\times10^{-5}$
nucleoside triphosphate catabolic process	0.00195344561366
sphingomyelin biosynthetic process	0.00195344561366
sphingolipid biosynthetic process	$6.71993518642\times10^{-6}$
adenylyltransferase activity	0.000777278323579
ceramide cholinephosphotransferase activity	0.00195344561366
inorganic anion transport	0.00195344561366
malate dehydrogenase (decarboxylating) (NADP+) activity	0.00195344561366
Cluster	**5**
fat-soluble vitamin metabolic process	$7.81206238682\times10^{-5}$
regulation of secretion	0.0071964719671
positive regulation of secretion	0.00178692760946
arachidonate 15-lipoxygenase activity	0.00230802061752
renal water homeostasis	0.00230802061752
epithelial cell differentiation	0.00570145764537
phenanthrene 9,10-monooxygenase activity	0.00671074030738
linoleate 13S-lipoxygenase activity	0.00230802061752
carbonyl reductase (NADPH) activity	0.00671074030738
alkane 1-monooxygenase activity	0.00230802061752

unsaturated fatty acid metabolic process	0.000165656207175
androsterone dehydrogenase activity	0.00671074030738
renal system process involved in regulation of systemic arterial blood pressure	0.00671074030738
leukotriene-B4 20-monooxygenase activity	0.00230802061752
estradiol 17-beta-dehydrogenase activity	0.00230802061752
diacylglycerol kinase activity	$4.79836380407\times10^{-6}$
3-oxo-5-alpha-steroid 4-dehydrogenase activity	0.00230802061752
icosanoid catabolic process	0.00230802061752
isoprenoid catabolic process	$6.70255389583\times10^{-5}$
long-chain fatty acid metabolic process	0.00587366284063
positive regulation of organic acid transport	0.00671074030738
retinoic acid catabolic process	0.000106660773032
trans-1,2-dihydrobenzene-1,2-diol dehydrogenase activity	0.00230802061752
exogenous drug catabolic process	0.000150923161808
leukotriene metabolic process	0.00081389867487
arachidonate 12-lipoxygenase activity	0.000106660773032
terpenoid metabolic process	0.00222599527077
cellular response to jasmonic acid stimulus	0.000412248000885
monocarboxylic acid binding	0.00299436362643
ketosteroid monooxygenase activity	0.00671074030738
lipoxygenase pathway	0.000412248000885
leukotriene B4 catabolic process	0.00230802061752
water homeostasis	0.00230802061752
arachidonic acid metabolic process	$6.64504225378\times10^{-6}$
Cluster	**6**
Ada2/Gcn5/Ada3 transcription activator complex	0.00142814931413
DNA replication	0.000199986431668
base-excision repair, gap-filling	0.00142814931413
acetyltransferase complex	0.00142814931413
protein phosphatase binding	0.00816357862462
nucleotide-excision repair, DNA gap filling	$5.13327096738\times10^{-5}$
DNA polymerase activity	0.000486962726909
phosphatidylinositol 3-kinase activity	0.00142814931413
intracellular signal transduction	0.00459787562859
positive regulation of transferase activity	0.00816357862462

phosphatidylinositol phosphate kinase activity	0.00142814931413
cortisol biosynthetic process	0.00816357862462
phosphatidylinositol 3-kinase complex	0.00142814931413
phosphatidylinositol kinase activity	$5.13327096738\times10^{-5}$
lipid phosphorylation	0.000948614883387
aldosterone biosynthetic process	0.00816357862462
glycerophospholipid biosynthetic process	0.002210818121
DNA biosynthetic process	0.000948614883387
glycerophospholipid metabolic process	$1.79969129337\times10^{-5}$
response to potassium ion	0.00142814931413
steroid 11-beta-monooxygenase activity	0.00418178252708
histone H3 acetylation	0.00142814931413
glucocorticoid metabolic process	0.00816357862462
Cluster	**7**
alcohol dehydrogenase (NAD) activity	$1.30243276319\times10^{-6}$
lysophospholipid acyltransferase activity	0.00225888789787
phospholipase A2 activity	0.000304285308076
aldehyde dehydrogenase [NAD(P)+] activity	0.000247813081953
negative regulation of carbohydrate metabolic process	$9.16123259115\times10^{-5}$
primary alcohol metabolic process	0.00113701020372
retinoic acid binding	$8.841039012\times10^{-12}$
regulation of carbohydrate metabolic process	0.00162342459554
regulation of cellular ketone metabolic process	0.00924744149146
negative regulation of lipid metabolic process	0.00431397226454
regulation of cellular carbohydrate metabolic process	0.00162342459554
monocarboxylic acid binding	$1.95553573334\times10^{-11}$
glycerophospholipid metabolic process	0.000269209645729
glucuronosyltransferase activity	$1.25566224085\times10^{-13}$
uronic acid metabolic process	$4.02189392901\times10^{-12}$
phosphatidylcholine acyl-chain remodeling	0.00402179979617
Cluster	**8**
L-ascorbic acid metabolic process	0.00985915492932
cellular response to cGMP	0.00985915492932
nucleoside diphosphate kinase activity	$1.2714274078\times10^{-12}$
negative regulation of ion transmembrane transporter activity	0.00985915492932
nucleoside triphosphate metabolic process	$3.47652203339\times10^{-5}$

nucleoside triphosphate biosynthetic process	$6.20325591449\times10^{-7}$
leukocyte differentiation	0.00985915492932
RNA biosynthetic process	$6.50991482942\times10^{-11}$
nucleoside diphosphate metabolic process	$2.49162731703\times10^{-6}$
cellular response to organic cyclic compound	0.00128683773729
positive regulation of immune response	0.00148133135366
regulation of type I interferon production	$8.30747724612\times10^{-7}$
positive regulation of cytokine production	0.00770714258585
transcription, DNA-templated	$4.05653288738\times10^{-12}$
adenylate kinase activity	0.00118724138837
positive regulation of defense response	$4.88320530324\times10^{-5}$
regulation of defense response	0.000387243057092
negative regulation of transmembrane transport	0.00985915492932
response to macrophage colony-stimulating factor	0.00985915492932
DNA-directed RNA polymerase II, core complex	$1.2714274078\times10^{-12}$
DNA-directed RNA polymerase III complex	$5.63592505998\times10^{-11}$
regulation of innate immune response	$8.83096626758\times10^{-6}$
phosphoric diester hydrolase activity	0.00647155226004
guanyl nucleotide binding	0.0078770144057
guanyl ribonucleotide binding	0.0078770144057
caffeine oxidase activity	0.00356929967823
nucleobase-containing compound kinase activity	$4.32982258136\times10^{-6}$
peptide disulfide oxidoreductase activity	0.00985915492932
Cluster	**9**
acylglycerol metabolic process	0.0059673109137
aldehyde dehydrogenase (NAD) activity	0.000128024871741
neutral lipid catabolic process	0.00855984137165
high-density lipoprotein particle remodeling	0.00266722736524
triglyceride lipase activity	0.00523416499316
Cluster	**10**
porphyrin-containing compound biosynthetic process	0.000759655470805
heme a biosynthetic process	0.00523983059161
glycosaminoglycan metabolic process	0.00209358399186
response to vitamin D	0.000369949528004
tetrapyrrole biosynthetic process	0.000759655470805
mannosyltransferase activity	0.000759655470805

aminoglycan catabolic process	0.000759655470805
cellular response to light stimulus	0.00140276492157
heparanase activity	0.00523983059161
response to UV	0.00630537318887
response to sterol	0.00523983059161
pantetheine hydrolase activity	0.00523983059161
osteoblast differentiation	0.00630537318887
response to interleukin-1	0.00332483835499
hyaluronoglucuronidase activity	0.00523983059161
Cluster	**11**
fatty acid biosynthetic process	0.000637582106715
regulation of intracellular transport	0.00914880405747
positive regulation of transmembrane transport	0.00283566756057
regulation of phosphatase activity	0.000313609983214
regulation of calcium ion transport into cytosol	0.000614149022136
positive regulation of ion transmembrane transporter activity	0.00283566756057
fatty acid synthase activity	0.00283566756057
very long-chain fatty acid metabolic process	$5.60751889367\times10^{-10}$
3-hydroxyacyl-CoA dehydratase activity	0.00469811878164
positive regulation of ion transport	0.00668035176894
butyrate-CoA ligase activity	0.00469811878164
carboxylic acid biosynthetic process	0.00561008145297
beta-galactoside alpha-2,6-sialyltransferase activity	0.00469811878164
protein kinase activity	0.00914880405747
regulation of cytokinesis	0.00283566756057
regulation of cyclic-nucleotide phosphodiesterase activity	0.000313609983214
regulation of muscle system process	0.0011927659888
protein kinase binding	0.00163000613914
Cluster	**12**
chitinase activity	0.00523983059161
regulation of interferon-gamma production	0.00523983059161
natriuretic peptide receptor activity	0.00523983059161
macromolecule deacylation	0.00523983059161
chitin metabolic process	0.00523983059161
palmitoyltransferase activity	$2.56777313428\times10^{-5}$
C-acyltransferase activity	0.00332483835499

regulation of interleukin-2 production	0.00523983059161
biliverdin reductase activity	0.00523983059161
Cluster	**13**
regulation of Wnt signaling pathway	0.00523983059161
protein methyltransferase activity	$2.35367281221\times10^{-13}$
N-methyltransferase activity	$2.35367281221\times10^{-13}$
Set1C/COMPASS complex	0.00523983059161
covalent chromatin modification	$3.89142096324\times10^{-9}$
positive regulation of biosynthetic process	0.00334715726731
ncRNA metabolic process	0.00630537318887
peptidyl-lysine methylation	$1.26323285166\times10^{-10}$
S-adenosylmethionine-dependent methyltransferase activity	$7.79376563287\times10^{-14}$
cellular response to decreased oxygen levels	0.00630537318887
pyrophosphatase activity	$1.64589927076\times10^{-6}$
regulation of macromolecule biosynthetic process	0.000980134018061
regulatory region DNA binding	0.000121382632547
sequence-specific DNA binding	0.000369949528004
RNA polymerase II core promoter proximal region sequence-specific DNA binding transcription factor activity involved in positive regulation of transcription	0.00523983059161
histone lysine methylation	$1.05771591485\times10^{-9}$
monovalent inorganic cation transport	0.00034430351978
MHC class I protein binding	0.00523983059161
regulation of RNA metabolic process	0.000563233893036
regulation of cellular biosynthetic process	0.00326765055607
regulation of gene expression	0.00284923374714
regulation of nucleobase-containing compound metabolic process	0.000472831162781
cellular protein modification process	0.00133042799092
positive regulation of nitrogen compound metabolic process	0.00504165744289
glutaminyl-tRNA synthase (glutamine-hydrolyzing) activity	0.000369949528004
tRNA aminoacylation	0.00332483835499
translation	0.000369949528004
Cluster	**14**
glucosyltransferase activity	0.00786699212521
UDP-glucosyltransferase activity	0.00480634205464

glucan metabolic process	0.00786699212521
prostanoid metabolic process	0.00167368710541
coenzyme A metabolic process	0.000422434124588
glucan biosynthetic process	0.000830624772471
phosphoglycerate mutase activity	$8.78398627689 \times 10^{-5}$
cellular glucan metabolic process	0.00786699212521
isocitrate dehydrogenase activity	0.00244709095957
purine nucleoside bisphosphate metabolic process	0.0007250193856
nucleoside bisphosphate biosynthetic process	0.00480634205464
nucleoside bisphosphate metabolic process	0.0007250193856
glycogen biosynthetic process	0.000830624772471
Cluster	**15**
histone-threonine phosphorylation	0.00435339308504
L-serine ammonia-lyase activity	0.00435339308504
regulation of cellular protein localization	0.00799174572515
regulation of blood vessel size	0.00435339308504
positive regulation of protein localization to nucleus	0.00253676367504
isopentenyl-diphosphate delta-isomerase activity	0.00435339308504
oxidoreduction coenzyme metabolic process	0.00476094968996
protein kinase activity	0.00777375319713
protein farnesylation	0.00435339308504
Cluster	**16**
hydrogen sulfide biosynthetic process	0.000758396531881
pyrimidine-containing compound biosynthetic process	0.00720163341439
tetrahydrofolate metabolic process	0.00720163341439
cysteine metabolic process	0.00439603751703
dihydrofolate reductase activity	0.000758396531881
Cluster	**17**
acylglycerol metabolic process	$3.58860130253 \times 10^{-5}$
acylglycerol biosynthetic process	0.000689514424748
choline kinase activity	0.00115236875817
primary amine oxidase activity	0.00115236875817
retinol dehydrogenase activity	$3.71592960405 \times 10^{-5}$
branched-chain amino acid catabolic process	0.00662177660088
phosphatidylcholine biosynthetic process	0.000689514424748
S-adenosylmethionine biosynthetic process	0.00115236875817

terpenoid metabolic process	0.00264006430028
ethanolamine kinase activity	0.00338314165604
ion channel activity	0.00338314165604
Cluster	**18**
acetate-CoA ligase activity	0.00386092780403
acyl-CoA biosynthetic process	0.000890458017772
tryptophan 2,3-dioxygenase activity	0.00386092780403
fructose-bisphosphate aldolase activity	0.000232944296515
indoleamine 2,3-dioxygenase activity	0.00386092780403
acetyl-CoA metabolic process	0.00406736214002
glycine hydroxymethyltransferase activity	0.00386092780403
Cluster	**19**
alpha-sialidase activity	0.000219299381388
organic anion transport	0.00510534828812
positive regulation of organelle organization	0.00510534828812
very-low-density lipoprotein particle assembly	0.00217371894418
negative regulation of chromatin modification	0.0070422535209
long-chain fatty acid transport	0.0070422535209
DNA-methyltransferase activity	0.00217371894418
N-acyltransferase activity	0.00959472054332
long-chain fatty acid-CoA ligase activity	$2.27515072337\times10^{-8}$
3-hydroxybutyrate dehydrogenase activity	0.0070422535209
branched-chain-amino-acid transaminase activity	0.0070422535209
cholesterol binding	0.0005785676842
glycosphingolipid metabolic process	0.00425002964653
protein-lipid complex assembly	0.00217371894418
positive regulation of chromatin modification	0.0070422535209
oligosaccharide catabolic process	$4.68505965401\times10^{-5}$
membrane lipid catabolic process	0.00425002964653
very long-chain fatty acid-CoA ligase activity	0.0005785676842
glycolipid catabolic process	0.00134586107042
methylcrotonoyl-CoA carboxylase activity	0.0070422535209

Table C.2. GO term enrichment of 19 enzyme clusters in *H. sapiens* metabolic network. First column: GO term. Second column: p-value of the enrichment.

C.2 GO Enrichment of Enzyme Sets Corresponding to Characteristic Topological Patterns

For the enzymes that are annotated with purine nucleotide metabolic process and that touch graphlets G_2 at orbit 6 in *H. sapiens* metabolic network, we find that the set of enzymes touching these graphlets at orbit 5 is statistically significantly enriched in GO terms listed in Table C.3.

For the enzymes annotated with purine nucleotide metabolic process and that touch graphlets G_5 at orbit 11 in *H. sapiens* metabolic network, we find that the set of enzymes touching graphlets G_5 at the remaining orbits (orbit 10 or 12), is statistically significantly enriched in GO terms listed in Table C.4.

For the enzymes that are annotated with purine nucleotide metabolic process and that touch graphlets G_2 at orbit 6 in *M. musculus* metabolic network, we find that the set of enzymes touching these graphlets at orbit 5 is statistically significantly enriched in GO terms listed in Table C.5.

For the enzymes annotated with purine nucleotide metabolic process and that touch graphlets G_5 at orbit 11 in *M. musculus* metabolic network, we find that the set of enzymes touching graphlets G_5 at the remaining orbits (orbit 10 or 12), is statistically significantly enriched in GO terms listed in Table C.6.

For the enzymes that are annotated with purine nucleotide metabolic process and that touch graphlets G_2 at orbit 6 in *D. melanogaster* metabolic network, we find that the set of enzymes touching these graphlets at orbit 5 is statistically significantly enriched in GO terms listed in Table C.7.

For the enzymes annotated with purine nucleotide metabolic process and that touch graphlets G_5 at orbit 11 in *D. melanogaster* metabolic network, we find that the set of enzymes touching graphlets G_5 at the remaining orbits (orbit 10 or 12), is statistically significantly enriched in GO terms listed in Table C.8.

For the enzymes that are annotated with ribose phosphate metabolic process and that touch graphlets G_2 at orbit 6 in *H. sapiens* metabolic network, we find that the set of enzymes touching these graphlets at orbit 5 is statistically significantly enriched in GO terms listed in Table C.9.

For the enzymes annotated with ribose phosphate metabolic process and that touch graphlets G_5 at orbit 11 in *H. sapiens* metabolic network, we find that the set of enzymes touching graphlets G_5 at the remaining orbits (orbit 10 or 12), is statistically significantly enriched in GO terms listed in Table C.10.

For the enzymes that are annotated with ribose phosphate metabolic process and

GO term	p-value
nucleotide catabolic process	0.000139137872827
nucleotide biosynthetic process	$2.09179133526\times10^{-6}$
nucleoside biosynthetic process	0.00183488292176
nucleoside diphosphate kinase activity	$1.79161463443\times10^{-10}$
nucleoside triphosphate metabolic process	$8.37254164876\times10^{-5}$
nucleoside triphosphate biosynthetic process	$5.66784749559\times10^{-5}$
ribonucleotide catabolic process	0.00017083973161
adenyl ribonucleotide binding	$6.82423649498\times10^{-5}$
transcription, DNA-templated	$2.05737837877\times10^{-9}$
RNA biosynthetic process	$1.6815517867\times10^{-9}$
purine nucleoside metabolic process	0.000288531256083
nucleoside diphosphate metabolic process	$2.83476893258\times10^{-5}$
adenyl nucleotide binding	$6.82423649498\times10^{-5}$
positive regulation of immune response	$1.19719222883\times10^{-5}$
regulation of type I interferon production	$1.51711257443\times10^{-5}$
positive regulation of cytokine production	0.00552320503685
ribose phosphate biosynthetic process	0.000348117418135
regulation of innate immune response	$9.81066176295\times10^{-5}$
nucleotide metabolic process	$2.86996816645\times10^{-7}$
ribonucleoside metabolic process	0.000423276719616
adenylate kinase activity	0.00724700883331
positive regulation of defense response	0.00051278407008
regulation of defense response	0.00363712307552
purine nucleotide metabolic process	$5.42007299598\times10^{-7}$
DNA-directed RNA polymerase II, core complex	$1.79161463443\times10^{-10}$
DNA-directed RNA polymerase III complex	$8.07663069757\times10^{-9}$
nucleotide-excision repair, DNA gap filling	0.00401747992595
RNA polymerase activity	0.00401747992595
phosphoric diester hydrolase activity	0.00803251987665
nucleoside monophosphate biosynthetic process	0.000903691572726
phosphorylation	0.00541458825394
nucleoside monophosphate metabolic process	$8.37254164876\times10^{-5}$
ribonucleotide metabolic process	$4.39674207042\times10^{-8}$
myeloid cell differentiation	0.00401747992595
nucleobase-containing compound kinase activity	$3.03581474603\times10^{-6}$
purine-containing compound catabolic process	0.000697255183771
ubiquitin protein ligase binding	0.00724700883331

Table C.3. GO term enrichment of enzymes touching orbit 5 in *H. sapiens* metabolic network when the enzymes touching orbit 6 of the same graphlet are annotated with purine nucleotide metabolic process. First column: GO term. Second column: p-value of the enrichment.

GO term	p-value
nucleotide catabolic process	$5.30681652056\times10^{-5}$
nucleotide biosynthetic process	$3.86453591128\times10^{-7}$
nucleoside biosynthetic process	0.000912669748425
nucleoside diphosphate kinase activity	$5.71358516055\times10^{-11}$
nucleoside triphosphate metabolic process	$3.14592083027\times10^{-5}$
nucleoside triphosphate biosynthetic process	$2.41254319037\times10^{-5}$
ribonucleotide catabolic process	6.987208625×10^{-5}
adenyl ribonucleotide binding	0.000136493119549
transcription, DNA-templated	$6.00953842245\times10^{-10}$
positive regulation of cytokine production	0.00302314673943
RNA biosynthetic process	$4.49683845716\times10^{-10}$
purine nucleoside metabolic process	0.000441586595457
nucleoside diphosphate metabolic process	$1.10888880278\times10^{-5}$
adenyl nucleotide binding	0.000136493119549
positive regulation of immune response	$5.76643246153\times10^{-6}$
regulation of type I interferon production	$8.62577299121\times10^{-6}$
regulation of innate immune response	$6.14091768376\times10^{-5}$
nucleotide metabolic process	$1.6392758484\times10^{-8}$
ribonucleoside metabolic process	0.00056245187314
adenylate kinase activity	0.00511901885652
positive regulation of defense response	0.000325326189435
regulation of defense response	0.00236973468809
purine nucleotide metabolic process	$7.77104436356\times10^{-8}$
DNA-directed RNA polymerase II, core complex	$5.71358516055\times10^{-11}$
DNA-directed RNA polymerase III complex	$3.09919767627\times10^{-9}$
nucleotide-excision repair, DNA gap filling	0.00304031248855
RNA polymerase activity	0.00304031248855
phosphoric diester hydrolase activity	0.00422834573084
ribose phosphate biosynthetic process	0.000154842644698
nucleoside monophosphate metabolic process	$3.14592083027\times10^{-5}$
ribonucleotide metabolic process	$5.35720856565\times10^{-9}$
nucleoside monophosphate biosynthetic process	0.000500439429898
nucleobase-containing compound kinase activity	$6.94098122089\times10^{-7}$
purine-containing compound biosynthetic process	0.00683599594482
purine-containing compound catabolic process	0.00029807956557

Table C.4. GO term enrichment of enzymes touching orbits 10 or 12 in *H. sapiens* metabolic network when the enzymes touching orbit 11 of the same graphlet are annotated with purine nucleotide metabolic process. First column: GO term. Second column: p-value of the enrichment.

GO term	p-value
nucleotide catabolic process	0.000410375245616
nucleotide biosynthetic process	$1.06858744076\times10^{-10}$
nucleoside biosynthetic process	0.000641527638692
cilium movement	0.00153598545297
nucleoside triphosphate metabolic process	$8.16637146794\times10^{-9}$
nucleoside triphosphate catabolic process	0.000105768986782
deoxyribonucleotide biosynthetic process	0.000790400765506
deoxyribonucleotide catabolic process	0.000790400765506
DNA-directed RNA polymerase I complex	$1.94190319335\times10^{-5}$
deoxyribose phosphate catabolic process	0.00562309567845
nucleoside diphosphate metabolic process	0.00836819230451
ribose phosphate biosynthetic process	$2.0590699866\times10^{-5}$
nucleotide metabolic process	$1.50512935448\times10^{-12}$
ribonucleoside metabolic process	0.000865009490838
purine nucleotide metabolic process	$6.50716147632\times10^{-11}$
nucleoside monophosphate metabolic process	$1.54795326908\times10^{-5}$
ribonucleotide metabolic process	$2.88948844729\times10^{-6}$
nucleoside monophosphate biosynthetic process	0.000813941504429
nucleoside monophosphate catabolic process	0.00562309567845
nucleobase-containing compound kinase activity	0.00215966432502
purine-containing compound biosynthetic process	$1.67162811926\times10^{-7}$
2'-deoxyribonucleotide metabolic process	$4.9414917433\times10^{-5}$

Table C.5. GO term enrichment of enzymes touching orbit 5 in *M. musculus* metabolic network when the enzymes touching orbit 6 of the same graphlet are annotated with purine nucleotide metabolic process. First column: GO term. Second column: p-value of the enrichment.

GO term	p-value
nucleotide catabolic process	0.000347705479615
nucleotide biosynthetic process	$7.37931937778\times10^{-12}$
nucleoside biosynthetic process	$7.64818267278\times10^{-5}$
cilium movement	0.00143246292186
nucleoside triphosphate metabolic process	$6.29348562153\times10^{-9}$
nucleoside triphosphate catabolic process	$9.42562613476\times10^{-5}$
deoxyribonucleotide biosynthetic process	0.000721469223498
deoxyribonucleotide catabolic process	0.000721469223498
DNA-directed RNA polymerase I complex	$1.72630529772\times10^{-5}$
deoxyribose phosphate catabolic process	0.00525553883157
purine nucleoside metabolic process	0.00252705625906
nucleoside diphosphate metabolic process	0.00769128555423
adenylyltransferase activity	0.000300256779766
nucleotide metabolic process	$1.14586118372\times10^{-12}$
ribonucleoside metabolic process	0.000151138328517
purine nucleotide metabolic process	$5.58242341242\times10^{-12}$
ribonucleotide metabolic process	$3.24050657752\times10^{-7}$
ribose phosphate biosynthetic process	$1.97342211239\times10^{-6}$
nucleoside monophosphate metabolic process	$1.26925178126\times10^{-5}$
nucleoside monophosphate biosynthetic process	0.000728873723372
nucleoside monophosphate catabolic process	0.00525553883157
nucleobase-containing compound kinase activity	0.00197589514898
purine-containing compound biosynthetic process	$1.54616328629\times10^{-8}$
2'-deoxyribonucleotide metabolic process	$4.30866121032\times10^{-5}$

Table C.6. GO term enrichment of enzymes touching orbits 10 or 12 in *M. musculus* metabolic network when the enzymes touching orbit 11 of the same graphlet are annotated with purine nucleotide metabolic process. First column: GO term. Second column: p-value of the enrichment.

GO term	p-value
ribonucleotide metabolic process	0.00457072512321
purine nucleotide metabolic process	0.00457072512321
DNA-directed RNA polymerase II, core complex	0.00064549982473
nucleotide metabolic process	0.000767128336139

Table C.7. GO term enrichment of enzymes touching orbit 5 in *D. melanogaster* metabolic network when the enzymes touching orbit 6 of the same graphlet are annotated with purine nucleotide metabolic process. First column: GO term. Second column: p-value of the enrichment.

GO term	p-value
nucleotide biosynthetic process	0.00149791055107
ribonucleotide metabolic process	0.000670604413007
purine nucleotide metabolic process	0.000670604413007
DNA-directed RNA polymerase II, core complex	0.000712275668912
nucleotide metabolic process	0.000122799296242

Table C.8. GO term enrichment of enzymes touching orbits 10 or 12 in *D. melanogaster* metabolic network when the enzymes touching orbit 11 of the same graphlet are annotated with purine nucleotide metabolic process. First column: GO term. Second column: p-value of the enrichment.

that touch graphlets G_2 at orbit 6 in *M. musculus* metabolic network, we find that the set of enzymes touching these graphlets at orbit 5 is statistically significantly enriched in GO terms listed in Table C.11.

For the enzymes annotated with ribose phosphate metabolic process and that touch graphlets G_5 at orbit 11 in *M. musculus* metabolic network, we find that the set of enzymes touching graphlets G_5 at the remaining orbits (orbit 10 or 12), is statistically significantly enriched in GO terms listed in Table C.12.

For the enzymes that are annotated with ribose phosphate metabolic process and that touch graphlets G_2 at orbit 6 in *D. melanogaster* metabolic network, we find that the set of enzymes touching these graphlets at orbit 5 is statistically significantly enriched in GO terms listed in Table C.13.

For the enzymes annotated with ribose phosphate metabolic process and that touch graphlets G_5 at orbit 11 in *D. melanogaster* metabolic network, we find that the set of enzymes touching graphlets G_5 at the remaining orbits (orbit 10 or 12), is statistically significantly enriched in GO terms listed in Table C.14.

For the enzymes that are annotated with cyclic nucleotide metabolic process and that touch graphlets G_2 at orbit 6 in *H. sapiens* metabolic network, we find that the set of enzymes touching these graphlets at orbit 5 is statistically significantly enriched in GO terms listed in Table C.15.

For the enzymes annotated with cyclic nucleotide metabolic process and that touch graphlets G_5 at orbit 11 in *H. sapiens* metabolic network, we find that the set of enzymes touching graphlets G_5 at the remaining orbits (orbit 10 or 12), is statistically significantly enriched in GO terms listed in Table C.16.

For the enzymes that are annotated with cyclic nucleotide metabolic process and that touch graphlets G_2 at orbit 6 in *M. musculus* metabolic network, we find that the set of enzymes touching these graphlets at orbit 5 is statistically significantly enriched in GO terms listed in Table C.17.

GO term	p-value
nucleotide catabolic process	0.000151048912701
nucleotide biosynthetic process	$2.41337586149\times10^{-6}$
nucleoside biosynthetic process	0.000373047937047
nucleoside diphosphate kinase activity	$1.97584948403\times10^{-10}$
nucleoside triphosphate metabolic process	$9.10172697846\times10^{-5}$
nucleoside triphosphate biosynthetic process	$6.09824402275\times10^{-5}$
ribonucleotide catabolic process	0.000184385388112
adenyl ribonucleotide binding	$7.46242449733\times10^{-5}$
transcription, DNA-templated	$2.28743024522\times10^{-9}$
RNA biosynthetic process	$1.88354098984\times10^{-9}$
nucleoside diphosphate metabolic process	0.00021598566696
adenyl nucleotide binding	$7.46242449733\times10^{-5}$
positive regulation of immune response	$1.27496842386\times10^{-5}$
regulation of type I interferon production	$1.59296817441\times10^{-5}$
positive regulation of cytokine production	0.00581366612092
regulation of innate immune response	0.000102160609657
nucleotide metabolic process	$3.64105515738\times10^{-7}$
ribonucleoside metabolic process	0.000118203526353
adenylate kinase activity	0.00746712993431
positive regulation of defense response	0.000533315242821
regulation of defense response	0.003773549597
purine nucleotide metabolic process	$6.38709502288\times10^{-7}$
DNA-directed RNA polymerase II, core complex	$1.97584948403\times10^{-10}$
DNA-directed RNA polymerase III complex	$8.7725930964\times10^{-9}$
nucleotide-excision repair, DNA gap filling	0.00411546724079
RNA polymerase activity	0.00411546724079
nucleoside monophosphate metabolic process	$9.10172697846\times10^{-5}$
phosphoric diester hydrolase activity	0.00848082409569
ribonucleotide metabolic process	$9.28383592225\times10^{-9}$
ribose phosphate biosynthetic process	$5.89233961904\times10^{-5}$
purine nucleoside metabolic process	0.00122950274736
phosphorylation	0.00588663490718
nucleoside monophosphate biosynthetic process	0.000950682855103
pyrimidine nucleotide metabolic process	0.00162483187895
myeloid cell differentiation	0.00411546724079
nucleobase-containing compound kinase activity	$6.01778139098\times10^{-7}$
purine-containing compound catabolic process	0.000749501000677
ubiquitin protein ligase binding	0.00746712993431

Table C.9. GO term enrichment of enzymes touching orbit 5 in *H. sapiens* metabolic network when the enzymes touching orbit 6 of the same graphlet are annotated with ribose phosphate metabolic process. First column: GO term. Second column: p-value of the enrichment.

GO term	p-value
nucleotide catabolic process	$5.82108330626\times10^{-5}$
nucleotide biosynthetic process	$4.54814658157\times10^{-7}$
nucleoside biosynthetic process	0.000167340542799
nucleoside diphosphate kinase activity	$6.36134478427\times10^{-11}$
nucleoside triphosphate metabolic process	$3.45545594969\times10^{-5}$
nucleoside triphosphate biosynthetic process	$2.61766391187\times10^{-5}$
ribonucleotide catabolic process	$7.61235419343\times10^{-5}$
adenyl ribonucleotide binding	0.00014931637621
transcription, DNA-templated	$6.75332678668\times10^{-10}$
positive regulation of cytokine production	0.00320341363031
RNA biosynthetic process	$5.0968240739\times10^{-10}$
nucleoside diphosphate metabolic process	$9.54776282008\times10^{-5}$
adenyl nucleotide binding	0.00014931637621
positive regulation of immune response	$6.18136753416\times10^{-6}$
regulation of type I interferon production	$9.1005857763\times10^{-6}$
ribose phosphate biosynthetic process	$2.3614712782\times10^{-5}$
regulation of innate immune response	$6.42005030653\times10^{-5}$
nucleotide metabolic process	$2.16929898489\times10^{-8}$
ribonucleoside metabolic process	0.000156609960835
adenylate kinase activity	0.00529124655973
positive regulation of defense response	0.000339700080058
regulation of defense response	0.00246848948203
purine nucleotide metabolic process	$9.37471139606\times10^{-8}$
DNA-directed RNA polymerase II, core complex	$6.36134478427\times10^{-11}$
DNA-directed RNA polymerase III complex	$3.39381722725\times10^{-9}$
nucleotide-excision repair, DNA gap filling	0.00312174942972
RNA polymerase activity	0.00312174942972
nucleoside monophosphate metabolic process	$3.45545594969\times10^{-5}$
phosphoric diester hydrolase activity	0.00449836362002
ribonucleotide metabolic process	$1.01764596749\times10^{-9}$
purine nucleoside metabolic process	0.0018640475526
nucleoside monophosphate biosynthetic process	0.000529484483356
pyrimidine nucleotide metabolic process	0.00104888931866
nucleobase-containing compound kinase activity	$1.23830740106\times10^{-7}$
purine-containing compound biosynthetic process	0.00729209044188
purine-containing compound catabolic process	0.000323450296654

Table C.10. GO term enrichment of enzymes touching orbits 10 or 12 in *H. sapiens* metabolic network when the enzymes touching orbit 11 of the same graphlet are annotated with ribose phosphate metabolic process. First column: GO term. Second column: p-value of the enrichment.

GO term	p-value
nucleotide catabolic process	0.000268745038165
nucleotide biosynthetic process	$3.1135982681\times10^{-11}$
nucleoside biosynthetic process	0.000422830941328
cilium movement	0.00128606728303
nucleoside triphosphate metabolic process	$4.20379231425\times10^{-9}$
nucleoside triphosphate catabolic process	$7.88684960144\times10^{-5}$
deoxyribonucleotide catabolic process	0.000626510099194
deoxyribonucleotide biosynthetic process	0.000626510099194
DNA-directed RNA polymerase I complex	$1.43920623612\times10^{-5}$
deoxyribose phosphate catabolic process	0.00473382212615
nucleoside diphosphate metabolic process	0.00674829929982
ribose phosphate biosynthetic process	$1.13397687737\times10^{-5}$
purine nucleoside metabolic process	0.00654776811725
nucleotide metabolic process	$1.00042196749\times10^{-12}$
ribonucleoside metabolic process	0.000509450579749
purine nucleotide metabolic process	$1.64278590731\times10^{-11}$
ribonucleotide metabolic process	$1.23028537691\times10^{-6}$
nucleoside monophosphate metabolic process	$9.32792813169\times10^{-6}$
nucleoside monophosphate biosynthetic process	0.000614324688629
nucleoside monophosphate catabolic process	0.00473382212615
nucleobase-containing compound kinase activity	0.00172179330687
purine-containing compound biosynthetic process	$6.16755195804\times10^{-8}$
2'-deoxyribonucleotide metabolic process	$3.4852049724\times10^{-5}$

Table C.11. GO term enrichment of enzymes touching orbit 5 in *M. musculus* metabolic network when the enzymes touching orbit 6 of the same graphlet are annotated with ribose phosphate metabolic process. First column: GO term. Second column: p-value of the enrichment.

GO term	p-value
nucleotide catabolic process	0.000224902995547
nucleotide biosynthetic process	$2.10231831943\times10^{-12}$
nucleoside biosynthetic process	$4.63487893747\times10^{-5}$
cilium movement	0.0011942323281
nucleoside triphosphate metabolic process	$3.1825688751\times10^{-9}$
nucleoside triphosphate catabolic process	$6.97673673813\times10^{-5}$
deoxyribonucleotide biosynthetic process	0.000568547733544
deoxyribonucleotide catabolic process	0.000568547733544
DNA-directed RNA polymerase I complex	$1.27003708446\times10^{-5}$
deoxyribose phosphate catabolic process	0.00440531978569
purine nucleoside metabolic process	0.00151141698945
nucleoside diphosphate metabolic process	0.00616618511956
ribose phosphate biosynthetic process	$9.90601465856\times10^{-7}$
adenylyltransferase activity	0.000223596825416
nucleotide metabolic process	$3.08419956241\times10^{-13}$
ribonucleoside metabolic process	$8.12021820856\times10^{-5}$
purine nucleotide metabolic process	$1.59239288422\times10^{-12}$
ribonucleotide metabolic process	$1.23845618205\times10^{-7}$
nucleoside monophosphate metabolic process	$7.5416965043\times10^{-6}$
nucleoside monophosphate biosynthetic process	0.000546062855761
nucleoside monophosphate catabolic process	0.00440531978569
nucleobase-containing compound kinase activity	0.00156610846777
purine-containing compound biosynthetic process	$5.08498365637\times10^{-9}$
purine-containing compound catabolic process	0.00963133376838
2'-deoxyribonucleotide metabolic process	$3.01172053138\times10^{-5}$

Table C.12. GO term enrichment of enzymes touching orbits 10 or 12 in *M. musculus* metabolic network when the enzymes touching orbit 11 of the same graphlet are annotated with ribose phosphate metabolic process. First column: GO term. Second column: p-value of the enrichment.

GO term	p-value
ribonucleotide metabolic process	0.00457072512321
purine nucleotide metabolic process	0.00457072512321
DNA-directed RNA polymerase II, core complex	0.00064549982473
nucleotide metabolic process	0.000767128336139

Table C.13. GO term enrichment of enzymes touching orbit 5 in *D. melanogaster* metabolic network when the enzymes touching orbit 6 of the same graphlet are annotated with ribose phosphate metabolic process. First column: GO term. Second column: p-value of the enrichment.

GO term	p-value
nucleotide biosynthetic process	0.00149791055107
ribonucleotide metabolic process	0.000670604413007
purine nucleotide metabolic process	0.000670604413007
DNA-directed RNA polymerase II, core complex	0.000712275668912
nucleotide metabolic process	0.000122799296242

Table C.14. GO term enrichment of enzymes touching orbits 10 or 12 in *D. melanogaster* metabolic network when the enzymes touching orbit 11 of the same graphlet are annotated with ribose phosphate metabolic process. First column: GO term. Second column: p-value of the enrichment.

GO term	p-value
adenylate kinase activity	0.000692977590416
nucleotide catabolic process	0.000714939731469
nucleotide biosynthetic process	$9.00436690527\times10^{-5}$
regulation of defense response	0.000197947467289
response to macrophage colony-stimulating factor	0.00747890606934
positive regulation of defense response	$2.43930059655\times10^{-5}$
cellular response to cGMP	0.00747890606934
purine nucleotide metabolic process	$5.25831408771\times10^{-6}$
DNA-directed RNA polymerase II, core complex	$7.29749594086\times10^{-13}$
DNA-directed RNA polymerase III complex	$1.22880594589\times10^{-11}$
nucleoside triphosphate metabolic process	0.00291343081481
nucleoside triphosphate biosynthetic process	0.000269648844699
nucleoside diphosphate kinase activity	$7.59053930821\times10^{-10}$
regulation of type I interferon production	$3.54009670578\times10^{-7}$
ribonucleotide catabolic process	0.00165540148065
leukocyte differentiation	0.00747890606934
ribose phosphate biosynthetic process	0.00509385926732
transcription, DNA-templated	$7.93587418002\times10^{-13}$
positive regulation of cytokine production	0.00382602665904
RNA biosynthetic process	$9.7428731749\times10^{-12}$
nucleoside monophosphate metabolic process	$9.59766265107\times10^{-6}$
ribonucleotide metabolic process	$3.05550410928\times10^{-6}$
nucleoside monophosphate biosynthetic process	0.000213699454149
purine nucleoside metabolic process	0.0063012644724
nucleoside diphosphate metabolic process	0.000605731832889
positive regulation of immune response	0.00077471919838
myeloid cell differentiation	0.000633642875923
nucleobase-containing compound kinase activity	$7.15134178209\times10^{-7}$
regulation of innate immune response	$4.36051067398\times10^{-6}$
nucleotide metabolic process	$2.02063799237\times10^{-6}$
ribonucleoside metabolic process	0.00428577005878

Table C.15. GO term enrichment of enzymes touching orbit 5 in *H. sapiens* metabolic network when the enzymes touching orbit 6 of the same graphlet are annotated with cyclic nucleotide metabolic process. First column: GO term. Second column: p-value of the enrichment.

GO term	p-value
adenylate kinase activity	0.000579914517673
nucleotide catabolic process	0.000519606088899
nucleotide biosynthetic process	$5.40359271366\times10^{-5}$
regulation of defense response	0.000158476250664
response to macrophage colony-stimulating factor	0.00682885189876
positive regulation of defense response	$1.93953590195\times10^{-5}$
cellular response to cGMP	0.00682885189876
purine nucleotide metabolic process	$2.72021858994\times10^{-6}$
DNA-directed RNA polymerase II, core complex	$1.22757359833\times10^{-12}$
DNA-directed RNA polymerase III complex	$7.56383844447\times10^{-12}$
nucleoside triphosphate metabolic process	0.00222392157498
nucleoside triphosphate biosynthetic process	0.00020076826701
nucleoside diphosphate kinase activity	$4.71678807124\times10^{-10}$
ribonucleotide catabolic process	0.00125471153481
leukocyte differentiation	0.00682885189876
ribose phosphate biosynthetic process	0.00403720368682
transcription, DNA-templated	$1.00697228334\times10^{-12}$
purine nucleoside metabolic process	0.00473063285663
RNA biosynthetic process	$5.49427170426\times10^{-12}$
nucleoside monophosphate metabolic process	$6.2309284623\times10^{-6}$
ribonucleotide metabolic process	$1.56794999617\times10^{-6}$
nucleoside monophosphate biosynthetic process	0.00016472712032
nucleoside diphosphate metabolic process	0.000454238515502
positive regulation of immune response	0.00062448193648
regulation of type I interferon production	$2.67152766131\times10^{-7}$
positive regulation of cytokine production	0.00302195801386
nucleobase-containing compound kinase activity	$3.88908336979\times10^{-7}$
purine-containing compound biosynthetic process	0.00871990785067
regulation of innate immune response	$3.45518869505\times10^{-6}$
nucleotide metabolic process	$7.60718349313\times10^{-7}$
ribonucleoside metabolic process	0.00310758973214

Table C.16. GO term enrichment of enzymes touching orbits 10 or 12 in *H. sapiens* metabolic network when the enzymes touching orbit 11 of the same graphlet are annotated with cyclic nucleotide metabolic process. First column: GO term. Second column: p-value of the enrichment.

GO term	p-value
long-term memory	0.00413160714693
nucleotide catabolic process	0.00422811604743
nucleotide biosynthetic process	0.000694616775687
cilium movement	0.00025788251794
negative regulation of embryonic development	0.00413160714693
purine nucleotide metabolic process	$1.04705806236\times10^{-6}$
nucleoside triphosphate metabolic process	0.00968562286096
regulation of ERBB signaling pathway	0.00413160714693
regulation of catecholamine metabolic process	0.00413160714693
ribose phosphate biosynthetic process	$5.88657444562\times10^{-5}$
nucleoside monophosphate metabolic process	$2.31287462396\times10^{-5}$
ribonucleotide metabolic process	$2.78588064939\times10^{-6}$
nucleoside monophosphate catabolic process	0.000984205886121
DNA-directed RNA polymerase I complex	$9.45224824966\times10^{-7}$
nucleoside diphosphate metabolic process	0.000907524707261
nucleoside salvage	0.00234787559214
pyruvate kinase activity	0.00413160714693
nucleobase-containing compound kinase activity	0.00448126108738
purine-containing compound biosynthetic process	$4.91084484089\times10^{-6}$
nucleotide metabolic process	$2.17323301732\times10^{-6}$

Table C.17. GO term enrichment of enzymes touching orbit 5 in *M. musculus* metabolic network when the enzymes touching orbit 6 of the same graphlet are annotated with cyclic nucleotide metabolic process. First column: GO term. Second column: p-value of the enrichment.

GO term	p-value
long-term memory	0.00396297012122
nucleotide catabolic process	0.000500120828581
nucleotide biosynthetic process	0.000590499578823
cilium movement	0.000242093792955
purine nucleotide metabolic process	$9.95288754746\times10^{-8}$
nucleoside triphosphate metabolic process	0.00109439493541
regulation of ERBB signaling pathway	0.00396297012122
regulation of catecholamine metabolic process	0.00396297012122
ribonucleotide catabolic process	0.00489106978343
ribose phosphate biosynthetic process	$5.02487646896\times10^{-5}$
nucleoside monophosphate metabolic process	$2.00447404655\times10^{-5}$
ribonucleotide metabolic process	$2.54199133409\times10^{-7}$
nucleoside monophosphate catabolic process	0.000924914183122
DNA-directed RNA polymerase I complex	$8.48773565432\times10^{-7}$
nucleoside diphosphate metabolic process	0.000837159004529
nucleoside salvage	0.00220872562233
pyruvate kinase activity	0.00396297012122
nucleobase-containing compound kinase activity	0.00422003021046
purine-containing compound biosynthetic process	$3.86233719252\times10^{-6}$
nucleotide metabolic process	$2.55472766386\times10^{-7}$

Table C.18. GO term enrichment of enzymes touching orbits 10 or 12 in *M. musculus* metabolic network when the enzymes touching orbit 11 of the same graphlet are annotated with cyclic nucleotide metabolic process. First column: GO term. Second column: p-value of the enrichment.

For the enzymes annotated with cyclic nucleotide metabolic process and that touch graphlets G_5 at orbit 11 in *M. musculus* metabolic network, we find that the set of enzymes touching graphlets G_5 at the remaining orbits (orbit 10 or 12), is statistically significantly enriched in GO terms listed in Table C.18.

For the enzymes that are annotated with cyclic nucleotide metabolic process and that touch graphlets G_2 at orbit 6 in *D. melanogaster* metabolic network, we find that the set of enzymes touching these graphlets at orbit 5 is statistically significantly enriched in GO terms listed in Table C.19.

For the enzymes annotated with cyclic nucleotide metabolic process and that touch graphlets G_5 at orbit 11 in *D. melanogaster* metabolic network, we find that the set of enzymes touching graphlets G_5 at the remaining orbits (orbit 10 or 12), is statistically significantly enriched in GO terms listed in Table C.20.

GO term	p-value
DNA-directed RNA polymerase II, core complex	0.000524728889933
nucleotide biosynthetic process	0.00858347329687
ribonucleotide metabolic process	0.00334584034794
DNA integrity checkpoint	0.00881651174264
DNA damage checkpoint	0.00881651174264
nucleotide metabolic process	0.000488867328955
purine nucleotide metabolic process	0.00334584034794

Table C.19. GO term enrichment of enzymes touching orbit 5 in *D. melanogaster* metabolic network when the enzymes touching orbit 6 of the same graphlet are annotated with cyclic nucleotide metabolic process. First column: GO term. Second column: p-value of the enrichment.

GO term	p-value
DNA-directed RNA polymerase II, core complex	0.000524728889933
nucleotide biosynthetic process	0.00858347329687
ribonucleotide metabolic process	0.00334584034794
DNA integrity checkpoint	0.00881651174264
DNA damage checkpoint	0.00881651174264
nucleotide metabolic process	0.000488867328955
e- purine nucleotide metabolic process	0.00334584034794

Table C.20. GO term enrichment of enzymes touching orbits 10 or 12 in *D. melanogaster* metabolic network when the enzymes touching orbit 11 of the same graphlet are annotated with cyclic nucleotide metabolic process. First column: GO term. Second column: p-value of the enrichment.