Chapter 5
Biological Network Alignment

Shahin Mohammadi and Ananth Grama

Abstract Recent experimental approaches to high-throughput screening, combined with effective computational techniques have resulted in large, high-quality databases of biochemical interactions. These databases hold the potential for fundamentally enhancing our understanding of cellular processes and for controlling them. Recent work on analyses of these databases has focused on computational approaches for aligning networks, identifying modules, extracting discriminating and descriptive components, and inferring networks. In this chapter, we focus on the problem of aligning a given set of networks with a view to identifying conserved subnetworks, finding orthologies, and elucidating higher level organization and evolution of interactions. Network alignment, in general, poses significant computational challenges, since it is related to the subgraph isomorphism problem (which is known to be computationally expensive). For this reason, effective computational techniques focus on exploiting structure of networks (and their constituent elements), alternate formulations in terms of underlying optimization, and on the use of additional data for simplifying the alignment process. We present a comprehensive survey of these approaches, along with important algorithms for various formulations of the network alignment problem.

1 Introduction

The emergence of high-throughput screening techniques coupled with computational approaches to network reconstruction and inference, have resulted in large databases of biochemical interactions. These interactions can be effectively

A. Grama (✉)
Department of Computer Science, Purdue University, 305 N. University Street,
West Lafayette, IN 47907-2107, USA
e-mail: ayg@purdue.edu

M. Koyutürk et al. (eds.), *Functional Coherence of Molecular Networks in Bioinformatics*, DOI 10.1007/978-1-4614-0320-3_5,

analyzed to gain novel insights into cellular processes, and identify suitable approaches to controlling these processes. One such analysis technique relies on aligning multiple networks with a view to understanding the conserved functional and organizational principles of biological systems.

The problem of network alignment takes as input one or more networks and establishes correspondences between nodes in the network(s) that optimize a given objective function. Here, the objective function is designed to reflect the conservation of interactions across two or more species. Such analysis for sequences (amino-acid or nucleotide sequences) has been used to great effect in understanding the structure and function of biomolecules. The basic idea that conserved subsequences are likely to share structure, function, and evolutionary trajectories provides the basis for large classes of computational techniques. Network alignment can similarly be used for identifying functionally coherent machinery – "shared function is likely to reflect in aligned subcomponents," and vice-versa. This principle can be used to "project" or "transfer" interaction machinery across organisms that share corresponding function, and to identify latent orthologies among constituent elements. Building further on this premise, alignment can also provide valuable insights into evolutionary trajectories and specialization. As more interaction databases become available, network alignment provides an essential tool for identifying descriptive (and discriminative) components corresponding to the phenotype. Clearly, network alignment in its various forms discussed in this chapter, is an important analysis tool for biochemical pathways.

Given extensive computational infrastructure for sequence alignment, it is natural to examine the relationship between sequence and network alignment [1, 2]. In this context, the two key questions relate to models and methods. Models provide a formal framework for alignment problems – namely, they quantify the fitness of an alignment (when one alignment is better than the other) and its significance (how likely is an alignment to correspond to biologically relevant artifacts). Methods, on the other hand, use these models to arrive at desirable alignments and their significance scores. For sequence analysis, BLAST is one of the most commonly used alignment methods, which relies on statistical measures like p-values to quantify significance of alignments. It is easy to see that sequences are special cases of networks – networks with linear connectivity. It follows then that the problem of alignment of *general* networks is at least as hard as sequence alignment (recall that most formulations of multiple sequence alignment are classified into the family of NP-Hard problems – problems for which subexponential time algorithms are not known).

An instance of the network alignment problem for two networks (or a network with itself) is the subgraph isomorphism problem. This problem relates to the identification of the largest common subnetwork of the two given networks. The subgraph isomorphism problem is known to belong to the class of NP-Hard problems as well. The consequent exponential time complexity of solving this problem renders general combinatorial approaches to solving this problem intractable for biochemical networks of interest. Consequently, the problem has motivated a rich class of models and methods that rely on applications' characteristics to solve the problem.

A second important aspect of the problem relates to quantification of significance values associated with alignments. The significance of an alignment quantifies the likelihood of obtaining the quality of an alignment by chance only. The smaller the likelihood, the more significant (hence, more likely to be biologically relevant) the alignment. Traditional approaches to quantifying significance rely either on analytical formulations or on simulations. The state-of-the-art in analytical modeling of networks is in relative infancy. Simulation based methods, on the other hand, suffer from slow convergence and high computational cost. Consequently, quantification of significance poses intriguing challenges, that continue to be investigated.

In the rest of this chapter, we will provide an overview of models, methods, validation techniques, and key data sources for alignment of biochemical networks.

2 Definitions and Notations

Biological networks are often modeled as graphs consisting of vertices (or nodes) and edges (or arcs). Formally, a graph G is defined as $G = (V, E)$, where V is a finite set of vertices and E is a finite set of edges, such that $E \subseteq (V \times V)$. A graph can be either *directed* or *undirected*. In an undirected graph, edges define a symmetric relation among graph vertices, meaning that a relation between vertices v_i and v_j also implies the same relation between v_j and v_i. In directed graphs, relations are not implicitly symmetric. In a directed graph with an edge (v_i, v_j), we refer to v_i and v_j as the source and sink of the edge, respectively.

Graphs can be represented in different ways (i.e., using different data structures). While these representations are logically equivalent, depending on the operations on the graph, some are more computationally efficient than others. One of the commonly used representations is the node adjacency matrix – given a graph G with n nodes, we construct a matrix $\mathbf{A}$ of dimension $n \times n$, in which entry a_{ij} specifies whether there exists an edge between nodes v_i and v_j in G.

In many applications, one also needs to encapsulate additional information about vertices (entities) or edges (relations) of the input network. *Attributed graphs* allow for embedding such information in the graphs. An *edge attributed* (also known as *edge colored*) graph is the one in which the edges have additional information, while the *node attributed* (also known as *node colored*) graph has additional information about the nodes. *Edge weighted graphs* or simply *weighted graphs* are special cases of edge attributed graphs, in which every edge has a real valued attribute (or weight). These weights can be stored in the adjacency matrix of graph G by allowing a_{ij} to store the weight of edge $(v_i, v_j) \in E$. Another way to encapsulate node (or edge) attributes, which is especially useful if we have multiple attributes, is to attach vectors to graph vertices and/or edges.

There are different kinds of biological data that are represented using graphs. *Protein–protein interaction (PPI)* networks are often used in a variety of analyses tasks. In these networks, each node represents a protein and each edge indicates a physical interaction between a pair of proteins. A PPI network can be modeled using an undirected, weighted or unweighted, graph. In the former, the weight usually indicates the probability or confidence of the PPI.

Metabolic networks are often used to understand chemical compositions and reactions. There are two complementary representations of a metabolic network, both of which rely on directed graphs. In the first one, each vertex represents a chemical compound (substrate), and there is an edge between a pair of vertices if they occur (either as substrates or products) in the same chemical reaction. In the second representation, each vertex represents a chemical reaction catalyzed by an enzyme E_i, and there is an edge between any pair of vertices i and j, representing enzymes E_i and E_j, respectively, if they share at least one chemical compound, either as substrate or as product. In other words, if E_i catalyzes a reaction in which compound A is produced, and E_j takes A as a substrate.

Other data, relating to signaling, gene regulations, and lethal interactions are also modeled as graphs. Cell signaling corresponds to the basic communication network of a cell. It governs how a cell perceives and responds to its physio-chemical environment, regulates basic processes such as development, growth and repair (at the tissue level), response to stress, etc. Nodes in these networks correspond to biomolecules (or complexes thereof) and edges correspond to signals. Nodes and edges in these networks are typically labeled to indicate the spatial localization, nature of signals, and type of biomolecules. Gene regulatory networks (GRNs) represent the interactions between genes (through their respective products, which are often not explicitly annotated in the network). Individual nodes correspond to genes and edges correspond to their regulatory roles. An edge from node (gene) i to j implies a regulatory relationship. Since a regulatory link may be positive (up-regulation) or negative (down-regulation), edges are sometimes categorized into up-or down-regulatory edges. GRNs are often modeled as networks of reactions – each modeled using an ordinary differential equation (based on chemical kinetic models). In such networks, rate constants are used to annotate edges. Other models such as Boolean Networks (genes, or nodes are restricted to binary states, that is, they can be on or off, and edges change the state of downstream nodes) and Bayesian Networks (recognizing the stochastic nature of the regulation process).

More recently, data from synthetic genetic arrays have been represented as networks coding synthetic lethality. Synthetic lethality refers to the observation that a combination of two or more gene mutations leads to cell death, while a single mutation to either of these genes does not. In synthetic lethality networks, nodes correspond to genes and edges reflect the existence of a synthetic lethal interaction between the two genes.

2.1 Network Alignment Problems

Given a set of graphs $\mathcal{G} = \{G_1, G_2, \ldots, G_k\}$, an alignment corresponds to a proper mapping between the nodes of input networks that maximizes the similarity between mapped entities. The *pairwise network alignment problem* is a special case of this problem with two input networks. Network alignment, in its general form, is a computationally hard problem, since it can be related to the subgraph-isomorphism problem, which is known to be NP-complete. Effective techniques for solving this problem rely on suitable formulations of the alignment problem, use of heuristics to solve these problems, or on the use of alternate data to guide the alignment process.

At a high level, the network alignment problem can be classified as *local alignment* or *global alignment*. The former is a relationship over a subset of the nodes in $V = \{V_1 \cup V_2 \ldots \cup V_k\}$, while the latter is defined as partitioning all nodes in V into disjoint subsets, also known as equivalence classes [2]. Global network alignment can be further classified into *one-to-one*, in which every subset has exactly n nodes, one from each input network, and *many-to-many*, in which the subsets are not restricted to have exactly one node from each input network.

The biological interpretation of the local alignment problem is that each subset of aligned nodes represents a conserved module. In a global one-to-one alignment, nodes in each subset can be interpreted as functional orthologs, while in many-to-many network alignment, each subset is a classification of all possible functional orthologs in given species into an equivalence class.

We start by denoting the set of all possible alignments as $\mathscr{A}$. It is common to represent each network alignment $A \in \mathscr{A}$ using an *alignment graph*, $G_A = (V_A, E_A)$, where every node in the alignment graph represents an equivalence class, while each edge represents a relationship between a pair of equivalence classes. To define the network alignment problem formally, we also need to define an *alignment scoring function*, $\phi : \mathscr{A} \to \Re$, which assigns to each alignment $A \in \mathscr{A}$, a real fitness value. Given an alignment scoring function, the global network alignment problem is formally defined as finding the maximum score global network alignment A_{opt}, while the local network alignment problem is defined as finding a set of maximal score local network alignments. The core of any alignment algorithm consists of an *alignment scoring function* together with a *search, or optimization method*.

Before we discuss alignment algorithms, we also introduce a general form for the *node scoring function*, $S : \{V_1 \cup V_2 \ldots \cup V_k\} * \{V_1 \cup V_2 \ldots \cup V_k\} \to \Re$, which assigns a similarity score to each pair of nodes in the input networks. Different node similarity functions have been proposed, based on the node attributes, as well as the local network topology around each node. An example of the former case is the BLAST score of the protein sequences corresponding to a pair of given nodes, while an example of the latter case is the scoring function proposed by Kuchaiev et al. [3], in which they use a vector representing the number of *graphlets* that each node takes part, to compare the topological similarity around each node.

3 Algorithms and Methods

Alignment problems have been modeled as diverse optimization problems, based on the underlying applications. In this section, we describe the mathematical models underlying these variants of alignment problems and discuss algorithms for these problems. An important and difficult problem associated with these algorithms is their validation. This difficulty stems from the noisy, incomplete, and statistically skewed nature of underlying data. We conclude the discussion in this section with an overview of validation techniques and databases available for analyses and validation.

3.1 Local Alignment

Local alignment corresponds to a relationship defined over a subset of vertices in the input networks. It is often used to extract conserved substructures (modules, pathways, complexes) from a set of species. A number of algorithms have been proposed for local alignment. We provide an overview of these methods in this section.

3.1.1 The Blast Family: PathBlast, NetworkBlast, and NetworkBlast-M

PathBlast [4], proposed by Kelley et al. [5], was among the first attempts at network alignment, with the goal of identifying conserved pathways in a pair of species. The method identifies high-scoring alignments between pairs of pathways, one from each input network, such that proteins in the first pathway map to their putative homologs in the same order in the second pathway. To accomplish this, PathBlast initially builds an alignment graph (see Sect. 2), where edges can be either a *match*, *gap*, or a *mismatch* edge. Let v_i^1 and v_i^2 denote the nodes from first and second species, respectively, in the equivalence class represented by node v_i in the alignment graph. A *match* edge occurs between nodes v_i and v_j in the alignment graph when v_i^1 and v_j^1 are connected in the first species, and v_i^2 and v_j^2 are connected in the second species. Otherwise, it can be either a *mismatch*, or a *gap* edge. The former occurs when neither v_i^1 and v_j^1, nor v_i^2 and v_j^2 are connected in their corresponding species, and the latter occurs when only one of the protein pairs in one of the species are connected.

The core of the PathBlast algorithm is a log probability score for evaluating each pathway P in the alignment graph. This score is computed by decomposing the pathway similarity score into a vertex scoring fraction and an edge scoring fraction. More formally, the scoring function is defined as follows:

$$S(P) = \sum_{v \in P} \frac{p(v)}{p_{\text{random}}} + \sum_{e \in P} \frac{q(e)}{q_{\text{random}}}. \tag{5.1}$$

Here $p(v)$ represents the probability of true homology between the protein pair from input networks represented by node v in the alignment graph. The quantity $q(e)$ represents the probability that interactions represented by e are real interaction, not false positive interactions. Probabilities p_{random} and q_{random} are evaluated as the expected values of $p(v)$ and $q(e)$, respectively. Using this scoring function, one can find the optimal alignment as the one in which the pathway scoring function is optimized over all pathways up to length L for networks of size n using randomized dynamic programming. The method has an expected time complexity $O(nL!)$, if the input networks are acyclic (i.e., do not contain any cycle).

PathBLAST is available through a web-interface at http://www.pathblast.org/. A user may specify a short protein interaction network for query against a target PPI network from a network database. Protein interactomes of yeast (*Saccharomyces cerevisiae*), the bacterial pathogen (*Helicobacter pylori*), bacterium (*Escherichia coli*), nematode worm (*Caenorhabditis elegans*), fruit fly (*Drosophila melanogaster*), mouse (*Mus musculus*), and human (*Homo sapiens*) are available as target species. The program returns a ranked list of matching paths from the target network along with a graphical view of these paths and the associated overlap.

Sharan et al. [6] extend the idea of PathBlast for extracting conserved protein complexes from a pair of input networks. Their algorithm, NetworkBlast, allows extraction of *all* conserved complexes across networks, as opposed to the single query model of PathBlast. The resulting computational problem is more general and difficult. NetworkBlast has also been generalized to NetworkBlast-M [7] for identifying conserved networks among multiple networks.

Sharan et al. initially evaluate the reliability of PPI and build a weighted network by assigning a confidence value to each interaction. They propose a *logistic regression model*, based on the method proposed by Bader et al. [8], and use the following three random variables to define their logistic distribution:

X_1: Number of times an interaction between the proteins is experimentally observed
X_2: Pearson correlation coefficient of expression measurements for the corresponding genes
X_3: Proteins' small world clustering coefficient.

Using these random variables, the probability of a true interaction T_{uv} is defined as:

$$Pr(T_{uv}|X) = \frac{1}{1+\exp(-\beta_0 - \sum_{i=1}^{3} \beta_i X_i)}, \tag{5.2}$$

where $\beta_0, \ldots, \beta_3$ are parameters of the distribution [6]. They then build an alignment graph, in which each node corresponds to a group of k similar proteins, that is, proteins from different species with BLAST E-values smaller than 10^{-7}. Each edge in the alignment graph represents a conserved interaction between the proteins that occur in its end nodes. An edge is considered conserved if and only if one of the following conditions is met:

- A pair of proteins directly interacts, and all other pairs include proteins with distance at most two in their corresponding networks.
- All protein pairs have distance exactly two in their corresponding networks.
- At least max $\{2, k-1\}$ protein pairs directly interact.

Finally, they devise a scoring scheme based on a likelihood model to fit the subnetwork to the given structure. Given a subset U of the vertices, O_U denotes the collection of all observations on vertex pairs in U, and O_{uv} denotes the set of available observations on the proteins u and v, that is, the set of experiments in which an interaction between u and v was, or was not, observed. Also, let T_{uv} denote the event that two proteins u and v interact, and F_{uv} denote the event that they do not interact.

One may formalize the log-likelihood ratio of a subgraph under a conserved subnetwork model, M_s, and under a null model, M_n, in a single species, as follows:

$$L(U) = \log \frac{Pr(O_U|M_s)}{Pr(O_U|M_n)} = \sum_{(u,v)\in U*U} \log \frac{\beta Pr(O_{uv}|T_{uv}) + (1-\beta)Pr(O_{uv}|F_{uv})}{p_{uv}Pr(O_{uv}|T_{uv}) + (1-p_{uv})Pr(O_{uv}|F_{uv})}, \tag{5.3}$$

where β is a high probability of interaction under the clique model, while p_{uv} is the probability of interaction between proteins u and v under the null model (random graph with the same degree distribution). To find the log-likelihood ratio of multiple complexes across different species, one may sum the log-likelihoods for single species.

Using this scoring function, the problem of identifying conserved subnetworks reduces to one of finding high scoring subgraphs. This problem is known to be NP-hard. Consequently, they adopt a greedy approach to this problem, which is based on an extension of high scoring seeds, similar to the BLAST algorithm. NetworkBlast is available via a web interface at http://www.cs.tau.ac.il/~bnet/networkblast.htm. It can also be downloaded as a stand-alone program from the same website.

3.1.2 MaWISh: Alignment Based on Network Evolution Models

Koyutürk et al. [9,10] propose an evolution-based scoring function, which quantifies the evolutionary distance of any pair of induced subgraphs in the input networks. They use this scoring function to align the input networks (see Box 5.1 for a detailed explanation of their scoring scheme). They reduce the local alignment problem into a *maximum weight induced subgraph problem* (MaWISh). Noting the NP-completeness of this problem by reduction from max-clique, they propose a greedy approach to approximate the solution. They initially match the *hub nodes* and iteratively expand the subgraph in the sparse product graph by adding nodes that share a matching edge with these nodes, to maximize their scoring function.

Koyutürk et al. [14] extend their method to multiple networks, by contracting the global alignment graph and then applying algorithms from frequent itemset extraction. The MaWISh software is currently available for download from http://compbio.case.edu/koyuturk/software/mawish.tar.gz.

Box 5.1: MAWISH network evolution based scoring scheme

A common model of evolution that explains preferential attachment is the duplication/divergence model, which is based on gene duplications [11–13]. According to this model, when a gene is duplicated in the genome, the node corresponding to the product of this gene is also duplicated together with its interactions. A protein loses many aspects of its functions rapidly after being duplicated. This translates to divergence of duplicated (paralogous) proteins in the interactome through elimination and emergence of interactions. Elimination of an interaction in a PPI network implies the loss of an interaction between two proteins due to structural and/or functional changes. Similarly, emergence of an interaction in a PPI network implies the introduction of a new interaction between two noninteracting proteins, caused by mutations that change protein surfaces. Examples of duplication, elimination, and emergence of interactions are illustrated in Fig. 5.1.

Using the duplication/divergence model, Koyutürk et al. [9, 10] propose a novel evolution-based scoring function. Given PPI networks $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$, a *protein subset pair* $P = \{S_1, S_2\}$ is defined as a pair of protein subsets $S_1 \subseteq V_1$ and $S_2 \subseteq V_2$. Given a pair of graphs G_1 and G_2, any protein subset pair P induces a local alignment $A(G_1, G_2, P) = \{\mathscr{M}, \mathscr{N}, \mathscr{D}\}$ of G_1 and G_2 with respect to similarity score function S (see Sect. 2), characterized by a set of duplications $\mathscr{D}$, a set of matches $\mathscr{M}$, and a set of mismatches $\mathscr{N}$. The biological analog of a *duplication* is the duplication of a gene in the course of evolution. Each duplication is associated with a score that reflects the divergence of function between the two proteins, estimated using their similarity. A *match* corresponds to a conserved interaction between two orthologous protein pairs, which is rewarded by a match score that reflects our confidence in both protein pairs being orthologous. A *mismatch*, on the other hand, is the lack of an interaction in the PPI network of one organism between a pair of proteins whose orthologs interact in the other organism. A mismatch may correspond to the emergence of a new interaction or the elimination of a previously existing interaction in one of the species after the

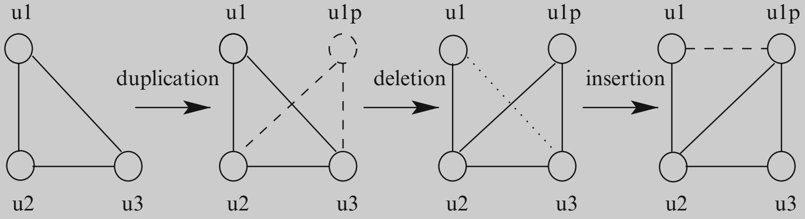

Fig. 5.1. Evolutionary events, and their effects on network topology

(continued)

Box 5.1 (continued)

split, or an experimental error. Thus, mismatches are penalized to account for the divergence from the common ancestor. The formal definitions of these three concepts are as follows:

Definition 5.1. Match, Mismatch, and Duplication Given protein interaction networks $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$, and a pairwise similarity function S, any protein subset pair $P = (S_1, S_2)$, induces a local alignment $A(G_1, G_2, P) = \{\mathscr{M}, \mathscr{N}, \mathscr{D}\}$, where:

$$\begin{aligned}\mathscr{M} &= \big\{u,v \in V_1, u',v' \in V_2 : 0 < S\left(u,u'\right),\ 0 < S\left(v,v'\right),\\ &\qquad (u,v) \in E_1 \wedge \left(u',v'\right) \in E_2\big)\big\} \qquad (5.4)\\ \mathscr{N} &= \big\{u,v \in V_1, u',v' \in V_2 : 0 < S\left(u,u'\right),\ 0 < S\left(v,v'\right),\\ &\qquad \big((u,v) \in E_1 \wedge \left(u',v'\right) \notin E_2\big) \vee \big((u,v) \notin E_1 \wedge (u',v') \in E_2\big)\big\} \qquad (5.5)\\ \mathscr{D} &= \{u,v \in V_1 : 0 < S(u,v)\} \cup \big\{u',v' \in V_2 : 0 < S\left(u',v'\right)\big\} \qquad (5.6)\end{aligned}$$

Matches $M \in \mathscr{M}$, mismatches $N \in \mathscr{N}$, and duplications $D \in \mathscr{D}$ are associated with scores $\mu(M)$, $\nu(N)$, and $\delta(D)$, respectively. Using this formulation of match, mismatch, and duplication, the evolutionary plausible scoring function to evaluate each network alignment can be defined as follows:

Definition 5.2. Alignment Score Given PPI networks G_1 and G_2, the score of alignment $A(G_1, G_2, P) = \{\mathscr{M}, \mathscr{N}, \mathscr{D}\}$ is defined as:

$$\sigma(A) = \sum_{M \in \mathscr{M}} \mu(M) - \sum_{N \in \mathscr{N}} \nu(N) - \sum_{D \in \mathscr{D}} \delta(D). \qquad (5.7)$$

Equation (5.7) can be used to evaluate the evolutionary distance of any given subset pair in the input networks.

3.1.3 Graemlin: Alignment with Equivalence Classes

Flannick et al. [2] propose an alternate method, Graemlin, which improves over previous methods by using heuristics from sequence alignment. They propose a formulation of network alignment, based on *equivalence classes*. In this model, the network alignment problem is posed as follows: given a set of input networks, a network alignment is defined as a set of subgraphs together with a symmetric mapping between the corresponding (aligned) vertices. For the alignment to be unique, this mapping should be transitive, meaning that $A \leftrightarrow B, B \leftrightarrow C \Rightarrow A \leftrightarrow C$; mathematically, such a symmetric-transitive relation is also known as equivalence relation. This definition classifies the aligned vertices into disjoint groups (*equivalence classes*). Each equivalence class consists of proteins evolved from a common ancestral protein, and unlike previous definitions, can contain multiple proteins in same species, also known as paralogs. This formulation allows them to modify the *progressive alignment* method adapted from sequence alignment, and to be able to scale linearly in the number of the networks compared. They also use a heuristic similar to *seed extension* in sequence alignment, to align the input networks efficiently, and to be able to trade-off speed versus sensitivity.

Using this formulation, Flannick et al. propose a scoring function composed of two parts, one to evaluate each equivalence class, and the other to evaluate each edge in the alignment. The former is more straightforward, while the latter is more involved, but provides the opportunity to search for arbitrary module structures. The scoring scheme is similar in both: find the probability distribution defined for two different models, namely the constrained alignment model $\mathscr{M}$ based on a given module structure and random model $\mathscr{R}$, and define the score function as the log-ratio of two probabilities. Equation (5.8) presents the Graemlin scoring function.

$$S = S_c + S_e, \text{ where } \begin{cases} S_c = \log\left(\frac{P_{\mathscr{M}}(c)}{P_{\mathscr{R}}(c)}\right) \\ S_e = \log\left(\frac{P_{\mathscr{M}}(e)}{P_{\mathscr{R}}(e)}\right) \end{cases} \tag{5.8}$$

Scoring of equivalence classes is based on construction of the most parsimonious ancestral history of the proteins in each equivalence class. This construction is based on sequence mutations, insertions, deletions, duplication, and divergence among proteins in each class. The probability of sequence mutations is estimated in a principled manner in their study; other events are determined heuristically. The alignment model M is trained by sampling pairs of proteins from within the same COG [15] group, while the random model R corresponds to picking random pairs in the network (see Flannick et al. [2] supplementary material for a detailed description).

Scoring of alignment edges is based on the concept of an *Edge Scoring Matrix (ESM)*, a symmetric matrix defined over a set of alphabets, Σ, in which every entry in the matrix is a probability distribution over edge weights. Graemlin first assigns alphabets to each equivalence class, then it scores each alignment edge using the cell in the ESM index by the labels assigned to two endpoints of the edge. This approach extends the previous methods in that it is capable of searching for conserved substructures with user-defined structure, not just pathways or complexes.

The next two steps use the score function to align a pair of networks, and to extend this approach to multiple alignment. Graemlin mimics the *seed extension* method, meaning that it tries to find a proper set of candidate seed vertices, and then extends them greedily. Unlike MaWISh, the seed vertices are chosen in a way that does not impose special topology (clique-like) on the subgraph structure. Seed selection in Graemlin is based on the concept of *d-clusters*, it first selects d-clusters for each node by finding $d-1$ nearest neighbors, where the distance between vertices is defined as the negative log of edge weights. It then finds the pairwise node similarity score of sample mappings between two d-clusters, one from each species, and reports the highest score among them. The d-clusters with mapping scores higher than the user defined threshold T are used as *seeds*. Parameters d and T are adjustable parameters that can be used to trade-off speed versus sensitivity in the algorithm. After computing the seeds, Graemlin greedily expands each equivalence class by coalescing vertices in the frontier of each equivalence class.

An extension of this approach to multiple alignment using an analog of the *progressive alignment* technique, commonly used in sequence alignment. Having an extended phylogenetic tree with species on the leaves, the technique successively aligns the closest pair of networks, and places three new networks in the parent node: one for the alignment network, and two other networks for unaligned subsets of the pair of networks.

Flannick et al. [2] construct ten weighted microbial PPI networks based on the SRINI algorithm [16]. These are publicly available at http://graemlin.stanford.edu/nets.tar.gz. Graemlin1.0 can be freely downloaded from http://graemlin.stanford.edu/graemlin-1.0.tar.gz as a stand-alone application.

3.1.4 Information Theoretic Network Alignment

Yet another method, motivated by information theory, is recently proposed by Chor et al. [17]. The fundamental idea in this method is to devise a computationally tractable measure that computes the disparity between two uniquely labelled graphs G_1 and G_2. This problem is then reduced to finding how many additional bits do we need to encode a graph G_2 given graph G_1 (known as description length of G_2 given G_1). To tackle this problem, Chor et al. impose the following key assumptions:

- *Shortest path conservation*: If a pair of nodes u and v are common in the vertex set of both networks, the length of the shortest path between them in the underlying graph of G_1 and G_2 must be similar.
- *Neighborhood conservation*: If a pair of nodes u in G_1 and v in G_2 are similar in some sense, like homolog proteins in PPI networks, but not identical, then the level one neighborhood of u and v must be highly similar.

Using these assumptions, they developed a measure, $D(G_2|G_1)$, which illustrates the number of additional bits needed for encoding the adjacency list of graph G_2 given graph G_1. This measure is not a distance metric, since it is clearly not symmetric. To devise a metric, they proposed the notion of *relative description length* as follows:

$$\mathrm{RDL}(G_1, G_2) = \frac{\mathrm{DL}(G_1|G_2)}{\mathrm{DL}(G_1)} + \frac{\mathrm{DL}(G_2|G_1)}{\mathrm{DL}(G_2)} \qquad (5.9)$$

Armed with the RDL metric, which computes the distance between graphs using an information theoretic method, they tackled the problem of finding conserved regions in networks. Conserved regions are defined as specific vertex-induced subgraphs in each network. More precisely, they first extracted pairs of "similar" nodes in networks, and then used this vertex set to induce subgraphs in corresponding input networks. To find the set of conserved nodes, they started with the set of common vertices, $V' = V_1 \cap V_2$, and proceed by comparing the level d neighborhood of each node $v \in V'$ in networks G_1 and G_2 using RDL metric. Any node that the RDL distance of its level d neighborhoods in G_1 and G_2 exceeds a threshold c will be filtered out from V'. Using V', edge sets E'_1 and E'_2 can be easily found by imposing V' on G_1 and G_2, respectively, and finding the induced subgraph in each network.

Chor et al. [17] successfully apply their method to both metabolic pathways extracted from KEGG database, and on a pair of PPI networks. Since PPI networks do not have unique labeling among networks, they use a heuristic to label the nodes. They define identical nodes in input networks as pairs of nodes in which the BLAST scores of their corresponding proteins have E-value $< \mathrm{e}^{-10}$. This is similar in nature to pruning the state space of mappings from beginning of the algorithm to a very small subset of total possible mappings, namely the most promising ones.

3.1.5 Network Queries: A Supervised Approach to the Network Alignment Problem

Network alignment and integration are focused on de novo discovery of biologically significant regions embedded in a network, based on the assumption that regions supported by multiple networks are functional. In contrast, a supervised approach to conserved module detection relies on a query subnetwork that is previously known to be functional. The objective of such methods is to identify subnetworks in a given network that are similar to the query. Among these methods, MetaPathwayHunter aims to identify metabolic pathways that match a query pathway in a database of pathways [18]. Similarly, Narayanan and Karp [19] aim to find matching pathways in PPI networks based on a match-and-split strategy. Bruckner et al. [20] propose a novel method, named Torque (TOpology-free netwoRk QUErying), which unlike most of the previous methods, does not restrict the topology of query network. Finally, Banks et al. [21] propose an extension of regular expressions on strings to networks, named *network scheme*.

3.2 Global Alignment

Global alignment algorithms aim to find a consistent relationship defined over *all* vertices of the input networks. Global alignment is commonly used to establish functional orthologs across species. A number of models and methods have been proposed for global alignment of networks.

3.2.1 Markov Random Field

One of the early efforts at global alignment of protein interaction networks is due to Bandyopadhyay et al. [22]. This study aims at solving the ambiguity in Inparanoid clusters with more than two proteins, to increase the accuracy of functional ortholog prediction. It is based on the idea that early paralogous proteins (out-paralogs) are more likely to change their interaction patterns and adopt new functions in the cell (for more information, please see Sect. 4.1).

This method uses topological information in PPI networks to maximize the number of conserved interactions to resolve ambiguity. The method relies on a probabilistic model and assigns a binary random variable, z_i to each node i in the alignment graph (representing a pair of aligned nodes in the input graphs). The variable indicates whether the corresponding protein pair represents true functional orthologs or not. Two nodes in the alignment graph, z_i and z_j, are connected if at least one of the protein pairs in the input graph (the protein pair represented by either i or j) are connected, and the other one has a common neighbor (or is also connected). The conditional probability distribution of Z_i can be defined as:

$$P(Z_i|Z_{N(i)}) = \frac{1}{1+\exp\{-\alpha_i + \sum_{j\in N(i)} \beta_{ij} Z_j\}}, \tag{5.10}$$

where $N(i)$ represents the neighbors of node i in the alignment graph. Simply stated, this formulation implies that a pair of proteins represented by node i in the alignment graph are more probable to be true functional orthologs when most of their neighbors are functional orthologs as well. To verify this formulation, one may observe that if we have only two proteins in the cluster, z_i will be 1, and for any pair of proteins in different clusters it is equal to 0. Bandyopadhyay et al. use a training data set to estimate parameters α and β, and use Gibbs sampling to evaluate the distribution function Z. Markov Random Field (MRF) based methods are successfully applied to alignment of protein interaction networks of yeast (*S. cerevisiae*) and fruit fly (*D. melanogaster*) (http://www.cellcircuits.org/Bandyopadhyay2006/).

3.2.2 IsoRank Family: Pairwise IsoRank, IsoRank-M, and IsoRank-N

The basic idea of the IsoRank family of methods, as explained in detail in Box 5.2, is to characterize the similarity of two nodes, v_i in G_1 and v_j in G_2, as a combination of node similarity and topological similarity. This quantity, denoted r_{ij}, is computed for all node pairs. The resulting similarity matrix, R, is used to align the input networks.

Singh et al. [23] propose a pairwise alignment technique based on similarity matrix R. They use a well-known algorithm for graph matching to align a pair of input graphs: they initially built a full weighted bipartite graph (nodes from G_1 in one part, nodes from G_2 in the other part, and edges representing similarity of nodes in G_1 to nodes in G_2). They then compute a *maximum weight bipartite match* using Hungarian algorithm [24], to find the one-to-one global alignment. Since the multiple graph matching problem, unlike bipartite graph matching, is known to be NP-complete, Singh et al. [25] extend this result to multiple network alignment by proposing heuristics for many-to-many alignment of input graphs based on the following greedy approach:

Initialization: Select the edge (v_i^{k1}, v_j^{k2}) with the highest score, where v_i^{k1} and v_j^{k2} are vertices in G_{k1} and G_{k2}, respectively. Initialize a new equivalence class with v_i^{k1} and v_j^{k2} as its initial members.

Expand to other species: In every other species, $\{G_1, \ldots, G_k)\} \backslash \{G_{k1}, G_{k2}\}$, if a node l exists in species G_{kx} such that:

- $R_{il}^{\langle k1,kx \rangle}$ and $R_{jl}^{\langle k2,kx \rangle}$ are the highest scores between l and any node in G_{k1} and G_{k2}, respectively, and
- Both $\beta_1 R_{ij}^{\langle k1,k2 \rangle} \le R_{il}^{\langle k1,kx \rangle}$, and $\beta_1 R_{ij}^{\langle k1,k2 \rangle} \le R_{jl}^{\langle k2,kx \rangle}$

then, add it to the primary class. This step ensures that the equivalence class has at most one node from each species.

Heuristic expansion: Add up to $r-1$ nodes from different parts of the graph to the equivalence class. Suppose v (from G_{ky}) is already in the equivalence class. Then, node v' (again from species G_{ky}) is added to the class if $\beta_2 R_{vw}^{\langle ky,kz \rangle} \le R_{v'w}^{\langle ky,kz \rangle}$, for every node $w \in G_{kz}$ which is already in the equivalence class ($w \neq v$).

Update Remaining: Remove from the alignment graph all of the nodes in the constructed equivalence class, and their corresponding edges.

Here, parameters β_1, $\beta_2 \in (0,1)$ and r are user-defined parameters. Also, $R_{ij}^{\langle p,q \rangle}$ represents the similarity between node v_i from species p, and node v_j from species q.

Liao et al. [26] propose an alternate heuristic for multiple alignment of networks. Their method, called IsoRankN (IsoRank-Nibble), is similar in concept to *PageRank-Nibble*, which approximates the Personalized PageRank vector. This approach constructs a full weighted k-partite graph with pairwise similarity scores as the weight on edges, and use a method based on spectral clustering to cluster the graph into low-conductance sets (similar to partitioning the graph into maximal weight subgraphs). All versions of IsoRank are available for download from http://groups.csail.mit.edu/cb/mna/.

3.2.3 Graemlin Family: Graemlin2.0

Flannick et al. [27] extend the concepts underlying Graemlin 1.0 by incorporating a general scoring framework. This framework is based on a user defined *feature vector*, and a *weight function* that can be learned from a set of true alignments. They also propose a hill-climbing method in Graemlin 2.0 and use this scoring function to align the input networks globally.

Box 5.2: IsoRank Algorithm

The core of all *IsoRank*-based algorithms is a method for computing the similarity matrix R, representing the functional similarity scores between any pair of nodes in two input networks. To compute these similarity scores, Singh et al. [23] propose an approach similar to PageRank. The method is based on the notion that a pair of nodes (i, j) represent a good match if the sequences corresponding to nodes i and j align well, and that their respective neighbors are also good matches. This recursive definition leads to the following formal definition of R:

$$R_{ij} = \sum_{v_u \in N(v_i), v_w \in N(v_j)} \frac{1}{|N(v_u)||N(v_w)|} R_{uw}, \tag{5.11}$$

where $N(v_i)$ represents the neighbors of node v_i in the input network. Using a matrix notation, this equation can be rewritten as:

$$R = AR$$
$$A[i, j][u, v] = \begin{cases} \frac{1}{|N(u)||N(v)|}, & \text{if } (i, u) \in E_1 \text{ and } (j, v) \in E_2; \\ 0, & \text{otherwise.} \end{cases}$$

This formulation describes an eigenvalue problem. Matrix A is a stochastic matrix, with principal eigenvalue of 1. One can use the simple *power method* for solving this problem. To incorporate sequence similarities into the formulation, the normalized node similarity matrix (calculated from an all-to-all BLAST bit scores) can be used. The corresponding modified eigenproblem is as follows:

$$R = \alpha AR + (1 - \alpha)E, \tag{5.12}$$

where E is the normalized node similarity matrix and parameter α represents the tradeoff between network and node similarity. Equation (5.12) is the base equation for all IsoRank-based algorithms.

The approach to learning the weight function is based on the definition of *loss function* $\mathscr{L}$, defined as $\mathscr{L} : \mathscr{A} * \mathscr{A} \rightarrow R^{+}$, which measures the distance of a given alignment from the gold standard alignment used for training. Intuitively, the learned weight vector should assign higher scores to alignments with smaller loss function values, and a score of zero for the correct alignment. The loss function grows as alignments diverge from the correct alignment.

To learn the weight function, Flannick et al. [27] use KEGG Ortholog (KO) groups [28] as the training set. Each training sample contains networks from a set of species, $G^{(i)} = G_1^{(i)}, \ldots, G_n^{(i)}$, with nodes that do not have a KO group removed; the correct alignment $a^{(i)}$ contains an equivalence class for each KO group. Let $[x]_{a^{(i)}}$ denote the equivalence class of $x \in V^{(i)} = \cup_j V_j^{(i)}$ in $a^{(i)}$, and $[x]_a$ denote the equivalence class of protein x under a. One possible definition for the loss function is as follows:

$$\mathscr{L}(a^{(i)},a) = \sum_{x \in V^{(i)}} |[x]_a \backslash [x]_{a^{(i)}}|. \tag{5.13}$$

Here, $A \backslash B$ represents the set difference between sets A and B. It counts the number of nodes aligned in a that are not aligned in the correct alignment $a^{(i)}$. To learn the weight function, the parameter learning problem is posed as the maximum margin structured learning problem. Given a training set and the loss function, the learned weight function, w, should score each training alignment $a^{(i)}$ higher than all other alignments a by at least $\mathscr{L}(a^{(i)},a)$. Formally we have the following definition:

$$\forall i, a \in \mathscr{A}^{(i)}, \mathbf{w}.\mathbf{f}(a) + \mathscr{L}(a^{(i)},a) \leq \mathbf{w}.\mathbf{f}(a^{(i)}), \tag{5.14}$$

where $\mathscr{A}^{(i)}$ is the set of all possible alignments of $G^{(i)}$. The optimal weight function w is then computed using a subgradient descent method.

After finding the optimum w, Graemlin2.0 uses a hill-climbing method for approximating the global alignment of input networks. It starts from an initial alignment containing every node in a separate equivalence class. It then iteratively updates the alignment by evaluating a series of local movements on vertices, computing the alignment score before and after the move, and performing the move that increases the score the most. There are four possible moves for each vertex under consideration:

- Do nothing
- Create a new equivalence class containing only that node
- Move the node to another equivalence class
- Merge the container equivalence class of that node with another equivalence class

This process terminates when an iteration does not increase the alignment score. Graemlin2.0 is available for download at http://graemlin.stanford.edu/graemlin-2.01.tar.gz Datasets needed for training and testing Graemlin2.0 can be downloaded from http://graemlin.stanford.edu/graemlin-2.0_test_files.tar.gz.

3.2.4 Methods Based on Integer Quadratic Programming Formulations

The global alignment problem can be explicitly posed as an *integer quadratic programming (IQP)* problem. Several approaches take this view to the global alignment problem and aim to solve this problem. Before we introduce the IQP formulation of the one-to-one global network alignment problem, we note that the pairwise alignment of a pair of input networks, $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$, can be formulated as a bipartite graph matching problem. We construct a bipartite graph as follows: let the vertices of the first part of the bipartite graph consist of the vertices in V_1, and the vertices in the second part consist of the vertices in V_2. Connect each node in the first part to every node in the second part, but not to any of the vertices within the same partition. Formally, let us denote the bipartite graph as $G_{Bi} = (V_{Bi}, E_{Bi})$, in which $V_{Bi} = \{V_1 \cup V_2\}$ and $E_{Bi} = \{(v_i, v_j) \in V_1 * V_2\}$. This graph is a complete bipartite graph, represented as $K_{m,n}$. We claim that every one-to-one network alignment between G_1 and G_2, is equivalent to a matching in the constructed bipartite graph, since any one-to-one network alignment assumes that each vertex in first network is mapped to at most one node in the second network, and correspondingly any matching in the bipartite graph, is a subset of E_{Bi} such that no two edges share the same endpoint. This is equivalent to the condition that each node in the first graph should be aligned with at most one node in the second graph.

Following this bijection, one can extend the concept of matching to *maximum weight matching*, to find an *optimal* one-to-one global network alignment. Having set the *appropriate* edge weights in the bipartite graph, one may argue that the maximum weight bipartite matching (which can be found using the Hungarian algorithm [24] in $O(\max\{m,n\}^3)$ time), is equivalent to the optimal one-to-one network alignment. The pairwise IsoRank algorithm (see Sect. 3.2.2) is an example of this class of problems – it defines the similarity score between nodes in input networks, namely the R matrix, in a way that captures both the node-based and topological similarities around each node, and uses this matrix to weight the edges in the bipartite graph to find the alignment. Integer quadratic programming, on the other hand, aims at explicitly finding the optimal matching and updating the maximum scores of the alignments, in a way that maximizes the node similarity score between matched nodes, as well as the conserved edges in a pair of networks.

Definition 5.3. Integer Quadratic Programming Formulation Given a pair of unweighted graphs, $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$, represented by their corresponding adjacency matrices $A = (a_{ij})_{m*m}$ and $B = (b_{ij})_{n*n}$, respectively, let the matching variable x_{ij} be equal to one, if node $v_i \in V_1$ is matched to node $v_j \in V_2$. The global network alignment can be formulated as an integer quadratic program as follows:

$$\begin{aligned}\text{Maximize}_X\{\phi(G_1,G_2)\} &= \lambda \sum_{i=1}^{m}\sum_{j=1}^{n} s_{ij}x_{ij} + (1-\lambda)\\ &\quad\times \sum_{i=1}^{m}\sum_{j=1}^{n}\sum_{k=1}^{m}\sum_{l=1}^{n} a_{ik}b_{jl}x_{ij}x_{kl} \end{aligned} \tag{5.15}$$

$$\text{Subject to} \quad \begin{cases} \sum_{j=1}^{n} x_{ij} \le 1, & \forall i \in \{1,\ldots,m\};\\ \sum_{i=1}^{m} x_{ij} \le 1, & \forall j \in \{1,\ldots,n\};\\ x_{ij} \in \{0,1\}, & \forall i \in \{1,\ldots,m\} \text{ and } \forall j \in \{1,\ldots,n\}. \end{cases}$$

Here, parameter λ adjusts the relative importance of node similarity and edge conservation. The first two constraints ensure that every node in each partition is mapped to at most one node in the other partition, while the last constraint is the integer constraint for variables x_{ij}. This formulation can also be expressed in closed matrix form. Denoting the matching variables x_{ij} and node similarity scores s_{ij} using matrices X and S, respectively, the above definition can rewritten as:

$$\text{Maximize}_X\{\phi(G_1,G_2)\} = \lambda XS + (1-\lambda)AXB^T \bullet X \tag{5.16}$$

$$\text{Subject to} \quad \begin{cases} X1_m \leq 1_n, X^T 1_n \leq 1_m & \text{Matching constraints;} \\ x_{ij} \in \{0,1\}, & \text{Integer constraint.} \end{cases}$$

Here, $\bullet$ is the inner-product operator between matrices, and 1_m and 1_n are vectors of all ones, of sizes m and n, respectively. To generalize the problem to arbitrary bipartite graphs, (E_{Bi} does not correspond to a complete bipartite graph), we formulate the problem differently. Let vector X_υ, of size $|E_{Bi}| = |V_1| * |V_2| = m * n$, denote the vectorization of X, and vector S_υ of the same size denote the vectorization of S. Let matrix C be a matrix of size $|E_{Bi}| * |E_{Bi}|$, in which element $C_{e_1,e_2} = 1$, for any $e_1 = (i_1, j_1), e_2 = (i_2, j_2) \in E_{Bi}$, if $(i_1, i_2) \in E_1$ and $(j_1, j_2) \in E_2$, and zero otherwise. An entry in matrix C indicates whether or not a pair of matchings, $i_1 \rightarrow j_1$ and $j_1 \rightarrow j_2$, result in a conserved edge in the input networks. Finally, let matrix D, of size $|V_{Bi}| * |E_{Bi}|$, be the unoriented incidence matrix of the bipartite graph. The general IQP can be written as follows:

$$\text{Maximize}_X\{\phi(G_1,G_2)\} = \lambda S_\upsilon^T X + (1-\lambda) X_\upsilon^T C X_\upsilon \tag{5.17}$$

$$\text{Subject to} \;:\; DX_\upsilon \leq \mathbf{1}, x \in (\{0,1\}^{m*n})^T,$$

where $\mathbf{1}$ is the vector of all ones, of size $|V_{Bi}| = |V_1| + |V_2| = m + n$.

Equation (5.15) was initially proposed by Li et al. [29], who showed that the constraints in the formulation have a unimodular property. This implies that the problem can be relaxed to quadratic programming with an integral solution in the general case. Furthermore, they proved the sufficient conditions to ensure that the quadratic programming will have an integer solution.

The associated algorithm, called MNAligner, is used to align PPI networks of yeast (*S. cerevisiae*) and fruit fly (*D. melanogaster*) (from [22]), as well as a pair of metabolic pathways for *E. Coli* and yeast (*S. cerevisiae*). To deal with the computational complexity associated with large networks, Li et al. apply a network clustering algorithm to input network first, and then apply their method to identify conserved regions in the smaller subgraphs. Matlab code for the algorithm is available from http://zhangroup.aporc.org/bioinfo/MNAligner or http://intelligent.eic.osaka-sandai.ac.jp/chenen/software/MNAligner.

The closed form in (5.16) and (5.17) is first introduced by Bayati et al. [30]. They also show that IsoRank is an approximate solution of the integer quadratic programming, that does not explicitly satisfy the constraints. They also propose a modification of the IsoRank formulation by restricting the number of edges in the bipartite graph by eliminating unpromising edges. This makes the algorithm more suitable for large sparse graphs (where the number of nodes in input graphs are in order of hundreds of thousands). Their implementation in Matlab as well as test cases are available for download at http://www.stanford.edu/~dgleich/publications/2009/netalign/.

Klau [31] proposes a similar formulation, albeit with different notation, and a different relaxation technique. In this approach, the IQP is first transformed into an equivalent linear integer program. A relaxation based on the Lagrangian decomposition is then used to solve this problem. The violation of constraints, together with their Lagrange multipliers, are integrated into the objective function. It is known that the solution to the Lagrangian linear program is an upper bound for the linear program, which itself is an upper bound for the network alignment problem. A heuristic is developed for reducing the gap between the fractional upper bound and integer solution. An implementation of this technique is available from https://www.mi.fu-berlin.de/wiki/pub/LiSA/Natalie/natalie-0.9.tgz.

Zaslavskiy et al. [32,33] also use a similar formulation, and propose two different methods for solving it. The first method, called GA, is based on the gradient descent method. GA starts from an initial solution and searches the state space of matchings for an optimal solution based on the gradient of the objective function ϕ. Like all other local search methods, this approach is suitable if we can start from a "good" initial solution that is close enough to the optimal solution. Otherwise, it gets stuck in local minima. The second algorithm, called PATH, is based on two relaxations of (5.17), one concave and one convex, over the set of doubly stochastic matrices. PATH starts by solving the convex relaxation, using the Frank–Wolfe method [34], and then iteratively solves a linear combination of convex and concave relaxation by gradually increasing the weight of the concave relaxation and following the path of the solutions thus created. This algorithm is implemented as part of the *Graph Matching (GraphM)* package. This package aims to collect various graph matching methods in a unified framework, and to organize them in a simple, easily extendible software package. The package is freely available from http://cbio.ensmp.fr/graphm/personal_dir/graphm-0.5.tar.gz.

3.3 Multiple Network Alignment: Complexity and Scalability

Increasing amounts of network data requires methods that scale up from aligning pairs of networks to multiple networks from different species. Existing methods have serious limitations with respect to scalability to large numbers of networks and most rely on heuristics. Trade-offs between computational cost of heuristics and their solution quality remains an open and active area of research.

NetworkBlast, proposed by Sharan et al. [7], is applied to the alignment of up to three networks. While this method is able to align multiple networks theoretically, in practice the running time grows exponentially in the number of species, which limits the number of graphs that can be simultaneously aligned. Kalaev et al. [35] improve the running time of this method from $O(n^k)$ to $O(n2^k)$, where n denotes the number of vertices and k denotes the number of networks. The intuition behind this method is to prevent the creation of alignment graph directly, and to build it implicitly as part of the algorithm. This avoids creation of nodes for every set of potentially orthologous proteins (recall that the size of the alignment graph grows exponentially in k). Note that the resulting algorithm still has exponential running time.

All IsoRank-based methods require a quadratic time complexity in the number of input species, k, multiplied by the running time for computing similarities between a pair of networks using the iterative procedure. IsoRank for aligning multiple graphs, as proposed by Singh et al. [25] (see Sect. 3.2.2), takes pairwise similarity matrices, and applies a greedy method to construct an alignment graph based on them. IsoRank-N [26], on the other hand, uses a spectral clustering mechanism to cluster the nodes in input networks based on the pairwise similarity matrices.

Flannick et al. [2] define equivalence classes for constructing the alignment graph, and are able to mimic the progressive sequence alignment technique to achieve linear runtime dependence in number of graphs. As mentioned in Sect. 3.1.3, this approach initially links species using a phylogenetic tree, and at each step merges the two closest networks to create a single alignment graph. This method has been successfully applied to up to ten microbial networks. Note, however, that this heuristic is sensitive to the quality of the phylogenetic tree used to establish the relationship between species.

3.4 Validation Methods

An important problem associated with validating network alignment algorithms is that assessment of the quality of an alignment is not straightforward. The basic concept underlying comparative network analysis is one of transferring "knowledge" from one species to other. This knowledge can be the functional annotation of proteins, functional modules, disease/phenotype, etc. Consequently, before we can evaluate a method, and its associated knowledge transfer, we need to define a unified framework to describe the knowledge, annotate entities, and transfer it among different species. *Ontologies*, which provide a hierarchical framework of categorized consensus vocabularies, provide facilities for formally describing the knowledge about various biological entities. This set of vocabularies can change from context to context, and even in the same context we might have several different frameworks. The most widely used vocabularies describing protein function are the Gene Ontology (GO) [36], Enzyme Commission (E.C.) [37], and MIPS Functional Catalogue (FunCat) [38].

GO consists of three individual, hierarchical ontologies containing terms that describe molecular function (biochemical activity), biological process (pathway), and cellular component (localization). GO terms associated with protein sequences carry evidence codes that describe the experimental or computational evidence for the annotation. E.C., which is commonly used for annotating enzymes in KEGG pathways, is a four-level hierarchy of enzyme nomenclature, describing biochemical activity. MIPS FunCat is a six-level hierarchical scheme used for genome annotation containing over 1,300 terms in 28 general categories [39]. There are different ontologies for describing disease implicated genes, based on their relation to different disease related pathways. As an example, NetPath [40], at this time contains ten immune and ten cancer signaling pathways. OMIM [41] is a frequently accessed database related to genetic variants associated with phenotypes.

Here, we primarily focus on methods for quality assessment in function prediction using comparative analysis. Knowledge relating to annotations is partial and one is interested in using methods such as network alignment to expand this knowledge. This enhancement is hard to assess, especially since the available knowledge is not reliable or even homogeneous. As an example, GO annotations have different *tags* based on their annotating methodology, and GO annotations tagged as IEA (electronic annotation), ISS (pure sequence-based annotation), or ND (annotation without documented evidence) are known to be unreliable. This heterogeneity and incompleteness in data makes it hard to define measures for evaluating the quality of different methods. Furthermore, cellular entities typically participate in different processes, and thus have multiple annotations. Considering all of the aforementioned limitations, one must consider a gold standard, and evaluate methods based on this gold standard.

Since there have been different methods proposed for evaluating the consistency of functional annotation mappings, we briefly review different approaches. These approaches are based on given mappings between nodes of the input networks. Singh et al. [25] propose the following methodology for computing *functional coherence* as their quality assessment measure: Given an ortholog list, they initially extract equivalence classes that have at least a fraction k of their proteins with at least one GO term, which they set $k = 80\%$ in their multiple alignment method using IsoRank (see Sect. 3.2.2). Next, they collect all of the GO terms corresponding to any protein in each remaining equivalence class (except those with ISS, IEA or ND tags). To compare these GO term lists, they map each GO term into a *standard form*, which they define as subset of GO terms that are at a distance of five from the root of the GO tree, and each GO term t is mapped to its ancestor(s) at this level. In this way, they not only map the annotation to a common level in the GO hierarchy, but also eliminate functional annotations that are not specific enough. Having the set of proteins in each equivalence class annotated with homogenized set of GO terms, they proposed an intra equivalence class scoring, followed by averaging of scores in different classes. To evaluate the functional coherence in each equivalence class, Singh et al. first define the similarity score between any pair of GO terms used to annotate proteins in each equivalence class as follows: let S_i and S_j be the set of proteins in the equivalence class annotated by standardized GO terms t_i and t_j, respectively. The pairwise similarity score between t_i and t_j is defined as:

$$\mathrm{sim}(t_i,t_j) = \frac{S_i \cap S_j}{S_i \cup S_j} \qquad (5.18)$$

Note that this similarity score is symmetric, and is bounded by 0 and 1. Next, to find the functional coherence in each equivalence class, they find the median over all possible pairwise combinations of GO terms in each equivalence class. Finally, as mentioned earlier, they average over functional enrichments of all classes.

They propose different methods for evaluating IsoRankN (see Sect. 3.2.2) – *consistency* and *coverage*. The former is defined as the mean entropy of the predicted clusters. More formally, consistency of a given cluster S_v^* is defined as:

$$H(S_v^*) = H(p_1, p_2, \ldots, p_d) = \sum_{i=1}^{d} p_i \log p_i, \tag{5.19}$$

where p_i is the fraction of the proteins in S_v^* with GO or KEGG group ID i. They also propose a normalization of entropy scores by the cluster size as, $H_{\text{norm}}(S_v^*) = \frac{1}{\log d} H(S_v^*)$. The coverage of an alignment method is measured by the number of clusters containing proteins from at least k species, where k is an adjustable parameter. An alternate definition for coverage is proposed by Kalaev et al. [35] based on the enrichment of predicted groups with respect to known ontologies derived from either GO or KEGG.

Flannick et al. [2] propose two different sets of measures to mimic *sensitivity* and *specificity*, respectively. They assess the former by counting the number of KEGG pathways in species that are aligned together correctly, meaning that at least three proteins in each pathway are aligned with their counterparts in the other species. To measure the specificity, they propose two methods. First, to compute the specificity based on GO terms, they assign to each protein all of its annotations from level 8 or deeper in the GO hierarchy, and then calculate the alignment enrichment using GO TermFinder [42]. The alignment is considered enriched, if the p-value of the alignment is less than 0.01. Second, they measure the specificity based on the fraction of nodes that have KEGG orthologs, but are aligned to any nodes other than their KEGG orthologs.

An alternate method for assessing alignment methods, is to measure the number of conserved edges. Conservation in this sense means that a pair of nodes v_i^1 and v_j^1 are aligned to their orthologs v_i^2 and v_j^2, and there is an edge both between v_i^1 and v_j^1, as well as v_i^2 and v_j^2, indicating that the alignment *conserved* those edges.

3.5 Databases

There are a number of databases for comparative network analysis. The first set of sources contain interactomes of different species. One of the most commonly studied interactomes, is the PPI network. The following databases are frequently used for PPI data:

- Biomolecular Interaction Network Database (BIND) [43] is a database of full descriptions of interactions, molecular complexes, and pathways. Development of the BIND 2.0 data model has led to the incorporation of virtually all components of molecular mechanisms, including interactions between any two molecules composed of proteins, nucleic acids, and small molecules. The BIND database can be accessed through http://www.bind.ca/.

- The Database of Interacting Proteins (DIP) [44] catalogs experimentally determined interactions between proteins. It combines information from a variety of sources to create a single, consistent set of PPIs. The data stored in DIP has been curated, both manually, by expert curators, and automatically, using computational approaches that utilize knowledge about the PPI networks extracted from the most reliable, core subset of DIP data. In addition to the interaction information, DIP includes additional data regarding the proteins participating in PPI networks. This database is available on http://dip.doe-mbi.ucla.edu/.
- IntAct [45] provides an open framework for storing, presenting, and analyzing protein interactions. The web interface provides both textual and graphical representations of protein interactions, and allows exploration of interaction networks in the context of the GO annotations of the interacting proteins. A web service allows direct computational access to interaction networks in XML. All IntAct services are accessible through http://www.ebi.ac.uk/intact.
- The Biological General Repository for Interaction Datasets (BioGRID) [46] is another curated database of protein–protein and genetic interactions. It aims to provide a comprehensive resource for protein-protein and genetic interactions for all major model organisms, while attempting to remove redundancy, to create a single mapping of interactions. It can be accessed from http://www.thebiogrid.org/.
- The Molecular INTeraction database (MINT) [47] extracts, curates, and stores experimental information about physical interactions between proteins from previously published results in peer-reviewed journals. This database is accessible from http://mint.bio.uniroma2.it/mint/.
- MPact [48] is a PPI database that is targeted to *yeast (S. cerevisiae)*. The complete dataset, as well as user-defined subnetworks can be retrieved in the PSI-MI format from http://mips.gsf.de/genre/proj/mpact.

DIP, IntAct, BioGRID, MINT, and MPact are participating databases in the International Molecular Exchange Consortium (IMEx), a group of the major public interaction data providers. The databases of IMEx work together to prevent duplications of effort, collecting data from nonoverlapping sources and sharing curated interaction data. There are also several databases related to cellular pathways, which are briefly reviewed here:

- Kyoto Encyclopedia of Genes and Genomes (KEGG) [28], which is publicly available at http://www.genome.ad.jp/kegg/, is a collection of online databases of genomes, enzymatic pathways, and biochemicals. The *pathway* database stores networks of molecular interactions in the cells, and their variants specific to select organisms. They cover different areas of interest including metabolism, genetics, cellular processes, human diseases, and drug development. The database also provides a standardized method for representing pathways that a protein takes part in using the *KEGG Orthology (KO)*.

- BioCyc [49] is a collection of databases publicly available at http://biocyc.org/. Databases within BioCyc describe genome and pathway information for individual organisms. EcoCyc and MetaCyc are the two databases within BioCyc, which are well curated from scientific literature.
- Netpath [40] is a curated resource of human signal transduction pathways, which can be accessed at http://www.netpath.org/. It currently consists of ten immune and ten cancer signaling pathways. These pathways contain information pertaining to PPIs, catalytic reactions, translocation events, and genes that are differentially regulated upon stimulation of receptors by their specific ligands.
- Reactome [50] is a curated, peer-reviewed resource of human biological processes publicly available at http://www.reactome.org/. The largest set of entries refers to human biology, however, it also covers a number of other organisms as well. GO is used to describe the subcellular locations of molecules and reactions, molecular functions, and the larger biological processes that a specific reaction is part of.
- NCI-Nature Pathway Interaction Database (PID) [51] is a free biomedical database of human cellular signaling pathways available at (http://pid.nci.nih.gov/). The database contains information about molecular interactions and reactions that take place in cells, with a specific focus on processes relevant to cancer research and treatment.

In addition to interactome and pathway databases, there are several sequence-related (genes/proteins) databases. Currently, UniProt [52], which is accessible at http://www.uniprot.org/, is the universal protein database. It is a central repository of protein sequences that integrates Swiss-Prot, a reliable database from European Bioinformatics Institute (EBI), and Swiss Institute of Bioinformatics (SIB), TrEMBL, a less reliable database that covers a wider range of proteins, and Protein Sequence Database (PSD), from Protein Information Resource (PIR). Three major databases storing gene sequences are:

- DNA Data Bank of Japan (DDBJ) [53], which is maintained by National Institute of Genetics (NIG) in the Shizuoka prefecture of Japan, which is publicly available at www.ddbj.nig.ac.jp/.
- EMBL Nucleotide Sequence Database [54], which is maintained by the European Bioinformatics Institute (EBI), available at http://www.ebi.ac.uk/embl.
- GenBank [55], maintained by National Center for Biotechnology Information (NCBI) as part of the International Nucleotide Sequence Database Collaboration, or INSDC, available at www.ncbi.nlm.nih.gov/genbank/.

These databases are the main repositories for gene sequences for all organisms, and are members of The International Nucleotide Sequence Database (INSD). They exchange newly submitted gene sequences frequently (daily) minimize inconsistencies. There are also a number of species-specific databases. These databases typically integrate information from different sources to construct a uniform database of all the information specific to a species. Some well-known species-specific databases include:

- Flybase [56], accessible at http://flybase.bio.indiana.edu/, is an online database of the biology and genome of the model organism *fruit fly (D. melanogaster)*. It contains a complete annotation of the *D. melanogaster*. It also includes a searchable bibliography of research on Drosophila genetics.
- The Arabidopsis Information Resource (TAIR) [57] maintains a database of genetic and molecular biology data for the model organism *plant (Arabidopsis thaliana)*, at http://www.arabidopsis.org/. Data available from TAIR includes the complete genome sequence, along with gene structure, gene product information, metabolism, gene expression, DNA and seed stocks, genome maps, genetic, and physical markers.
- Mouse Genome Database (MGD) [58] is an integrated data resource for mouse genetic, genomic, and biological information, at http://www.informatics.jax.org/. MGD includes a variety of data, ranging from gene characterization and genomic structures, to orthologous relationships between mouse genes and those of other mammalian species, to maps (genetic, cytogenetic, physical), descriptions of mutant phenotypes, characteristics of inbred strains, and information about biological reagents such as clones and primers. Data is accessed via search/retrieval Web forms and displayed as tables, text, and graphical maps, with supporting primary data. A rich set of hypertext links is provided, such as those from gene and clone information to DNA and protein sequence databases (GenBank, EMBL, DDBJ, SWISS-PROT), from bibliographic data to PubMed, from phenotypes to Online Mendelian Inheritance in Man (OMIM), and from gene homology records to the genomic databases of other species.
- Rat Genome Database (RGD) [59], accessible at http://rgd.mcw.edu/, stores genomic, genetic, functional, physiological, pathway and disease data for the laboratory rat, as well as comparative data for rat, which is a major model organism for the study of human disease.
- Saccharomyces Genome Database (SGD) [60] stores information about the chromosomal features and gene products of the budding yeast *S. cerevisiae*, which can be publicly accessed at http://www.yeastgenome.org/.

4 Applications

Network alignment has been successfully applied to a variety of problems, including function prediction for unannotated proteins, investigating cellular machinery, comparative analysis of evolutionary events, and integrating biological networks with prior data sources for disease diagnoses.

4.1 Projecting Functional Annotations

While high-throughput methodologies for assessing protein function are emerging, computational methods are essential for complementing these experimental techniques. Evolutionary events and analyses have been shown to be effective in studying biomolecular functions across species. Comparative analyses across evolutionarily close species, such as humans (*H. sapiens*) and mice (*M. musculus*), and model organisms such as yeast (*S. cerevisiae*), nematode worm (*C. elegans*), and fruit fly (*D. melanogaster*) (because of their short life cycle), have provided critical insight into structure and function of various proteins [61].

Understating phylogenetic relationships among proteins can help in predicting their structure and function. Two proteins with similar sequences are known to be *homologous*. If a pair of homologous proteins have evolved from a common ancestor by *speciation* event(s), they are referred to as *orthologs*. Proteins can also be separated by *duplication* event(s) – such proteins are called *paralogs*. Paralogous proteins, contrary to orthologous proteins, can, and usually do, diverge in their function after duplication. They can be classified into two different classes: *in-paralogs* (also known as *recent* paralogs), in which pairs of proteins are duplicated after a speciation event, and *out-paralogs* (also known as *ancient* paralogs), in which the duplicated event precedes the speciation event. In the former case, proteins are more likely to be true functional orthologs, since there is shorter distance between the duplicated ancestor and its descendants.

Early computational methods for predicting protein functions are primarily sequence-based. Sequences of proteins from different species are compared to find homologous proteins. Two examples of sequence-based models are Clusters of Orthologous Groups (COG) [15] and Inparanoid [62]. COG defines functional orthologs using sets of proteins that contain best BLAST matches across a minimum of three species. The Inparanoid approach is a sequence-based method for finding functional annotation. It uses clustering to derive ortholog families, leaving some of the orthology relations ambiguous. When the homology is not ambiguous, especially in cases where the function is essential to the evolutionary fitness, the pair of homologous proteins usually are functional orthologs, carrying the same set of functions. On the other hand, when we have multiple homologous proteins in different organisms, there is an ambiguity about the true functional orthologs, since these may result from different evolutionary events.

An all-versus-all BLAST method for predicting protein functions is often unable to distinguish between out-paralogs and in-paralogs, and thus results in false-positives. Different methods have been proposed to remedy this problem. Comparative network analysis is one of the most promising methods. The motivation behind the use of comparative network analysis is that out-paralogs, which are ancient, had more time to diverge in their patterns of interactions. One may use these differences in interaction patterns to make a decision regarding the elimination of out-paralogous proteins. Most of the global alignment methods discussed in Sect. 3.2, are used to transfer functional annotations based on this hypothesis.

4.2 Conserved Functional Modules Across Species

Biological systems can often be decomposed into smaller subsets, known as modules. Modules are a sets of cohesive entities that are loosely connected to the rest of the system [63]. Hartwell et al. [64] hypothesized that biological processes within individual cells are carried out by such modules, also called "functional modules." These are discrete entities composed of various molecules, whose functions are separable from others, and whose functions are manifested in their interaction patterns.

It has been shown that most of cellular interactome, including PPI, metabolic, and GRNs, have modular structure. Protein complexes in PPI networks [65, 66], metabolic pathways in metabolic networks [67], and signal transduction pathways in GRNs [68, 69], are examples of modules that have specific functions in their corresponding networks. However, both decomposition and functional annotation of the modules pose significant challenges from points of view of model and method development. Comparative network analysis is known to be one of the most powerful methods for decomposing networks of multiple species to extract their common functional modules. These methods are based on the idea that conservation of specific substructures across species implies their functional coherence (for a brief overview of other methods, please see Sect. 5.1).

Most of the local alignment methods discussed in Sect. 3.1 can be used to extract conserved substructures in the networks of multiple species, and to predict functional modules. For example, the use of PathBLast [5] to align protein interaction networks of two distantly related species, yeast (*S. cerevisiae*) and bacterium (*H. pylori*), uncovers remarkable conserved pathways among them. It is also used for aligning protein interaction networks of *P. falciparum* with other eukaryotes (yeast (*S. cerevisiae*), fruit fly (*D. melanogaster*), and worm (*C. elegans*)) and yeast (*S. cerevisiae*), and bacterium (*H. pylori*), to identify their conserved pathways (please see Sect. 3.1.1 for more details). MAWISH [9, 10] and NetworkBlast-M [7] are used to find conserved protein complexes, and to identify relationships among different biological process in distinct organisms. Graemlin1.0 [2] is used for transferring functional annotations of known modules in one species to another. It used two approaches – the first one transfers functional annotation from a protein, to other aligned proteins whose annotations are unknown. This is similar to sequence based methods, however, network-based methods have been shown to be more accurate since they rely on both sequence and interactions. The second approach is specific to network alignment – it works on the thesis that the function of landmark proteins is shared by other proteins in the alignment.

4.3 Studying Evolutionary Events

Evolution manifests itself in variations, for example, mutations on the genome that impact structure and function of associated proteins [68]. These changes in turn affect protein interactions, metabolic reactions, and genetic interactions [64]. Despite the key roles of these interaction networks in structural and functional characterization, study of evolutionary trajectories remains an active area of investigation. The evolution of protein interaction networks depends on the modification of genes that produce proteins and the way the general structure of the network has been impacted over the evolutionary history [71]. Phylogenetic analysis of protein interaction networks is most commonly performed through network comparison, based on the idea that a common ancestor is shared by all distinct organisms [1].

Network alignment can be used in different levels to uncover evolutionary relations. At lowest level, each protein can be represented by its tertiary structure graph, and the corresponding structures can be compared using network alignment. As mentioned earlier, evolution affects cellular process by mutations on nucleotide sequences. These result in changes in amino acid sequences and consequently in their structure and function. Conversely, identifying similarities and differences in protein structures can uncover their evolutionary history [72]. Comparative network analysis can also be used to overcome the ambiguity problem among homolog proteins that pure sequence-based methods often fail to do. As discussed in Sect. 4.1, discriminating in-paralogs and out-paralogs, which is an important component of finding the order of duplication/speciation events, can be efficiently performed using network alignment of protein interactions of different species.

Comparative analysis can also be applied at a modular level to help understand evolutionary events. For example, evolutionary biologists have extensively studied genes related to aging, and on understanding mechanisms leading to cell death. The role of biological networks in explaining the complex traits provides exciting new avenues to extend current efforts focused on the study of individual genes. Promislow [73] examined the subset of yeast proteins related to senescence, and collates them with subsets of other traits. He found that proteins (and corresponding genes) related to senescence have higher connectivity compared to other proteins; four of five examined traits were unrelated to senescence, and they did not have notable connectivity in comparison to those related to senescence. His final conclusion was that genes associated with aging produce proteins that are more highly connected and have greater pleiotropy (are associated with multiple phenotypes that seem completely independent from each other). Wagner et al. [74] illustrated that the rate of evolution in highly connected nodes between modules is significantly higher than the nodes found in a module. Finally, Yosef et al. [75] developed a framework for investigating the evolutionary trajectory of protein complexes, besides the role of the self-interacting protein's duplication in complex evolution. In this study, phylogenetic analysis is used to age the proteins in each complex, which were identified by network alignment using NetworkBLAST (see Sect. 3.1.1) or network clustering using MCL algorithms. They also investigate whether the members of a protein complex evolve together. Their results show that complex proteins emerge early in evolution, and evolve together over history.

Network alignment can also be used as a compare function between the interactomes of different species. This in turn can be interpreted as the distance in their cellular structure. From this point of view, network alignment can be used as the pairwise distance of a set of species, to link them and reconstruct the phylogenetic tree. Most of the previously mention methods, for example, IsoRank and IQP-based methods (see Sect. 3.2), align the entire interactome as a single object, and give a unique alignment score to each pair of networks. However, due to the computational cost of aligning large networks, all of these algorithms must deal with intractable graph comparisons, besides the high degree of noise in interaction data. Towfic et al. [76] propose an algorithm for aligning large biological networks based on the alignment of their subgraphs, which are scored by a graph kernel. The computed score of these alignments can be used for clustering species and recovering phylogenetic information. Erten et al. [70] alleviate these difficulties by designing a slightly different approach, named MOPHY, to extract evolutionary trajectory based on conserved modules. The fundamental idea behind MOPHY that cohesive interaction patterns have strong tendency to be conserved over evolutionary history. MOPHY initially identifies the modular subgraphs in different networks independently. It then maps these modules to networks of other species to understand the conservation and divergence of different modular processes in these networks. Finally, it uses these modules to compute overall phylogeny of the networks by comparing the module maps together using feature vectors. Results from this algorithm show that modularity based analysis can be used to gain deeper insight into functional evolution, and phylogenetic analysis of individual module.

4.4 Disease Discovery

The availability of complete genomes, proteomic, and metabolic data combined with phenotype characterization provide significant new avenues for understanding and treating disease [77]. It has been shown that the function of a gene in disease depends on the locus of its protein in protein interaction networks [78]. These intricate relationships in cellular networks establish the role of network analysis.

Different classes of human diseases such as cancer types, autoimmune disorders, hormone diseases, genetic disorders, infectious diseases, neurological disorders,and mental illnesses are caused by defects in genetic structure or cellular metabolism. This can be the result of the pathogen's infection or genetic variations such as missing, mutation, or extra copy of a gene. Understanding the genetic makeup of diseases is the initial step toward analyzing different diseases and intervention.

A variety of biochemical networks have been shown to conform to a scale-free structure. While robustness is one of the key attributes of scale-free networks, targeted variations at specific loci and positions lead to dysfunctional behavior. In fact, this characteristic of biological networks supports the observation that several mutations must occur for the onset of a disease like cancer [80]. Studying the structural changes to networks caused by diseases, is an essential component of understanding underlying mechanisms, and consequently their cure.

Many of the methods described in this chapter are directly applicable to disease discovery and characterization as well. As shown earlier, function prediction is carried out by first annotating proteins in different networks that have known GO or KEGG annotations, and then transferring (projecting) these annotations to their putative orthologs after aligning networks. Similarly, one may annotate proteins, genes, or other cell components in the networks by their known relations to different diseases, and transfer this knowledge to putative orthologs to predict disease implicated genes. In a similar fashion, functional module discovery is also readily extendable to disease discovery – one may annotate pathways, complexes, and disease related modules in general, and transfer this knowledge to the aligned substructures. Finally, network-based phylogenetic studies can be used for understanding disease. Phylogenetic trees are useful in uncovering the evolution of viral strains [81]. Such studies can be used to explain how some viruses, for example, canine (a virus that transfers from cats to dogs), can jump from one species to another. This analysis leads to a better understanding of viruses such as avian influenza that can transfer to humans from other species such as birds or pigs. Using phylogenetic methods to identify the relationships between HRV (rhinovirus) strains, as the pathogen for common cold, may lead to novel therapies, and more effective drugs, by elucidating structure, function, interactions, and context [82]. Investigation of human tumor subtypes using phylogenetic methods leads to identification of differentiation-related genes [83].

Understanding pathogens and uncovering the way they affect the normal cells and turn them to infected cells is a fundamental challenge. Comparative network analysis provides an important tool for such analysis. The single cell parasite (*P. falciparum*), is responsible for one million deaths every year around the world from malaria. One of the key challenges in dealing with malaria is that falciparum becomes resistant to the anti-malarial drugs. Falciparum is a human parasite, therefore it causes disruption of pathways active in falciparum without harming normal functional human pathways. Consequently, pathways that are different between the parasite and the human cell provide promising therapeutic targets. Comparative network analysis can help in revealing conserved pathways between falciparum and other eukaryotes, which implicitly help in finding the conserved pathways of falciparum and human, and assist in drug design.

To find conserved pathways, Suthram et al. [84] aligned the protein interactome of falciparum with the protein interactome of yeast (*S. cerevisiae*), worm (*C. elegans*), fruit fly (*D. melanogaster*), and the bacterial pathogen (*H. pylori*), using the PathBlast algorithm (see Sect. 3.1.1). Results from their study showed that falciparum has just three conserved complexes with yeast and no conserved complex with other species. However, yeast, fly, and worm have significant numbers of conserved complex among each other. While this is preliminary research, it shows that falciparum is significantly different from other model organisms and this poses challenges.

Regulatory enzymes have essential roles in cellular metabolism. Among them, phosphorylation is a key event in regulation. Phosphorylation sites are, however, short and changeable, unlike proteins domains that are conserved over longer periods of time. Investigating conservation of phosphorylation sites by sequence similarity is too hard and inefficient. To overcome this problem, Heng et al. [85] investigated the conservation of protein phosphorylation events at sequence, and networks levels for a set of species – human (*H. sapiens*), yeast (*S. cerevisiae)*, nematode worm (*C. elegans*), and fruit fly (*D. melanogaster*). At sequence level, they found *core sites* by identifying conserved phosphorylation sites that are positionally conserved between human and at least one target species. Among a total of 23,977 human phosphorylation sites found across 6,456 phospho-proteins encoded by 6,293 genes, they identify a subset of 479 core sites that are conserved between human, and at least one target species in 344 proteins encoded by 337 human genes. However, phosphorylation sites are often positioned in disordered regions, which are changeable. Therefore they cannot be used to show evolutionary conservation at sequence level. Hung et al. constructed a kinase-substrate network for target model organisms, and applied a network alignment method to extract the conserved human kinase-substrate network, also known as *core net*. Among a total 25,563 human interactions between 113 kinases and 5,515 substrates, 1,255 interactions between 27 human kinases and 778 substrates encoded by 759 genes were found. In this study, 1,105 interactions (88% of interactions) and 698 substrates were not found in core sites. Finally, they illustrated significant overlap between human genes coding phospho-proteins and cancer-associated genes as well as OMIM genes [41].

There have been several other studies relating heterogeneous networks – phenotypic and genotypic networks, and applying comparative network analysis to investigate their structural similarities. Wu et al. [86] introduced AlignPI, a method that exploits known gene-disease associations by aligning the phenotypic similarity network with the human PPI network. To align these, human disease phenome and interactome are modeled as graphs. In the former, nodes correspond to disease phenotype and the latter is a network of genes with interaction between their proteins products. In addition, these two networks are connected by the interactions between their genes and related phenotype. The link between networks is constructed based on the gene-phenotype relationships. Finally, they used NetworkBlast to identify locally dense regions of the PPI network and their associated disease clusters. They find that there is a conserved gene module in gene interactome for each known disease module in human phenome. In other words, for each set of phenotypically related diseases, there is a set of associated causative genes for these diseases. Another important study in this area is the work of Goh et al. [87]. In this study, the authors investigate three different networks – *gene disease* (network of disease genes with links between the genes that are involved in one disorder), *human disease* (network of human diseases with links between diseases that share a disease gene), and *diseasome* (network of human disease and disease genes with link between disease and its causal gene). They discovered high level relationships among human disease and their related causal genes. Their result indicates that similar disorders are usually caused by similar genes.

5 Related Efforts

In this section, we briefly overview the relationships between network alignment and other well-known computational problems. We first study the relationship between network alignment and other network-related methods for finding functional modules, including network motifs, network clustering, and network querying. We then discuss other problems that have similar principles and formulations as the network alignment problem. This comparison is especially useful since there are extensive studies in different fields that share similar goals.

5.1 Network Alignment and Other Network Analyses Problems

Network alignment, especially local network alignment, is a powerful technique for finding conserved modules across species (see Sect. 4.2). The basic motivating idea behind identification of conserved modules, is that the existence of specific connected subgraphs that recur in a specific network or across networks is consistent with the tenets of evolutionary theory. Each of these subgraphs, defined by a particular pattern of interactions among vertices, may correspond to a molecular machinery in which specific functions are achieved efficiently [88]. In addition to network alignment, there are other methods for finding functional modules. These methods often use alternate definitions for functional modules.

Modules are often defined independent of function and based on frequency of occurrence. These connected vertex induced subgraphs, which occur at significantly higher frequencies in networks (as compared to random networks), are also known as *network motifs*. *Network clustering* is another approach used for identifying functional modules. The network clustering problem is based on coupling graph vertices in *a single network*, such that vertices in each group are tightly coupled with each other, but are loosely coupled with other vertices. The underlying idea is that functionally related proteins interact with each other, thus need to be in the close proximity. Protein complexes in PPI networks are examples of this class of functional modules.

The network alignment methods discussed in this chapter aim to find a functional mapping between the nodes of the networks being compared. An alternate approach to comparative network analysis is to directly compare the topological properties of these networks. In an attempt to understand the topological characteristics of PPI networks, Pržulj [89] used graphlet distributions across multiple species, where a graphlet refers to a small subgraph with a specified topology. Extensive studies on PPI networks of 14 eukaryotic species showed that this approach outperforms standard topological measures in understanding the functional relationships between different PPI networks.

5.2 Network Alignment vs. Graph Matching

There are different formulations of similarity between two labeled graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$, with adjacency matrices A and B, respectively. The most restricted definition is graph isomorphism: G_1 and G_2 are called *isomorphic*, if there exists a mapping $\pi : V_1 \to V_2$ that maps E_1 to E_2. More formally, two graphs are isomorphic if there exists a permutation matrix P such that $B = PAP^T$. This definition is strict, and cannot be used if the number of vertices in the input graphs is not equal. The *subgraph isomorphism problem* and *maximum common subgraph* are two extensions of the graph isomorphism problem to a more general case where the number of vertices is not equal. In the former, we aim to map the smaller graph to a subset of larger graph, while in the latter, we aim to find a pair of maximum subgraphs in input graphs that are isomorphic. We cannot readily use these extensions because biological networks are usually noisy and contain false positive/negative edges. A more flexible definition of graph similarity is needed.

If one can define the similarity between a pair of nodes in V_1 and V_2, the similarity between G_1 and G_2 can be formulated as a *graph matching problem*. Formally, a matching in graph $G = (V,E)$ is defined as a subset of E without any common vertices, which is also known as independent edge set in G. In other words, any subset of edges in E defines a graph matching in G, if every vertex in V has degree 1 (for detailed definition of graph similarity scores and their relationships to graph matching, please refer to [90]).

An alternate formulation is specifically useful in pairwise one-to-one network alignment. To find the graph similarity between graphs G_1 and G_2, one needs to construct a complete bipartite graph $G_{Bi} = (V_{Bi}, E_{Bi})$, in which $V_{Bi} = \{V_1 \cup V_2\}$, and $E_{Bi} = \{(v_i, v_j) : \forall v_i \in V_1, v_j \in V_2\}$. With this bijection, each one-to-one network alignment between the species, can be viewed as a matching in G_{Bi}, since each alignment assumes that every vertex is at most mapped to one other node, and correspondingly any matching in G_{Bi}, which is a subset of E_{Bi}, does not contain any edges that share the same endpoint. There is a two way connection between matchings and one-to-one network alignment: starting from a matching in G_{Bi}, one can construct the corresponding network alignment of input networks and vice versa. To see this, one may observe that each edge in G_{Bi} defines a potential alignment between a pair of nodes in input networks, while the matching constraint enforces these potential alignments to be one-to-one. This implies that each node is aligned to one other node, at most, from each network.

Among all possible matchings in G_{Bi} (i.e., all pairwise network alignments between G_1 and G_2), one is usually interested in an optimal matching. This is usually achieved by appropriately weighting each edge in the bipartite graph G_{Bi} based on node similarities, as well as local topological similarities, and finding the *maximum weight matching* in G_{Bi}. There are both *exact* and *approximate* algorithms for different versions of the graph matching problem. Among these is a well-known exact polynomial algorithm, known as the Hungarian algorithm [24]. This algorithm has time complexity $O(\max\{V_1, V_2\}^3)$.

5.3 Network Alignment vs. Graph Kernels

Kernel methods are often used in pattern discovery. They can be applied to general data types, including sequences and *graphs* to identify general relations, such as clustering, correlation, and classification. The core of any kernel method is a *kernel function* for measuring the similarity of any given pair of input objects, by mapping them into another space, named *feature space*. Computations in this space are typically easier and input data is more separable. This mapping is defined implicitly, by specifying a kernel function $\kappa : X * X \rightarrow \Re$, as the inner product for the feature space. This is defined as $\kappa(x_1,x_2) = < \phi(x_1), \phi(x_2) >$, where $\phi(.)$ is the embedding function. Note that one does not need to know the mapping $\phi(.)$ explicitly. It suffices to be able to evaluate the kernel function, which is often much easier than computing the coordinates of the points explicitly.

Kernel methods provide useful tools in the analysis of biological networks, since they can be applied at various levels. At the lowest level, one is interested in a kernel function k_v over the set of vertices in *a single graph*. Defining such a kernel function can help us in clustering nodes together based on their similarity. One of the frequently studied examples of this class of problems is the prediction of protein functions in a single network by assigning a kernel function to them that captures both the similarity of the node attributes (amino acid sequences) and local network structure. Diffusion kernels, which are based on random walks in the input graph, are among the most well-known methods for assigning node similarities. At the next level, one is interested in defining k_v for vertices in *multiple graphs*. To compute this kernel, the product graph of the input graphs can be constructed, and the kernel function can be defined based on the random walks in the product graph. The IsoRank method (see Sect. 3.2.2) is similar in nature to this class of problems. At the next level, we aim to define *graph kernels*, instead of *vertex kernels*. Graph kernels can be used to find the similarity between networks, and thus they can be used as a tool for clustering different species based on their similarities. Graph kernels also provide powerful methods for reconstructing phylogenetic trees based on a hierarchical clustering.

Acknowledgments The authors thank Ron Pinter (Technion) and Mehmet Koyutürk (Case Western Reserve) for many useful suggestions.

References

1. Ideker, R.S.T.: Modeling cellular machinery through biological network comparison. Nature Biotechnology **24** (2006) 427–433
2. Flannick, J., Novak, A., Srinivasan, B., McAdams, H., Batzoglou, S.: Graemlin: general and robust alignment of multiple large interaction networks. Genome Research **16**(9) (2006) 1169–1181
3. Kuchaiev, O., Milenković, T., Memišević, V., Hayes, W., Pržulj, N.: Topological network alignment uncovers biological function and phylogeny. Journal of The Royal Society Interface (2010)
4. Kelley, B.P., Yuan, B., Lewitter, F., Sharan, R., Stockwell, B.R., Ideker, T.: PathBLAST: a tool for alignment of protein interaction networks. Nucleic Acids Research **32**(Web-Server-Issue) (2004) 83–88

5. Kelley, B.P., Sharan, R., Karp, R.M., Sittler, T., Root, D.E., Stockwell, B.R., Ideker, T.: Conserved pathways within bacteria and yeast as revealed by global protein network alignment. PNAS **100(20)** (2003)
6. Sharan, R., Ideker, T., Kelley, B.P., Shamir, R., Karp, R.M.: Identification of protein complexes by comparative analysis of yeast and bacterial protein interaction data. Journal of Computational Biology **12**(6) (2005) 835–846
7. Sharan, R., Suthram, S., Kelley, R.M., Kuhn, T., McCuine, S., Uetz, P., Sittler, T., Karp, R.M., Ideker, T.: Conserved patterns of protein interaction in multiple species. Proceedings of the National Academy of Sciences of the United States of America **102**(6) (2005) 1974–1979
8. Bader, J.S., Chaudhuri, A., Rothberg, J.M., Chant, J.: Gaining confidence in high-throughput protein interaction networks. Nature Biotechnology **22**(1) (2003) 78–85
9. Koyutürk, M., Grama, A., Szpankowski, W.: Pairwise local alignment of protein interaction networks guided by models of evolution. In: RECOMB. (2005) 48–65
10. Koyutürk, M., Kim, Y., Topkara, U., Subramaniam, S., Szpankowski, W., Grama, A.: Pairwise alignment of protein interaction networks. Journal of Computational Biology **13(2)** (2006) 182–199
11. Pastor-Satorras, R., Smith, E., Solé, R.V.: Evolving protein interaction networks through gene duplication. J Theo. Bio. **222** (2003) 199–210
12. Vázquez, A., Flammini, A., Maritan, A., Vespignani, A.: Modeling of protein interaction netwokrs. ComPlexUs **1** (2003) 38–44
13. Wagner, A.: How the global structure of protein interaction networks evolves. Proc. R. Soc. Lond. Biol. Sci. **270**(1514) (2003) 457–466
14. Koyutürk, M., Kim, Y., Subramaniam, S., Szpankowski, W., Grama, A.: Detecting conserved interaction patterns in biological networks. Journal of Computational Biology **13**(7) (2006) 1299–1322
15. Tatusov, R., Fedorova, N., Jackson, J., Jacobs, A., Kiryutin, B., Koonin, E., Krylov, D., Mazumder, R., Mekhedov, S., Nikolskaya, A., Rao, B.S., Smirnov, S., Sverdlov, A., Vasudevan, S., Wolf, Y., Yin, J., Natale, D.: The COG database: an updated version includes eukaryotes. BMC Bioinformatics **4**(1) (2003) 41
16. Srinivasan, B.S., Novak, A.F., Flannick, J., Batzoglou, S., McAdams, H.H.: Integrated protein interaction networks for 11 microbes. In: RECOMB. (2006) 1–14
17. Chor, B., Tuller, T.: Biological Networks: Comparison, Conservation, and Evolution via Relative Description Length. Journal of Computational Biology **14**(6) (2007) 817–838
18. Pinter, R.Y., Rokhlenko, O., Yeger-Lotem, E., Ziv-Ukelson, M.: Alignment of metabolic pathways. Bioinformatics **21**(16) (August 2005) 3401–3408
19. Narayanan, M., Karp, R.M.: Comparing protein interaction networks via a graph match-and-split algorithm. Journal of computational biology: a journal of computational molecular cell biology **14**(7) (September 2007) 892–907
20. Bruckner, S., Hüffner, F., Karp, R.M., Shamir, R., Sharan, R.: Topology-free querying of protein interaction networks. Journal of computational biology : a journal of computational molecular cell biology **17**(3) (March 2010) 237–252
21. Banks, E., Nabieva, E., Peterson, R., Singh, M.: NetGrep: fast network schema searches in interactomes. Genome Biology **9**(9) (2008)
22. Bandyopadhyay, S., Sharan, R., Ideker, T.: Systematic identification of functional orthologs based on protein network comparison. Genome Research **16** (2006) 428–435
23. SinghF, R., Xu, J., Berger, B.: Pairwise global alignment of protein interaction networks by matching neighborhood topology. In: RECOMB. (2007) 16–31
24. Kuhn, H.W.: The Hungarian method for the assignment problem. Naval Research Logistic Quarterly **2** (1955) 83–97
25. Singh, R., Xu, J., Berger, B.: Global alignment of multiple protein interaction networks. In: Pacific Symposium on Biocomputing. (2008) 303–314
26. Liao, C.S., Lu, K., Baym, M., Singh, R., Berger, B.: IsoRankN: spectral methods for global alignment of multiple protein networks. Bioinformatics **25**(12) (2009) i253–i258

27. Flannick, J., Novak, A.F., Do, C.B., Srinivasan, B.S., Batzoglou, S.: Automatic parameter learning for multiple network alignment. In: RECOMB. (2008) 214–231
28. Kanehisa, M., Goto, S., Kawashima, S., Okuno, Y., Hattori, M.: The KEGG resource for deciphering the genome. Nucleic acids research **32**(Database issue) (2004) D277–280
29. Zhenping, L., Zhang, S., Wang, Y., Zhang, X.S., Chen, L.: Alignment of molecular networks by integer quadratic programming. Bioinformatics **23**(13) (2007) 1631–1639
30. Bayati, M., Gerritsen, M., Gleich, D., Saberi, A., Wang, Y.: Algorithms for large, sparse network alignment problems. In: ICDM. (2009) 705–710
31. Klau, G.W.: A new graph-based method for pairwise global network alignment. BMC Bioinformatics **10**(S-1) (2009)
32. Zaslavskiy, M., Bach, F.R., Vert, J.P.: Global alignment of protein-protein interaction networks by graph matching methods. Bioinformatics **25**(12) (2009) i259–1267
33. Zaslavskiy, M., Bach, F., Vert, J.P.: A path following algorithm for the graph matching problem. IEEE Trans. Pattern Anal. Mach. Intell. **31**(12) (2009) 2227–2242
34. Frank, M., Wolfe, P.: An algorithm for quadratic programming. Naval Research Logistics Quarterly **3** (1956) 95–110
35. Kalaev, M., Bafna, V., Sharan, R.: Fast and accurate alignment of multiple protein networks. In: RECOMB. (2008) 246–256
36. Ashburner, M., Ball, C.A., Blake, J.A., Botstein, D., Butler, H., Cherry, J.M., Davis, A.P., Dolinski, K., Dwight, S.S., Eppig, J.T., Harris, M.A., Hill, D.P., Issel-Tarver, L., Kasarskis, A., Lewis, S., Matese, J.C., Richardson, J.E., Ringwald, M., Rubin, G.M., Sherlock, G.: Gene ontology: tool for the unification of biology. the gene ontology consortium. Nature genetics **25**(1) (2000) 25–29
37. Tipton, K.F., Nomenclature Committee of the International Union of Biochemistry and Molecular Biology (NC-IUBMB). Enzyme nomenclature. Recommendations 1992. Supplement: corrections and additions. European journal of biochemistry/FEBS, **223**(1) (1994) 1–5
38. Ruepp, A., Zollner, A., Maier, D., Albermann, K., Hani, J., Mokrejs, M., Tetko, I., Güldener, U., Mannhaupt, G., Münsterkötter, M., Mewes, H.W.: The FunCat, a functional annotation scheme for systematic classification of proteins from whole genomes. Nucleic Acids Res **32**(18) (2004) 5539–5545
39. Hawkins, T., Kihara, D.: Function prediction of uncharacterized proteins. J. Bioinformatics and Computational Biology **5**(1) (2007) 1–30
40. Kandasamy, K., Mohan, S.S., Raju, R., Keerthikumar, S., Kumar, G.S.S., Venugopal, A.K., Telikicherla, D., Navarro, J.D., Mathivanan, S., Pecquet, C., Gollapudi, S.K.K., Tattikota, S.G.G., Mohan, S., Padhukasahasram, H., Subbannayya, Y., Goel, R., Jacob, H.K.K., Zhong, J., Sekhar, R., Nanjappa, V., Balakrishnan, L., Subbaiah, R., Ramachandra, Y., Rahiman, B.A., Prasad, T.K.K., Lin, J.X.X., Houtman, J.C.C., Desiderio, S., Renauld, J.C.C., Constantinescu, S.N., Ohara, O., Hirano, T., Kubo, M., Singh, S., Khatri, P., Draghici, S., Bader, G.D., Sander, C., Leonard, W.J., Pandey, A.: NetPath: a public resource of curated signal transduction pathways. Genome biology **11**(1) (2010)
41. Hamosh, A., Scott, A.F., Amberger, J.S., Bocchini, C.A., Mckusick, V.A.: Online Mendelian Inheritance in Man (OMIM), a knowledgebase of human genes and genetic disorders. Nucleic acids research **33**(Database issue) (2005)
42. Boyle, E.I., Weng, S., Gollub, J., Jin, H., Botstein, D., Cherry, J.M., Sherlock, G.: GO: TermFinder-open source software for accessing gene ontology information and finding significantly enriched gene ontology terms associated with a list of genes. Bioinformatics **20**(18) 3710+
43. Bader, G.D., Donaldson, I., Wolting, C., Ouellette, B.F.F., Pawson, T., Hogue, C.W.V.: BIND–The Biomolecular Interaction Network Database. Nucl. Acids Res. **29**(1) (2001) 242–245
44. Xenarios, I., Rice, D.W., Salwinski, L., Baron, M.K., Marcotte, E.M., Eisenberg, D.: DIP: The Database of Interacting Proteins. Nucleic acids research **28**(1) (2000) 289–291
45. Hermjakob, H., Montecchi-Palazzi, L., Lewington, C., Mudali, S., Kerrien, S., Orchard, S.E., Vingron, M., Roechert, B., Roepstorff, P., Valencia, A., Margalit, H., Armstrong, J., Bairoch, A., Cesareni, G., Sherman, D.J., Apweiler, R.: IntAct: an open source molecular interaction database. Nucleic Acids Research **32**(Database-Issue) (2004) 452–455

46. Stark, C., Breitkreutz, B.J., Reguly, T., Boucher, L., Breitkreutz, A., Tyers, M.: BioGRID: a general repository for interaction datasets. Nucl. Acids Res. **34** (2006) D535–539
47. Chatr-aryamontri, A., Ceol, A., Palazzi, L.M.M., Nardelli, G., Schneider, M.V.V., Castagnoli, L., Cesareni, G.: MINT: the Molecular INTeraction database. Nucleic acids research **35**(Database issue) (2007) D572–574
48. Güldener, U., Münsterkötter, M., Oesterheld, M., Pagel, P., Ruepp, A., Mewes, H.W., Stümpflen, V.: MPact: the MIPS protein interaction resource on yeast. Nucleic Acids Research **34**(Database-Issue) (2006) 436–441
49. Karp, P.D., O.C.M.K.C.G.L.K.: Expansion of the biocyc collection of pathway/genome databases to 160 genomes. Nucleic Acids Research **33** (2005) 6083–9
50. Joshi-Tope, G., Gillespie, M., Vastrik, I., D'Eustachio, P., Schmidt, E., de Bono, B., Jassal, B., Gopinath, G.R., Wu, G.R., Matthews, L., Lewis, S., Birney, E., Stein, L.: Reactome: a knowledgebase of biological pathways. Nucleic acids research **33**(Database issue) (2005) D428–432
51. Schaefer, C.F.F., Anthony, K., Krupa, S., Buchoff, J., Day, M., Hannay, T., Buetow, K.H.: PID: the pathway interaction database. Nucleic Acids Research **37**(Database issue) (2009) 674–679
52. Bairoch, A., Apweiler, R., Wu, C.H., Barker, W.C., Boeckmann, B., Ferro, S., Gasteiger, E., Huang, H., Lopez, R., Magrane, M., Martin, M.J., Natale, D.A., O'Donovan, C., Redaschi, N., Yeh, L.S.L.: The Universal Protein Resource (UniProt). Nucleic Acids Research **33**(Database-Issue) (2005) 154–159
53. Tateno, Y., Imanishi, T., Miyazaki, S., Fukami-Kobayashi, K., Saitou, N., Sugawara, H., Gojobori, T.: DNA Data Bank of Japan (DDBJ) for genome scale research in life science. Nucleic Acids Research **30**(1) (2002) 27–30
54. Stoesser, G., Tuli, M.A., Lopez, R., Sterk, P.: The EMBL Nucleotide Sequence Database. Nucl. Acids Res. **27**(1) (1999) 18–24
55. Benson, D.A., Karsch-Mizrachi, I., Lipman, D.J., Ostell, J., Sayers, E.W.: GenBank. Nucleic Acids Research **37**(Database-Issue) (2009) 26–31
56. Gelbart, W.M., Crosby, M.A., Matthews, B., Rindone, W.P., Chillemi, J., Twombly, S.R., Emmert, D., Ashburner, M., Drysdale, R.A., Whitfield, E., Millburn, G.H., de Grey, A., Kaufman, T., Matthews, K., Gilbert, D., Strelets, V.B., Tolstoshev, C.: FlyBase: a Drosophila database. The FlyBase consortium. Nucleic Acids Research **25**(1) (1997) 63–66
57. Rhee, S.Y., Beavis, W.D., Berardini, T.Z., Chen, G., Dixon, D.A., Doyle, A., Garcia-Hernandez, M., Huala, E., Lander, G., Montoya, M., Miller, N., Mueller, L.A., Mundodi, S., Reiser, L., Tacklind, J., Weems, D.C., Wu, Y., Xu, I., Yoo, D., Yoon, J., Zhang, P.: The Arabidopsis Information Resource (TAIR): a model organism database providing a centralized, curated gateway to Arabidopsis biology, research materials and community. Nucleic Acids Research **31**(1) (2003) 224–228
58. Bult, C.J., Blake, J.A., Richardson, J.E., Kadin, J.A., Eppig, J.T.: The Mouse Genome Database (MGD): integrating biology with the genome. Nucleic Acids Research **32**(Database-Issue) (2004) 476–481
59. Twigger, S.N., Lu, J., Shimoyama, M., Chen, D., Pasko, D., Long, H., Ginster, J., Chen, C.F., Nigam, R., Kwitek, A.E., Eppig, J.T., Maltais, L., Maglott, D.R., Schuler, G.D., Jacob, H.J., Tonellato, P.J.: Rat Genome Database (RGD): mapping disease onto the genome. Nucleic Acids Research **30**(1) (2002) 125–128
60. Hong, E.L., Balakrishnan, R., Dong, Q., Christie, K.R., Park, J., Binkley, G., Costanzo, M.C., Dwight, S.S., Engel, S.R., Fisk, D.G., Hirschman, J.E., Hitz, B.C., Krieger, C.J., Livstone, M.S., Miyasato, S.R., Nash, R.S., Oughtred, R., Skrzypek, M.S., Weng, S., Wong, E.D., Zhu, K.K., Dolinski, K., Botstein, D., Cherry, J.M.: Gene Ontology annotations at SGD: new data sources and annotation methods. Nucleic Acids Research **36**(Database-Issue) (2008) 577–581
61. Dolinski, K., Botstein, D.: Orthology and functional conservation in eukaryotes. Annual Review of Genetics **41**(1) (2007) 465–507
62. O'Brien, K.P., Remm, M., Sonnhammer, E.L.L.: Inparanoid: a comprehensive database of eukaryotic orthologs. Nucl. Acids Res. **33** (2005) D476–480

63. Pereira-Leal, J.B., Levy, E.D., Teichmann, S.A.: The origins and evolution of functional modules: lessons from protein complexes. Phil. Trans. R. Soc. **361**(1467) (2006) 507–517
64. Hartwell, L.H., Hopfield, J.J., Leibler, S., Murray, A.W.: From molecular to modular cell biology. Nature **402** (1999) C47–C51
65. Spirin, V., Mirny, L.A.: Protein complexes and functional modules in molecular networks. Proc Natl Acad Sci U S A **100**(21) 12123–8+
66. Pereira-Leal, J., Enright, A., Ouzounis, C.: Detection of functional modules from protein interaction networks. Proteins **54** (2004) 49–57
67. Ravasz, E., Somera, A.L., Mongru, D.A., Oltvai, Z.N., Barabasi, A.L.: Hierarchical organization of modularity in metabolic networks. Science **297**(5586) (2002) 1551–1555
68. Ihmels, J., Friedlander, G., Bergmann, S., Sarig, O., Ziv, Y., Barkai, N.: Revealing modular organization in the yeast transcriptional network. Nat Genet **31**(4) (2002) 370–377
69. Segal, E., Shapira, M., Regev, A., Pe'er, D., Botstein, D., Koller, D., Friedman, N.: Module networks: identifying regulatory modules and their condition-specific regulators from gene expression data. Nat Genet **34**(2) (2003) 166–176
70. Erten, S., Li, X., Bebek, G., Li, J., Koyutürk, M.: Phylogenetic analysis of modularity in protein interaction networks. BMC Bioinformatics **10**(1) (2009)
71. Robertson, D.L., Lovell, S.C.: Evolution in protein interaction networks: co-evolution, rewiring and the role of duplication. Biochemical Society transactions **37** (2009) 768–771
72. Huan, J., Bandyopadhyay, D., Wang, W., Snoeyink, J., Prins, J., Tropsha, A.: Comparing graph representations of protein structure for mining family-specific residue-based packing motifs. Journal of Computational Biology **12** (2005) 657–671
73. Promislow, D.E.: Protein networks, pleiotropy and the evolution of senescence. In: Proc Biol Sci. Volume 271. (2004) 1225–1234
74. Wagner, G.P., Pavlicev, M., Cheverud, J.M.: The road to modularity. Nat Rev Genet **8**(12) (2007) 921–31
75. Yosef, N., Kupiec, M., Ruppin, E., Sharan, R.: A complex-centric view of protein network evolution. Nucl. Acids Res. **37**(12) (2009)
76. Towfic, F., Greenlee, M.H.W., Honavar, V.: Aligning biomolecular networks using modular graph kernels. In: WABI. Volume 5724 of Lecture Notes in Computer Science, Springer (2009) 345–361
77. Joseph, L., Isaac, K., Albert-Laszlo, B.: Human disease classification in the postgenomic era: a complex systems approach to human pathobiology. Molecular Systems Biology **3**(124) (2007)
78. Chavali, S., Barrenas, F., Kanduri, K., Benson, M.: Network properties of human disease genes with pleiotropic effects. BMC Systems Biology **4**(1) (2010)
79. Lin, Z.: Bioinformatics Basics: Applications in Biological Science and Medicine. Edited by Lukas K. Buehler and Hooman H. Rashidi. Brief Bioinform **9**(3) (2008) 256–257
80. Zhu, X., Gerstein, M., Snyder, M.: Getting connected: analysis and principles of biological networks. Genes and Development **21**(9) (2007) 1010–1024
81. Lei, G., Ji, Q.: Whole genome molecular phylogeny of large dsdna viruses using composition vector method. BMC Evol Bio **7**(41) (2007)
82. Palmenberg, A.C., Spiro, D., Kuzmickas, R., Wang, S., Djikeng, A., Rathe, J.A., Fraser-Liggett, C.M., Liggett, S.B.: Sequencing and analyses of all known human rhinovirus genomes reveals structure and evolution. Science (2009)
83. Riester, M., Stephan-Otto Attolini, C., Downey, R.J., Singer, S., Michor, F.: A differentiation-based phylogeny of cancer subtypes. PLoS Comput Biol **6**(5) (2010)
84. Suthram S, Sittler T, I.T.: The Plasmodium protein network diverges from those of other eukaryotes. Nature (2005)
85. Tan, C.S.H.S., Bodenmiller, B., Pasculescu, A., Jovanovic, M., Hengartner, M.O., Jørgensen, C., Bader, G.D., Aebersold, R., Pawson, T., Linding, R.: Comparative analysis reveals conserved protein phosphorylation networks implicated in multiple diseases. Science signaling **2**(81) (2009)
86. Wu, X., Liu, Q., Jiang, R.: Align human interactome with phenome to identify causative genes and networks underlying disease families. Bioinformatics **25**(1) (2009)

87. Goh, K.I., Cusick, M.E., Valle, D., Childs, B., Vidal, M., Barabási, A.L.: The human disease network. PNAS **104**(21) (2007) 8685–8690
88. Milo, R., Shen-Orr, S., Itzkovitz, S., Kashtan, N., Chklovskii, D., Alon, U.: Network motifs: simple building blocks of complex networks. Science (New York, N.Y.) **298**(5594) (2002) 824–827
89. Pržulj, N.: Biological network comparison using graphlet degree distribution. Bioinformatics **26** (March 2010) 853–854
90. Zager, L.: Graph similarity and matching. Master's thesis, MIT (2005)