PROBABILISTIC RANDOM WALK MODELS FOR COMPARATIVE NETWORK ANALYSIS

A Dissertation

by

HYUNDOO JEONG

Submitted to the Office of Graduate and Professional Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Chair of Committee,	Byung-Jun Yoon
Co-Chair of Committee,	Xiaoning Qian
Committee Members,	Edward R. Dougherty
	P. R. Kumar
	Won-Bo Shim
Head of Department,	Miroslav M. Begovic

August 2017

Major Subject: Electrical Engineering

ABSTRACT

Graph-based systems and data analysis methods have become critical tools in many fields as they can provide an intuitive way of representing and analyzing interactions between variables. Due to the advances in measurement techniques, a massive amount of labeled data that can be represented as nodes on a graph (or network) have been archived in databases. Additionally, novel data without label information have been gradually generated and archived. Labeling and identifying characteristics of novel data is an important first step in utilizing the valuable data in an effective and meaningful way. Comparative network analysis is an effective computational means to identify and predict the properties of the unlabeled data by comparing the similarities and differences between well-studied and less-studied networks. Comparative network analysis aims to identify the matching nodes and conserved subnetworks across multiple networks to enable a prediction of the properties of the nodes in the less-studied networks based on the properties of the matching nodes in the well-studied networks (i.e., transferring knowledge between networks).

One of the fundamental and important questions in comparative network analysis is how to accurately estimate node-to-node correspondence as it can be a critical clue in analyzing the similarities and differences between networks. Node correspondence is a comprehensive similarity that integrates various types of similarity measurements in a balanced manner. However, there are several challenges in accurately estimating the node correspondence for large-scale networks. First, the scale of the networks is a critical issue. As networks generally include a large number of nodes, we have to examine an extremely large space and it can pose a computational challenge due to the combinatorial nature of the problem. Furthermore, although there are matching nodes and conserved subnetworks in different networks, structural variations such as node insertions and deletions make it

difficult to integrate a topological similarity.

In this dissertation, novel probabilistic random walk models are proposed to accurately estimate node-to-node correspondence between networks. First, we propose a context-sensitive random walk (CSRW) model. In the CSRW model, the random walker analyzes the context of the current position of the random walker and it can switch the random movement to either a simultaneous walk on both networks or an individual walk on one of the networks. The context-sensitive nature of the random walker enables the method to effectively integrate different types of similarities by dealing with structural variations. Second, we propose the CUFID (Comparative network analysis Using the steady-state network Flow to IDentify orthologous proteins) model. In the CUFID model, we construct an integrated network by inserting pseudo edges between potential matching nodes in different networks. Then, we design the random walk protocol to transit more frequently between potential matching nodes as their node similarity increases and they have more matching neighboring nodes. We apply the proposed random walk models to comparative network analysis problems: global network alignment and network querying. Through extensive performance evaluations, we demonstrate that the proposed random walk models can accurately estimate node correspondence and these can lead to improved and reliable network comparison results.

DEDICATION

To my father, mother, brother, niece and my love Kyoung Hwa.

ACKNOWLEDGMENTS

First of all, I would like to express my sincere and deepest gratitude to my advisor, Professor Byung-Jun Yoon, for his priceless advice and support for my graduate studies. I will always be thankful for how he put my best interest first in order to successfully further my career. I will forever remember him as my life's greatest advisor, teacher, and mentor. I would also like to acknowledge my co-advisor, Professor Xiaoning Qian, for his guidance and encouragement in my research. I respect his enthusiasm and integrity for research and hope to one day follow in his ways.

I would also like to acknowledge my proposal and defense committee members: Professor Edward R. Dougherty, Professor P. R. Kumar, Professor Won-Bo Shim, Professor I-Hong Hou, and Professor Yoonsuck Choe. I also recognize and appreciate Dr. A. Datta, Dr. Charles D. Johnson and Dr. Noushin Ghaffari at the Center for Bioinformatics and Genomics Systems Engineering.

I would like to thank my parents for their unconditional love, support, and encouragement. My mother's constant prayers for my health and wellbeing and my father's endless dedication in support of our family have served as a great inspiration to me throughout my studies. I would also like to thank my brother for his concern and consistent prayers for me. I would like to give a special thanks to my lovely niece, Ha Yool. In every difficult moment or hardship that came my way, her smile and laughter would lift my spirits. I would like to thank my beloved Kyoung Hwa for her encouragements and prayers for me.

Last but not least, I would like to thank God for his guidance in my lifetime. My life without God is nothing. Soli Deo gloria. All glory to God.

CONTRIBUTORS AND FUNDING SOURCES

Contributors

This work was supported by a dissertation committee consisting of Professor Byung-Jun Yoon, Professor Xiaoning Qian, Professor Edward R. Dougherty, and Professor P. R. Kumar of the Department of Electrical and Computer Engineering and Professor Won-Bo Shim of the Department of Plant Pathology and Microbiology.

Funding Sources

Graduate study was supported by a National Science Foundation (NSF) CAREER: Models and Algorithms for Comparative Analysis of Biological Networks (NSF award CCF-1149544) and EAGER: Identifying Blockmodel Functional Modules across Multiple Networks (NSF award CCF-1447235).

NOMENCLATURE

BLAST	Basic Local Alignment Search Tool
CE	Conserved Edges
CI	Conserved Interactions
CN	Correct Nodes
COI	Conserved Orthologous Interactions
CSRW	Context-Sensitive Random Walk
CUFID	Comparative network analysis Using the steady-state network Flow to IDentify orthologous proteins
FDR	False Discovery Rate
GO	Gene Ontology
GOC	Gene Ontology Consistency
HMM	Hidden Markov Model
IC	Information Content
MEA	Maximum Expected Accuracy
MNE	Mean Normalized Entropy
MWBM	Maximum Weighted Bipartite Matching
PCT	Probabilistic Consistent Transformation
PPI	Protein-Protein Interactions
PPR	Personalized PageRank
SPE	SPEcificity

TABLE OF CONTENTS

Page

ABSTRACT ii

DEDICATION iv

ACKNOWLEDGMENTS v

CONTRIBUTORS AND FUNDING SOURCES vi

NOMENCLATURE vii

TABLE OF CONTENTS viii

LIST OF FIGURES x

LIST OF TABLES xii

1. INTRODUCTION 1

    1.1 Background 1
    1.2 Outline of the dissertation 2

2. ESTIMATION OF NODE-TO-NODE CORRESPONDENCE BETWEEN DIFFERENT GRAPHS 4

    2.1 Context-sensitive random walk model 4
        2.1.1 Motivation and overall approach 5
        2.1.2 Proposed random walk model 6
        2.1.3 Performance assessments 8
        2.1.4 Conclusions 11
    2.2 Network alignment through the context-sensitive random walk model . . 11
        2.2.1 Background and motivation 11
        2.2.2 Methods 14
        2.2.3 Results 18
        2.2.4 Conclusions 28
    2.3 Network querying through the context-sensitive random walk model . . . 29
        2.3.1 Background and motivtion 29
        2.3.2 Methods 32

2.3.3 Results . . . 39
2.3.4 Conclusions . . . 49

3. ESTIMATION OF NODE-TO-NODE CORRESPONDENCE BY MEASURING THE STEADY-STATE NETWORK FLOW USING A MARKOV MODEL 51

3.1 CUFID model . . . 51
3.1.1 Problem formulation . . . 51
3.1.2 Motivation and overall approach . . . 52
3.1.3 Methods . . . 54
3.2 Network alignment through the CUFID model . . . 61
3.2.1 Methods . . . 61
3.2.2 Results . . . 63
3.2.3 Conclusions . . . 74
3.3 Network querying through the CUFID model . . . 74
3.3.1 Methods . . . 76
3.3.2 Results . . . 86
3.3.3 Conclusions . . . 95

4. SUMMARY AND CONCLUSIONS . . . 97

REFERENCES . . . 99

APPENDIX A. LIST OF DATABASES FOR COMPARATIVE NETWORK ANALYSIS . . . 112

APPENDIX B. SOFTWARE AVAILABILITY . . . 114

LIST OF FIGURES

FIGURE Page

2.1 Illustration of the context-sensitive random walk model. 5

2.2 Performance dependence on pairwise node similarity. 10

2.3 The total number of conserved orthologous interactions and conserved interactions. 22

2.4 Equivalence class coverage for 5-way network alignment. 23

2.5 Equivalence class coverage for 8-way network alignment. 24

2.6 Computation time for aligning real PPI networks. 28

2.7 Illustration for the query network and conserved subnetwork in the target network. 33

2.8 Example for the pre-processing: removing non-homologous nodes. 35

2.9 Example for the pre-processing: inserting pseudo edges. 35

2.10 Number of matches for each query and target species pair (i.e., query species – target species). 44

2.11 Number of significant hits and significant functionally coherent (FC) hits for the 863 query complexes. 45

2.12 Number of hits and FC hits for querying 863 biological complexes. 47

2.13 Computation time of 863 querying results for each querying algorithm. . . 49

3.1 Illustration of how node correspondence is measured based on the steady-state network flow. 54

3.2 Illustration for constructing the integrated network from a network pair. . 55

3.3 Illustration of the steady-state network flow. 59

3.4 Illustration of the main difference between CSRW model and CUFID model. 60

3.5 GOC scores of various pairwise network alignment algorithms. 68

3.6 Illustration of a typical network querying problem. 75

3.7 Illustration for constructing the integrated network by combining the query and target networks. 79

3.8 Estimating the steady-state network flow based on the CUFID model. . . 81

3.9 The number of hits and the number of meaningful hits are shown for each network querying algorithm. 89

3.10 The number of specific hits for each network querying algorithm. 91

3.11 The specificity of the predictions made by different network querying algorithms. 92

3.12 Computation time for each algorithm. 94

LIST OF TABLES

TABLE Page

2.1 Performance comparison of different scoring methods. 9

2.2 Performance comparison for pairwise network alignment. 21

2.3 Performance comparison for 5-way network alignment. 21

2.4 Performance comparison for 8-way network alignment. 21

2.5 Mean computation time for aligning PPI networks in the NAPAbench. . . 25

2.6 Pairwise network alignment results for real PPI networks. 26

2.7 Multiple network alignment results for real PPI networks (for 3 species). . 26

2.8 Significant SPE for the ontology aspect of "cellular component". 46

2.9 SPE for the ontology aspect of "cellular component". 48

3.1 Memory complexity to construct a transition probability matrix. 61

3.2 Pairwise alignment results for the IsoBase dataset. Protein functionality is determined based on the KEGG Orthology (KO) group annotations. . . . 67

3.3 Number of conserved interactions (CI) obtained by different network alignment algorithms. 69

3.4 Number of conserved orthologous interactions (COI) obtained by different network alignment algorithms. 70

3.5 CPU time of the tested network alignment algorithms (in seconds). 72

3.6 The number of identified nodes and the number of annotated nodes. . . . 93

A.1 List of available databases for PPI network analysis. 112

A.2 Databases for known biological complexes. 112

A.3 Databases for proteins. 113

B.1 List of softwares proposed in this dissertation. 114

1. INTRODUCTION

1.1 Background

Graph-based system and data analysis techniques have become a critical tool in many fields as it can provide an intuitive way of representing interactions between variables and analyzing them [1, 2, 3, 4]. In recent years, graph-based techniques have been widely applied to the analysis of social networks [5, 6], images [7, 8], and biological networks [9, 10]. Additionally, we can infer the properties of the less-studied system by comparing it with the well-studied systems and finding the corresponding elements. To this aim, given multiple graphs, one question that is of practical importance is how the nodes in a given graph can be mapped to nodes in the other graphs based on the similarity between nodes and the topological similarity between graphs. Considering that each node may have a number of similar nodes in the other graphs and that the graphs may have significant differences in their topology, quantitatively estimating this overall similarity between nodes – or the *node correspondence* – is theoretically challenging. Furthermore, estimating these similarities can pose computational challenges, especially for large graphs, due to the combinatorial nature of the problem.

So far, several methods have been proposed for measuring the node correspondence between graphs, where random walk based methods have been popular as they are intuitive and can be efficiently implemented [10, 11, 12, 13, 14, 15]. These methods perform a simultaneous random walk on the two graphs to be compared, where the random walk scheme is designed such that the walker more frequently visits (or stays longer at) node pairs that have higher similarity and are surrounded by a larger number of similar node pairs. The stationary probability of the resulting (semi-)Markov model gives us the long-run proportion of time that the random walker simultaneously visits (and stays at) a given

node pair, which can be used as the correspondence score between the two nodes. This score provides a simple and intuitive way of measuring the overall similarity between two nodes in different graphs by integrating the node similarity and the topological similarity [10]. Recently, these random walk models have been applied to the comparative analysis of large-scale biological networks [12, 13].

In this study, we have studied effective methods for comparative network analysis based on a graphical representation of systems so that we can transfer the knowledge of the well-analyzed system into the less-studied system. We have proposed novel random walk models that can significantly improve the accuracy of the estimation of the node-to-node correspondence between different graphs. Additionally, we have verified the effectiveness of the proposed method on biological networks. Although we mainly present the performance evaluations using biological networks, the proposed random walk models and algorithms can be applied to various types of networks. Note that two terms, network and graph, are utilized interchangeably in this dissertation.

1.2 Outline of the dissertation

In this dissertation, we propose novel probabilistic random walk models and present their applications to comparative network analysis using biological networks. In the chapter 2, we propose the context-sensitive random walk model to estimate node-to-node correspondence between graphs through a long-run behavior of a random walker. In the chapter 3, we propose a novel random walk model, called the CUFID model, to estimate node correspondences by measuring the steady-state network flow between networks. We will show that the CUFID model further improves the estimation accuracy of the node correspondences with the reduced computational complexity. In the chapter 2 and 3, we present the potential applications of the proposed random walk models in global network alignment and network querying problem. We will demonstrate the effectiveness of the

proposed random walk models through extensive performance evaluations using synthetic networks and real biological networks.

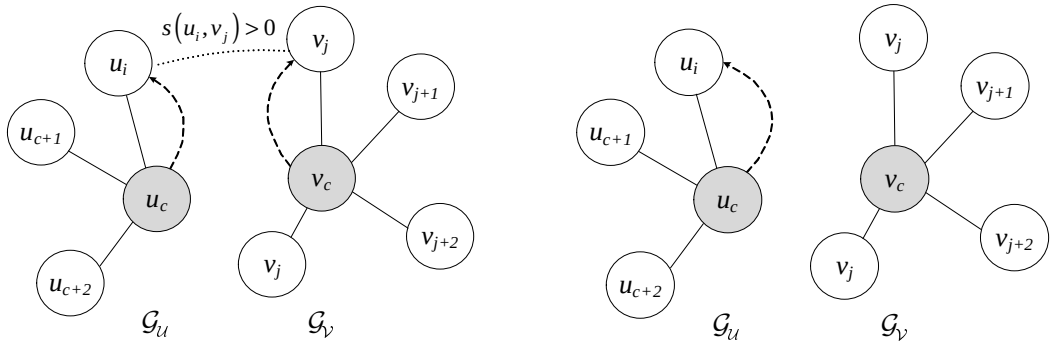
2. ESTIMATION OF NODE-TO-NODE CORRESPONDENCE BETWEEN DIFFERENT GRAPHS *

In this chapter, we propose a novel random walk model that can significantly improve the accuracy of the estimation of the node correspondence between different graphs. The proposed random walker performs a random walk on the two graphs to be compared, where it can switch its mode between a simultaneous walk on both graphs and an individual walk on one of the graphs. The mode switching is determined by the presence (or absence) of similar node pairs among the current neighbors. Through extensive simulations, we show that the proposed model leads to an enhanced node-correspondence scoring method that clearly outperforms existing methods.

2.1 Context-sensitive random walk model

Consider two graphs $\mathcal{G}_\mathcal{U} = (\mathcal{U}, \mathcal{D})$ and $\mathcal{G}_\mathcal{V} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{G}_\mathcal{U}$ consists of a set $\mathcal{U} = \{u_1, u_2, \cdots\}$ of nodes and a set $\mathcal{D} = \{d_{ij}\}$ of edges between nodes u_i and u_j and $\mathcal{G}_\mathcal{V}$ consists of a set $\mathcal{V} = \{v_1, v_2, \cdots\}$ of nodes and a set $\mathcal{E} = \{e_{\ell m}\}$ of edges between nodes v_ℓ and v_m. We assume that a nonnegative pairwise node similarity score $s(u_i, v_j)$ is given for every node pair (u_i, v_j). Our goal is to estimate the node correspondence score $c(u_i, v_j)$ for every node pair (u_i, v_j) that quantifies the overall similarity between these nodes by integrating the pairwise node similarity scores and the topological similarity between the two graphs in a reasonable manner. In other words, we want the node correspondence score $c(u_i, v_j)$ to be proportional to the posterior alignment probability $P\left[u_i \sim v_j | \mathcal{G}_\mathcal{U}, \mathcal{G}_\mathcal{V}\right]$

*Part of this chapter is reprinted with a permission from "Hyundoo Jeong and Byung-Jun Yoon. Effective estimation of node-to-node correspondence between different graphs. *IEEE Signal Processing Letters*" [16] © [2015] IEEE and "Hyundoo Jeong and Byung-Jun Yoon. Accurate multiple network alignment through context-sensitive random walk. *BMC Systems Biology*, 9(Suppl. 1):S7, 2015" [17] © [2015] BioMed Central and "Hyundoo Jeong and Byung-Jun Yoon. SEQUOIA: Significance enhanced network querying through context-sensitive random walk and minimization of network conductance. *BMC Systems Biology*," 11(Suppl. 3):20, [18] © [2017] BioMed Central.



(a) Simultaneous walk on both graphs

(b) Individual walk on either graph ($\mathcal{G}_\mathcal{U}$ in the example shown)

Figure 2.1: Illustration of the context-sensitive random walk model. The shaded nodes show the current position of the random walker on the two graphs. The dashed arrows indicate the movement of the random walker at the next time step [16] © [2015] IEEE.

of u_i and v_j given $\mathcal{G}_\mathcal{U}$ and $\mathcal{G}_\mathcal{V}$.

2.1.1 Motivation and overall approach

We propose a novel random walk model to measure the node correspondence score $c(u_i, v_j)$. Our random walk model is motivated by the pair hidden Markov model (pair-HMM), which has been widely used for the comparative analysis of biological sequences (e.g., sequence alignment) due to its simplicity and effectiveness [19, 20].

Unlike traditional HMMs, which generate a single symbol sequence, the pair-HMM generates a pair of aligned symbol sequences. A typical pair-HMM has three different states: M, I_1, and I_2. At the M state (indicates a "matched" symbol pair), the HMM emits an aligned symbol pair. On the other hand, at the I_k state (indicates an "inserted" symbol in either sequence), the HMM only emits a symbol to sequence-k alone that is aligned to a gap symbol in the other sequence. Given two (unaligned) symbol sequences, we can use the forward-backward algorithm to predict the alignment probabilities between symbols in the two sequences based on the pair-HMM [20].

Similarly, the proposed random walk model has three different internal states, M, I_U, and I_V, where each state corresponds to a different "mode" of random walk. At a M state, which corresponds to "matched" node pair, the random walker makes a simultaneous walk on both graphs, moving into a pair of matched nodes. This is illustrated in Figure 2.1a. On the other hand, at state I_U (or state I_V), the random walker makes an "individual" walk on graph $\mathcal{G}_\mathcal{U}$ (or $\mathcal{G}_\mathcal{V}$). Figure 2.1b illustrates the individual walk at state I_U. The random walker can switch its mode between a simultaneous walk and an individual walk, in a context dependent way by examining the neighborhood. In the presence of node pairs in the immediate neighborhood with a positive node similarity score, the random walker will make a simultaneous move on both graphs by randomly moving into one of the similar node pairs (M state). Otherwise, the random walker will make a transition to either state I_U or I_V and make a random move only on the corresponding graph.

Based on this random walk model, we estimate the steady state probabilities of this random walk, or in other words, the long-run proportion of time that the random walker will simultaneously visit a given node pair. Finally, from these steady state probabilities, we estimate the actual proportion of time that the random walker spends at a given node pair by "entering" the nodes *simultaneously* (i.e., at state M), which we used as the correspondence score for the node pair. It should be noted that this last step is crucial, since we are not interested in the case when the random walker happens to stay at a node pair as a result of an individual move on one of the graphs. In such cases, the simultaneous visit of the two nodes is coincidental and is not a direct result of the relevance between the given nodes.

2.1.2 Proposed random walk model

Let $\mathcal{G}_\mathcal{X} = (\mathcal{X}, \mathcal{E}_\mathcal{X})$ be the product graph of $\mathcal{G}_\mathcal{U}$ and $\mathcal{G}_\mathcal{V}$, where the nodes in the graph $\mathcal{G}_\mathcal{X}$ correspond to node pairs (u_i, v_j), $u_i \in \mathcal{U}$ and $v_i \in \mathcal{V}$. Two nodes in the prod-

uct graph $\mathcal{G}_{\mathcal{X}}$ are connected if and only if the corresponding nodes are connected in both $\mathcal{G}_{\mathcal{U}}$ and $\mathcal{G}_{\mathcal{V}}$. Joint random walk on the two graphs $\mathcal{G}_{\mathcal{U}}$ and $\mathcal{G}_{\mathcal{V}}$, both simultaneous walk and individual walk, can be viewed as a random walk on this product graph $\mathcal{G}_{\mathcal{X}}$. We define $\mathcal{M} = \{(u_i, v_j) \,|s\,(u_i, v_j) > 0, u_i \in \mathcal{U}, v_j \in \mathcal{V}\}$ as the set of similar node pairs, where $s\,(u_i, v_j)$ is the pairwise node similarity score for the node pair (u_i, v_j). Suppose that the random walker is currently located at (u_c, v_c) for some $u_c \in \mathcal{U}$ and $v_c \in \mathcal{V}$. Let us define the set of similar node pairs in the neighborhood of (u_c, v_c) as $\mathcal{N}(u_c, v_c) = \{(u_i, v_j) \,|u_i \in \mathcal{N}(u_c), v_j \in \mathcal{N}(v_c), (u_i, v_j) \in \mathcal{M}\}$, where $\mathcal{N}(u_c)$ is the set of neighbors of node u_c in graph $\mathcal{G}_{\mathcal{U}}$ and $\mathcal{N}(v_c)$ is the set of neighbors of node v_c in graph $\mathcal{G}_{\mathcal{V}}$.

If there are similar node pairs in the current neighborhood, hence $\mathcal{N}(u_c, v_c) \neq \varnothing$, the random walker makes a simultaneous move on both graphs, from (u_c, v_c) to (u_i, v_j), according to the following transition probabilities:

$$P\big[(u_i, v_j) \,|\, (u_c, v_c)\big] = \frac{s\,(u_i, v_j)}{\sum_{(u_{i'}, v_{j'}) \in \mathcal{N}(u_c, v_c)} s\,(u_{i'}, v_{j'})}. \tag{2.1}$$

On the other hand, if there is no similar node pair in the neighborhood, hence $\mathcal{N}(u_c, v_c) = \varnothing$, the random walker randomly selects either $\mathcal{G}_{\mathcal{U}}$ or $\mathcal{G}_{\mathcal{V}}$ and performs an individual walk only on the selected graph. The probability that each graph will be selected is proportional to its size (i.e., number of nodes in the graph), and in the selected graph, the random walker will move into one of the neighboring nodes with equal probability. The resulting transition probabilities are given by

$$P\big[(u_i, v_c) \,|\, (u_c, v_c)\big] = \frac{|\mathcal{U}|}{|\mathcal{U}| + |\mathcal{V}|} \times \frac{1}{|\mathcal{N}\,(u_c)|} \tag{2.2a}$$

$$P\big[(u_c, v_j) \,|\, (u_c, v_c)\big] = \frac{|\mathcal{V}|}{|\mathcal{U}| + |\mathcal{V}|} \times \frac{1}{|\mathcal{N}\,(v_c)|} \tag{2.2b}$$

for $u_i \in \mathcal{N}(u_c)$ and $v_j \in \mathcal{N}(v_c)$. Note that $|\mathcal{U}|$ and $|\mathcal{V}|$ denote the number of nodes in the graph $\mathcal{G}_\mathcal{U}$ and $\mathcal{G}_\mathcal{V}$, respectively. From (2.1), (2.2a), and (2.2b), we can construct the transition probability matrix $\mathbf{P}$ for the random walk on the product graph $\mathcal{G}_\mathcal{X}$. In practice, the matrix $\mathbf{P}$ will be often sparse, as the original graphs $\mathcal{G}_\mathcal{U}$ and $\mathcal{G}_\mathcal{V}$ that arise in practical applications will be typically sparse. This property makes it easy to compute the steady state probability $\pi(u_i, v_j)$ of the random walk using the power method [12, 13, 21]. Given $\pi(u_i, v_j)$, we finally compute the actual proportion of time $\hat{\pi}(u_i, v_j)$ that the random walker spends at (u_i, v_j) by entering the node pair through a simultaneous random walk (i.e., at state M) as follows:

$$\hat{\pi}(u_i, v_j) = \sum\nolimits_{(u_p,u_q)\in\mathcal{N}(u_i,u_j)} \pi(u_p, u_q) \cdot P\big[(u_i, v_j) \,|\, (u_p, v_q)\big], \tag{2.3}$$

for all $(u_i, v_j) \in \mathcal{M}$. Finally, we define the correspondence score between two nodes u_i and v_j as $c(u_i, v_j) \equiv \hat{\pi}(u_i, v_j)$, where $u_i \in \mathcal{G}_\mathcal{U}$ and $v_j \in \mathcal{G}_\mathcal{V}$. As we will demonstrate in the following section, the proposed scoring scheme effectively quantifies the overall similarity between nodes in different graphs by seamlessly integrating the pairwise node similarity and the topological similarity between graphs.

2.1.3 Performance assessments

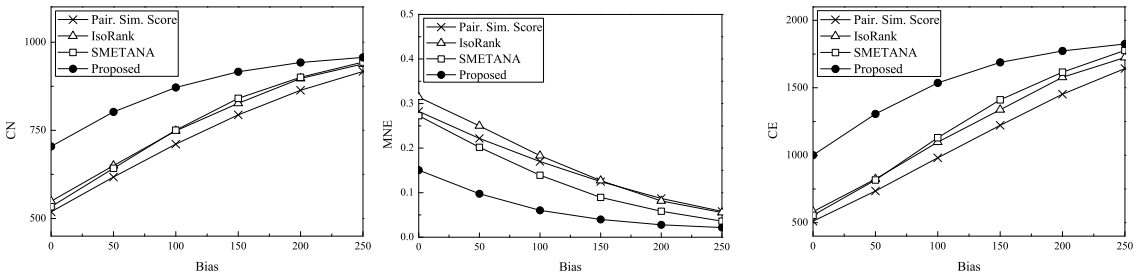
In order to demonstrate the effectiveness of the proposed scoring method, we performed extensive simulations based on synthetic graphs [22]. To evaluate the performance, we computed the node correspondence scores using the proposed scheme, and used the scores to predict the graph alignment through greedy one-to-one mapping. More specifically, we started from an empty alignment and built up the graph alignment by iteratively adding one node pair at a time according to its correspondence score in a descending order. Given the final alignment, we define the equivalence class as the set of nodes that are aligned to each other. A given equivalence class is said to be correct if the aligned nodes

	Pair. Sim. Score	IsoRank	SMETANA	CSRW
CN	519	549	533.4	704.4
MNE	0.28	0.31	0.27	0.15
CE	510.2	581.5	554.3	1,000.4

Table 2.1: Performance comparison of different scoring methods [16] © [2015] IEEE.

have the same label, indicating that they belong to the same functional class. We computed three different metrics to assess the goodness of the predicted alignment: correct nodes (CN), mean normalized entropy (MNE), and conserved edges (CE). CN is the total number of aligned nodes that belong to the correct equivalence class. The coherence of the node mapping can be accessed by MNE. MNE for a given equivalence class $\mathbf{C}$ can be computed by $H(\mathbf{C}) = -\frac{1}{\log d}\sum_{i=1}^{d} p_i \log p_i$, where p_i is the relative proportion of nodes in $\mathbf{C}$ with label i and d is the total number of different labels. A mapping with higher coherence will lead to a lower entropy. CE counts the total number of conserved edges between aligned nodes in the predicted graph alignment. CE can be used to assess the performance of detecting conserved topological structures across graphs. For comparison, we repeated similar experiments by using two state-of-the-art scoring schemes used in IsoRank [12] (parameter α was set to 0.6 as in the original paper) and SMETANA [13].

Using the NAPAbench package [22], we generated 10 pairs of synthetic graphs based on the crystal growth model [23], where each pair consists of a graph with 750 nodes and another graph with 1,000 nodes. On average, the smaller graphs had around 3,000 edges and the larger graphs had around 4,000 edges. For every pair of graphs, the true correspondence between the nodes in the two graphs are known, hence we can evaluate the effectiveness of the proposed scheme. Table 2.1 shows the performance of different scoring methods. The proposed method clearly outperforms all other methods. For example, the proposed scoring method finds around 30 percent more correct nodes compared to the



(a) CN (b) MNE (c) CE

Figure 2.2: Performance dependence on pairwise node similarity: correct nodes (left), mean normalized entropy (center), and conserved edges (right) [16] © [2015] IEEE.

scoring methods in IsoRank [12] and SMETANA [13]. Furthermore, the proposed method yields a more coherent mapping as indicated by the lower MNE. It is also important to note that our proposed method results in significantly higher CE, which implies that the resulting node correspondence scores capture the topological similarity between graphs more effectively.

Next, we evaluated the influence of the pairwise node similarity scores on the performance of each method. For this purpose, we introduced an additional bias term to further separate the distribution of the pairwise node similarity score between nodes with the same label and the score distribution for nodes with different labels. A higher bias makes it easier to predict the correspondence between nodes in different graphs based on the pairwise node similarity score alone (i.e., without taking topological similarity into account). Figure 2.2 shows that the proposed method significantly outperforms other scoring methods for a wide range of bias. As we would expect, the performance difference between the proposed method and the other methods decreases with an increasing bias, as it becomes easier to distinguish relevant nodes from irrelevant ones.

2.1.4 Conclusions

In this subchapter, we proposed a context-sensitive random walk model for scoring the correspondence between nodes that belong to two different graphs. The proposed method utilizes a novel random walk model that switches between two different modes of random walk – simultaneous walk on both graphs and individual walk on either graph – in a context dependent manner. The node correspondence scores are estimated based on the steady stationary probabilities of the random walk. Simulation results show that the proposed scoring method significantly outperforms previous methods that rely on different random walk models in terms of accuracy and robustness. Our scoring scheme can provide an effective and computationally efficient foundation for comparative analysis of graphs, including biological networks and social networks.

2.2 Network alignment through the context-sensitive random walk model

2.2.1 Background and motivation

With the availability of large-scale protein-protein interactions (PPI) networks, comparative network analysis tools have been gaining increasing interests as they provide useful means of investigating the similarities and differences between different networks. As demonstrated in [9, 24], PPI networks of different species embed various conserved functional modules – such as signaling pathways and protein complexes – which can be detected through network querying [11, 25, 26] and network alignment algorithms [12, 13, 27, 28, 29, 30, 31, 32, 33]. Comparative network analysis methods allow us to transfer existing knowledge on well-studied organism to less-studied ones and they have the potential to detect potential functional modules conserved across different organisms and species [9, 10, 24].

There exist several different types of comparative network analysis methods, among which global network alignment methods specifically aim to predict the best overall map-

ping among two or more biological networks. In order to obtain biologically meaningful results, where functionally similar biomolecules across networks are accurately mapped to each other, we should consider both the molecule-level similarity between the individual molecules as well as the similarity between their interaction patterns. The former is often called the "node similarity" while the latter is typically referred to as the "topological similarity." Examination of conserved functional modules shows that many of the molecular interactions in such modules are also well conserved, clearly showing the importance of taking the topological similarity into account when comparatively analyzing biological networks. Biological networks, such as PPI networks, are typically represented as graphs, where the nodes represent individual biomolecules (e.g., proteins) and interactions (e.g., protein binding) between biomolecules are represented by edges connecting the corresponding nodes. Given these graph representations of biological networks, the network alignment problem can be formulated as an optimization problem whose goal is to find the optimal mapping – either one-to-one or many-to-many – among a set of graphs that maximizes a scoring function that assesses the goodness of a given mapping. This is essentially a combinatorial optimization problem with a exponentially large search space, which makes finding the optimal mapping practically infeasible for large networks. As a result, existing network alignment methods employ various heuristic techniques to make the network alignment problem computationally tractable.

Several global network alignment algorithms have been proposed so far [12, 13, 27, 28, 29, 30, 31, 34, 32, 33], many of which focus on the pairwise network alignment [35]. For example, GRAAL [29] analyzes the graphlet degree signature for two PPI networks, where it can generalize the degree of node by counting the number of graphlets for each node, and then align the two networks using a seed-and-extend approach. MI-GRAAL [30] extends GRAAL by integrating further sources of information (e.g., clustering coefficient or functional similarity) to measure the similarity between two networks. PINALOG [31]

is another example of pairwise network alignment algorithm, which constructs the initial mapping for protein nodes that form dense subgraphs in the respective networks. This initial mapping is further extended by subsequently finding similar nodes in the neighborhood. HubAlign [34] first assigns weights to the nodes and edges in the PPI networks based on their topological importance (i.e., likelihood to be a hub), and then calculates the alignment score for every pair of proteins based on the global topological property and sequence information. Then, the algorithm constructs a global network alignment using a greedy seed-and-extension approach. Recently, a number of multiple network alignment algorithms have been proposed [13, 32, 33]. For example, SMETANA [13] tries to estimate probabilistic node correspondence scores using a semi-Markov random walk model, and then uses the estimated scores to predict the maximum expected accuracy (MEA) alignment of the given networks. Given a set of networks, NetCoffee [32] generates all possible combinations of bipartite graphs for these networks, and updates the edges in each bipartite graph based on the sequence similarity of the proteins and the topological structure of the networks. Then, the algorithm finds candidate edges (i.e., mappings) in the bipartite graphs and combines qualified edges through simulated annealing. BEAMS [33] is another recent multiple network alignment algorithm, which first extracts the so-called "backbones", or the minimal set of disjoint cliques in the filtered similarity graph, and then iteratively merges these backbones to maximize the overall alignment score.

In this subchapter, we propose a novel multiple network alignment algorithm based on a context-sensitive random walk (CSRW) model. The employed CSRW model adaptively switches between different modes of random walk in a context-sensitive manner by sensing and analyzing the present neighborhood of the random walker. This context-sensitive behavior improves the quantitative estimation of the potential correspondence between nodes belonging to different networks, ultimately, improving the overall accuracy of the multiple network alignment as we will demonstrate through extensive performance evalu-

ation based on real and synthetic biological networks.

2.2.2 Methods

Let us assume that we have a set of N PPI networks $\mathbf{G} = \{\mathcal{G}_1, \mathcal{G}_2, \ldots, \mathcal{G}_N\}$. Each network $\mathcal{G}_n = (\mathcal{V}_n, \mathcal{E}_n)$ has a set of nodes $\mathcal{V}_n = \{v_1, v_2, \ldots\}$ and edges $\mathcal{E}_n = \{e_{i,j}\}$, where $e_{i,j}$ represents the interaction between nodes v_i and v_j in the network $\mathcal{G}_n$. For each pair of PPI networks $\mathcal{G}_\mathcal{U} = (\mathcal{U}, \mathcal{D})$ and $\mathcal{G}_\mathcal{V} = (\mathcal{V}, \mathcal{E})$, we denote the pairwise node similarity score for a node pair (u_i, v_j), where $u_i \in \mathcal{U}$ and $v_j \in \mathcal{V}$, as $s(u_i, v_j)$. In this study, we use the BLAST bit score between proteins as their node similarity score, but other types of similarity scores based on structural or functional similarity can be also utilized if available.

Suppose $\mathcal{A}^*$ is the true alignment of the networks in the set $\mathbf{G}$, which is unknown and needs to be predicted. As in [13, 36], we can define the accuracy of a given network alignment $\mathcal{A}$ as follows:

$$accuracy(\mathcal{A}, \mathcal{A}^*) = \frac{1}{|\mathcal{A}|} \sum_{u_i \sim v_j \in \mathcal{A}} \mathbf{1}(u_i \sim v_j \in \mathcal{A}^*), \tag{2.4}$$

where $\mathbf{1}(\cdot)$ is an indicator function, whose value is 1 if the mapping $u_i \sim v_j$ is included in the true alignment $\mathcal{A}^*$ and 0 otherwise. The given measure assesses the goodness of the alignment $\mathcal{A}$ based on the relative proportion of correctly aligned nodes. Of course, since the true alignment is not known, the accuracy of a network alignment $\mathcal{A}$ cannot be measured using (2.4), hence we cannot directly use this measure to compare different potential alignments to choose the best one. A reasonable alternative would be to estimate the expected accuracy as follows:

$$\mathbf{E}_{\mathcal{A}^*}[accuracy(\mathcal{A}, \mathcal{A}^*)] = \frac{1}{|\mathcal{A}|} \sum_{u_i \sim v_j \in \mathcal{A}} P(u_i \sim v_j | \mathbf{G}), \tag{2.5}$$

where $P(u_i \sim v_j|\mathbf{G})$ is the posterior alignment probability between the nodes u_i and v_j given the set of networks $\mathbf{G}$. Based on this measure, our objective is then to predict the maximum expected accuracy (MEA) network alignment $\tilde{\mathcal{A}}^*$ of the networks in $\mathbf{G}$ as follows:

$$\tilde{\mathcal{A}}^* = \max_{\mathcal{A}} \mathbf{E}_{\mathcal{A}^*} \left[accuracy \left(\mathcal{A}^*, \mathcal{A} \right) \right]. \tag{2.6}$$

A similar MEA approach [37] has been formerly adopted by a number of multiple sequence alignment algorithms, including ProbCons [36], ProbAlign [38], and PicXAA [39, 40, 41]. The MEA framework has been shown to be very effective in constructing accurate alignment of multiple biological sequences, making it one of the most popular approaches for a sequence alignment. Recently, the MEA approach has been also applied to comparative network analysis, where RESQUE [11] performs MEA-based network querying and SMETANA [13] performs MEA-based multiple network alignment.

In order to find the alignment that maximizes the expected accuracy defined in (2.5), we first need an accurate method for estimating the posterior node alignment probability $P(u_i \sim v_j|\mathbf{G})$. For this purpose, we adopt a proposed context-sensitive random walk model [16].

Suppose we want to measure the correspondence between nodes that belong to two different networks $\mathcal{G}_\mathcal{U} = (\mathcal{U}, \mathcal{D})$ and $\mathcal{G}_\mathcal{V} = (\mathcal{V}, \mathcal{E})$, both of which are included in $\mathbf{G}$, the set of PPI networks to be aligned. For every node pair (u_i, v_j), where $u_i \in \mathcal{U}$ and $v_j \in \mathcal{V}$, our goal is to quantify the level of confidence – which we refer to as the *node correspondence score* – using the CSRW model discussed earlier. For this purpose, based on the transition probabilities given by (2.1), (2.2a), and (2.2b), we can construct the transition probability matrix $\mathbf{P}$ that corresponds to the context-sensitive random walk for a simultaneous walk and individual walk on the two networks $\mathcal{G}_\mathcal{U}$ and $\mathcal{G}_\mathcal{V}$. Given $\mathbf{P}$, we can estimate the long-run proportion of time that the random walker spends in each pair of

nodes (u_i, v_j) by computing the steady state probability π. In practice, since real PPI networks typically have a relatively small number of interactions (therefore only few edges for most nodes), the resulting transition probability matrix for the CSRW is sparse, which makes it relatively straightforward to compute the steady state distribution using the power method [12, 13, 21].

In order to increase the computational efficiency of the proposed network alignment method, instead of using the original transition probability matrix $\mathbf{P}$, we use a reduced matrix $\widetilde{\mathbf{P}}$. The reduced matrix $\widetilde{\mathbf{P}}$ is obtained by removing the rows and columns in $\mathbf{P}$ that correspond to node pairs in $\mathcal{I}$ while keeping only the rows and columns that correspond to node pairs in $\mathcal{M}$. After the reduction, $\widetilde{\mathbf{P}}$ is re-normalized to make it a legitimate stochastic matrix. In practice, since the CSRW is designed to spend more time at node pairs with higher similarity, the random walker spends a relatively small amount of time at node-pairs that belong to the set $\mathcal{I}$, and using the reduced matrix $\widetilde{\mathbf{P}}$ instead of $\mathbf{P}$ only minimally affects the estimated long-run proportion of time spent at $(u_i, v_j) \in \mathcal{M}$.

We make one further modification to the CSRW in [16] by allowing the random walker to restart at a new position at each time step with a fixed restart probability λ. Note that a similar "random walk with restart" approach was used by IsoRank [12] and IsoRankN [27], although these algorithms do not utilize the CSRW adopted in our method. We allow the random walker to select its restart position according to the pairwise node similarity, such that node pairs with higher node similarity have higher chance to be the restart position of the random walker. To this aim, we normalize the pairwise node similarity scores so that they sum up to 1. Our final node correspondence score vector $\mathbf{c}$ is obtained from a linear combination of the steady-state distribution of the context-sensitive random walker $\tilde{\pi}$ (estimated using the reduced transition probability matrix $\widetilde{\mathbf{P}}$) and the

normalized node similarity score vector $\mathbf{s}$ as follows:

$$\mathbf{c} = \lambda \mathbf{s} + (1 - \lambda)\, \tilde{\pi}. \tag{2.7}$$

The above formulation, obtained by allowing the CSRW to restart the random walk at a new position, is especially useful when comparing real PPI networks, which are often incomplete and contain many isolated nodes. Simulation results show that the incorporation of the restart scheme can make our CSRW-based alignment method more robust, especially when the available topological data are either unreliable or insufficient for detecting the similarities between networks.

In order to determine the restart probability λ, we first analyze the structure of the reduced product graph of $\mathcal{G}_{\mathcal{U}}$ and $\mathcal{G}_{\mathcal{V}}$ that contains only similar node pairs included in $\mathcal{M}$. Intuitively, it is desirable to increase the restart probability λ if the networks are disconnected and decrease the probability if the networks are well connected. For example, if all the nodes in the reduced product graph are completely disconnected, it is desirable to restart the random walker at every step. Additionally, when we consider the following two cases – (i) most nodes in the product graph are connected and there are only a few disconnected nodes; (ii) the product graph is equally divided into N connected subnetworks of identical size – it would be desirable to assign a higher λ to the latter case. Based on these intuitions, we set the restart probability λ as the ratio of the total number of nodes in the top $K\%$ smallest subnetworks to the total number of nodes in the reduced product graph. In this work, we used $K = 99\%$ to determine the restart probability λ.

Once we have computed the node correspondence scores in (2.7) for every pair of networks in $\mathbf{G}$, we take a greedy approach as in [13] to construct the multiple network alignment. The overall alignment process is as follows. First, in order to improve the reliability of the node correspondence scores, we selectively apply the probabilistic con-

sistent transformation (PCT) defined in [13]. If λ is larger than a predefined threshold λ_t, we do not apply PCT to the node correspondence scores. A large λ implies that the product graph is ill connected (e.g., containing a large number of isolated nodes), in which case applying the PCT would not be helpful and may in fact make the scores less reliable. This is because the PCT in [13] was developed based on the assumption that the product graphs for all network pairs are relatively well connected. After the potential score refinement step through PCT, we begin with an empty alignment and greedily add aligned node pairs (u_i, v_j) to the network alignment, starting from the pairs with the highest node correspondence scores, until there is no other node pair left that can be added without creating inconsistencies in the network alignment. Assuming that the node correspondence scores in (2.7) obtained by the context-sensitive random walk model with restart accurately reflect the true correspondence between nodes – such that the score is proportional to the posterior node alignment probability – the proposed network alignment scheme can be viewed as a heuristic way to find the MEA alignment of the networks in $\mathbf{G}$.

2.2.3 Results

To assess the performance of the proposed method, we tested the proposed network alignment method based on PPI networks in NAPAbench [22] and IsoBase [42]. NAPAbench is a network alignment benchmark that consists of 3 different datasets, referred to as the pairwise alignment dataset, 5-way alignment dataset, and 8-way alignment dataset. Each dataset contains three different subsets of 10 network families, each subset created using a different network growth model – CG (crystal growth), DMC (duplication-mutation-complementation), and DMR (duplication with random mutation). Each network family consists of 2, 5, or 8 PPI networks depending on the alignment dataset. For network families in the pairwise alignment dataset, each family contains one network with 3,000 nodes and the other with 4,000 nodes. In the 5-way network alignment dataset, a

network family consists of 5 networks with 1,000, 1,500, 2,000, 2,500, and 2,500 nodes. Finally, in the 8-way alignment dataset, every network family consists of 8 networks, where each network contains 1,000 nodes. To evaluate the performance of the proposed method on real PPI networks, we utilized IsoBase datasets [42], which was constructed by integrating the following databases: BioGRID [43], DIP [44], HPRD [45], MINT [46], and IntAct [47]. IsoBase contains the PPI networks of five species: *H. sapiens*, *M. musculus*, *D. melanogaster*, *C. elegans*, and *S. cerevisiae*. Currently, the PPI network of *H. sapiens* in [42] has 22,369 proteins and 43,757 interactions, the PPI network of *M. musculus* has 24,855 proteins and 452 interactions, the PPI network of *D. melanogaster* has 14,098 proteins and 26,726 interactions, the PPI network of *C. elegans* has 19,756 proteins and 5,853 interactions, and the PPI network of *S. cerevisiae* has 6,659 proteins and 38,109 interactions. In our analysis, we excluded the *M. musculus* network as it currently contains only a small number of interactions.

Based on our simulations, we report the following performance metrics: correct nodes (CN), specificity (SPE), mean normalized entropy (MNE), conserved interaction (CI), coverage and computation time. CN is the total number of nodes in the correct equivalence classes. Given a network alignment, an equivalence class is defined as the set of aligned nodes, and if all nodes in the equivalence class have the same functionality the given equivalence class is said to be correct. SPE is the relative number of correct equivalence classes to the total number of equivalence classes in a network alignment. For each equivalence class $\mathbf{C}$, the normalized entropy can be computed by $H(\mathbf{C}) = -\frac{1}{\log d}\sum_{i=1}^{d} p_i \log p_i$, where p_i is the relative proportion of nodes in $\mathbf{C}$ with functionality i and d is the total number of different functionalities in the given equivalence class. As a result, a network alignment that accurately maps functionally similar nodes, hence being functionally consistent, will have lower mean normalized entropy. CI is defined as the total number of edges between equivalence classes. We also count the total number of edges between correct equivalence

classes, which we refer to as the conserved orthologous interactions (COI), to assess the biological relevance of the conserved interactions that have been identified by the network alignment method. Finally, for 5-way and 8-way alignment datasets, we measure the equivalence class coverage and the node coverage, where the former is the number of equivalence classes that include nodes from k different networks, and the latter is the number of nodes in an equivalence class whose equivalence class coverage is k. For the performance evaluation based on real PPI networks in IsoBase, we determined the functionality of each protein using the KEGG protein annotation [48, 49]. Note that nodes without any functional annotation in each equivalence class and equivalence classes that consist of a single node or nodes from a single network were removed before computing the performance metrics.

We compared the performance of the proposed multiple network alignment method against a number of state-of-the-art algorithms: SMETANA [13], IsoRankN [27], PINALOG [31], NetCoffee [32], and BEAMS[33]. NetCoffee was not included in pairwise network alignment experiments, since it requires at least 3 networks. For multiple network alignment experiments, PINALOG was excluded as the algorithm can only handle pairwise alignments. For IsoRankN, we set the parameter α to 0.6 as in the original paper [27]. For BEAMS, we set the filtering threshold to 0.4 for IsoBase and 0.2 for NAPAbench as in the original paper [33], and set the parameter α to 0.5. The parameter α for NetCoffee was set to 0.5. We used default parameters for SMETANA (i.e., $n_{\max} = 10$, $\alpha = 0.9$, and $\beta = 0.8$), and the same parameters were used in the proposed network alignment method as well. Finally, in the proposed method, we used $\lambda_t = 0.7$ to determine whether or not to apply PCT to the estimated node correspondence scores.

All experiments were performed on a personal computer with a 2.4GHz Intel i7 processor and 8GB memory.

We first evaluated the performance of the proposed algorithm using the NAPAbench

	DMC			DMR			CG		
	CN	SPE	MNE	CN	SPE	MNE	CN	SPE	MNE
Proposed	5,593.9	0.958	0.039	5,305.3	0.939	0.055	4,893.2	0.942	0.054
SMETANA	5,164.5	0.926	0.068	4,900.6	0.916	0.078	4,846.2	0.949	0.048
BEAMS	5,076.5	0.826	0.150	5,176.7	0.840	0.138	5,441.2	0.870	0.112
PINALOG	3,779	0.726	0.274	3,533.4	0.683	0.317	4,325	0.788	0.212
IsoRankN	3,816.5	0.827	0.163	3,905.2	0.836	0.155	3,863.2	0.832	0.159

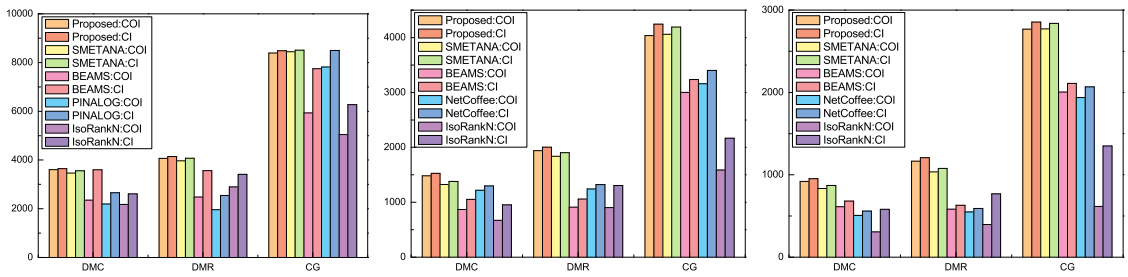
Table 2.2: Performance comparison for pairwise network alignment [17] © [2015] BMC.

	DMC			DMR			CG		
	CN	SPE	MNE	CN	SPE	MNE	CN	SPE	MNE
Proposed	7,536.7	0.940	0.047	7,410.3	0.934	0.053	7,177.6	0.919	0.060
SMETANA	7,273.2	0.912	0.069	7,181.8	0.915	0.068	7,331.6	0.935	0.048
BEAMS	6,842.2	0.863	0.104	6,882	0.873	0.096	7,376.5	0.921	0.062
NetCoffee	6,431.2	0.894	0.090	6,395.7	0.890	0.093	6,150.2	0.854	0.120
IsoRankN	5,559	0.920	0.147	5,462.3	0.793	0.162	5,688.4	0.828	0.132
Proposed (all 5 species)	4,476.9	0.931	0.048	4,017.9	0.916	0.060	3,644.8	0.900	0.068
SMETANA (all 5 species)	4,062.3	0.891	0.077	3,704.9	0.889	0.080	3,778.9	0.922	0.052
BEAMS (all 5 species)	2,858.4	0.814	0.121	3,095.2	0.838	0.104	3,510.3	0.918	0.052
NetCoffee (all 5 species)	2,960.4	0.867	0.106	2,973.3	0.855	0.113	2,841.2	0.796	0.156
IsoRankN (all 5 species)	1,668.1	0.728	0.179	1,595.4	0.677	0.215	2,233.5	0.742	0.168

Table 2.3: Performance comparison for 5-way network alignment [17] © [2015] BMC.

	DMC			DMR			CG		
	CN	SPE	MNE	CN	SPE	MNE	CN	SPE	MNE
Proposed	6,621.3	0.901	0.080	6,467.2	0.891	0.090	6,345.4	0.884	0.090
SMETANA	6,336.7	0.869	0.106	6,195.2	0.860	0.114	6,481.2	0.897	0.079
BEAMS	6,083.1	0.825	0.163	6,063.5	0.826	0.162	6,537.6	0.877	0.111
NetCoffee	5,127.2	0.757	0.206	5,084.1	0.750	0.213	4,944.1	0.724	0.239
IsoRankN	4,069.1	0.644	0.268	3,916.7	0.623	0.284	3,860	0.612	0.291
Proposed (all 8 species)	4,116	0.961	0.034	3,473.7	0.930	0.059	3,689.5	0.945	0.043
SMETANA (all 8 species)	3,686.7	0.920	0.066	3,348.9	0.907	0.075	3,785.6	0.960	0.031
BEAMS (all 8 species)	2,897.9	0.905	0.095	3,054.7	0.901	0.099	3,475.1	0.989	0.011
NetCoffee (all 8 species)	3,300.8	0.837	0.136	3,331.8	0.822	0.148	3,317.8	0.800	0.172
IsoRankN (all 8 species)	2,002.8	0.569	0.284	1,775.8	0.542	0.303	2,161.6	0.536	0.303

Table 2.4: Performance comparison for 8-way network alignment [17] © [2015] BMC.



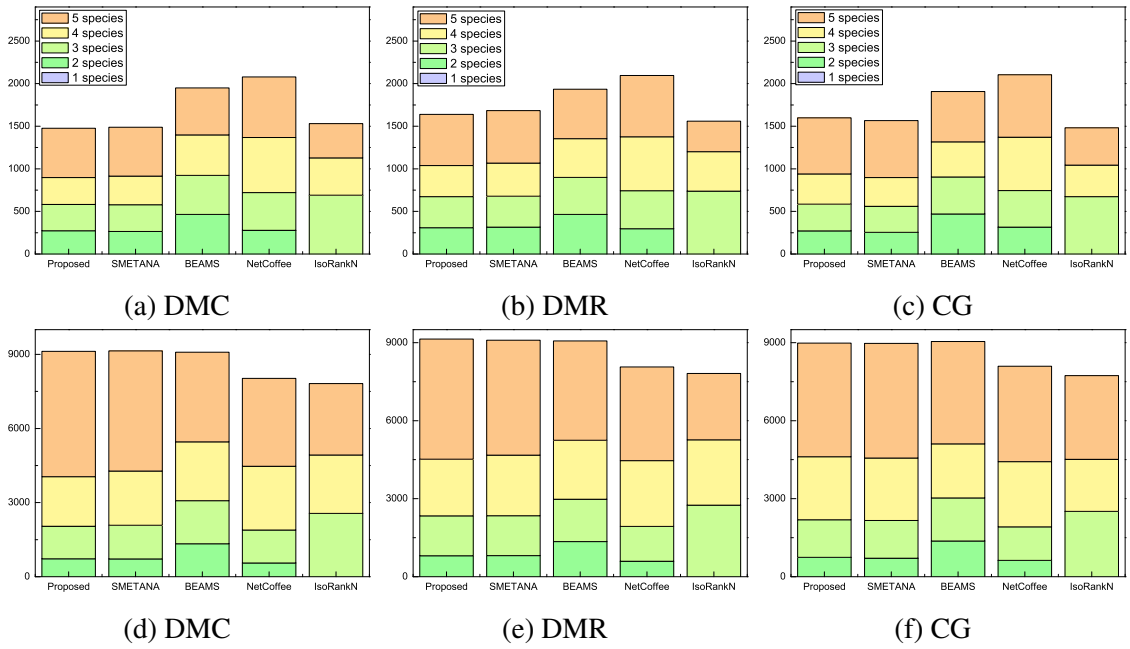
(a) Pairwise alignment. (b) 5-way alignment. (c) 8-way alignment.

Figure 2.3: The total number of conserved orthologous interactions (COI) and conserved interactions (CI) [17] © [2015] BMC.

network alignment benchmark and compared it to other leading algorithms. The evaluation results are summarized in Table 2.2, 2.3, and 2.4, which show the average CN, SPE, and MNE of various network alignment algorithms.

As we can see in Table 2.2, in most cases, the proposed algorithm yields a significantly higher CN and SPE compared to other algorithms, which shows that the algorithm is capable of finding conserved nodes with both high sensitivity and specificity. Furthermore, the mean normalized entropy (MNE) is also much lower, indicating that the proposed algorithm yields network alignment results that are more functionally coherent. This table shows that BEAMS yields higher CN for the CG dataset, although its SPE is lower and its MNE is higher than the proposed method. Both SMETANA and the proposed algorithm shows similar performance on the CG dataset, but we can also see that the proposed algorithm consistently outperforms SMETANA on the DMC/DMR datasets.

Multiple network alignment results obtained using the 5-way alignment dataset and the 8-way alignment dataset show similar trends. Tables 2.3 and 2.4 show that, in most cases, our proposed algorithm outperforms other algorithms with higher CN, higher SPE, and lower MNE. For multiple networks alignment, we further compared different network alignment algorithms based on their capability of predicting equivalence classes that



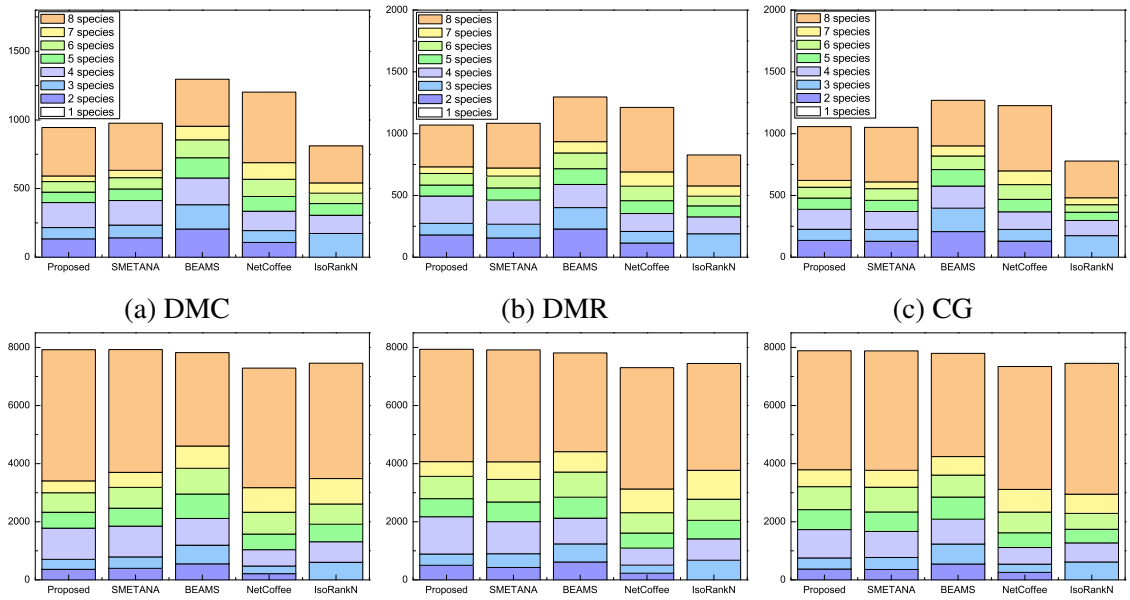
(a) DMC (b) DMR (c) CG

(d) DMC (e) DMR (f) CG

Figure 2.4: Equivalence class coverage for 5-way network alignment: (a) DMC; (b) DMR; (c) CG, and node coverage for 5-way network alignment: (d) DMC; (e) DMR; (f) CG [17] © [2015] BMC.

span all networks, since one of the main goals of multiple network alignment is to find functionally homologous proteins that are conserved in the networks of all target species. Simulation results show that, in most cases, our proposed method also yields much higher CN and SPE as well as lower MNE for equivalence classes that span all networks.

Next, we compared the number of conserved (orthologous) interactions identified by different network alignment algorithms. As Figure 2.3 shows, the proposed method was able to identify the largest number of conserved interactions as well as conserved orthologous interactions in most cases, resulting in higher CI and COI. The performance of SMETANA was comparable to the proposed method, while other algorithms typically resulted in lower CI and COI. It is worth noting that more than 95% of the conserved interactions that were detected by our proposed network alignment algorithm were between correct equivalence classes (i.e., conserved orthologous interactions). This certainly



(a) DMC (b) DMR (c) CG

(d) DMC (e) DMR (f) CG

Figure 2.5: Equivalence class coverage for 8-way network alignment: (a) DMC; (b) DMR; (c) CG, and node coverage for 8-way network alignment: (d) DMC; (e) DMR; (f) CG [17] © [2015] BMC.

shows that our method can effectively detect biologically meaningful conserved interactions through network alignment.

We also analyzed the overall coverage of the predicted alignment results for the 5-way and 8-way network alignments. The results are shown in Figure 2.4 for the 5-way network alignment and in Figure 2.5 for the 8-way network alignment. For the 5-way network alignment, we can see that around 40% of the equivalence classes predicted by the proposed method contained nodes from all 5 networks. SMETANA shows a similar level of coverage, while for the remaining algorithms, only about 30% of the predicted equivalence classes included nodes from all 5 networks. The overall node coverage also shows similar trends. The 8-way alignment results summarized in Figure 2.5 show that the proposed algorithm can effectively find equivalence classes with good coverage, which include nodes from a large number of networks. For example, we can see that around 40%

Algorithms	Pairwise	5-way	8-way	Average
Proposed	117.8	273.1	178.7	189.8
SMETANA	6.9	58.0	70.7	45.2
BEAMS	42.4	134.8	333.8	170.3
PINALOG	77.1	·	·	77.1
NetCoffee	·	132.7	225.7	179.2
IsoRankN	1083.7	3326.1	2694.8	2368.2

Table 2.5: Mean computation time for aligning PPI networks in the NAPAbench datasets (in seconds) [17] © [2015] BMC.

of the equivalence classes predicted by the proposed method contained nodes from all 8 networks.

Table 2.5 shows the mean computation time of the respective algorithms for aligning the network families in the NAPAbench datasets. As we can see in Table 2.5, SMETANA requires the least amount of time for aligning the networks in NAPAbench, while IsoRankN needs the most computation time. In our simulations, we observed that NetCoffee runs relatively fast, although its computation time varies significantly depending on the network structure. For example, it took much longer to align networks in the DMR dataset using NetCoffe, compared to networks in the DMC or CG datasets.

For further evaluation, we performed additional experiments using real PPI networks in IsoBase. Table 2.6 shows the pairwise network alignment performance of the tested algorithms for several PPI network pairs. As we can see in this table, the proposed algorithm consistently performs fairly well in all cases, outperforming the other algorithms. We can make similar observations in Table 2.7, which summarizes the performance evaluation results for aligning 3 PPI networks. The proposed algorithm attains high CN, high SPE, and low MNE across all cases, showing that it can effectively compare and accurately align real PPI networks. BEAMS shows good performance on multiple alignment of real networks that is comparable to the proposed method, with a slightly lower SPE

	H.sa-S.ce			D.me-S.ce			C.el-S.ce		
	CN	SPE	MNE	CN	SPE	MNE	CN	SPE	MNE
Proposed	1307	0.689	0.310	1725	0.727	0.277	1543	0.796	0.196
SMETANA	1190	0.671	0.331	1579	0.709	0.295	1443	0.771	0.222
BEAMS	1306	0.649	0.347	1636	0.675	0.320	1499	0.742	0.247
PINALOG	1100	0.682	0.324	1368	0.722	0.289	640	0.737	0.266
IsoRankN	1367	0.765	0.238	1641	0.777	0.230	1458	0.843	0.155
Node Similarity	1486	0.740	0.259	1832	0.779	0.224	1670	0.831	0.163
	D.me-H.sa			D.me-C.el			C.el-H.sa		
	CN	SPE	MNE	CN	SPE	MNE	CN	SPE	MNE
Proposed	2681	0.724	0.279	2714	0.855	0.146	1995	0.771	0.224
SMETANA	2274	0.671	0.331	2458	0.827	0.175	1684	0.737	0.255
BEAMS	2612	0.658	0.338	2738	0.808	0.192	1941	0.691	0.300
PINALOG	1172	0.604	0.412	672	0.689	0.317	482	0.677	0.325
IsoRankN	2635	0.759	0.246	2488	0.851	0.150	1881	0.783	0.216
Node Similarity	2932	0.750	0.251	2897	0.875	0.125	2185	0.770	0.227

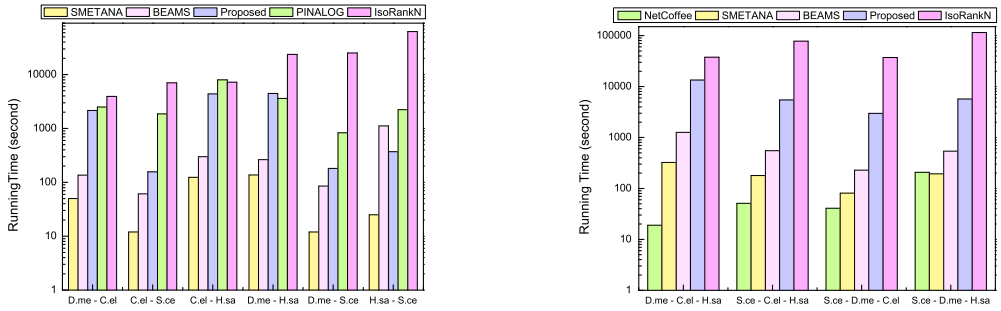
Table 2.6: Pairwise network alignment results for real PPI networks [17] © [2015] BMC.

	D.me-C.el-H.sa			S.ce-C.el-H.sa			S.ce-D.me-C.el			S.ce-D.me-H.sa		
	CN	SPE	MNE	CN	SPE	MNE	CN	SPE	MNE	CN	SPE	MNE
Proposed	4,331	0.705	0.289	3,077	0.709	0.281	3,581	0.746	0.247	3,637	0.672	0.326
SMETANA	3,871	0.663	0.331	2,625	0.657	0.333	3,227	0.714	0.279	3,108	0.616	0.380
BEAMS	4,354	0.676	0.316	3,084	0.671	0.320	3,606	0.727	0.267	3,629	0.627	0.366
NetCoffee	1,471	0.552	0.451	1,234	0.575	0.426	1,477	0.593	0.414	1,877	0.540	0.465
IsoRankN	4,423	0.717	0.279	3,131	0.711	0.282	3,464	0.749	0.245	3,752	0.684	0.313
NodeSimilarity	4,775	0.746	0.248	3,457	0.737	0.256	3,920	0.798	0.197	4,132	0.719	0.278
Proposed (all 3-species)	3,926	0.702	0.290	2,387	0.724	0.265	2,624	0.715	0.271	2,540	0.681	0.315
SMETANA (all 3-species)	3,442	0.671	0.323	2,106	0.677	0.312	2,378	0.685	0.301	2,225	0.630	0.363
BEAMS (all 3-species)	3,867	0.687	0.304	2,277	0.711	0.278	2,573	0.718	0.272	2,441	0.672	0.318
NetCoffee (all 3-species)	747	0.518	0.478	578	0.528	0.465	713	0.538	0.462	1,167	0.516	0.489
IsoRankN (all 3-species)	3,757	0.753	0.241	2,323	0.775	0.215	2,470	0.732	0.258	2,510	0.726	0.267

Table 2.7: Multiple network alignment results for real PPI networks (for 3 species) [17] © [2015] BMC.

and a slightly higher MNE. Additionally, although BEAMS and IsoRankN achieve higher CN in some cases, the proposed method consistently yields higher CN than these methods

with comparable SPE and MNE when we consider multiple network alignment results for regions that are conserved across all networks. Another observation we can make in Table 2.6 is that IsoRankN performs very well on real PPI networks compared to the other more recent algorithms. This is especially interesting, if we consider the fact that the performance of IsoRankN lagged behind the other algorithms according to the large-scale evaluations using NAPAbench. One possible explanation is that, for constructing the network alignment, IsoRankN relies on node similarity (i.e., sequence similarity in this case) more strongly compared to the other algorithms. In order to find out whether this is indeed a plausible explanation, we performed network alignment experiments solely using node similarity scores (i.e., without considering network topology), where we constructed the network alignment in a greedy manner by iteratively adding protein pairs with the highest node similarity scores. The alignment results are shown in Tables 2.6 and 2.7 right below the results for IsoRankN (labeled as "Node Similarity"). Surprisingly, these results show that this simple greedy network alignment approach that uses node similarity alone outperforms IsoRankN in most cases and surpasses all the other algorithms in all cases. In fact, currently available PPI networks are known to be very incomplete and these network typically contain a large number of isolated nodes. They are suspected to include a large number of spurious interactions while still missing many potential protein-protein interactions [50, 51]. Furthermore, only a small proportion of proteins in these PPI networks have reliable functional annotations (e.g., according to KEGG orthology), making it difficult to reliably assess the quality of a predicted network alignment. As a result, for current PPI networks, utilization of topological similarity between networks may not be necessarily helpful for improving the overall quality of the network alignment across the entire network. Moreover, since only a few large real PPI networks are available at the moment, we risk overtraining network alignment algorithms if they are mainly evaluated solely based on real PPI networks.



(a) Pairwise network alignment. (b) Multiple network alignment (for 3 species).

Figure 2.6: Computation time for aligning real PPI networks (in seconds) [17] © [2015] BMC.

Figure 2.6 shows the computation time for aligning the PPI networks in IsoBase. SMETANA required the least computation time for pairwise network alignment and NetCoffee was the fastest among all for aligning the PPI networks of 3 species. Although IsoRankN yielded accurate alignment results for real PPI networks in IsoBase, it also required the largest amount of computation time in most cases. Figure 2.6 shows that our proposed network alignment algorithm requires relatively longer running time compared to other algorithms, in exchange for the improved alignment accuracy.

2.2.4 Conclusions

In this subchapter, we proposed a novel network alignment algorithm based on a context-sensitive random walk model. The CSRW model provides an effective mathematical framework for comparing different biological networks and quantifying the node-to-node correspondence between nodes that belong to different networks. In our proposed method, we combined the CSRW model with a restart scheme, where the restart probability is automatically adjusted based on the characteristics of the networks under comparison. Furthermore, the proposed network alignment algorithm employs adaptive probabilistic consistency transformation, where the PCT is adaptively activated or deactivated

based on the overall structure of the given networks. As we have shown through extensive performance evaluations based on biologically realistic PPI networks in NAPAbench as well as real PPI networks in IsoBase, the novel network alignment algorithm proposed in this subchapter can significantly improve the overall accuracy of pairwise as well as multiple network alignment.

2.3 Network querying through the context-sensitive random walk model

2.3.1 Background and motivtion

Protein-protein interaction (PPI) plays pivotal roles in understanding biological systems. Diverse functional modules in cells, such as signaling pathways and protein complexes, involve numerous proteins and their functions are governed by the intertwined interactions among these proteins. For this reason, to better understand the functions and roles of proteins in cells, it is critically important to investigate how groups of proteins collaborate with each other to perform certain biological functions and achieve common goals, in addition to studying the functions of individual proteins. Recent advances in technologies for high throughput measurement of protein-protein interactions have enabled genome-scale studies of protein interactions, and systematic analyses of the available PPI networks may reveal new functional network modules and unveil novel functionalities of the proteins that are involved in such modules. Recent investigations of PPI networks show that functionally important network modules (e.g., molecular complexes and pathways) are often well conserved across networks of different species [9, 24]. These observations clearly point to comparative network analysis [10] as a promising solution for effectively analyzing large-scale PPI networks, detecting common functional modules that are embedded in the networks, and predicting the functions of proteins that comprise these modules.

Network querying is one possible way of comparatively analyzing biological networks,

which can be especially useful when prior knowledge of functional modules is available for a given species. As implied in its name, network querying aims to find out whether a target network (typically, belonging to another species) contains network modules that resemble the module that is being used as the query [10]. This provides an efficient way of transferring knowledge between species, since we could use computational means to predict potential network modules in a new (or less-studied) species that may have similar functions, structures, and underlying mechanisms to well-studied modules in other species.

Several network querying algorithms have been proposed [11, 25, 52, 53, 54, 55, 56, 57, 58]. PathBLAST [52] is one of pioneering network querying algorithms, but it can search only linear pathways and the computational complexity limits the size of the query network. QPath [53] can search much longer pathways than PathBLAST and QNet [25] can search both linear pathways and tree structure, but both algorithms still requires high computational complexity and searching capability is limited to either a pathway or a tree. PathMatch [54] solves a network querying problem by finding the longest weighted path in a directed acyclic graph (target network) and GraphMatch [54] finds highest scoring subgraphs in a target network using an exact algorithm. SAGA [55] solves an approximated graph matching based on the fragment index, where it is the index on a small substructure of graphs in a database, and SAGA employs a flexible model for node gaps/mismatches and network structural variations. NatalieQ [56] identifies the querying results by solving the integer linear programming through Lagrangian relaxation combined with a branch-and-bound approach. TORQUE [57] proposed a topology-free network querying algorithm. That is, it only requires a set of proteins in the query network and it does not necessary to provide the topological structure of the query network. TORQUE finds a connected set of matching proteins through a dynamic and integer linear programming based on a sequence similarity of proteins. RESQUE [11] estimates the node-to-node cor-

respondence through a semi-Markov random walk (SMRW) model[14]. Then, RESQUE iteratively removes less relevant nodes in the target network and identifies the best matching subnetwork through either a Hungarian method or identifying the largely connected subnetwork. Corbi [58] estimates a matching probability of nodes in the query and target network through a conditional random field (CRF), and identifies the matching subnetwork through iterative bi-directional mapping.

Most of the aforementioned network querying methods consider both *node similarity* and *topological similarity* between the query and the target networks to detect matching subnetworks in the target network. Node similarity between nodes that belong to different networks is typically measured based on sequence similarity. Topological similarity between (sub)networks are measured in various ways to capture the molecular interaction patterns that are conserved across networks. Incorporating both types of similarities has been shown to be crucial in making biologically relevant predictions about conserved functional modules [9, 10, 24, 59]. However, one important aspect of network module detection that is often neglected in network querying is that such modules are often well separated from the rest of the network. In fact, this separability has played critical roles in "non-comparative" network analysis methods that aim to detect modules or sub-communities in a given network [60, 61, 62], since molecules in a functional module tend to be densely connected to other molecules in the same module but loosely connected to nodes that are not part of the module. Although identifying densely connected subnetwork modules is not the main objective of network querying, explicitly incorporating separability criterion into comparative network analysis methods has strong potentials to enhance the quality of the predictions [63].

In this subchapter, we propose a novel network querying algorithm called **SEQUOIA** (Significance Enhanced QUerying Of InterAction networks). The proposed algorithm is built on the following important concepts: (i) effective estimation of *node correspondence*

– or overall functional similarity between nodes in different networks – by sensibly combining sequence similarity and interaction pattern similarity through a random walk model; and (ii) minimization of network conductance of potential network modules, thereby identifying matching modules in the target network that are well separated from the rest of the network. In our proposed algorithm, we first estimate the node correspondence based on a context-sensitive random walk model [16, 17], and select a seed network based on the estimated node correspondence scores. Then, the seed network is iteratively extended by adding the nodes that maximally reduce the conductance of the subnetwork. Finally, the significance enhanced querying result is achieved by keeping the nodes with acceptable extension reward scores, which are updated for every node at each extension step. Through extensive evaluations based on real biological complexes, we show that SEQUOIA can remarkably enhance the biological significance of the network querying results by estimating the node correspondence based on the CSRW model and minimizing the conductance of matching network modules.

2.3.2 Methods

Suppose that we have a query protein-protein interaction (PPI) network represented by a graph $\mathcal{G}_\mathcal{Q} = (\mathcal{V}_\mathcal{Q}, \mathcal{E}_\mathcal{Q})$, which has a set of nodes $\mathcal{V}_\mathcal{Q} = \{v_1, v_2, ...\}$ and set of edges $\mathcal{E}_\mathcal{Q} = \{e_{i,j}\}$. A protein in the query network is represented as a node $v_i \in \mathcal{V}_\mathcal{Q}$ in the graph $\mathcal{G}_\mathcal{Q}$ and the interaction between two proteins v_i and v_j is represented by an edge $e_{i,j}$, whose weight $w_{i,j}$ reflects the strength (or confidence) of the interaction. Similarly, suppose we are also given a target PPI network represented by a graph $\mathcal{G}_\mathcal{T} = (\mathcal{V}_\mathcal{T}, \mathcal{E}_\mathcal{T})$. We define the size of a network as the number of nodes in the given network, hence the size of the query network is $|\mathcal{V}_\mathcal{Q}|$ and that of the target network is $|\mathcal{V}_\mathcal{T}|$. Typically, in a network querying problem, the size of the target network is significantly larger than the query network (*i.e.*, $|\mathcal{V}_\mathcal{Q}| \ll |\mathcal{V}_\mathcal{T}|$). We assume that a pairwise node similarity score $s(v_q, v_t)$ is available

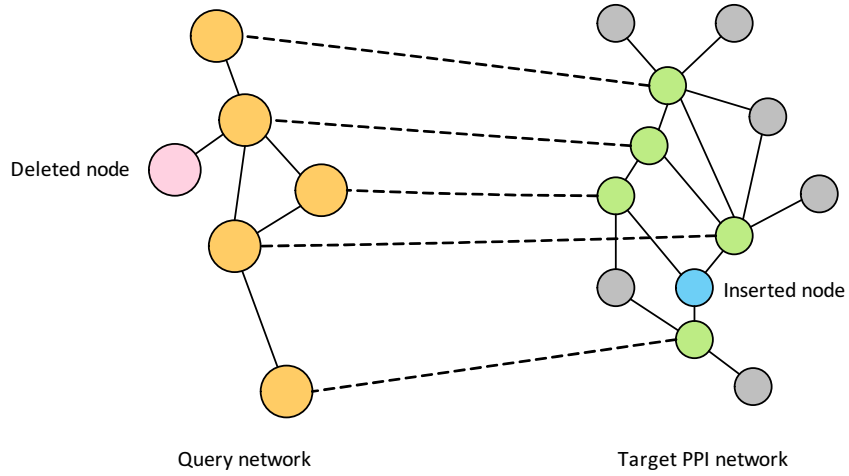


Figure 2.7: Illustration for the query network and conserved subnetwork in the target network. Gray colored nodes in the target network are irrelevant to the query network. Pink colored node is deleted in the target network, and blue colored node is inserted in the target network. Note that the inserted node in the target network is deleted in the query network, and vice versa [18] © [2017] BMC.

$\forall v_q \in \mathcal{V}_\mathcal{Q}$ and $\forall v_t \in \mathcal{V}_\mathcal{T}$, reflecting the molecular level similarity between the proteins in the query network and the target PPI network. In this study, we use the BLAST bit score as the pairwise node similarity score as in most network querying and alignment algorithms.

The main objective of network querying is to find the conserved subnetwork $\hat{\mathcal{G}}_\mathcal{T} = \left(\hat{\mathcal{V}}_\mathcal{T}, \hat{\mathcal{E}}_\mathcal{T}\right)$ within the target PPI network $\mathcal{G}_\mathcal{T} = (\mathcal{V}_\mathcal{T}, \mathcal{E}_\mathcal{T})$ that bears the largest overall functional similarity to the given query network $\mathcal{G}_\mathcal{Q}$. Therefore, we can formulate the network querying problem as the following optimization problem:

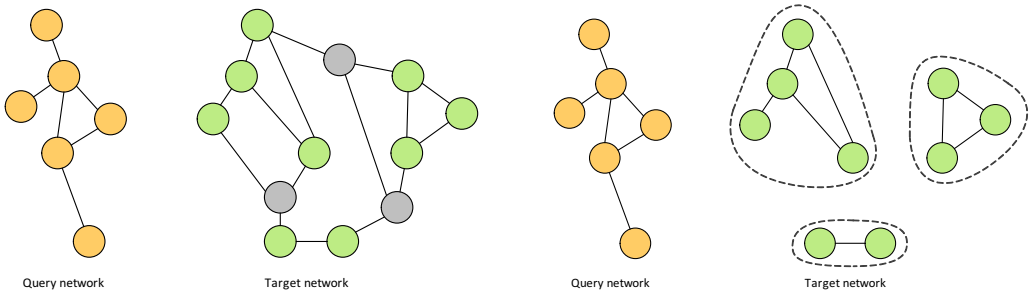
$$\hat{\mathcal{G}}_\mathcal{T}^* = \underset{\hat{\mathcal{G}}_\mathcal{T} \in \mathbf{G_T}}{\arg\max}\, f\left(\hat{\mathcal{G}}_\mathcal{T}, \mathcal{G}_\mathcal{Q}\right), \tag{2.8}$$

where $\mathbf{G_T}$ is the set of all possible subnetworks of the target PPI network, and $f\left(\mathcal{G}_x, \mathcal{G}_y\right)$ is a function that measures the overall functional similarity between two networks $\mathcal{G}_x$ and $\mathcal{G}_y$.

The network querying problem can be reformulated as a subgraph isomorphism prob-

lem, whose goal is to find a bijection between two graphs. In order to find a one-to-one mapping, deleted nodes can be modeled as dummy nodes so that an inserted node in the query network can be mapped to a dummy node in the target network, and vice versa. The subgraph isomorphism problem is known to be NP-complete [64], hence the existence of a polynomial time algorithm for solving the problem is unknown. Furthermore, it is also not straightforward to quantitatively estimate the overall functional similarity $f(\mathcal{G}_x, \mathcal{G}_y)$ between two networks $\mathcal{G}_x$ and $\mathcal{G}_y$ in such a way that is biologically meaningful. As a result, it is practically challenging to effectively formulate the optimization problem in (2.8) and solve the problem for large-scale networks in a computationally efficient manner [11, 25, 57]. A reasonable way to estimate this functional similarity is to define $f(\mathcal{G}_x, \mathcal{G}_y)$ by sensibly combining the node similarity and the topological similarity between the networks under comparison [10]. Given a reasonable $f(\mathcal{G}_x, \mathcal{G}_y)$, heuristic optimization schemes may have to be employed to make the optimization problem (2.8) computationally tractable.

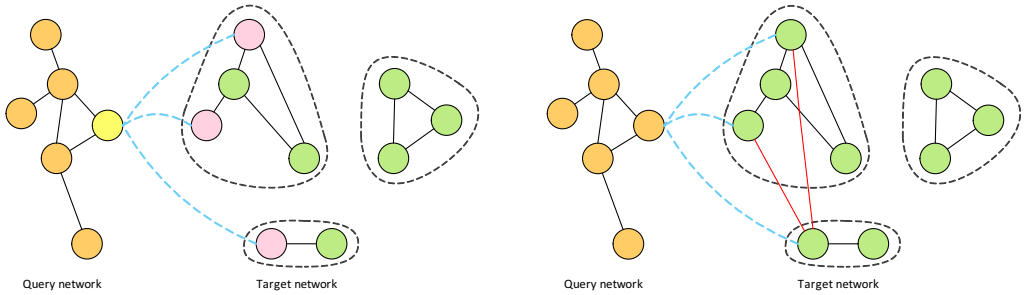
Before computing the node correspondence scores based on the CSRW model, we perform two pre-processing steps. First, we reduce the target network by removing potential non-homologous nodes. Specifically, we remove every node v_t in the target network whose node similarity $s(v_q, v_t)$ never exceeds a given threshold T_h for any of the query nodes $v_q \in \mathcal{V}_Q$. In this study, we set the threshold T_h as 0, such that a node is kept in the target network if it has at least one query node with nonzero similarity score. Removing target nodes that do not have any homologous node in the query network can significantly reduce the computation time as well as the memory requirement. Second, since removing non-homologous nodes may make the target network disconnected, we insert a pseudo-edge between nodes that are likely to share similar functionalities, motivated by the fact that proteins with direct interactions are more likely to share similar functionalities [65]. For this purpose, we assumed that any two nodes in the target network are likely to share



(a) Query and target network including non-homologous nodes. Gray colored nodes represent the non-homologous nodes.

(b) Ill-connected target network after removing non-homologous nodes.

Figure 2.8: Example for the pre-processing: removing non-homologous nodes [18] © [2017] BMC.



(a) Before inserting pseudo edges: pink colored proteins in the target network share the common potential homologous protein (yellow colored) in the query network.

(b) Target network with pseudo edges. Red colored edges are inserted pseudo edges between two proteins having a common potential homologous node in the query network.

Figure 2.9: Example for the pre-processing: inserting pseudo edges [18] © [2017] BMC.

similar functionalities and may potentially have a direct interaction if they have a common node in the query network with high node similarity. However, to refrain from inserting too many false-positive pseudo edges, we only insert a pseudo edge if the two nodes under consideration belong to different subnetworks that are disconnected from each other.

After pre-processing the target network, the CSRW model is used to estimate the correspondence between nodes in the query and the target networks. The resulting node

correspondence score matrix $\mathbf{C}$ is normalized to obtain the normalized score matrix $\bar{\mathbf{C}}$ using the normalization method proposed in [13]:

$$\bar{\mathbf{C}} = \frac{1}{2}\left[\mathbf{J_L} \cdot \mathbf{C} + \mathbf{C} \cdot \mathbf{J_R}\right]. \tag{2.9}$$

The matrix $\bar{\mathbf{C}}$ is a $|\mathcal{V}_\mathcal{Q}| \times |\mathcal{V}_\mathcal{T}|$ dimensional matrix containing the normalized node correspondence scores, $\mathbf{J_L}$ is a $|\mathcal{V}_\mathcal{Q}| \times |\mathcal{V}_\mathcal{Q}|$ dimensional diagonal matrix with the diagonal term $\mathbf{J_L}(q,q) = 1 \Big/ \sum_{t=1}^{|\mathcal{V}_\mathcal{T}|} c(v_q, v_t)$, and $\mathbf{J_R}$ is a $|\mathcal{V}_\mathcal{T}| \times |\mathcal{V}_\mathcal{T}|$ dimensional diagonal matrix with the diagonal term $\mathbf{J_R}(t,t) = 1 \Big/ \sum_{q=1}^{|\mathcal{V}_\mathcal{Q}|} c(v_q, v_t)$. This normalization step aims to estimate the *relative* significance between corresponding nodes, which has been shown to be useful for comparing networks of different size [13]. Based on the normalized correspondence score $\bar{\mathbf{C}}$, we iteratively select N_Q seed nodes in the target network based on the following rule:

$$\arg\min_{v_t} \left[\prod_{v_q \in \mathcal{V}_\mathcal{Q}} \left(1 - \bar{c}(v_q, v_t)\right)\right]. \tag{2.10}$$

The above selection rule aims to identify the nodes in the target network that have a large number of highly corresponding nodes in the query network. The score $\bar{c}(v_q, v_t)$ will be close to 1 for a highly corresponding node pair (v_q, v_t). Therefore, the product $\prod_{v_q \in \mathcal{V}_\mathcal{Q}} \left(1 - \bar{c}(v_q, v_t)\right)$ will approach 0 for a target node v_t (*i.e.*, a potential seed node) that has a large number of query nodes $v_q \in \mathcal{V}_\mathcal{Q}$ with a high node correspondence score $\bar{c}(v_q, v_t)$. This is based on an assumption that a target node with a larger number of relevant nodes in the query network may be more likely to be involved in similar functions as the query network compared to a node that has fewer corresponding nodes. After selecting the N_Q seeds, we find the largest connected subnetwork based on the N_Q seed nodes, which is referred to as the seed network. In this work, we set $N_Q = |\mathcal{V}_\mathcal{Q}|$ so that the size of the seed network does not exceed the size of the query network.

Once the seed network is obtained, we iteratively extend the network by adding nodes that can make the extended network well-separated from the rest of the network. To this aim, we estimate the conductance of the subnetwork and define the extension reward score for each node as follows. First, given a network $\mathcal{G} = (\mathcal{V}_\mathcal{G}, \mathcal{E}_\mathcal{G})$, suppose that we have a Gaussian surface enclosing the subnetwork $\mathcal{H} = (\mathcal{V}_\mathcal{H}, \mathcal{E}_\mathcal{H})$ such that $\mathcal{H} \subseteq \mathcal{G}$. Then the conductance φ of the subnetwork $\mathcal{H}$ is defined as the number of edges that pass through the surface divided by the volume of the subnetwork (*i.e.*, the number of edges that are enclosed by the surface) [66, 67]. The conductance of the subnetwork $\mathcal{H}$ is given by

$$\phi\left(\mathcal{H}\right) = \frac{|\{e_{i,j}|i \in \mathcal{V}_\mathcal{H}, j \in \mathcal{V}_{\bar{\mathcal{H}}}\}|}{\min\left(vol\left(\mathcal{V}_\mathcal{H}\right), vol\left(\mathcal{V}_{\bar{\mathcal{H}}}\right)\right)}, \tag{2.11}$$

where $\bar{\mathcal{H}} = (\mathcal{V}_\mathcal{G} \backslash \mathcal{V}_\mathcal{H}, \mathcal{E}_\mathcal{G} \backslash \mathcal{E}_\mathcal{H})$, and $vol\left(\mathcal{V}_\mathcal{X}\right) = \sum_{u \in \mathcal{V}_\mathcal{X}} d\left(u\right)$, where $d\left(u\right)$ is the degree of the node u. In a network querying problem, since the conserved subnetwork is typically significantly smaller than the rest of the target PPI network, the volume of the querying result is also much smaller than the volume of the rest of the target network, *i.e.*, $vol\left(\mathcal{V}_\mathcal{H}\right) \ll vol\left(\mathcal{V}_{\bar{\mathcal{H}}}\right)$. Hence, the conductance of the subnetwork $\mathcal{H}$ becomes

$$\phi\left(\mathcal{H}\right) = \frac{|\{e_{i,j}|i \in \mathcal{V}_\mathcal{H}, j \in \mathcal{V}_{\bar{\mathcal{H}}}\}|}{vol\left(\mathcal{V}_\mathcal{H}\right)} = \frac{|\{e_{i,j}|i \in \mathcal{V}_\mathcal{H}, j \in \mathcal{V}_{\bar{\mathcal{H}}}\}|}{|\{e_{i,j}|i, j \in \mathcal{V}_\mathcal{H}\}|}. \tag{2.12}$$

Second, we define the extension reward score for a given node as the number of newly added neighboring nodes during the extension step. That is, in each extension step, when we add a new node, all neighboring nodes in the extended subnetwork will get an extra extension reward score of 1. Based on the extension reward score, we can measure the contribution of each node towards making the subnetwork dense. A node with a higher extension reward score interacts with a larger number of newly added nodes, playing a more significant role in making the subnetwork dense after adding the new nodes.

Algorithm 1: SEQUOIA network querying algorithm

Data: Query and target network, pairwise node similarity score
Result: Best matching subnetwork in the target network for the given query
begin

Data pre-processing: i) Removing non-homologous nodes and ii) Inserting pseudo-edges
Compute the normalized node correspondence $\bar{\mathbf{C}}$ using Eq. (2.9)
Select the seed network $\mathcal{G}_{\mathcal{S}} = \{\mathcal{V}_{\mathcal{S}}, \mathcal{E}_{\mathcal{S}}\}$ using Eq. (2.10)
while $|\mathcal{G}_{\mathcal{S}}| \leq 2 \cdot N_Q$ *or* $\varphi_{current} \leq \beta \cdot \varphi_{previous}$ **do**
    Find the set of neighboring nodes $\mathcal{N}$ of the network $\mathcal{G}_{\mathcal{S}}$
    Compute the conductance φ_t for the extended network $\{\mathcal{V}_{\mathcal{S}} \cup v_t\}$, for each $v_t, \forall v_t \in \mathcal{N}$
    Find the node $v_{t^*} = \arg\min_t \varphi_t$
    Extend the network $\mathcal{G}_{\mathcal{S}}$, i.e., $\mathcal{V}_{\mathcal{S}} = \{\mathcal{V}_{\mathcal{S}} \cup v_{t^*}\}$ and $\mathcal{E}_{\mathcal{S}} = \{\mathcal{E}_{\mathcal{S}} \cup e_{i,j}\}, \forall i \in \mathcal{V}_{\mathcal{S}}, \forall j \in v_{t^*}$
    Update the current conductance $\varphi_{current} = \varphi_{t^*}$
    Update the extension reward score $r(v_t) = r(v_t) + 1, \forall v_t \in \mathcal{N}(v_{t^*})$
end
Remove nodes in $\mathcal{G}_{\mathcal{S}}$ whose extension reward score is 0 while keeping the initial seed nodes.

end

In each extension step, we add the node which is densely connected to the nodes within the extending network and loosely connected to the nodes out of the extending network, in order to minimize the conductance defined in (2.12). We repeat the extension steps until there is no more neighboring node that can reduce the current conductance by more than 5 percent or until the size of extending network exceeds twice the size of the query network, whichever occurs first. Once the extension process comes to an end, we remove all nodes whose extension reward score does not exceed a certain threshold. This is to enhance the functional coherence of the final querying result, since nodes with fewer interactions are relatively less likely to share similar functionalities with other neighbors. However, the original seed nodes are kept in the final result, even if their extension reward score is not large, since those nodes have high node correspondence to nodes in the query network. In

this study, we set the threshold for node removal as 0, so that nodes that do not interact with any of the newly added nodes are removed in the final querying result. The overall procedure of the proposed SEQUOIA network querying algorithm is summarized in Algorithm 1.

2.3.3 Results

To assess the performance of SEQUOIA, we carried out network querying experiments based on the real PPI networks of three different species – *H. sapiens* (human), *S. cerevisiae* (yeast), and *D. melanogaster* (fly) – obtained from [68]. PPI networks in [68] were originally obtained from the STRING database [69], but interactions between proteins without experimental validation were removed. The human PPI network contains 12,575 proteins and 86,890 interactions, the fly PPI network contains 8,624 proteins and 39,466 interactions, and the yeast PPI network contains 6,136 proteins and 166,229 interactions.

As the query networks, we used protein complexes obtained from [57], comprised of complexes in three species: *H. sapiens*, *S. cerevisiae*, and *D. melanogaster*. Furthermore, we expanded the query set by adding the latest version of human complexes obtained from CORUM [70], and yeast complexes from SGD [71] (as of Jan. 5, 2015). Finally, as in [57, 11], we selected connected complexes of size 5$\sim$25 and used them as our query networks (863 complexes in total). We assessed the performance of SEQUOIA based on the 863 real protein complexes, where 293 human complexes were searched against the fly PPI network, 289 human complexes were searched against the yeast PPI network, 141 yeast complexes were searched against the human PPI network, and 140 yeast complexes were searched against the fly PPI network. Since there are only a small number of test cases for querying fly complexes against human and yeast PPI networks, we excluded those experiments in this study.

The performance of SEQUOIA was compared against several state-of-the-art algo-

rithms, which include: RESQUE [11], Corbi [58], NatalieQ [56], HubAlign [34], and LocalAli [72]. Although HubAlign and LocalAli are global and local network alignment algorithms, respectively, we used those algorithms to identify conserved subnetworks as network querying can be viewed as a special case of pairwise network alignment. For Corbi, we used the default parameters for the gap penalty and set the option for the query type as 1, which is for general network querying. For HubAlign, we used the default parameters (*i.e.*, $\lambda = 0.1$ and $\alpha = 0.7$). We also used the default parameter for NatalieQ. For LocalAli, we set the minimum number of extension (-minext) to 0 and the maximum number of extension (-maxext) to 25, since the size of the query networks ranged between 5 to 25. Default values were used for other parameters. Since LocalAli identifies multiple local complexes as its output, we selected the complex with the best score as the querying result of LocalAli.

To assess various aspects of the network querying algorithms, we defined several performance metrics. First, we used the matching score to count the number of matches for each query and target species pair [73]. Given two biological complexes Q and C, the matching score is computed based on the Jaccard index between the nodes in the two biological complexes as follows:

$$match\_score\,(Q, C) = \frac{|\mathcal{V}_Q \cap \mathcal{V}_C|}{|\mathcal{V}_Q \cup \mathcal{V}_C|}, \tag{2.13}$$

where $\mathcal{V}_X$ is the set of nodes in the complex X. If the matching score is greater than the threshold, the two complexes were regarded to be a match. As in [73], we set the threshold for the matching score as 0.5. To count the number of matches, we used the known biological complexes as our gold standard reference $\mathcal{C} = \{C_1, C_2, ..., C_N\}$. Given the querying result Q_i, if there is at least one matching complex C_j in the gold standard reference, we counted Q_i as a match. Then, we report the total number of matches for each

query and target species pair. That is, given the querying results $\mathcal{Q} = \{Q_1, Q_2, ..., Q_M\}$ for the M query complexes, we count the total number of querying results

$$|\{Q_i | match\_score\,(Q_i, C_j) \geq 0.5, \forall C_j \in \mathcal{C}, \forall Q_i \in \mathcal{Q}\}|\,. \tag{2.14}$$

Next, we defined two different types of hits that respectively measure: 1) the accuracy of the obtained querying results and 2) the capability of detecting novel functional network modules with strong biological significance. The former counts the number of querying results whose annotation is identical to the functional annotation of the query network so that it can assess the capability of a given algorithm to identify the conserved functional modules. The latter counts the number of querying results with strong biological significance, regardless of whether or not they have the same functional annotation as the query, so that it can be used to assess the ability of the network querying algorithm to predict novel potential functional modules in the target PPI network.

To evaluate the accuracy of the querying results, we picked the most significantly enriched GO term of the query network (referred to as the significant GO term). Note that the most significantly enriched GO term denotes the GO term with the lowest false discovery rate (FDR) corrected p-value. To this aim, we performed GO enrichment tests for the query network and the querying result. If the significant GO term in the query is also enriched in the network querying result and if its FDR corrected p-value is less than a threshold, we regarded the querying result as a significant hit. However, a higher number of significant hits do not necessarily imply that the network querying algorithm yields accurate results, since the querying results may potentially include a large number of functionally irrelevant proteins (i.e., proteins whose annotation does not include the significant GO term). For this reason, in order to assess the accuracy of the querying results, we additionally defined two important performance metrics: the significant specificity (SPE) and the significant

functionally coherent (FC) hit. Significant SPE is defined as the relative proportion of the proteins annotated with the significant GO term among the proteins included in the querying result. Based on this definition, an accurate querying result with fewer irrelevant proteins will have a higher significant SPE. Significant FC hits were defined as hits that satisfy the following two conditions: 1) FDR corrected p-value should be less than a certain threshold, and 2) at least 50% of the proteins included in the querying result should be annotated with the significant GO term. A network querying algorithm that can yield a larger number of significant FC hits can be viewed as being more accurate and being capable of making better predictions that are biologically more significant.

Next, in order to assess the capability of detecting novel potential functional network modules, we investigated the biological significance of the querying results. To this aim, we performed the GO enrichment test only for the querying result (i.e., not for the query network) and selected the GO term with the smallest FDR corrected p-value as the most significantly enriched GO term. If the FDR corrected p-value of the most significantly enriched GO term of the querying result is less than a threshold, we regarded the querying result as a hit. A querying result with a small FDR corrected p-value can be viewed as being biologically significant, even if the most significantly enriched GO term of the querying result and that of the query network do not match. As a result, for a given network querying algorithm, we can assess its capability of detecting potential functional network modules by measuring the number of hits. Furthermore, we defined the specificity as the relative proportion of proteins (in the querying result) that are annotated with the most significantly enriched GO term among all proteins included in the querying result. As before, we defined a hit as being functionally coherent (FC) – hence called a FC hit – if the FDR corrected p-value is less than a certain threshold and if more than 50% of the proteins in the retrieved result are annotated with the most significantly enriched GO term.

We used the latest version of GO::TermFinder [74] for the GO enrichment test, and

analyzed the querying results based on three different ontology aspects: 1) cellular component (CC, GO:0005575), 2) biological process (BP, GO:0008150), and 3) molecular function (MF, GO:0003674). In the following, we mainly present the assessment results based on the ontology aspect of "cellular component". The ontology and annotation files for the three species considered in our study have been downloaded from Gene Ontology Consortium [75, 76] (as of Feb. 9 2015). Then, we removed all GO terms without experimental evidence. That is, we only used GO terms having one of the following evidence codes: 'EXP', 'IDA', 'IPI', 'IMP', 'IGI', and 'IEP'. Additionally, due to the hierarchical structure of GO terms, certain GO terms are annotated to a large number of proteins, where such commonly appearing GO terms would not be very informative. In order to use the GO terms that are informative, we computed the information content (IC) for each GO term as recommended in [75]. IC is defined as

$$IC\left(g\right) = -\log_2 \frac{|g|}{|root\left(g\right)|}, \tag{2.15}$$

where $|g|$ is the total number of proteins with the GO term g, and $|root\left(g\right)|$ is the number of proteins under the root GO term of the GO term g. Note that there are three root GO terms: cellular component (CC, GO:0005575), biological process (BP, GO:0008150), and molecular function (MF, GO:0003674). In this study, we only used the GO terms whose information content is at least 2.

Figure 2.10 shows the number of matches for each query-target species pair. The figure shows that SEQUOIA yields the largest number of matches among all tested algorithms for all query-target pairs. When querying human complexes against the fly and the yeast PPI networks, SEQUOIA clearly outperforms other methods. When querying yeast complexes against the human and the fly PPI networks, NatalieQ shows comparable performance to SEQUOIA, although SEQUOIA still yields a larger number of matches compared to all

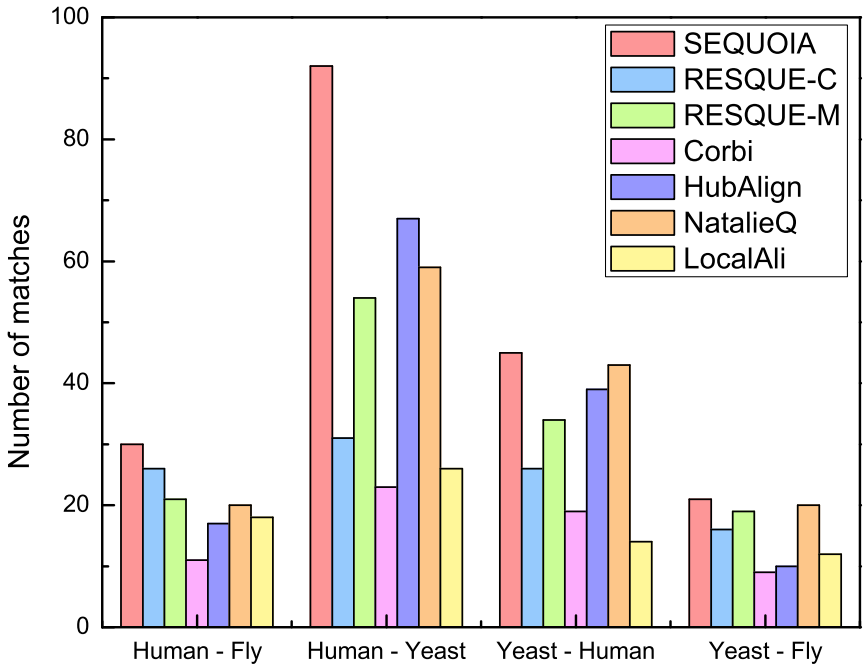
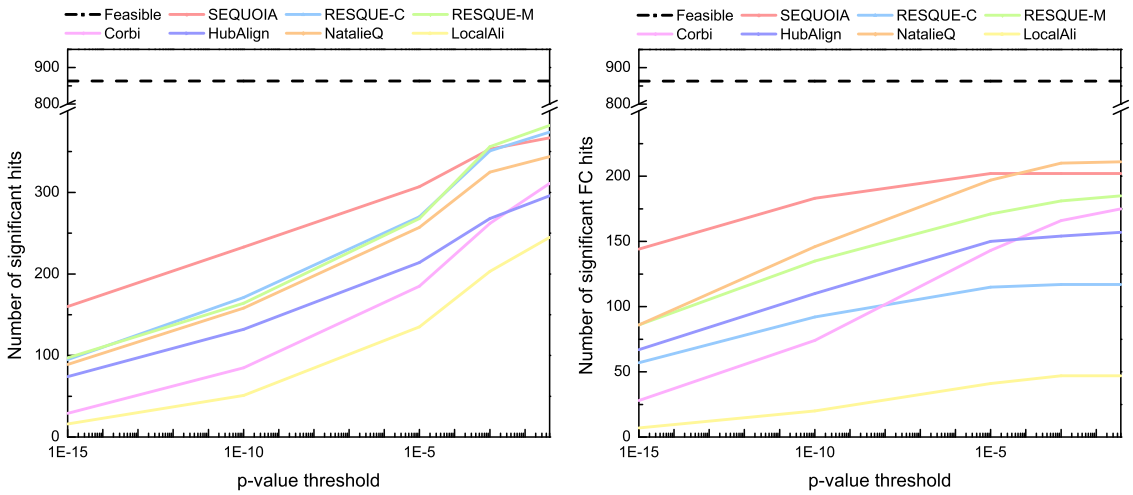


Figure 2.10: Number of matches for each query and target species pair (i.e., query species – target species) [18] © [2017] BMC.

other methods. Overall, SEQUOIA resulted in 188 matches, which is almost 32 percent more compared to the number of matches achieved by the next best algorithm, NatalieQ.

Figure 2.11 shows the number of significant hits and significant FC hits for all 863 querying results. As we can see in Figure 2.11a, SEQUOIA yields a larger number of significant hits compared to other algorithms. This means that SEQUOIA can more accurately identify conserved functional network modules with the significant GO term, (i.e., the most significantly enriched GO term in the query network). RESQUE family yielded similar number of significant hits at the p-value threshold of 0.05, but SEQUOIA outperformed both RESQUE-C and RESQUE-M when a smaller p-value threshold was used. Except for SEQUOIA and RESQUE-C, the number of nodes in the querying result is generally smaller than that in the query network for other tested algorithms. As a consequence, many algorithms may fail to identify inserted nodes and yield fewer significant hits.



(a) Number of significant hits.

(b) Number of significant FC hits.

Figure 2.11: Number of significant hits and significant functionally coherent (FC) hits for the 863 query complexes [18] © [2017] BMC.

Figure 2.11b shows that SEQUOIA yields a larger number of significant FC hits compared to other algorithms. This implies that SEQUOIA produces more accurate querying results that are functionally more coherent. Compared to SEQUOIA, the number of significant FC hits for Corbi decreases quickly as the p-value threshold decreases. Interestingly, although RESQUE family shows similar performance in terms of the number of significant hits, the number of significant FC hits for RESQUE-C is much smaller than RESQUE-M. This result shows that using a more sophisticated method to predict the best matching subnetwork would be needed to obtain better querying results that are functionally more coherent. In fact, RESQUE-C uses a relatively simple approach to find the best matching subnetwork, which is to find the largest connected subnetwork in the reduced target network, and this may increase the chances of including a larger number of functionally irrelevant nodes in the final querying result. SEQUOIA results in higher significant hits as well as higher significant FC hits by minimizing the network conductance of the matching subnetwork and filtering out potentially irrelevant nodes based on the extension reward

	Identified nodes	Annotated nodes†	Significant SPE
SEQUOIA	9,537	2,568	0.269
RESQUE-C	10,213	2,115	0.207
RESQUE-M	7,000	1,941	0.277
Corbi	4,761	1,149	0.241
HubAlign	7,342	1,526	0.208
NatalieQ	5,452	1,745	0.320
LocalAli	6,220	892	0.143

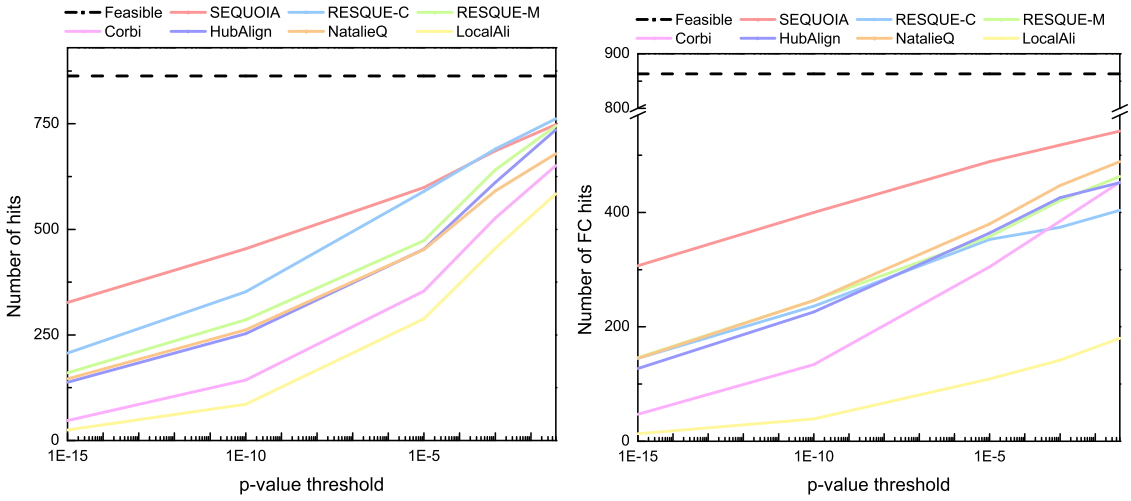
† Annotation corresponding to the most significantly enriched GO term in the query network.

Table 2.8: Significant SPE for the ontology aspect of "cellular component" [18] © [2017] BMC.

score.

The number of identified nodes included in the querying results and the number of nodes annotated with the most significant GO term are summarized in Table 2.8. The table shows that NatalieQ and RESQUE-M achieve higher significant SPE compared to SEQUOIA, but it should be noted that SEQUOIA can identify a much larger number of "annotated nodes" while keeping relatively higher significant SPE compared to other algorithms. The total number of identified nodes is comparable for SEQUOIA and RESQUE-C, although SEQUOIA results in a much higher significant SPE compared to RESQUE-C. From the perspective of potential knowledge transfer from a well-studied species to a less-studied species, the ability to achieve higher significant SPE is critical, as it implies that the network querying algorithm may be able to annotate the proteins in the querying result more accurately.

Figure 2.12 shows the number of hits and the number of FC hits for various FDR corrected p-value thresholds. Feasible hits in each figure correspond to the total number of query complexes, which is the maximum number of hits that can be achieved. As we can see in Figure 2.12a, SEQUOIA clearly outperforms other algorithms for various p-value thresholds. For example, at a p-value threshold of 1E-10, SEQUOIA yields 29%



(a) Number of hits.

(b) Number of FC hits.

Figure 2.12: Number of hits and FC hits for querying 863 biological complexes [18] © [2017] BMC.

more hits than RESQUE-C, which is the next best algorithm. These results indicate that SEQUOIA has stronger potentials to identify novel protein complexes compared to other state-of-the-art algorithms.

Next, we compared the number of FC hits for different network querying algorithms. Figure 2.12b shows that SEQUOIA clearly outperforms other algorithms. For example, SEQUOIA can identify 11% more FC hits than NatalieQ at a p-value threshold of 0.05 and almost twice as many FC hits compared to RESQUE and NatalieQ at a p-value threshold of 1E-15. LocalAli and NatalieQ fail to yield querying results in some test cases (*i.e.*, these algorithms cannot identify any protein node in the target network). LocalAli and NatalieQ may not perform robustly under certain conditions (*e.g.*, for certain query topology), which may result in a smaller number of hits. The results in Figure 2.12b show that SEQUOIA's performance is more robust compared to many other algorithms, and that SEQUOIA can more effectively detect conserved network modules with high functional coherence.

Finally, we also evaluated the functional coherence of the querying results for each

	Identified nodes	Annotated nodes[‡]	SPE
SEQUOIA	9,537	5,531	0.580
RESQUE-C	10,213	5,002	0.492
RESQUE-M	7,000	3,856	0.551
Corbi	4,761	2,486	0.522
HubAlign	7,342	3,822	0.521
NatalieQ	5,452	3,324	0.610
LocalAli	6,220	2,170	0.349

[‡] Annotation corresponding to the most significantly enriched GO term in the querying result.

Table 2.9: SPE for the ontology aspect of "cellular component" [18] © [2017] BMC.

algorithm. To this aim, we selected the most significantly enriched GO term in the querying result obtained by each algorithm for each query, and compute the relative proportion of proteins annotated with the most significantly enriched GO term. The results are summarized in Table 2.9. With the exception of NatalieQ, SEQUOIA achieves the highest SPE compared to all other algorithms. Although NatalieQ results in the highest SPE, SEQUOIA can identify about 66% more annotated nodes (*i.e.*, proteins annotated with the most significant GO term) compared to NatalieQ, while achieving a comparable SPE. This indicates that SEQUOIA can effectively identify a larger number of protein nodes that are functionally coherence than the other tested algorithms.

For RESQUE, we used the MATLAB script version 1.0 and MATLAB version 2014b. Executable binaries for NatalieQ, HubAlign, and LocalAli were obtained by compiling their source code using a C++ compiler. For Corbi, we used its R package and tested the algorithm on Windows. Except for Corbi, all other algorithms were tested on Mac OS X. All computer simulations were performed on a desktop computer equipped with a 2.4 GHz Intel i7 processor and 8 GB memory. For certain queries, NatalieQ and LocalAli may require a very long time (which is significantly longer than the average computation time), and such outliers were excluded when drawing the box plot for readability. As shown in Figure 2.13, the computation time of SEQUOIA is comparable to that of the RESQUE

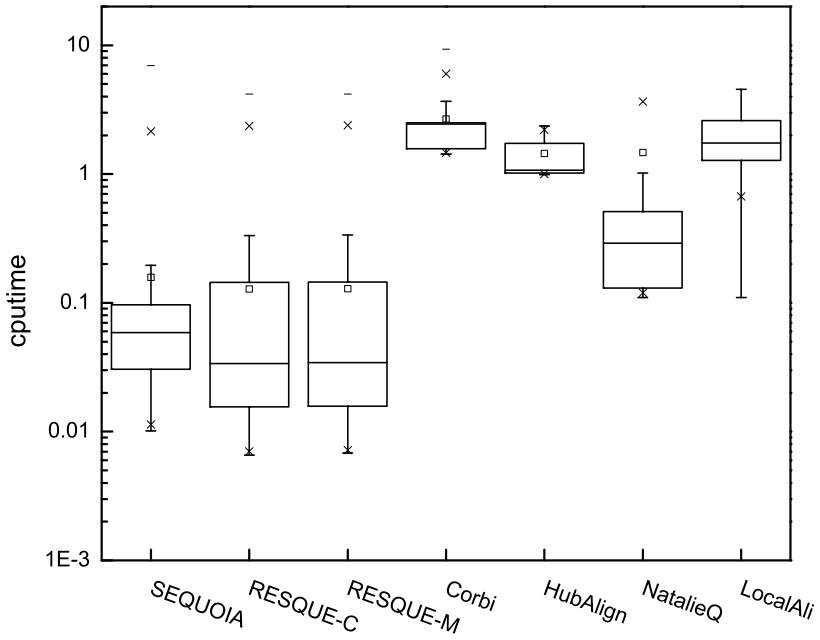


Figure 2.13: Computation time of 863 querying results for each querying algorithm [18] © [2017] BMC.

family, but it is much faster compared to other algorithms. On average, SEQUOIA yields the querying result in less than 0.06 second, and in 98% of the test cases, the algorithm needs less than a second to find the subnetwork that best matches the query.

2.3.4 Conclusions

In this subchapter, we proposed SEQUOIA, a novel network querying algorithm that can enhance the biological significance of the query results. In order to identify conserved subnetwork regions in the target network that are similar to a given query network, the algorithm compares the two networks and estimates the node correspondence scores by using the context-sensitive random walk model. Inspired by the pair hidden Markov model that has been widely used in the comparative sequence analysis, the CSRW model effectively captures the similarities between graphs by explicitly accounting for potentially inserted/deleted nodes. Based on the estimated CSRW node correspondence scores,

SEQUOIA identifies high-scoring regions (referred to as the seed networks) in the target network that bear considerable similarity with the query network. The seed network is further extended by adding neighboring nodes that reduce the network conductance of the extended network by the largest amount. This extension step identifies nearby proteins that are densely connected to other nodes in the potential network module, thereby effectively recruiting proteins that are likely to share similar functions with other proteins in the module. The final query result is obtained after pruning the matching subnetwork by removing any irrelevant nodes, thereby enhancing the separability and coherence of the identified network module. As we have shown through extensive numerical simulations based on 863 real biological complexes, our network querying algorithm SEQUOIA yields accurate query results with enhanced biological significance. The source code and datasets can be downloaded from http://www.ece.tamu.edu/∼bjyoon/SEQUOIA

3. ESTIMATION OF NODE-TO-NODE CORRESPONDENCE BY MEASURING THE STEADY-STATE NETWORK FLOW USING A MARKOV MODEL *

3.1 CUFID model

In this chapter, we propose a novel probabilistic random walk model for comparing PPI networks and effectively predicting the correspondence between proteins, represented as network nodes, that belong to conserved functional modules across the given PPI networks. The basic idea is to estimate the steady-state network flow between nodes that belong to different PPI networks based on a Markov random walk model. The random walker is designed to make random moves to adjacent nodes within a PPI network as well as cross-network moves between potential orthologous nodes with high sequence similarity. Based on this Markov random walk model, we estimate the steady-state network flow – or the long-term relative frequency of the transitions that the random walker makes – between nodes in different PPI networks, which can be used as a probabilistic score measuring their potential correspondence. Subsequently, the estimated scores can be used for detecting orthologous proteins in conserved functional modules through comparative network analysis.

3.1.1 Problem formulation

Suppose that we have a pair of PPI networks with the graph representations $\mathcal{G}_X = (\mathcal{U}, \mathcal{D})$ and $\mathcal{G}_Y = (\mathcal{V}, \mathcal{E})$, in which nodes represent proteins in each PPI network (i.e., $u_i \in \mathcal{U}$ or $v_j \in \mathcal{V}$), and edges ($d_{ij} \in \mathcal{D}$ or $e_{ij} \in \mathcal{E}$) indicate that the corresponding protein u_i (or v_i) binds with the protein u_j (or v_j). The edge weights in the PPI networks can

*Part of this chapter is reprinted with a permission from "Hyundoo Jeong, Xiaoning Qian, and Byung-Jun Yoon. Effective comparative analysis of protein-protein interaction networks by measuring the steady-state network flow using a Markov model. *BMC Bioinformatics*, 17(Suppl. 13):395, 2016" [77] ©[2016] BioMed Central.

indicate the strength or confidence of the interactions between the proteins. Given a pair of nodes across the PPI networks, we assume that the pairwise node similarity score $s(u_i, v_j)$, $u_i \in \mathcal{U}$ and $v_j \in \mathcal{V}$ can be computed, for example, based on the sequence similarity between the proteins. In this study, we utilized BLAST bit scores between proteins as the pairwise node similarity scores. However, other types of similarity measurements (or their combinations) could be also used as the pairwise node similarity score in case such measurements can be easily obtained.

Given a pair of PPI networks $\mathcal{G}_X$ and $\mathcal{G}_Y$, one objective of comparative network analysis is to derive the optimal one-to-one mapping A^* between nodes in different PPI networks. One possible criterion that could be used to find such a mapping is the maximum expected accuracy (MEA) criterion, which aims to maximize the expected number of correctly mapped nodes. Provided that we can derive a pairwise node alignment probability $\Pr[u_i \sim v_j | \mathcal{G}_X, \mathcal{G}_Y]$, $u_i \in \mathcal{U}$ and $v_j \in \mathcal{V}$, the optimal one-to-one mapping can be found by:

$$A^* = \arg\max_{A} \sum_{\forall (u_i \sim v_j) \in A} \Pr[u_i \sim v_j | \mathcal{G}_X, \mathcal{G}_Y] \tag{3.1}$$

according to the MEA criterion. This MEA approach has been widely used by many multiple sequence alignment algorithms [36, 38, 39, 41, 40] and it has been shown to be useful for network alignment [13, 17] and network querying [11] as well.

3.1.2 Motivation and overall approach

Based on the above problem setting, to obtain confident network comparison results, it is crucial to accurately estimate the pairwise node-to-node correspondences. To obtain biologically meaningful comparison results, it is necessary that the pairwise node correspondence is proportional to both the pairwise node similarity (i.e., sequence similarity) and the topological similarity between the subnetwork regions surrounding the nodes in the respective networks. This is based on the observation that orthologous proteins typically

have a high level of compositional similarity and often display similar interaction patterns to their neighboring nodes [9, 24]. To accurately estimate the pairwise node-to-node correspondences by effectively integrating these two different types of similarities, we propose to utilize the concept of steady-state network flow (i.e., the amount of 'water' that flows through a given channel in the network). Similar concepts have been previously adopted in various engineering applications to find the solutions to similar assignment problems. For example, in digital communication systems, the water-filling algorithm [78] is utilized to compute the optimal allocation of resources. Conceptually, it pours 'water' into an OFDM (orthogonal frequency division multiplexing) channel, and the 'water level' in the OFDM channel is utilized to find the optimal solution of the transmit power for each subcarrier. In digital image processing, the so-called watershed method [79] is used to find edges or contours of objects in the given image. The watershed method assumes that 'water' flows along the image gradient (e.g., intensity differences) and eventually reaches the local minima so that the 'water level' in the image provides the solution for the desired image segmentation.

In the proposed random walk model called CUFID (**C**omparative network analysis **U**sing the steady-state network **F**low to **ID**entify orthologous proteins) model, we measure the steady-state network flow in an integrated network that is obtained by combining the PPI network pair to be compared. More specifically, edges are inserted between nodes in different networks that have positive pairwise node similarity, and the pairwise node similarity score is assigned as the edge weight. Suppose we pour 'water' on the integrated network and that the amount of water flow is proportional to the edge weight. If a given pair of nodes in different PPI networks have higher pairwise node similarity and if their neighboring nodes also have higher pairwise node similarity, there would be a larger water flow between the pair of nodes in the long run. However, if the nodes have similar topological structure (i.e., in terms of the number of interacting nodes in the respective networks)

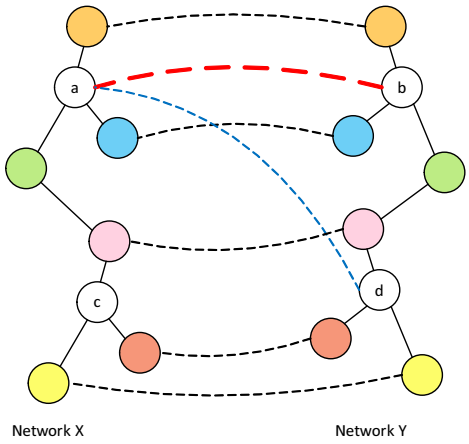


Figure 3.1: Illustration of how node correspondence is measured based on the steady-state network flow. Straight lines represent protein-protein interactions, and dotted lines indicate pairwise node similarity. In this example, there is a larger steady-state network flow between the node pair (a,b) than the node pair (a,d) because the node pair (a,b) has higher pairwise node similarity and as the nodes have similar interacting nodes in the respective networks. In contrast, although the node pair (a,d) has positive pairwise node similarity, the neighboring nodes in the respective networks are not similar, which leads to a smaller steady-state network flow between the nodes (a,d) compared to the flow between (a,b) [77] © [2016] BMC.

but if their neighboring nodes are not similar, there will be relatively small water flow between the pair of nodes. As a result, the water flow between nodes across different PPI networks provides an intuitive way of measuring the overall similarity of the nodes – or functional correspondence between the proteins. As will be shown later, the resulting node correspondence score obtained based on the concept of water flow in the integrated network can serve as an effective building block for constructing an accurate and biologically meaningful network alignment and querying.

3.1.3 Methods

In order to effectively estimate the node correspondence by integrating both the pairwise node similarity and topological similarity using a Markov random walk model, we first construct the integrated network $G = (V, E)$ by combining $\mathcal{G}_X = (\mathcal{U}, \mathcal{D})$ and $\mathcal{G}_Y =$

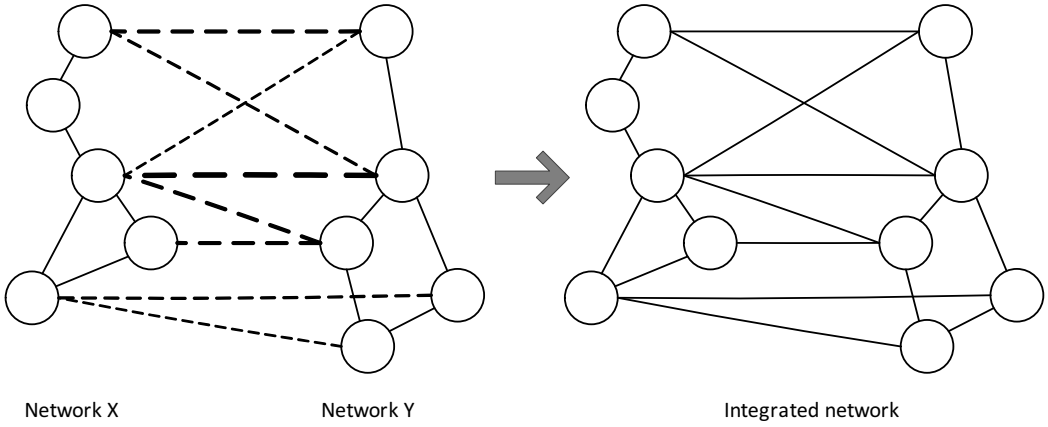


Figure 3.2: Illustration for constructing the integrated network from a network pair. Dotted lines in the left figure indicate the pairwise node similarity, in which the thickness of each line is proportional to the pairwise node similarity. To construct the integrated network, we insert an edge between each pair of nodes across different networks if they have positive pairwise node similarity [77] © [2016] BMC.

$(\mathcal{V}, \mathcal{E})$. Nodes of the integrated network G are the union of the nodes of $\mathcal{G}_X$ and $\mathcal{G}_Y$ (i.e., $V = \{\mathcal{U}, \mathcal{V}\}$), and edges are the union of the edges in the two networks $\mathcal{G}_X$ and $\mathcal{G}_Y$, and additional weighted (pseudo) edges $\mathcal{P}$ such that $\mathcal{P} = \{e_{i,j} | u_i \in \mathcal{U}, v_j \in \mathcal{V}, s(u_i, v_j) > s_t\}$, (i.e., $E = \{\mathcal{D}, \mathcal{E}, \mathcal{P}\}$). On this integrated network G, we allow the random walker to randomly move from the current node to any of its neighboring nodes at each time step. We define two different types of random moves based on their starting and ending points. First, if the random walker moves from a node in $\mathcal{U}$ to a node in $\mathcal{U}$ (or from a node in $\mathcal{V}$ to a node in $\mathcal{V}$), we define it as an *intra-network* random move, as the random walk takes place in the same PPI network. In this event, the random walker performs a random movement over the edges representing protein-protein interactions. Second, if the random walker moves from a node in $\mathcal{U}$ to a node in $\mathcal{V}$ (or from a node in $\mathcal{V}$ to a node in $\mathcal{U}$), we refer to this as a *cross-network* random move. In this case, the random walker can transit across the networks through the pseudo edges. The intra-network random move mainly aims to capture the topological similarity between the two PPI networks while the cross-

network random move aims to incorporate the pairwise node similarity between nodes that originally belong to different PPI networks.

The transition probabilities of the resulting random walker are determined as follows. Suppose the two networks $\mathcal{G}_X = (\mathcal{U}, \mathcal{D})$ and $\mathcal{G}_Y = (\mathcal{V}, \mathcal{E})$ have weighted edges, where the respective adjacency matrices are given by:

$$A_X[i,j] = \begin{cases} d_{ij}, & (u_i, u_j) \in \mathcal{D} \\ 0, & otherwise \end{cases}, \tag{3.2a}$$

$$A_Y[i,j] = \begin{cases} e_{ij}, & (v_i, v_j) \in \mathcal{E} \\ 0, & otherwise \end{cases}. \tag{3.2b}$$

First of all, to compute the transition probability of the intra-network random moves, we transform the edge weighted adjacency matrix into a legitimate stochastic matrix by normalizing each row. That is, the transition probability of the random walker is proportional to the weight of the edge that connects the node at the current position of the random walker and the neighboring node (in the same PPI network) to which it wants to move. The resulting transition probability of any intra-network random move is given by

$$P_k[i,j] = \frac{1}{\sum\limits_{\forall j} A_k[i,j]} \cdot A_k[i,j], k = X, Y. \tag{3.3}$$

Eq. (3.3) can be rewritten in a simple matrix form, which is given by

$$\mathbf{P_X} = \mathbf{D_X^{-1}} \cdot \mathbf{A_X} \text{ and } \mathbf{P_Y} = \mathbf{D_Y^{-1}} \cdot \mathbf{A_Y}, \tag{3.4}$$

where $\mathbf{D_X}$ is a $|\mathcal{U}| \times |\mathcal{U}|$ dimensional diagonal matrix such that $D_X[i,i] = \sum\limits_{\forall j} A_X[i,j]$, and $\mathbf{D_Y}$ is a $|\mathcal{V}| \times |\mathcal{V}|$ dimensional diagonal matrix such that $D_Y[i,i] = \sum\limits_{\forall j} A_Y[i,j]$.

Next, suppose that the transition probability of the cross-network random move between two nodes in different networks is proportional to their pairwise node similarity score. That is, from the current position of the random walker in a given PPI network, the random walker is more likely to move to a node in the other PPI network with higher pairwise node similarity. This will increase the 'network flow' between nodes that have higher node similarity. The transition probability for a cross-network random move from a node u_i in $\mathcal{G}_X$ to a node v_j in $\mathcal{G}_Y$ is then given by

$$\Pr\left[v_j|u_i\right] = P_{X\to Y}\left[i,j\right] = \frac{1}{\sum\limits_{\forall v_j} s\left[u_i, v_j\right]} \cdot s\left[u_i, v_j\right]. \tag{3.5}$$

In a matrix form, Eq. (3.5) can be written as:

$$\mathbf{P_{X\to Y}} = \mathbf{D_S^{-1}} \cdot \mathbf{S}, \tag{3.6}$$

where $\mathbf{S}$ is a $|\mathcal{U}| \times |\mathcal{V}|$ dimensional matrix for the pairwise node similarity score, and $\mathbf{D_S}$ is a $|\mathcal{U}| \times |\mathcal{U}|$ dimensional diagonal matrix such that $D_S\left[i,i\right] = \sum\limits_{\forall j} s\left[i,j\right]$. Similarly, the transition probability of a cross-network random move from a node v_i in $\mathcal{G}_Y$ to a node u_j in $\mathcal{G}_X$ is given by:

$$\Pr\left[u_j|v_i\right] = P_{Y\to X}\left[i,j\right] = \frac{1}{\sum\limits_{\forall u_j} s^{\mathrm{T}}\left[v_i, u_j\right]} \cdot s^{\mathrm{T}}\left[v_i, u_j\right], \tag{3.7}$$

where $s^{\mathrm{T}}\left[v_i, u_j\right]$ is a $[v_i, u_j]$-th element of the transposed matrix of $\mathbf{S}$. Eq. (3.7) can be written in a matrix form as follows:

$$\mathbf{P_{Y\to X}} = \mathbf{S}^{\mathrm{T}} \cdot \mathbf{D_{S^{\mathrm{T}}}^{-1}}, \tag{3.8}$$

where $\mathbf{S}^{\mathrm{T}}$ is a $|\mathcal{V}| \times |\mathcal{U}|$ dimensional matrix for the pairwise node similarity score, and $\mathbf{D}_{\mathbf{S}^{\mathrm{T}}}$ is a $|\mathcal{U}| \times |\mathcal{U}|$ dimensional diagonal matrix such that $D_{\mathbf{S}^{\mathrm{T}}}[i,i] = \sum_{\forall j} s^{\mathrm{T}}[i,j]$. In fact, the transition probability matrices $\mathbf{P}_{\mathbf{X}\to\mathbf{Y}}$ and $\mathbf{P}_{\mathbf{Y}\to\mathbf{X}}$ are normalized pairwise node similarity score matrices in the row-wise and column-wise manner.

Finally, we can get the $(|\mathcal{U}| + |\mathcal{V}|) \times (|\mathcal{U}| + |\mathcal{V}|)$ dimensional overall transition probability matrix for the Markov random walker over the integrated network G, given by

$$\mathbf{P} = \begin{bmatrix} \mathbf{P}_{\mathbf{X}} & \mathbf{P}_{\mathbf{X}\to\mathbf{Y}} \\ \mathbf{P}_{\mathbf{Y}\to\mathbf{X}} & \mathbf{P}_{\mathbf{Y}} \end{bmatrix}. \tag{3.9}$$

Based on the proposed random walk protocol, the random walker transits more frequently between the pair of nodes (u_i, v_j) if the node u_i and the node v_j have a higher pairwise node similarity and also if their neighboring nodes also have higher pairwise node similarity (i.e., higher topological similarity). So, as a result, the random walker will spend more time on an edge that connects a pair of nodes (u_i, v_j), $u_i \in \mathcal{U}$ and $v_j \in \mathcal{V}$ as their overall similarity (or node correspondence) increases. Hence, we can effectively estimate the pairwise node alignment probability – which should be proportional to the desired node correspondence – by measuring the steady-state network flow through each (pseudo) edge in $\mathcal{P}$ such that $\mathcal{P} = \{e_{i,j} | u_i \in \mathcal{U}, v_j \in \mathcal{V}, s(u_i, v_j) > s_t\}$

To compute the steady-state network flow, we first compute the steady-state probability $\pi(x)$ of the random walker for every node $x \in \mathcal{U} \cup \mathcal{V}$ in the integrated network. This is equivalent to the long-run proportion of time that the random walker spends at a given node x. The steady-state probability is equivalent to the eigenvector of the transition probability matrix $\mathbf{P}$ that corresponds to unit eigenvalue. This eigenvector, hence the steady-state probability, can be easily obtained through the power method, as the transition probability matrix $\mathbf{P}$ will be generally sparse for real PPI networks [13, 17]. The

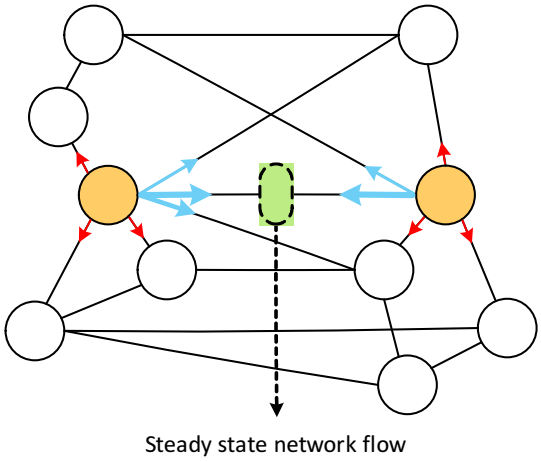


Figure 3.3: Illustration of the steady-state network flow. Note that the red colored arrows indicate the intra-network random moves, while the blue colored arrows represent the cross-network random moves [77] © [2016] BMC.

steady-state probability $\pi(x)$ can be viewed as the amount of 'water' at the node x in the long-run, and since the amount of the water flow is proportional to the edge weight, we can obtain the steady-state network flow along the edge (u_i, v_j) as follows:

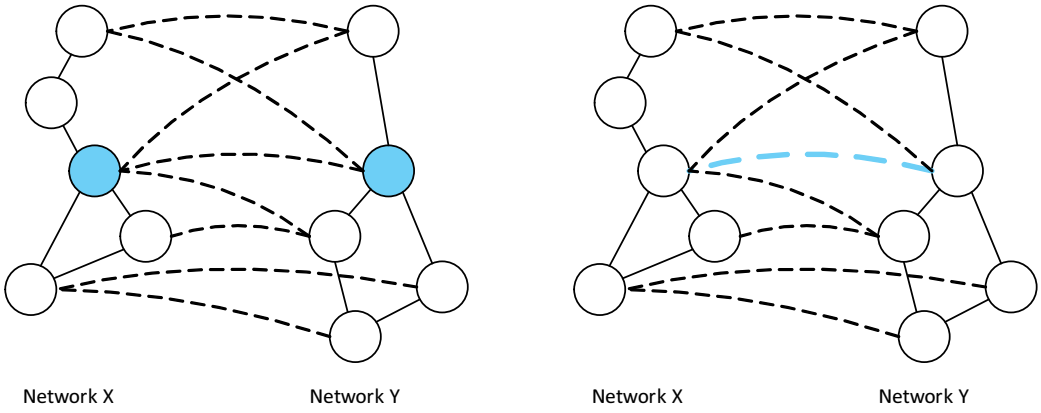
$$\begin{aligned} c(u_i, v_j) &= \pi(u_i) \cdot \Pr[v_j|u_i] + \pi(v_j) \cdot \Pr[u_i|v_j] \\ &= \pi(u_i) \cdot \frac{s(u_i,v_j)}{\sum\limits_{\forall v_j} s(u_i,v_j)} + \pi(v_j) \cdot \frac{s(u_i,v_j)}{\sum\limits_{\forall u_i} s(u_i,v_j)}. \end{aligned} \tag{3.10}$$

This equation can be rewritten in a matrix form as follows:

$$\mathbf{C} = \pi_{\mathbf{X}} \cdot \mathbf{P}_{\mathbf{X}\to\mathbf{Y}} + \mathbf{P}^{\mathrm{T}}_{\mathbf{Y}\to\mathbf{X}} \cdot \pi_{\mathbf{Y}}, \tag{3.11}$$

where $\mathbf{C}$ is a $|\mathcal{U}| \times |\mathcal{V}|$ dimensional matrix for the steady-state network flow (i.e., pairwise node correspondence scores), $\pi_{\mathbf{X}}$ is a $|\mathcal{U}| \times |\mathcal{U}|$ dimensional diagonal matrix such that $\pi_X[i,i] = \pi(u_i)$, $u_i \in \mathcal{U}$, and $\pi_{\mathbf{Y}}$ is a $|\mathcal{V}| \times |\mathcal{V}|$ dimensional diagonal matrix such that $\pi_Y[i,i] = \pi(v_j)$, $v_j \in \mathcal{V}$.

We briefly compare the major differences between CSRW model [16] and CUFID model [77]. First, although we designed that the staying time of the random walker is proportional to the node-to-node correspondences, the random walker in the CSRW model stays longer at the pair of potential matching nodes, but the random walker spends more time at the pseudo edge connecting the potential matching nodes in the CUFID model.



(a) CSRW model. (b) CUFID model.

Figure 3.4: Illustration of the main difference between CSRW model and CUFID model. Node correspondence can be estimated by measuring the long-run proportion of time that the random walker stays at the blue colored node pair (CSRW models) or blue colored edges (CUFID model).

Second, since constructing the transition probability matrix for the random walker is the main bottleneck to compute node-to-node correspondences, we briefly compare the memory requirement to construct the transition probability matrix for each model. Suppose that we have two networks and each has M and N nodes, respectively. The memory complexity for each method is summarized in Table 3.1

	CSRW model	CUFID model
Memory complexity	$O\left(\lvert M \cdot N \rvert \times \lvert M \cdot N \rvert\right)$	$O\left(\lvert M + N \rvert \times \lvert M + N \rvert\right)$

Table 3.1: Memory complexity to construct a transition probability matrix.

3.2 Network alignment through the CUFID model

In this subchapter, we propose a novel network alignment algorithm, called **CUFID-align** (**C**omparative network analysis **U**sing the steady-state network **F**low to **ID**entify orthologous proteins). The algorithm estimates the node correspondence by measuring the steady-state network flow of a random walk model over an *integrated network* of the given PPI networks. To accurately estimate the node correspondence based on the steady-state network flow, in a way that effectively captures the biological significance, we adopt the CUFID model such that the relative frequency that the random walker makes transitions between a pair of nodes in different PPI networks is proportional to the pairwise node similarity and the topological similarity between the surrounding network regions. The proposed scheme effectively captures the functional correspondence between nodes across different networks and the estimated node correspondence scores can lead to accurate network alignment results, as will be demonstrated through performance assessment based on real PPI networks.

3.2.1 Methods

Suppose that we have a pair of PPI networks $\mathcal{G}_X = (\mathcal{U}, \mathcal{D})$ and $\mathcal{G}_Y = (\mathcal{V}, \mathcal{E})$. To obtain biologically meaningful alignment results, we first estimate node-to-node correspondences $\mathbf{C}$ through Eq. (3.11). Then, as in SMETANA [13] and SMETANA-CSRW [17], we utilize the following probabilistic consistency transformation (PCT) given by:

$$\tilde{\mathbf{C}} = \alpha \cdot \mathbf{C} + (1 - \alpha) \cdot \mathbf{P}_{\mathbf{X}} \cdot \mathbf{C} \cdot \mathbf{P}_{\mathbf{Y}}^{\mathrm{T}}, \tag{3.12}$$

to update the estimated node correspondence scores. The above PCT assumes that, given a pair of nodes, if their neighboring nodes have high correspondence, the node pair has increased chance to be aligned. That is, updating the estimated node correspondence score by utilizing the neighbor's node correspondence could increase the overall accuracy of the node correspondence score. However, the PCT also has the potential risk of creating or increasing false positive node correspondence. That is, some node pairs with zero (or insignificant) correspondence scores can have positive (or increased) node correspondence scores after performing the PCT if they have neighboring nodes with positive correspondence scores, because PCT propagates the node correspondence scores to neighboring nodes. Therefore, to suppress false positive node alignments, we only keep the transformed scores that are larger than the 90 percentile ($= \beta$). Furthermore, we also keep the original scores $c\left[i,j\right]$ even if they are smaller than the threshold β. That is,

$$\bar{c}\left[i,j\right] = g\left(\tilde{c}\left[i,j\right]\right) = \begin{cases} \tilde{c}\left[i,j\right], & \text{if } \tilde{c}\left[i,j\right] \geq \beta \text{ or } c\left[i,j\right] > 0 \\ 0, & otherwise \end{cases}. \tag{3.13}$$

After transforming and removing node correspondence scores lower than a specific threshold using Eq (3.13), we obtain the final node correspondence scores $\bar{\mathbf{C}}$, which will be used to construct the network alignment.

After computing the transformed node correspondence score $\bar{\mathbf{C}}$, we use the scores to construct the network alignment based on the MEA criterion, based on the assumption that the pairwise node alignment probability is proportional to the obtained node correspondence score:

$$\Pr\left[u_i \sim v_j | \mathcal{G}_X, \mathcal{G}_Y\right] \propto \bar{c}\left(u_i, v_j\right). \tag{3.14}$$

Finally, to find the optimal solution of Eq. (3.20) based on the derived pairwise node alignment probability, we construct the maximum weighted bipartite matching (MWBM)

between $\mathcal{G}_X$ and $\mathcal{G}_Y$, using an efficient implementation of the MWBM algorithm included in the GAIMC library [80]. The overall procedure of the proposed network alignment algorithm is summarized in Algorithm 2.

Algorithm 2: CUFID-align

Data: A pair of PPI networks ($\mathcal{G}_X$ and $\mathcal{G}_Y$) and pairwise node similarity scores
Result: One-to-one alignment A between proteins in different PPI networks
begin
1. $A = \emptyset$ // Empty alignment
2. Construct the transition probability matrix using Eq. (3.9)
3. Compute the node correspondence $\mathbf{C}$ using Eq. (3.11)
4. Compute the transformed node correspondence $\bar{\mathbf{C}}$ using Eqs. (3.12) and (3.13)
5. Construct the maximum weighted bipartite matching between $\mathcal{G}_X$ and $\mathcal{G}_Y$ based on $\bar{\mathbf{C}}$

end

3.2.2 Results

We assessed the performance of CUFID-align based on the IsoBase dataset [42], which includes PPI networks of five different species: *H. sapiens* (human), *M. musculus* (mouse), *D. melanogaster* (fly), *C. elegans* (worm), and *S. cerevisiae* (yeast). PPI networks in IsoBase dataset were constructed by integrating five different databases: BioGRID [43], DIP [44], HPRD [45], MINT [46], and IntAct [47]. In IsoBase, the *H. sapiens* network has 22,369 proteins and 43,757 interactions; the *M. musculus* network has 24,855 proteins and 452 interactions; the *D. melanogaster* network has 14,098 proteins and 26,726 interactions; the *C. elegans* network has 19,756 proteins and 5,853 interactions; and the *S. cerevisiae* network has 6,659 proteins and 38,109 interactions.

We assessed the quality of the predicted network alignment based on the following metrics: correct nodes (CN), specificity (SPE), gene ontology consistency (GOC) scores,

conserved interactions (CI), conserved orthologous interactions (COI), and computation time. Note that CN, SPE, and GOC scores assess the biological significance of the alignment, and CI and COI assess the topological quality of the alignment. If the aligned nodes have the same functional annotation based on the KEGG Orthology (KO) group annotations [48], we considered the node alignment to be correct. CN counts the total number of correctly aligned nodes in a given network alignment. SPE is the relatively ratio of the total number of correctly aligned node pairs to the total number of aligned node pairs.

To further assess the functional consistency of a given network alignment A, we used GOC scores, which can be computed by

$$GOC\left(A\right) = \sum_{\forall(u_i \sim v_j) \in A} goc\left(u_i, v_j\right) = \sum_{\forall(u_i \sim v_j) \in A} \frac{\left|GO\left(u_i\right) \cap GO\left(v_j\right)\right|}{\left|GO\left(u_i\right) \cup GO\left(v_j\right)\right|}, \tag{3.15}$$

where $GO\left(x\right)$ denotes the set of all GO terms assigned to the protein x. To compute the GOC scores, we downloaded the latest version of GO annotations for each species from GO consortium [76] (Feb. 10, 2016 version). We only used GO terms that have experimental evidence (i.e., those that include the codes 'EXP', 'IDA', 'IPI', 'IMP', 'IGI', and 'IEP'). Additionally, similar to [81], we removed every GO term whose information content (IC) was smaller than 2, in order to compute GOC scores based on more informative GO annotations. IC is defined as

$$IC\left(c\right) = -\log_2 \frac{\left|c\right|}{\left|root\left(c\right)\right|}, \tag{3.16}$$

where $\left|c\right|$ is the number of proteins having the particular GO term c, and $\left|root\left(c\right)\right|$ is the total number of proteins under the root GO term of the particular GO term c, where three root GO terms are molecular function (MF, GO:0003674), biological process (BP, GO:0008150), and cellular component (CC, GO:0005575). Note that if at least one pro-

tein in the aligned protein pair does not have a functional annotation such as KO group annotations or GO terms, the aligned protein pair was removed before computing the performance metrics CN, SPE, and GOC scores.

To assess the topological quality of the constructed network alignment, we counted the number of conserved interactions (CI) as follows:

$$\sum_{\forall(u_i,u_j)\in\mathcal{D}} \mathbf{1}\left[(u_i,u_j)\in\mathcal{D}\right]\cdot\mathbf{1}\left[\left(f\left(u_i\right),f\left(v_j\right)\right)\in\mathcal{E}\right], \tag{3.17}$$

where $\mathbf{1}\left[\cdot\right]$ is the indicator function whose value is 1 if the statement in the bracket is true and 0 otherwise, and $f\left(x\right)$ denotes the corresponding protein aligned to the protein x. However, the conserved interactions may not be necessarily be significant from a biological perspective if the aligned proteins connected by the conserved interactions are not orthologous. Considering the large size of typical PPI networks, simply aiming at a network alignment that maximizes the number of conserved interactions may risk overfitting the network topology without clear biological significance, which can be especially problematic when PPI networks are incomplete and noisy. For this reason, in order to assess the biological significance of the topological mapping in a given network alignment, we counted the number of conserved orthologous interactions, which is the number of conserved interactions between orthologous protein pairs (COI). This is given by:

$$\sum_{\forall(u_i,u_j)\in\mathcal{D}} h\left(u_i,u_j\right)\cdot\mathbf{1}\left[(u_i,u_j)\in\mathcal{D}\right]\cdot\mathbf{1}\left[\left(f\left(u_i\right),f\left(u_j\right)\right)\in\mathcal{E}\right], \tag{3.18}$$

where

$$h\left(u_i,u_j\right)=\begin{cases}1, & \text{if } \left[goc\left(u_i,f\left(u_i\right)\right)\cdot goc\left(u_j,f\left(u_j\right)\right)\right]>0\\ 0, & otherwise\end{cases}. \tag{3.19}$$

We compared the performance of CUFID-align against a number of state-of-the-art

alignment methods: IsoRank [12], PINALOG [31], HubAlign [34], SMETANA [13], and SMETANA-CSRW [17] Additionally, to verify the effectiveness of the network-based approach over the conventional approach that uses sequence similarity alone, we compared the various network-based methods and with a method that finds the best mapping between networks solely based on the sequence similarity between the proteins. More specifically, given a network pair, we tried to predict the network alignment by using maximum weighted bipartite matching based on BLAST bit scores. Since both SMETANA and SMETANA-CSRW yield many-to-many mappings by default, we used the parameter $n_{max} = 1$ to obtain one-to-one mappings. Other than this, the default parameters were used in our experiments (i.e., $\alpha = 0.9$ and $\beta = 0.8$). For HubAlign, we used the default parameters (i.e., $\lambda = 0.1$, $d = 10$, and $\alpha = 0.7$). For IsoRank, we set the parameter $\alpha = 0.6$ as recommend in the original paper. For CUFID-align, we set the parameter $\alpha = 0.9$ and $\beta = 90$ percentile of the transformed correspondence score. We performed all experiments on a desktop computer equipped with a 3.2 GHz Intel i5 quad-core processor and 8 GB memory.

We assessed the performance of CUFID-align by predicting the alignment for every pair of PPI networks in the IsoBase dataset. CN and SPE are summarized in Table 3.2. As we can see, CUFID-align and BLAST-MWBM achieve higher CN in all test cases. This means that CUFID-align and BLAST-MWBM can generally align a larger number of proteins that have the same functional annotations (i.e., KEGG orthologous group annotations) than the other state-of-the-art network alignment methods. Interestingly, the sequence-similarity-based approach can identify a larger number of correct nodes (CN) than most of the other network-based approaches. However, as will be shown later, it is clearly biased and the method performs very poorly in terms of the topological quality of the predicted network alignment. CN for PINALOG and HubAlign may depend on the average degrees of the PPI networks (i.e., $|\mathcal{E}|/|\mathcal{V}|$). That is, if one of the PPI networks has

	Yeast – Fly		Yeast – Worm		Yeast – Human		Yeast – Mouse		Fly – Worm	
	CN [1]	SPE [2]	CN	SPE	CN	SPE	CN	SPE	CN	SPE
CUFID-align	1,708	0.748	1,548	0.834	1,330	0.736	1,304	0.794	2,616	0.873
SMETANA-CSRW	1,610	0.757	1,426	0.850	1,224	0.733	1,192	0.802	2,444	0.870
SMETANA	1,530	0.733	1,422	0.843	1,134	0.710	1,182	0.782	2,338	0.852
PINALOG	1,368	0.722	640	0.737	1,100	0.682	76	0.400	672	0.689
HubAlign	1,326	0.681	98	0.170	1,082	0.633	42	0.231	102	0.201
IsoRank	1,414	0.712	650	0.703	1,142	0.702	76	0.369	918	0.818
BLAST-MWBM[3]	1,712	0.776	1,544	0.836	1,334	0.768	1,280	0.792	2,680	0.885
	Fly – Human		Fly – Mouse		Worm – Mouse		Worm – Human		Human – Mouse	
	CN	SPE	CN	SPE	CN	SPE	CN	SPE	CN	SPE
CUFID-align	2,528	0.754	2,364	0.788	1,818	0.807	1,858	0.791	5,178	0.983
SMETANA-CSRW	2,358	0.763	2,146	0.768	1,610	0.811	1,722	0.803	5,002	0.978
SMETANA	2,096	0.706	2,112	0.764	1,578	0.803	1,570	0.780	4,876	0.972
PINALOG	1,172	0.604	118	0.567	66	0.458	482	0.677	282	0.972
HubAlign	354	0.219	34	0.230	24	0.188	32	0.063	144	0.667
IsoRank	1,736	0.725	146	0.566	72	0.456	644	0.793	286	0.979
BLAST-MWBM	2,580	0.766	2,374	0.781	1,824	0.808	1,884	0.794	5,140	0.982

[1] CN: ccorrect nodes.
[2] SPE: specificity.
[2] BLAST-MWBM: maximum weighted bipartite matching of PPI networks only using the BLAST bit score.

Table 3.2: Pairwise alignment results for the IsoBase dataset. Protein functionality is determined based on the KEGG Orthology (KO) group annotations [77] © [2016] BMC.

a much lower average degree, the overall quality of the network alignment may be significantly degraded. Note that human, yeast, and fly PPI networks have relatively higher average degrees, and mouse and worm PPI networks have relatively lower average degrees. Since PINALOG and HubAlign adopt a seed-and-extension approach, the search space for aligning addition protein pairs is restricted to the neighboring nodes of the seed network. Hence, it would be possible that PINALOG and HubAlign may align proteins even though there is no orthologous protein pair in the search space (i.e., the current set of neighboring nodes), which may affect the quality of the final alignment.

When it comes to the specificity of the alignment results, random walk based methods (CUFID-align, SMETANA-CSRW, and SMETANA) achieve relatively higher SPE compared to PINALOG and HubAlign. SPE of HubAlign appears to be more sensitive than the other methods with respect to the average degrees of the PPI networks. CUFID-

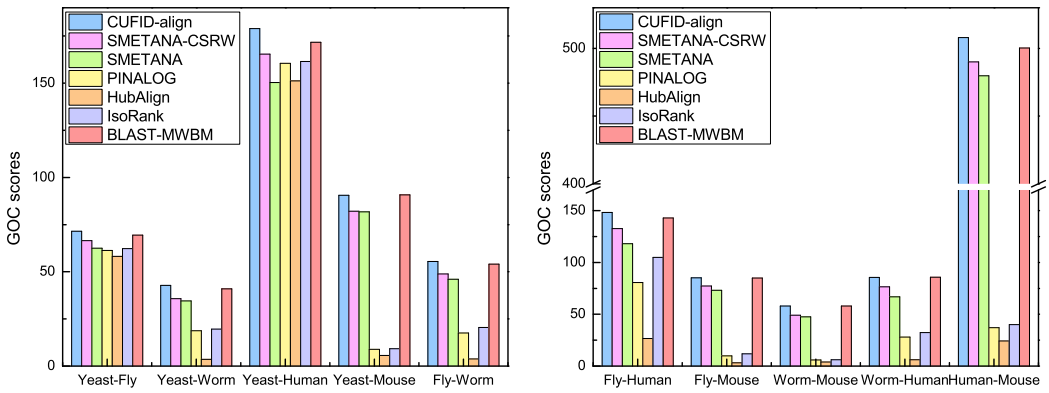


Figure 3.5: GOC scores of various pairwise network alignment algorithms [77] © [2016] BMC.

align, SMETANA-CSRW, and SMETANA achieve similar SPE, often higher than those of PINALOG and HubAlign. This means that CUFID-align can in general more accurately align protein pairs that have the same functional annotations compared to PINALOG and HubAlign.

Since proteins can have multiple functions, we further evaluated the functional consistency of the alignment results based on the GOC scores, where higher GOC scores indicate that the obtained alignments are functionally more coherent. As we can see in Figure 3.5, CUFID-align achieves higher GOC scores than the other compared algorithms in all test cases. Again, if the network pairs have higher average degrees, PINALOG and HubAlign show comparable GOC scores. However, probably due to the restricted search space of the seed-and-extend approach, GOC scores of PINALOG and HubAlign tend to be smaller than the other methods when the average degree of one of the PPI networks is relatively smaller than that of the other. In comparison, CUFID-align is more robust to the change of topological properties such as the average degrees of the PPI networks to be aligned.

	Yeast – Fly	Yeast – Worm	Yeast – Human	Yeast – Mouse	Fly – Worm
CUFID-align	1,721	486	3,421	56	347
SMETANA-CSRW	337	110	2,468	31	107
SMETANA	504	171	2,377	37	116
PINALOG	2,982	1,000	6,231	225	666
HubAlign	836	4,013	2,659	545	3,276
IsoRank	1,436	764	3,165	176	558
BLAST-MWBM	246	89	1,317	14	70
	Fly – Human	Fly – Mouse	Worm – Mouse	Worm – Human	Human – Mouse
CUFID-align	1,547	59	18	459	318
SMETANA-CSRW	710	41	8	198	336
SMETANA	965	50	16	283	337
PINALOG	2,730	88	47	917	358
HubAlign	9,317	491	459	3,743	532
IsoRank	1,471	106	130	569	350
BLAST-MWBM	441	12	2	138	253

Table 3.3: Number of conserved interactions (CI) obtained by different network alignment algorithms [77] © [2016] BMC.

The above results show that CUFID-align can accurately predict matching proteins in different species that have similar functionalities, according to the functional annotations of proteins that are currently available. The results also imply that the proposed algorithm may provide a useful tool for predicting the functions of unknown proteins in less studied species through network alignment with species that have been better studied.

Next, to assess the topological quality of the network alignment results, we compared the number of conserved interactions (CI) predicted by different methods. Table 3.3 shows the CI for all compared methods. As we can see in Table 3.3, CUFID-align can identify a larger number of conserved interactions than SMETANA-CSRW and SMETANA, but it is smaller than HubAlign and PINALOG. In fact, our results show that PINALOG and HubAlign outperform the other methods in terms of CI. One interesting observation is that although PINALOG and HubAlign can identify a large number of conserved interactions compared to CUFID-align, GOC scores for PINALOG and HubAlign are much smaller than CUFID-align as shown in Figure 3.5. Since both PINALOG and HubAlign adopt

	Yeast – Fly	Yeast – Worm	Yeast – Human	Yeast – Mouse	Fly – Worm
CUFID-align	91	10	743	5	3
SMETANA-CSRW	91	8	749	6	2
SMETANA	86	10	705	8	4
PINALOG	129	15	970	19	4
HubAlign	57	2	634	15	5
IsoRank	94	11	741	10	4
BLAST-MWBM	74	8	556	4	2
	Fly – Human	Fly – Mouse	Worm – Mouse	Worm – Human	Human – Mouse
CUFID-align	202	13	1	21	111
SMETANA-CSRW	196	10	1	26	139
SMETANA	230	14	1	26	123
PINALOG	180	22	2	27	134
HubAlign	67	15	4	5	98
IsoRank	185	17	1	18	142
BLAST-MWBM	112	6	0	15	94

Table 3.4: Number of conserved orthologous interactions (COI) obtained by different network alignment algorithms [77] © [2016] BMC.

a seed-and-extension approach, the algorithms only align protein nodes if they are connected to the seed network alignment. PINALOG and HubAlign may have a higher risk for overfitting the prediction outcomes to the topological structure of the PPI networks compared to the other methods, and they may not as effectively deal with the inserted or deleted nodes as the random walk based methods, which may be problematic when handling PPI networks that are incomplete and/or contain many errors (e.g., many false positive interactions). As the GOC scores were low for PINALOG and HubAlign, despite the high CI they attained, we wanted to further evaluate the biological significance of the conserved interactions in the predicted network alignment results. For this purpose, we counted the number of conserved interactions between orthologous protein pairs. Table 3.4 summarizes the number of conserved orthologous interactions (COI) predicted by different algorithms. Note that, for this experiment, we did not consider the alignment of networks whose average degrees differ significantly, since there will be only a small number of conserved orthologous interactions in such cases. Table 3.4 shows that CUFID-

align achieves comparable or higher COI compared to PINALOG and HubAlign except for the alignment between the yeast and human PPI networks.

We also compared the network-based approaches with the sequence-similarity-based approach. As we can see in Table 3.2 and Figure 3.5, a simple sequence-similarity-based approach can construct network alignments with high functional coherence, and that the node similarity score may provide useful guidelines for identifying orthologous proteins. However, these results should be taken with a grain of salt, since they are likely due to the fact that the current functional annotations of proteins are often based on sequence similarity between proteins. As shown in Table 3.3 and Table 3.4, BLAST-MWBM – which uses BLAST bit score and MWBM without using any network information – can identify a much smaller number of CIs and COIs compared to the network-based methods. These results imply that strong dependence on sequence similarity for constructing a network alignment has the potential risk of getting biased results that may fail to capture important protein interactions that are conserved across different species, which may be critical in deciphering the underlying cellular mechanisms that involve those interactions. In contrast, network-based methods, including CUFID-align, that incorporate topological information for constructing network alignments can make accurate and balanced predictions that identify both orthologous proteins as well as conserved interactions. Our results clearly show the importance of effective integration of node similarity and topological similarity for effective comparative analysis of PPI networks.

Finally, Table 3.5 shows the computation time for each method. As we can see in this table, CUFID-align needs the least computation time among all compared methods in most test cases. Computation time of HubAlign largely depends on the average degrees of the PPI networks because HubAlign takes a seed-and-extension approach, whose search space is strongly affected by the average degrees of the PPI networks to be aligned. Computation time of SMETANA-CSRW is proportional to the size of the PPI networks. The bottleneck

	Yeast – Fly	Yeast – Worm	Yeast – Human	Yeast – Mouse	Fly – Worm
CUFID-align	6.22	4.79	11.22	5.70	12.88
SMETANA-CSRW	243.64	163.24	448.29	435.94	3,002.20
SMETANA	6.65	5.81	11.47	9.12	26.11
HubAlign	451.24	75.67	571.23	5.30	55.87
PINALOG	997.85	1,654.66	1,984.03	2,202.15	2,141.00
IsoRank	1,737.07	369.52	3401.29	64.47	181.55
	Fly – Human	Fly – Mouse	Worm – Mouse	Worm – Human	Human – Mouse
CUFID-align	18.93	18.38	25.71	28.36	68.59
SMETANA-CSRW	6,104.70	6,420.80	6,383.70	6,084.10	4,9185.00
SMETANA	63.43	60.85	53.24	56.28	454.11
HubAlign	532.31	4.92	1.91	84.45	8.99
PINALOG	3,127.35	1,611.39	101.86	6,764.56	4,864.16
IsoRank	1433.27	37.77	16.92	326.79	77.64

Table 3.5: CPU time of the tested network alignment algorithms (in seconds) [77] © [2016] BMC.

for SMETANA-CSRW is the step for constructing the transition probability matrix of the context-sensitive random walker (CSRW), whose computation time is proportional to the size of the two PPI networks that need to be aligned. PINALOG requires a relatively long computation time compared to other methods in most cases, as shown in Table 3.5.

In this work, we have focused on the steady-state network flow approach and its application to the pairwise network alignment problem. However, the problem of multiple network alignment has been gaining wide interest in the research community and its practical importance has been increasing as the number of available PPI networks for different species continue to increase. Although it is beyond the scope of this subchapter, we expect the extension of CUFID-align for multiple network alignment will be relatively straightforward. First of all, to this aim, we can modify the transition probability matrix in Eq. (3.9) by concatenating the normalized adjacency matrices and node similarity score matrices for the multiple PPI networks to be aligned. Following the construction of this extended transition probability matrix, the steps for computing the node correspondence scores – shown in Eq. (3.11) and Eq. (3.13) – can be modified by constructing diagonal matrices

and inserting corresponding the matrices into the diagonal terms. The extended version of CUFID-align for multiple PPI network alignment is expected to have distinctive advantages over other existing multiple PPI network alignment algorithms. First, it may be able to estimate the 'global' node correspondence scores more accurately. Currently, most multiple PPI network alignment algorithms estimate the node correspondence scores for every PPI network pair in the interest of computational complexity. The estimated pairwise node correspondence scores are later updated based on additional transformations to make them more suitable for multiple network alignment. However, considering that the ultimate goal is in constructing the alignment of multiple networks, it would be preferable to estimate the node correspondence scores (or equivalently, node alignment probabilities) $\Pr[u_i \sim v_j|\mathbf{G}]$ considering all networks, rather than just estimating $\Pr[u_i \sim v_j|\mathcal{G}_X, \mathcal{G}_Y]$ based on the given network pair, where $u_i \in \mathcal{G}_X$, $v_j \in \mathcal{G}_Y$, and $\mathbf{G}$ is the set of all PPI networks including $\mathcal{G}_X$ and $\mathcal{G}_Y$. Since the aforementioned extension of CUFID-align estimates the node correspondence scores based on an integrated network that combines all networks in $\mathbf{G}$, it has the potential to accurately compute the posterior node-to-node alignment probability given all the networks. Computation of such 'global' node correspondence score may lead to improved multiple network alignment results. Second, the extended version of CUFID-align will still be computationally very efficient, as most steps in CUFID-align only require simple matrix operations even if extended to multiple networks. Finally, the extended approach will require relatively low computational resources (especially, in terms of memory). For example, suppose that there are N PPI networks, where the number of nodes in the i-th network G_i is V_i. To align the N PPI networks, IsoRankN will need the pairwise node correspondence scores for each of the $\binom{N}{2}$ network pairs, where for each pair, the algorithm will need to construct a $|V_i \cdot V_j| \times |V_i \cdot V_j|$ dimensional matrix. However, CUFID-align can compute the global node correspondence

scores by constructing a single $\left|\sum_{i=1}^{N} V_i\right| \times \left|\sum_{i=1}^{N} V_i\right|$ dimensional matrix. We are currently working on extending CUFID-align for multiple network alignment.

3.2.3 Conclusions

In this subchapter, we proposed CUFID-align, a novel network alignment algorithm based on the concept of steady-state network flow of a Markov random walk model on an integrated network. Given a pair of PPI networks, CUFID-align constructs an integrated network and a Markov random walk model on the resulting network such that the steady-state network flow between a pair of nodes in different PPI networks increases when the nodes have higher pairwise node similarity (typically measured based on sequence similarity) and topological similarity. For this purpose, the Markov random walk model is designed to make more frequent transitions between protein nodes that have higher overall similarity, thereby making the steady-state network flow – which reflects the long-run behavior of the random walker – an effective measure of the correspondence between nodes that belong to different networks. As we have shown in our performance assessment results using real PPI networks in the IsoBase database, CUFID-align can accurately align proteins with identical functional annotations at a relatively low computational cost. Our results show that CUFID-align may provide an effective means of computationally annotating the functions of proteins through comparative analysis of PPI networks. The source code and datasets can be downloaded from http://www.ece.tamu.edu/∼bjyoon/CUFID

3.3 Network querying through the CUFID model

In this subchapter, we propose a novel network querying algorithm to identify the conserved subnetwork in the target PPI network, considering both molecular and topological/structural properties. Proposed network querying algorithm addresses two major challenges in a network querying problem: complexity problem and structural variation of the conserved networks. To tackle the complexity problem, we adopt the CUFID

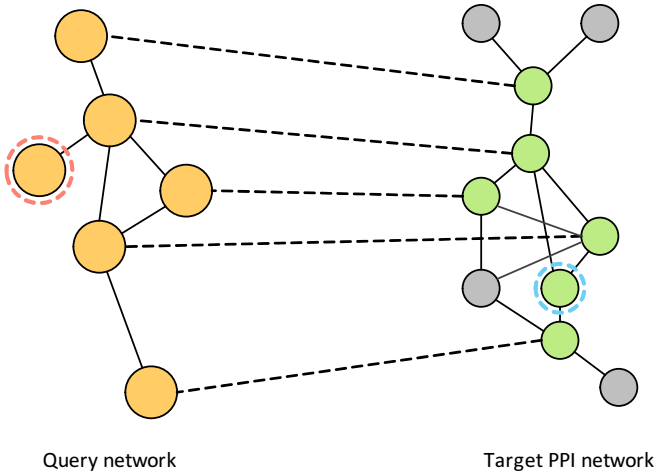


Figure 3.6: Illustration of a typical network querying problem. The node marked in red in the query network is deleted in the target network, and the node marked in blue is not present in the query but inserted in the target network. Note that the terms insertion/deletion are relative, and an inserted node in one network can be viewed as a deleted node in the other network.

(**C**omparative network analysis **U**sing the steady-state network **F**low to **ID**entify orthologous proteins) model [77]. Typically, as a network querying algorithm requires to examine a large-scale target network in order to find the best matching subnetwork that is similar to the given small query network, computationally efficient method to scan a large searching space is necessary. Additionally, we utilize a seed-and-extension approach in order to deal with structural variations of conserved networks. As illustrated in Figure 3.6, although the conserved subnetwork performs the similar biological functions, there are inserted and deleted nodes and edges, and these structural changes make it difficult to solve a network querying problem through a classical bipartite matching problem.

In the proposed method, we first estimate the node-to-node correspondence (i.e., biological relevance or matching probability) between query and target networks. Then, based on the estimated node correspondence scores, we select the largely connected seed network through a maximum weighted bipartite matching algorithm. Next, we iteratively

extend the seed network by including the node that meets the following two conditions: 1) larger association probability and 2) minimizing a conductance of the extending seed network from the rest of the target network. The association probability could estimate the frequency of interactions between the nodes in the extending seed network and the neighboring nodes of the seed network. Including the neighboring node with more interactions to the nodes in the seed network can be advantageous to lead functionally consistent querying results because proteins having a direct interactions have more chances to share and perform the similar biological functions [65]. Note that since we only consider the nodes in the target network as a candidate for a network extension, the searching space for the network extension is limited to the nodes in the target network. In the extension step, we list all candidate nodes based on the association probability and select the winning node that can maximally minimize the conductance of the extending seed network. This rule selects the node having a higher probability to frequently interact with the nodes in the extending seed network as well as rarely interact to the rest of the network. Finally, after completing the extension steps, we remove less relevant nodes in the fully extended network based on the personalized PageRank vector [82] in order to increase the functional consistency of the querying result.

3.3.1 Methods

Suppose that we have a query network and it can be represented as a graph $\mathcal{G}_{\mathcal{Q}} = (\mathcal{V}_{\mathcal{Q}}, \mathcal{E}_{\mathcal{Q}})$. For example, a node $v_i \in \mathcal{V}_{\mathcal{Q}}$ indicates a protein in the query network and an edge $e_{i,j} \in \mathcal{E}_{\mathcal{Q}}$ represents the interaction or binding between protein v_i and protein v_j. Similarly, suppose that a target network is given and represented by a graph $\mathcal{G}_{\mathcal{T}} = (\mathcal{V}_{\mathcal{T}}, \mathcal{E}_{\mathcal{T}})$. We assume that a pairwise node similarity score $s(v_q, v_t)$ is given for $\forall v_q \in \mathcal{V}_{\mathcal{Q}}$ and $\forall v_t \in \mathcal{V}_{\mathcal{T}}$, where it is proportional to the molecular level similarity of two proteins (v_q, v_t). In this study, we considered protein-protein interactions (PPI) networks, and we

utilized BLAST bit scores as pairwise node similarity scores but other types of similarity measurements or their combination can be utilized. Generally, in a network querying problem, the size of the target network is significantly larger than the size of the query network, i.e., $|\mathcal{V}_{\mathcal{Q}}| \ll |\mathcal{V}_{\mathcal{T}}|$, where the size of the network is the number of nodes in the network.

The goal of network querying is to identify the conserved subnetworks that are expected to perform the same or similar biological functions to the given query network. Hence, the network querying problem is formulated as the following optimization problem:

$$\hat{\mathcal{G}}^*_{\mathcal{T}} = \underset{\forall \hat{\mathcal{G}}_{\mathcal{T}} \in \mathbf{G_T}}{\arg\max} f\left(\hat{\mathcal{G}}_{\mathcal{T}}, \mathcal{G}_{\mathcal{Q}}\right), \tag{3.20}$$

where $\mathbf{G_T}$ is a feasible set of all subnetworks in the target PPI network, and $f\left(\mathcal{G}_{\mathcal{X}}, \mathcal{G}_{\mathcal{Y}}\right)$ is the function that can quantitatively estimate the functional similarity or biological relevance of two biological networks $(\mathcal{G}_{\mathcal{X}}, \mathcal{G}_{\mathcal{Y}})$.

Network querying can be viewed as a subgraph isomorphism problem, where it determines whether one graph (query network) is isomorphic to the subgraph of the target graph (target PPI network). Solving the network querying problem as the subgraph isomorphism problem, considering possible node (or edge) insertion and deletion in each network, is NP-complete [64] Additionally, identifying the conserved subnetwork in the target network is practically difficult because of the following reasons: 1) it is not easy to compute node correspondence scores as the scale of the biological network is very large (i.e., scalability problem), 2) quantitatively estimating the functional similarity $f\left(\mathcal{G}_{\mathcal{X}}, \mathcal{G}_{\mathcal{Y}}\right)$ of two biological networks is difficult, and 3) we have no prior knowledge whether the conserved subnetwork is larger or smaller than the query network because of the structural variations in biological networks. That is, we have no prior knowledge for the exact number of inserted/deleted nodes.

To overcome these challenges, we propose a heuristic network querying algorithm based on the CUFID (**C**omparative network analysis **U**sing the steady-state network **F**low to **ID**entify orthologous proteins) model [77] and a seed-and-extension approach. In the proposed network querying algorithm called CUFID-query, we first compute the node-to-node correspondence scores through the CUFID model. The CUFID model can effectively deal with the complexity problem as it can estimate the node correspondences for large-scale networks with a low computational cost. Based on the intuition that two proteins in different networks would be an orthologous pair if they have a high molecular similarity as well as the similar interaction patterns to its neighboring nodes [24, 12], the CUFID model can effectively estimate a biological relevance between the nodes in the query and target network by integrating the molecular and topological similarities in a balanced manner. After computing the node correspondence scores, we induce a seed network using the seed nodes that can be identified through a maximum weighted bipartite matching algorithm. Note that the seed network $\mathcal{G}_{\mathcal{S}} = (\mathcal{V}_{\mathcal{S}}, \mathcal{E}_{\mathcal{S}})$ is always smaller than the query network (i.e., $|\mathcal{V}_{\mathcal{S}}| \leq |\mathcal{V}_{\mathcal{Q}}|$). Then, we iteratively extend the seed network using a probabilistic model, where it is designed to select the nodes that can have more interactions to the nodes in the seed network and minimize the conductance of the extending seed network from the rest of the target network. Finally, we removed less relevant nodes based on the personalized PageRank vector. Due to the structural variations between conserved functional modules, solving a subgraph isomorphic problem may not the best way to find the solution to a network querying problem in a practical point of view, and a seed-and-extension approach could be a reasonable alternative. However, since the approach is not the optimal and less relevant nodes could be included in the network extension steps, effective post-processing to remove less relevant nodes can increase the accuracy of a querying result.

First, to estimate the node correspondence through the CUFID model, we construct the integrated network by combining networks to be compared. Specifically, as shown

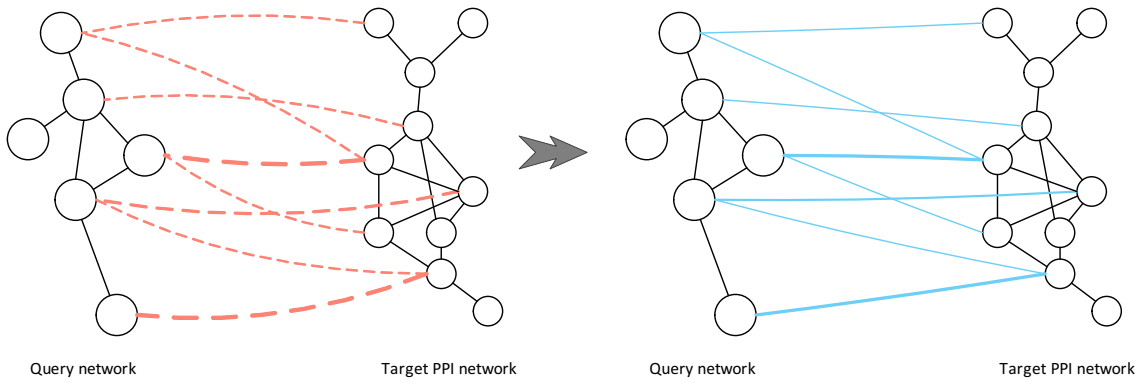


Figure 3.7: Illustration for constructing the integrated network by combining the query and target networks. Dotted lines indicate positive node similarity scores between pairs of nodes, where the thickness of each line is proportional to the similarity score. We insert a pseudo-edge between a node in the query network and a node in the target network if the corresponding proteins have a positive node similarity score.

in Figure 3.7, we insert the pseudo-edges connecting nodes in the query and target networks if their pairwise node similarity score is greater than a threshold s_t. That is, the integrated network can be represented as a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ denotes the union of the nodes in the query and target networks (i.e. $\mathcal{V} = \{\mathcal{V}_\mathcal{Q}, \mathcal{V}_\mathcal{T}\}$); $\mathcal{E}$ is the union of the edges in the two networks; and the inserted pseudo-edges such that $\mathcal{E}_\mathcal{P} = \{e_{i,j} | v_i \in \mathcal{V}_\mathcal{Q}, v_j \in \mathcal{V}_\mathcal{T}, s(v_i, v_j) > s_t\}$, (i.e., $\mathcal{E} = \{\mathcal{E}_\mathcal{Q}, \mathcal{E}_\mathcal{T}, \mathcal{E}_\mathcal{P}\}$). Then, we allow a random walker to transit within and across the networks to be compared.

If the random walker performs a random movement over the edges representing protein-protein interactions, it can move to its neighboring nodes belonging to the same PPI network. That is, at the current position of the random walker, it can transit to its neighboring nodes only if they are connected through the edges either $\mathcal{E}_\mathcal{Q}$ or $\mathcal{E}_\mathcal{T}$ indicating the protein-protein interactions. The transition probability for the random walk within either the query or target PPI networks is given by

$$\mathbf{P_Q} = \mathbf{D_Q^{-1}} \cdot \mathbf{A_Q} \text{ and } \mathbf{P_T} = \mathbf{D_T^{-1}} \cdot \mathbf{A_T}, \tag{3.21}$$

where $\mathbf{A_Q}$ (or $\mathbf{A_T}$) is an adjacency matrix of the query (or target) network and $\mathbf{D_Q}$ (or $\mathbf{D_T}$) is a diagonal matrix such that $\mathbf{D_Q} = \sum_{\forall j} \mathbf{A_Q}[i,j]$ (or $\mathbf{D_T} = \sum_{\forall j} \mathbf{A_T}[i,j]$).

The random walker can also transit across the query and target networks through the pseudo-edges $\mathcal{E}_\mathcal{P}$. When the random walker transits from the query network to the target PPI network, the transition probability of the random walker for this event is given by

$$\mathbf{P_{Q \to T}} = \mathbf{D_S^{-1}} \cdot \mathbf{S}, \tag{3.22}$$

where $\mathbf{S}$ is a $|\mathcal{V}_\mathcal{Q}| \times |\mathcal{V}_\mathcal{T}|$ dimensional matrix for the pairwise node similarity score such that $S[i,j] = s(v_i, v_j), \forall v_i \in \mathcal{V}_\mathcal{Q}, \forall v_j \in \mathcal{V}_\mathcal{T}$, and $\mathbf{D_S}$ is a $|\mathcal{V}_\mathcal{Q}| \times |\mathcal{V}_\mathcal{Q}|$ dimensional diagonal matrix such that $\mathbf{D_S} = \sum_{\forall j} \mathbf{S}[i,j]$.

Similarly, if the random walker jumps from the target PPI network to the query network, the transition probability is given by

$$\mathbf{P_{T \to Q}} = \mathbf{S}^\mathrm{T} \cdot \mathbf{D_{S^T}^{-1}}. \tag{3.23}$$

We can construct the overall transition probability matrix for the random walker over the integrated network $\mathcal{G}$ by concatenating the above probability matrices as follows:

$$\mathbf{P} = \begin{bmatrix} \mathbf{P_Q} & \mathbf{P_{Q \to T}} \\ \mathbf{P_{T \to Q}} & \mathbf{P_T} \end{bmatrix}, \tag{3.24}$$

with necessary normalization to make the matrix $\mathbf{P}$ a stochastic matrix. We can compute the corresponding steady-state probability π of the random walker, where it is equivalent to the expected time of the random walker staying at the particular node in long term. Since real PPI networks have generally sparse interactions, the steady-state probability can be easily obtained through a power method [77].

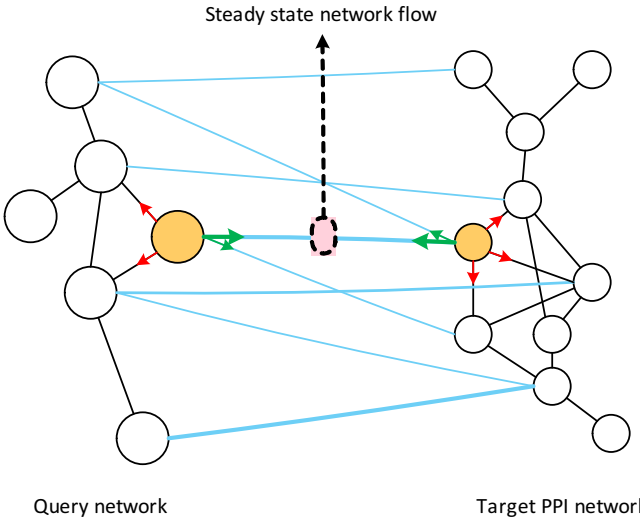


Figure 3.8: Estimating the steady-state network flow based on the CUFID model. Red arrows indicate the random walk within the query or the target network, while the green arrows represent the random walk across two networks. The correspondence between two nodes – one in the query network and the other in the target network – can be estimated by measuring the steady-state network flow through the pseudo-edges connecting the nodes.

Finally, as shown in Figure 3.8, the node-to-node correspondence between the query and target networks can be obtained by estimating the steady-state network flow (i.e., traversal of the random walker) across the pseudo-edges connecting the nodes in the query and target networks, which is given by

$$\mathbf{C} = \bar{\pi}_{\mathcal{Q}} \cdot \mathbf{P}_{\mathbf{Q}\to\mathbf{T}} + \mathbf{P}^{\mathrm{T}}_{\mathbf{T}\to\mathbf{Q}} \cdot \bar{\pi}_{\mathcal{T}}, \tag{3.25}$$

where $\bar{\pi}_{\mathcal{Q}}$ is a $|\mathcal{V}_{\mathcal{Q}}| \times |\mathcal{V}_{\mathcal{Q}}|$ dimensional diagonal matrix such that $\bar{\pi}_{\mathcal{Q}}[i,i] = \pi(v_i), \forall v_i \in \mathcal{V}_{\mathcal{Q}}$ and $\bar{\pi}_{\mathcal{T}}$ is a $|\mathcal{V}_{\mathcal{T}}| \times |\mathcal{V}_{\mathcal{T}}|$ dimensional diagonal matrix such that $\bar{\pi}_{\mathcal{T}}[i,i] = \pi(v_j), \forall v_j \in \mathcal{V}_{\mathcal{T}}$.

The proposed network querying algorithm – CUFID-query – has three main steps. First, we compute the node-to-node correspondence between the query and target net-

works through the CUFID model. Next, we select the seed network (i.e., high scoring subnetwork) and iteratively extend the seed network in the target network until it meets the stop conditions. Finally, we remove less relevant nodes based on the personalized PageRank (PPR) vector of the induced network.

Once we obtained the node correspondence between the query and target networks through Eq. (3.25), we select the seed nodes by maximum weighted bipartite matching implemented in the MATLAB GAIMC toolbox [80]. Then, we construct the induced seed network based on the selected seed nodes (i.e., the matching nodes in the target network corresponding to the nodes in the query network). If the induced network is disconnected, we will use the largest connected network as the seed network. If all the seed nodes are disconnected, we will select a single node with the maximum correspondence score as the seed.

Next, we iteratively extend the seed network by adding a node based on the association probability and conductance minimization criteria. To this aim, we define the association probability as the likelihood that the random walker starting from a node in the seed network will return to the seed network within 2 hops by passing through the neighboring nodes of the seed network. When a neighboring node has a higher association probability, it can have more interactions to the seed network and it is more likely to share the similar biological functions to the nodes in the seed network because interacting proteins tend to share the similar biological functions [65]. To compute the association probability, we compute the initial steady-state probability $\pi_{\mathcal{S}}$ of the random walker for the seed network. Given the seed network $\mathcal{G}_{\mathcal{S}} = (\mathcal{V}_{\mathcal{S}}, \mathcal{E}_{\mathcal{S}})$, the steady-state probability for the node v_i in the seed network is given by [83]

$$\pi_{\mathcal{S}}(v_i) = \frac{d(v_i)}{\sum_{v_i \in \mathcal{G}_{\mathcal{S}}} d(v_i)}, \tag{3.26}$$

where $d(v_i)$ is the degree of the node v_i.

Then, for each neighboring node v_n such that $v_n = \{v_x | v_x \in \mathcal{N}(v_i), \forall v_i \in \mathcal{V}_\mathcal{S}\}$, the probability of the random walker jumps to the neighboring node v_n from any node in the seed network is given by

$$\mathrm{P}_1(v_n) = \sum_{v_i \in \{\mathcal{V}_\mathcal{S} \cap \mathcal{N}(v_n)\}} \frac{\pi_\mathcal{S}(v_i)}{d(v_i)}, \tag{3.27}$$

where $\mathcal{N}(v_x)$ is the neighboring nodes of the node v_x.

Finally, the association probability for the neighboring node v_n is given by

$$\mathrm{P}_2(v_n) = \mathrm{P}_1(v_n) \cdot \frac{r(v_n)}{d(v_n)}, \tag{3.28}$$

where $r(v_n)$ is the number of edges connecting v_n and the nodes in the seed network (i.e., $|\{e_{n,j} | v_n, v_j \in \mathcal{G}_\mathcal{S}\}|$).

We select top K candidate nodes having the highest association probability, and select the finalist to be included to extend the seed network based on the conductance minimization criterion. Conductance minimization criterion has been widely utilized in the non-comparative network analysis algorithms [84, 83] because proteins in the functional module typically tend to be densely connected to each other while sparsely connected to the rest of the network. Given a subnetwork $\mathcal{G}_\mathcal{S}$ in the target network (i.e., $\mathcal{G}_\mathcal{S} \subset \mathcal{G}_\mathcal{T}$), the conductance of the subnetwork is given by [82]

$$\varphi(\mathcal{G}_\mathcal{S}) = \frac{|\{e_{i,j} | v_i \in \mathcal{V}_\mathcal{Q}, v_j \in \mathcal{V} \backslash \mathcal{V}_\mathcal{Q}\}|}{\min(vol(\mathcal{G}_\mathcal{S}), 2m - vol(\mathcal{G}_\mathcal{S}))}, \tag{3.29}$$

where m is the number of undirected edges and $vol(\mathcal{G}) = \sum_{v_i \in \mathcal{G}} d(v_i)$. Since the conserved subnetwork is typically much smaller than the target network (i.e., $\mathcal{G}_\mathcal{S} \ll \mathcal{G}_\mathcal{T}$), Eq. (3.29)

becomes

$$\begin{aligned}\varphi\left(\mathcal{G}_{\mathcal{S}}\right) &= \frac{\left|\{e_{i,j}|v_i \in \mathcal{V}_{\mathcal{Q}}, v_j \in \mathcal{V}\backslash\mathcal{V}_{\mathcal{Q}}\}\right|}{vol\left(\mathcal{G}_{\mathcal{S}}\right)} \\ &= \frac{\left|\{e_{i,j}|v_i \in \mathcal{V}_{\mathcal{Q}}, v_j \in \mathcal{V}\backslash\mathcal{V}_{\mathcal{Q}}\}\right|}{\left|\{e_{i,j}|v_i \in \mathcal{V}_{\mathcal{Q}}, v_j \in \mathcal{V}_{\mathcal{Q}}\}\right|}.\end{aligned} \tag{3.30}$$

In the extension steps, we first select the top 20 nodes with the highest association probability, and we finally select one node that can maximally minimize the conductance of the seed network. We iteratively extend the seed network until either one of the following stopping conditions is satisfied: 1) the size of the extending seed network exceeds the limits; 2) there are no neighboring nodes that can decrease the conductance of the extending network more than 10 percent.

Once the seed network is fully grown, we finally refine the extended seed network by removing the less relevant nodes based on the personalized PageRank (PPR) vector. For this purpose, we construct the induced network $\mathcal{G}_{\mathcal{I}}$ based on the extended seed network and its neighboring nodes (i.e., $\mathcal{G}_{\mathcal{I}} = (\mathcal{V}_{\mathcal{I}}, \mathcal{E}_{\mathcal{I}})$, where $\mathcal{V}_{\mathcal{I}} = \{\mathcal{V}_{\mathcal{S}}, \mathcal{N}\left(\mathcal{V}_{\mathcal{S}}\right)\}$ and $\mathcal{E}_{\mathcal{I}} = \{\mathcal{E}_{\mathcal{S}}, \mathcal{E}_{\mathcal{A}}\}$ such that $\mathcal{E}_{\mathcal{A}} = \{e_{i,j}|v_i \in \mathcal{V}_{\mathcal{S}}, v_j \in \mathcal{N}\left(\mathcal{V}_{\mathcal{S}}\right)\}$). Then, we compute the PPR vector for the induced network $\mathcal{G}_{\mathcal{I}}$. The standard PPR vector $\mathbf{r}$ is a unique solution of the following equation: [82]

$$\mathbf{r} = \alpha \cdot \mathbf{s} + (1 - \alpha) \cdot \mathbf{r} \cdot \mathbf{M}, \tag{3.31}$$

where α is a teleportation constant and we set α as 0.5, $\mathbf{M}$ is the normalized adjacency matrix of the induced network $\mathcal{G}_{\mathcal{I}}$ and $\mathbf{s}$ is a preference vector. We set the preference vector $\mathbf{s}$ as follows:

$$s\left(v_i\right) = \begin{cases} 1/|\mathcal{V}_{\mathcal{S}}|, & v_i \in \mathcal{V}_{\mathcal{S}} \\ 0, & otherwise. \end{cases} \tag{3.32}$$

Once we obtain the PPR vector for the induced network $\mathcal{G}_{\mathcal{I}}$, we iteratively select the nodes with the highest PPR vector values until the cumulative sum becomes 0.5. In this

Algorithm 3: CUFID-query

Data: Query $\mathcal{G}_\mathcal{Q}$ and target $\mathcal{G}_\mathcal{T}$ networks and pairwise node similarity scores
Result: List of nodes in the querying results
begin

Compute the node correspondence score $\mathbf{C}$ using Eq. (3.25)
Select seed nodes using a maximum weighted bipartite matching algorithm
Identifying the seed network $\mathcal{G}_\mathcal{S} = (\mathcal{V}_S, \mathcal{E}_\mathcal{S})$ by finding the largest connected network based on the seed nodes
Set $\varphi_{old} = \infty$
while $|\mathcal{V}_\mathcal{S}| \leq 2 \cdot |\mathcal{V}_\mathcal{Q}|$ or $\varphi_{new} \leq \beta \cdot \varphi_{old}$ **do**
    Find the set $\mathcal{K}$ of top K candidate nodes based on 2-hop returning probability
    Compute the conductance φ_t for the induced network $\{\mathcal{V}_\mathcal{S} \cup v_t\}$ for each v_t, $\forall v_t \in \mathcal{K}$
    $v_{t^*} = \arg\min_t \varphi_t$
    Set $\varphi_{new} = \varphi_{t^*}$
    Check stopping conditions
    Update $\mathcal{G}_\mathcal{S}$ such that $\mathcal{V}_\mathcal{S} = \{\mathcal{V}_\mathcal{S} \cup v_{t^*}\}$ and $\mathcal{E}_\mathcal{S} = \{\mathcal{E}_\mathcal{S} \cup e_{i,j} | \forall v_i \in \mathcal{V}_\mathcal{S}, j = v_{t^*}\}$
    Set $\varphi_{old} = \varphi_{t^*}$
end
Compute personalized PageRank vector using Eq. (3.31)
Remove less relevant nodes based on PPR vector and return the largest connected network

end

pruning step, it would be possible that the nodes in the extended seed network could be removed and other neighboring nodes would be included in the final querying results. Note that this pruning process could make the querying results disconnected. If the identified network is fragmented by the pruning step, CUFID-query only returns the largest connected network as the querying results. The steps of CUFID-query are summarized in Algorithm 3. We briefly compare SEQUOIA [18] and CUFID-query as they both adopt similar seed-and-extension approaches. One important difference between the seed exten-

sion steps in the two algorithms is that SEQUOIA extends the intermediate networks only based on the conductance minimization principle while CUFID-query adopts the conductance minimization principle and simultaneously uses the association probability to select additional nodes. Furthermore, in the post-processing step, SEQOIA only removes irrelevant nodes in the extended seed network, but CUFID-query can recruit new nodes that are originally not included the extended seed network by utilizing the PPR vector of the induced network $\mathcal{G}_{\mathcal{I}}$.

3.3.2 Results

To assess the performance of CUFID-query, we performed experiments based on the known biological complexes and real-world PPI networks for three species: *H. sapiens* (human), *S. cerevisiae* (yeast), and *D. melanogaster* (fly). We obtained target PPI networks from STRING v10 [85]. Then, we extracted the protein-protein interactions classified as a 'binding' (direct interaction) and removed the protein-protein interactions without an experimental validation. We further removed protein-protein interactions with the confidence score less than 400 that indicate a medium level confidence. After the aforementioned preprocessing, the human PPI network includes 12,049 proteins and 95,209 interactions, the mouse PPI network includes 10,428 proteins and 112,541 interactions, and the yeast PPI network includes 5,726 proteins and 88,308 interactions. To obtain the pairwise node similarity score for each network pair, we computed BLAST bit scores between amino acids sequences for each protein pair through BLAST version 2.3. Note that the amino acid sequences for each species were obtained from STRING v10.

We obtained the known biological complexes for human and mouse from CORUM [70], and known biological complexes for yeast are obtained from SGD [71] (accessed at Feb. 1 2017). Then, we extracted the connected networks with the size of 4 to 25. We obtained overall 1,242 test cases, where the 371 human complexes were queried against the mouse

PPI network, the 349 human complexes were queried against the yeast PPI network, the 64 mouse complexes were queried against the human PPI network, the 54 mouse complexes were queried against the yeast PPI network, the 201 yeast complexes were queried against the human PPI network, the 203 yeast complexes were queried against the mouse PPI network.

To assess the biological significance of the querying results, we performed a GO enrichment test for the querying results. To this aim, we downloaded the GO ontology and annotation files for each species from Gene Ontology Consortium [76] (accessed at Feb. 2 2017), and we only used GO terms with the following experimental evidence codes: 'EXP', 'IDA', 'IPI', 'IMP', 'IGI', and 'IEP'. Additionally, we retained GO terms whose information contents (IC) is greater than 2 in order to perform GO enrichment test based on the more informative terms as recommended in [75]. IC is given by

$$IC\left(g\right) = -\log_2 \frac{|g|}{|root\left(g\right)|}, \tag{3.33}$$

where $|g|$ is the number of proteins that are annotated with the particular GO term g, and $|root\left(g\right)|$ is the number of proteins belonging to the root GO term of the GO term g. Note that, due to the hierarchical structure, every GO term belongs to one of the root terms: biological process (BP, GO:0008150), cellular component (CC, GO:0005575), and molecular function (MF, GO:0003674). We used the latest version of GO::TermFinder [74] to perform the GO enrichment test for the querying results.

We compared the performance of CUFID-query against state-of-the-art algorithms: SEQUOIA [18], NatalieQ [56], Corbi [58], RESQUE [11], and HubAlign [34]. We used default parameters for NatalieQ. In the R package for Corbi, we used a function for a network querying with the default parameters and set the query type as a general querying because we cannot get the results when we set the query type as a heuristic querying.

Although HubAlign is a pairwise network alignment algorithm, we used HubAlign to compare the performance of network querying algorithm because network querying can be classified as a special case of a local network alignment.

To assess the performance of the querying algorithms, we defined various performance metrics. First, since network querying algorithms can be utilized to predict novel biological complexes, we performed GO enrichment test for the querying results through GO::TermFinder [74], and if the false discovery rate (FDR) corrected p-value of the querying result is smaller than 0.01, we considered that the querying result is biologically significant so that it has a potential to be a functional module. Then, we counted the number of hits, defined as the querying results whose FDR corrected p-values are smaller than 0.01. Among these hits, we also counted the number of meaningful hits that are connected querying results whose FDR corrected p-value is smaller than 0.01.

Next, we defined a specific hit as the querying result that is highly overlapped with the know biological complexes. To determine whether the querying result is well-matched to the known biological complexes, we computed the match score of the querying result by comparing it to the known biological complexes $\mathcal{R} = \{\mathcal{G}_1, \mathcal{G}_1, ..., \mathcal{G}_N\}$. Given two biological complexes $\mathcal{G}_\mathcal{X}$ and $\mathcal{G}_\mathcal{Y}$, the matching score is a Jaccard similarity index, which is given by [73]

$$match\_score\left(\mathcal{G}_\mathcal{X}, \mathcal{G}_\mathcal{Y}\right) = \frac{|\mathcal{V}_\mathcal{X} \cap \mathcal{V}_\mathcal{Y}|}{|\mathcal{V}_\mathcal{X} \cup \mathcal{V}_\mathcal{Y}|}. \tag{3.34}$$

Given a querying result $\mathcal{G}_{\mathcal{Q}^*}$, we computed the match score $match\_score\left(\mathcal{G}_{\mathcal{Q}^*}, \mathcal{G}_\mathcal{X}\right)$ for all $\mathcal{G}_\mathcal{X}$ in $\mathcal{R}$, and if there is at least one known complex that yields the match score greater than a threshold m_t, we considered the query result as a specific hit. In this study, we used a threshold m_t of 0.5 as in [73].

We also checked the specificity of the querying results because a querying result may contain irrelevant nodes even though it can detect the functional modules. Querying results

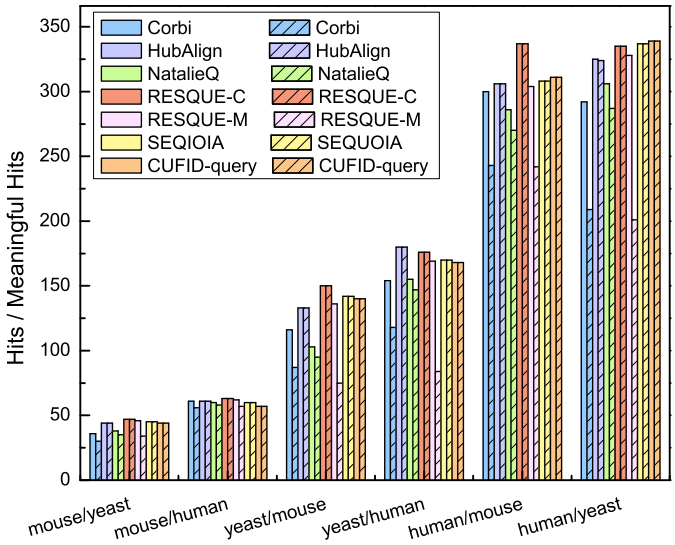


Figure 3.9: The number of hits and the number of meaningful hits are shown for each network querying algorithm. The bars shown in solid colors indicate hits and the shaded bar indicate meaningful hits. Labels in the horizontal axis show the (query species)/(target PPI network) pairs.

including many irrelevant nodes may decrease the reliability of the querying algorithm, and it may not be appropriate in practical applications as it requires additional biological experiments for validation. To this aim, a specificity was defined as the ratio of the number of annotated nodes to the overall number of nodes in the querying result. In this experiment, we selected the enriched GO term with the smallest FDR corrected p-value, and counted the number of nodes annotated with the GO term.

Finally, we also compared the running time of each method in order to compare the computational complexity.

Figure 3.9 shows the number of hits and meaningful hits for all the query and target pairs. As shown in Figure 3.9, although RESQUE-C can identify a slightly larger number of hits, CUFID-query achieves a comparable number of hits to the other methods. CUFID-query, SEQUOIA, HubAlign, and RESQUE-M show the similar performances in terms of hits. Among six methods, the sizes of the querying results for RESQUE-C are mostly

larger than those of other methods. Including more proteins in the query results can lead to more enriched GO terms with biological significance because biological complexes can be overlapped and proteins can perform multiple functions. As a result, RESQUE-C has a higher chance to achieve a higher number of hits than the other methods. Although RESQUE-C achieves the largest number of hits, we will show later that RESQUE-C includes a larger number of irrelevant nodes in the querying results that can decrease the specificity of the querying results. HubAlign and RESQUE-M show the comparable performance to CUFID-query, but we will also present that they can identify a relatively smaller number of annotated nodes. When considering one of goals for network querying, predicting and annotating functions of proteins in the target network based on the functions of the query network, identifying more annotated proteins is much advantageous. Results in Figure 3.9 implies that CUFID-query has a strong potential to identify a novel functional module conserved in the target PPI network.

Next, when considering meaningful hits, CUFID-query outperforms Corbi, NatalieQ, and RESQUE-M for all query-target pairs by achieving 52, 42, and 18 percent more meaningful hits, respectively. Although RESQUE-M records a similar number of hits to CUFID-query, the number of meaningful hits is much smaller than that of CUFID-query because RESQUE-M does not guarantee the connected querying results. Similarly, Corbi and NatalieQ may also identify disconnected subnetworks as their querying results, which can decrease the number of meaningful hits. Identifying a connected querying result is practically important because interactions between proteins can trigger or inhibit a particular cellular mechanism and disconnected querying results may not be helpful to decipher and interpret the functions of proteins and their relationships. That is, achieving a higher number of meaningful hits instead of any hits is more important in practice. Based on these results, CUFID-query is advantageous to identify and predict protein-protein interactions that cause particular biological processes.

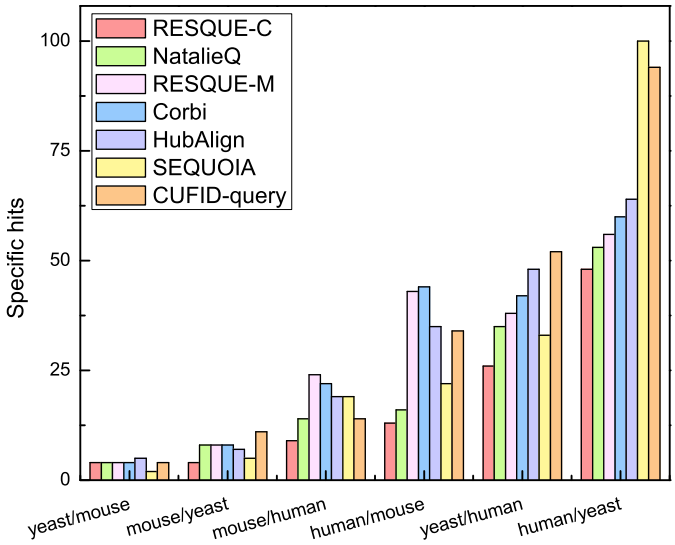
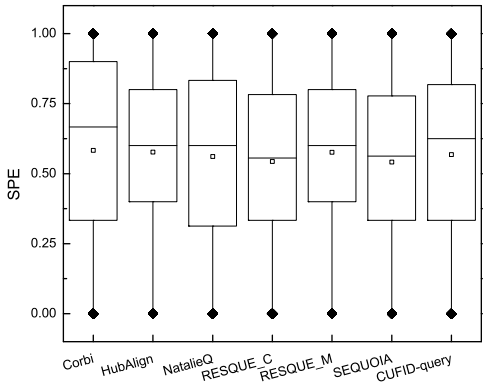


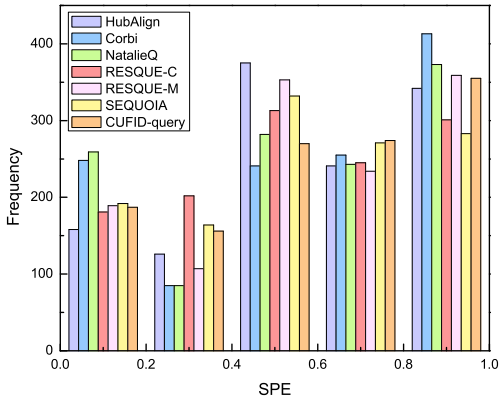
Figure 3.10: The number of specific hits for each network querying algorithm. Labels in the horizontal axis show the (query species)/(target PPI network) pairs.

Figure 3.10 shows the number of specific hits for each network querying algorithm. Except the case comparing mouse and human, CUFID-query achieves a higher number of specific hits. When querying yeast complexes against the human PPI network, CUFID-query clearly outperforms all competing methods. Although RESQUE-C achieves the largest number of hits and meaningful hits, it records the least number of specific hits because RESQUE-C includes a large number of less relevant nodes as we mentioned before. Overall, CUFID-query achieves 16 percent more specific hits than the next best algorithm, Corbi. Since the main goal of network querying is identifying the conserved subnetworks in the target network that are similar to the given query network, achieving a higher number of specific hits is more appropriate for the goal. These results mean that CUFID-query has a strong potential to accurately identify the known biological complex conserved in the target PPI network.

We also checked the specificity of each network querying algorithm. Although the querying algorithm can identify the highly relevant subnetworks to the given query net-



(a) Each box plot shows the specificity of a given network querying algorithm. Note that the square corresponds to the mean value and the black diamonds indicate the outliers.



(b) Histogram showing the specificity of each algorithm.

Figure 3.11: The specificity of the predictions made by different network querying algorithms.

work, if it includes a larger number of less relevant nodes, it is difficult to exactly select the conserved subnetwork corresponding to a particular biological process. Figure 3.11 shows a box plot and histogram of the specificity for each method. As shown in Figure 3.11a, although Corbi and NatalieQ achieve the highest median value, the difference of the mean specificity for each method is negligible. CUFID-query still achieves higher specificity than HubAlign and RESQUE families. Interestingly, there are a number of outliers at either 0 or 1. Based on the box plot for the specificity, it is difficult to select the best algorithm in terms of the specificity because of the outliers. However, Figure 3.11b shows that, although Corbi and NatalieQ can identify more querying results whose specificity is greater than 0.8, there are also a remarkably larger number of querying results whose specificity is smaller than 0.2. However, for CUFID-query, there are a relatively smaller number of querying results with low specificity, and there are a comparable number of querying results achieving fairly high specificity. This result indicates that querying results of the proposed method is comparably accurate and it includes a relatively smaller

	human/mouse			human/yeast		
	Annotated	Identified	% Annotated	Annotated	Identified	% Annotated
NatalieQ	1,233	2,454	0.502	1,382	1,753	0.788
Corbi	1,357	2,493	0.544	1,305	1,693	0.771
HubAlign	1,343	2,544	0.528	1,692	2,461	0.688
RESQUE-C	2,019	4,530	0.446	2,735	3,773	0.725
RESQUE-M	1,395	2,553	0.546	1,669	2,317	0.720
SEQUOIA	1799	3767	0.478	2807	3842	0.731
CUFID-query	1,548	2,946	0.525	2,234	2,969	0.752
	mouse/human			mouse/yeast		
	Annotated	Identified	% Annotated	Annotated	Identified	% Annotated
NatalieQ	223	366	0.609	161	193	0.834
Corbi	229	355	0.645	157	193	0.813
HubAlign	245	372	0.659	206	329	0.626
RESQUE-C	383	712	0.538	368	499	0.737
RESQUE-M	246	372	0.661	227	290	0.783
SEQUOIA	296	542	0.546	336	491	0.684
CUFID-query	277	447	0.620	274	397	0.690
	yeast/human			yeast/mouse		
	Annotated	Identified	% Annotated	Annotated	Identified	% Annotated
NatalieQ	767	1,250	0.614	394	1,223	0.322
Corbi	790	1,230	0.642	424	1,246	0.340
HubAlign	1,003	1,654	0.606	571	1,683	0.339
RESQUE-C	1,265	2,490	0.508	772	2,488	0.310
RESQUE-M	881	1,574	0.560	543	1,571	0.346
SEQUOIA	1,171	2,234	0.524	704	2,337	0.301
CUFID-query	942	1,507	0.625	531	1,541	0.345

	Overall		
	Annotated	Identified	% Annotated
NatalieQ	4,160	7,239	0.575
Corbi	4,262	7,210	0.591
HubAlign	5,060	9,043	0.560
RESQUE-C	7,542	14,492	0.520
RESQUE-M	4,961	8,677	0.572
SEQUOIA	7,113	13,213	0.538
CUFID-query	5,806	9,807	0.592

Table 3.6: The number of identified nodes and the number of annotated nodes are summarized for different network querying algorithms and different query/target network pairs.

number of less relevant nodes.

Next, we also investigated the number of identified nodes and annotated nodes. As

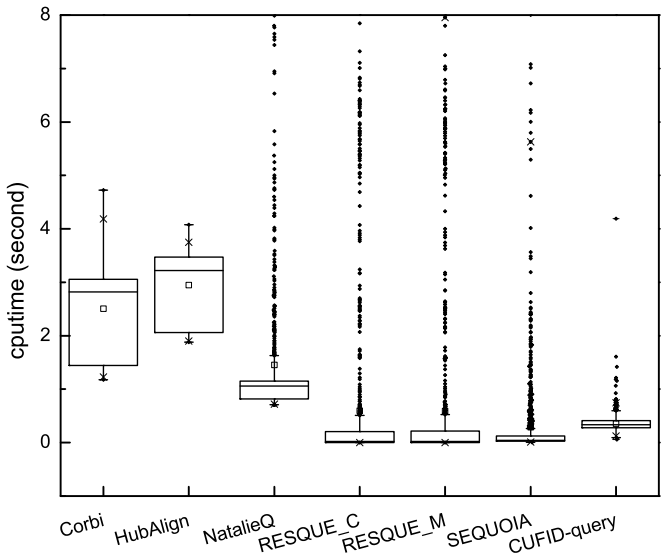


Figure 3.12: Computation time for each algorithm. Note that the black dots outside the whiskers correspond to outliers.

a network querying algorithm can be utilized to predict the functions of proteins in the identified network by transferring the knowledge about the functions of the querying network, identifying more annotated nodes would be advantageous for annotating functions of proteins in the target network (i.e., transferring the prior knowledge of the query network). Table 3.6 shows that RESQUE-C can identify a larger number of annotated nodes but the size of querying results is also relatively larger than the ones obtained by other methods. Hence, this causes the lowest percentage of annotated nodes for RESQUE-C. CUFID-query and Corbi show the similar percentages of annotated nodes, but CUFID-query can identify more annotated nodes than Corbi. This means that CUFID-query is more effective to accurately annotate protein functions in the novel biological complex (i.e., identified subnetwork in the target network).

To compare the computational complexity of each method, we compared the running time of each method. In this experiment, we tested all network querying algorithms using

the same machine equipped with intel i7 dual core processor (2.9 GHz) and 16 GB memory. Figure 3.12 shows that CUFID-query and RESQUE family are much faster than other algorithms, and NatalieQ records the next best in terms of the computation time. Interestingly, although the average of the computation time for NatalieQ and RESQUE family is very fast, there are a number of outliers. That is, they require unexpectedly longer time for network querying in some cases. These may depend on the topological structure of the query and target networks. That is, particular topological structures may require longer computation time for querying. Although CUFID-query also has outliers, the most cases complete the querying within a few seconds for all 1,242 test cases. In addition to the computation time, NatalieQ fails to identify querying results for 71 queries among 1,242 queries (i.e., NatalieQ can not find any matching nodes for 71 queries). This means NatalieQ may not be robust for a particular topological structure, but CUFID-query finds querying results for all 1,242 queries and records a stable running time, where it implies the robustness of CUFID-query.

3.3.3 Conclusions

In this subchapter, we propose a novel network querying algorithm, CUFID-query. We utilize the CUFID model in order to estimate the correspondence (or biological relevance) between nodes in the query and large-scale target networks. In the CUFID model, we first construct the integrated network by inserting pseudo-edges between nodes in the query and target networks, and we design a random walker whose random transition through a pseudo-edge is proportional to both node and topological similarities. Hence, we can effectively estimate the node correspondence by measuring a steady-state network flow across the pseudo-edges with a reduced computational cost. Based on the estimated node correspondence scores through the CUFID model, CUFID-query identifies the seed network (i.e., high correspondence region in the target network). Then, we iteratively extend

the seed network by adding a selected node, based on the association probability and the conductance minimization criterion. Finally, in case that the seed-and-extension approach may include irrelevant nodes, we remove less relevant nodes based on the personalized PageRank vector for the induced network. Through an extensive performance evaluation using 1,242 known biological complexes and large-scale PPI networks, we have shown that CUFID-query leads to accurate and functionally consistent querying results.

4. SUMMARY AND CONCLUSIONS

In this dissertation, we proposed novel probabilistic random walk models to estimate node-to-node correspondences between large-scale networks. To validate the effectiveness, we applied the proposed random walk models to comparative network analysis such as a global network alignment problem and a network querying problem.

First, we proposed the context-sensitive random walk (CSRW) model. In this model, we addressed challenges in comparative network analysis. That is, the main concern is how to effectively integrate the different types of similarities in a balanced manner by dealing with structural variations such node insertions/deletions. The concept of the context-sensitive random walk model is motivated by the pair-HMM that is widely accepted in the biological sequence alignment problem because we can identify conceptually similar counterparts between comparative sequence and network analysis. In the context-sensitive random walk model, the random walker can switch its mode of movement based on the context (i.e., similarities of the neighboring nodes) of the current position of the random walker. The context-sensitive nature of this model enabled to effectively incorporate different similarities even though there are structural variations across networks.

Second, we proposed a CUFID (**C**omparative network analysis **U**sing the steady-state network **F**low to **ID**entify orthologous proteins) model. The CUFID model adopts a concept of 'water' flow between networks. That is, it constructs the integrated network by inserting pseudo edges between potential matching nodes in different networks. Then, we design the random walker that the frequency of transitions across pseudo edges is proportional to the node level similarity as well as the number of (potential) matching neighboring nodes. As a result, the CUFID model further improves the CSRW model in terms of accuracy as well as computational complexity. More importantly, the CUFID model can

be easily extended to estimate global node correspondences for multiple networks.

We applied CSRW model and CUFID model to comparative network analysis problems using real protein-protein interactions (PPI) networks. In a global network alignment problem, we estimated node-to-node correspondences between different PPI networks by using CSRW and CUFID models. Then, we derived the maximum expected accuracy (MEA) alignments. Extensive performance validation using various metrics shows that the CSRW and CUFID models can accurately estimate the node-to-node correspondences for large-scale biological networks and it can lead accurate and reliable comparison results. We also applied the proposed models to a network querying problem in order to confirm whether the proposed models can accurately estimate node correspondences even though the size of networks to be compared is significantly different. Here, to deal with the structural variations (i.e., node insertions and deletions) across the conserved subnetworks, we adopt a seed-and extension approach to identify low conductance subnetworks. We select the seed network based on the estimated node correspondence scores through proposed random walk models, and iteratively extended the seed network by adding the node that can maximally minimize the conductance of the extending seed network. Finally, we removed less-relevant nodes. Experimental results using real biological complexes show that proposed method can lead reliable querying results with enhanced biological significance.

REFERENCES

[1] S. Umeyama, "An eigendecomposition approach to weighted graph matching problems," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 10, no. 5, pp. 695–703, 1988.

[2] C. Ding, T. Li, M. Jordan, *et al.*, "Nonnegative matrix factorization for combinatorial optimization: Spectral clustering, graph matching, and clique finding," in *IEEE International Conference on Data Mining (ICDM)*, pp. 183–192, IEEE, 2008.

[3] M. Bayati, M. Gerritsen, D. F. Gleich, A. Saberi, and Y. Wang, "Algorithms for large, sparse network alignment problems," in *IEEE International Conference on Data Mining (ICDM)*, pp. 705–710, IEEE, 2009.

[4] D. Koutra, H. Tong, and D. Lubensky, "Big-Align: Fast bipartite graph alignment," in *2013 IEEE International Conference on Data Mining (ICDM)*, pp. 389–398, IEEE, 2013.

[5] G. Palla, I. Derényi, I. Farkas, and T. Vicsek, "Uncovering the overlapping community structure of complex networks in nature and society," *Nature*, vol. 435, no. 7043, pp. 814–818, 2005.

[6] L. Backstrom and J. Leskovec, "Supervised random walks: predicting and recommending links in social networks," in *Proceedings of the 4th ACM International Conference on Web Search and Data Mining*, pp. 635–644, ACM, 2011.

[7] O. Duchenne, F. Bach, I.-S. Kweon, and J. Ponce, "A tensor-based algorithm for high-order graph matching," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 12, pp. 2383–2395, 2011.

[8] X. Qian and B.-J. Yoon, "Shape matching based on graph alignment using hidden markov models," in *IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP)*, pp. 934–937, IEEE, 2010.

[9] R. Sharan and T. Ideker, "Modeling cellular machinery through biological network comparison," *Nature Biotechnology*, vol. 24, no. 4, pp. 427–433, 2006.

[10] B.-J. Yoon, X. Qian, and S. M. E. Sahraeian, "Comparative analysis of biological networks: Hidden markov model and markov chain-based approach," *IEEE Signal Processing Magazine*, vol. 29, no. 1, pp. 22–34, 2012.

[11] S. M. E. Sahraeian and B.-J. Yoon, "RESQUE: Network reduction using semi-markov random walk scores for efficient querying of biological networks," *Bioinformatics*, vol. 28, no. 16, pp. 2129–2136, 2012.

[12] R. Singh, J. Xu, and B. Berger, "Global alignment of multiple protein interaction networks with application to functional orthology detection," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 105, no. 35, pp. 12763–12768, 2008.

[13] S. M. E. Sahraeian and B.-J. Yoon, "SMETANA: accurate and scalable algorithm for probabilistic alignment of large-scale biological networks," *PLoS ONE*, vol. 8, no. 7, p. e67995, 2013.

[14] S. M. E. Sahraeian and B.-J. Yoon, "A novel low-complexity hmm similarity measure," *IEEE Signal Processing Letters*, vol. 18, no. 2, pp. 87–90, 2011.

[15] M. D. Collins, J. Xu, L. Grady, and V. Singh, "Random walks based multi-image segmentation: Quasiconvexity results and gpu-based solutions," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1656–1663, IEEE, 2012.

[16] H. Jeong and B.-J. Yoon, "Effective estimation of node-to-node correspondence between different graphs," *IEEE Signal Processing Letters*, vol. 22, no. 6, pp. 661–665, 2015.

[17] H. Jeong and B.-J. Yoon, "Accurate multiple network alignment through context-sensitive random walk," *BMC Systems Biology*, vol. 9, no. suppl. 1, p. S7, 2015.

[18] H. Jeong and B.-J. Yoon, "SEQUOIA: significance enhanced network querying through context-sensitive random walk and minimization of network conductance," *BMC Systems Biology*, vol. 11, no. 3, p. 20, 2017.

[19] R. Durbin, *Biological sequence analysis: probabilistic models of proteins and nucleic acids*. Cambridge University Press, 1998.

[20] B.-J. Yoon, "Hidden Markov models and their applications in biological sequence analysis," *Current Genomics*, vol. 10, no. 6, p. 402, 2009.

[21] L. Page, S. Brin, R. Motwani, and T. Winograd, "The PageRank citation ranking: bringing order to the Web.," *Technical Report, Stanford InfoLab.*, 1999.

[22] S. M. E. Sahraeian and B.-J. Yoon, "A network synthesis model for generating protein interaction network families," *PLoS ONE*, vol. 7, no. 8, p. e41474, 2012.

[23] W. K. Kim and E. M. Marcotte, "Age-dependent evolution of the yeast protein interaction network suggests a limited role of gene duplication and divergence," *PLoS Computational Biology*, vol. 4, no. 11, p. e1000232, 2008.

[24] R. Sharan, S. Suthram, R. M. Kelley, T. Kuhn, S. McCuine, P. Uetz, T. Sittler, R. M. Karp, and T. Ideker, "Conserved patterns of protein interaction in multiple species," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 102, no. 6, pp. 1974–1979, 2005.

[25] B. Dost, T. Shlomi, N. Gupta, E. Ruppin, V. Bafna, and R. Sharan, "QNet: a tool for querying protein interaction networks," *Journal of Computational Biology*, vol. 15, no. 7, pp. 913–925, 2008.

[26] Q. Huang, L.-Y. Wu, and X.-S. Zhang, "An efficient network querying method based on conditional random fields," *Bioinformatics*, vol. 27, no. 22, pp. 3173–3178, 2011.

[27] C.-S. Liao, K. Lu, M. Baym, R. Singh, and B. Berger, "IsoRankN: spectral methods for global alignment of multiple protein networks," *Bioinformatics*, vol. 25, no. 12, pp. i253–i258, 2009.

[28] J. Flannick, A. Novak, B. S. Srinivasan, H. H. McAdams, and S. Batzoglou, "Graemlin: general and robust alignment of multiple large interaction networks," *Genome Research*, vol. 16, no. 9, pp. 1169–1181, 2006.

[29] O. Kuchaiev, T. Milenković, V. Memišević, W. Hayes, and N. Pržulj, "Topological network alignment uncovers biological function and phylogeny," *Journal of the Royal Society Interface*, vol. 7, no. 50, pp. 1341–1354, 2010.

[30] O. Kuchaiev and N. Pržulj, "Integrative network alignment reveals large regions of global network similarity in yeast and human," *Bioinformatics*, vol. 27, no. 10, pp. 1390–1396, 2011.

[31] H. T. Phan and M. J. Sternberg, "PINALOG: a novel approach to align protein interaction networks–implications for complex detection and function prediction," *Bioinformatics*, vol. 28, no. 9, pp. 1239–1245, 2012.

[32] J. Hu, B. Kehr, and K. Reinert, "NetCoffee: a fast and accurate global alignment approach to identify functionally conserved proteins in multiple networks," *Bioinformatics*, vol. 30, no. 4, pp. 540–548, 2013.

[33] F. Alkan and C. Erten, "BEAMS: backbone extraction and merge strategy for the global many-to-many alignment of multiple ppi networks," *Bioinformatics*, vol. 30, no. 4, pp. 531–539, 2014.

[34] S. Hashemifar and J. Xu, "HubAlign: an accurate and efficient method for global alignment of protein–protein interaction networks," *Bioinformatics*, vol. 30, no. 17, pp. i438–i444, 2014.

[35] S. Panni and S. E. Rombo, "Searching for repetitions in biological networks: methods, resources and tools," *Briefings in Bioinformatics*, vol. 30, no. 10, pp. 1343–1352, 2013.

[36] C. B. Do, M. S. Mahabhashyam, M. Brudno, and S. Batzoglou, "ProbCons: Probabilistic consistency-based multiple sequence alignment," *Genome Research*, vol. 15, no. 2, pp. 330–340, 2005.

[37] M. Hamada and K. Asai, "A classification of bioinformatics algorithms from the viewpoint of maximizing expected accuracy (mea)," *Journal of Computational Biology*, vol. 19, no. 5, pp. 532–549, 2012.

[38] U. Roshan and D. R. Livesay, "ProbAlign: multiple sequence alignment using partition function posterior probabilities," *Bioinformatics*, vol. 22, no. 22, pp. 2715–2721, 2006.

[39] S. M. E. Sahraeian and B.-J. Yoon, "PicXAA: greedy probabilistic construction of maximum expected accuracy alignment of multiple sequences," *Nucleic Acids Research*, vol. 38, no. 15, pp. 4917–4928, 2010.

[40] S. M. E. Sahraeian and B.-J. Yoon, "PicXAA-R: efficient structural alignment of multiple RNA sequences using a greedy approach," *BMC Bioinformatics*, vol. 12 suppl. 1, p. S38, 2011.

[41] S. M. E. Sahraeian and B.-J. Yoon, "PicXAA-Web: a web-based platform for non-progressive maximum expected accuracy alignment of multiple biological sequences," *Nucleic Acids Research*, vol. 39, pp. 8–12, Jul 2011.

[42] D. Park, R. Singh, M. Baym, C.-S. Liao, and B. Berger, "IsoBase: a database of functionally related proteins across ppi networks," *Nucleic Acids Research*, vol. 39, no. suppl. 1, pp. D295–D300, 2011.

[43] B.-J. Breitkreutz, C. Stark, T. Reguly, L. Boucher, A. Breitkreutz, M. Livstone, R. Oughtred, D. H. Lackner, J. Bähler, V. Wood, *et al.*, "The BioGRID interaction database: 2008 update," *Nucleic Acids Research*, vol. 36, no. suppl. 1, pp. D637–D640, 2008.

[44] L. Salwinski, C. S. Miller, A. J. Smith, F. K. Pettit, J. U. Bowie, and D. Eisenberg, "The database of interacting proteins: 2004 update," *Nucleic Acids Research*, vol. 32, no. suppl. 1, pp. D449–D451, 2004.

[45] T. K. Prasad, R. Goel, K. Kandasamy, S. Keerthikumar, S. Kumar, S. Mathivanan, D. Telikicherla, R. Raju, B. Shafreen, A. Venugopal, *et al.*, "Human protein reference database: 2009 update," *Nucleic Acids Research*, vol. 37, no. suppl. 1, pp. D767–D772, 2009.

[46] A. Ceol, A. C. Aryamontri, L. Licata, D. Peluso, L. Briganti, L. Perfetto, L. Castagnoli, and G. Cesareni, "MINT, the molecular interaction database: 2009 update," *Nucleic Acids Research*, pp. D532–D539, 2009.

[47] B. Aranda, P. Achuthan, Y. Alam-Faruque, I. Armean, A. Bridge, C. Derow, M. Feuermann, A. Ghanbarian, S. Kerrien, J. Khadake, *et al.*, "The IntAct molecular interaction database in 2010," *Nucleic Acids Research*, vol. 38, no. suppl. 1, pp. D525–D531, 2010.

[48] M. Kanehisa and S. Goto, "KEGG: Kyoto encyclopedia of genes and genomes," *Nucleic Acids Research*, vol. 28, no. 1, pp. 27–30, 2000.

[49] M. Kanehisa, M. Araki, S. Goto, M. Hattori, M. Hirakawa, M. Itoh, T. Katayama, S. Kawashima, S. Okuda, T. Tokimatsu, and Y. Yamanishi, "KEGG for linking genomes to life and the environment," *Nucleic Acids Research*, vol. 36, no. suppl. 1, pp. D480–D484, 2008.

[50] L. Hakes, J. W. Pinney, D. L. Robertson, and S. C. Lovell, "Protein-protein interaction networks and biology–what's the connection?," *Nature Biotechnology*, vol. 26, no. 1, pp. 69–72, 2008.

[51] O. Kuchaiev, M. Rašajski, D. J. Higham, and N. Pržulj, "Geometric de-noising of protein-protein interaction networks," *PLoS Computational Biology*, vol. 5, no. 8, p. e1000454, 2009.

[52] B. P. Kelley, B. Yuan, F. Lewitter, R. Sharan, B. R. Stockwell, and T. Ideker, "PathBLAST: a tool for alignment of protein interaction networks," *Nucleic Acids Research*, vol. 32, no. suppl. 2, pp. W83–W88, 2004.

[53] T. Shlomi, D. Segal, E. Ruppin, and R. Sharan, "QPath: a method for querying pathways in a protein-protein interaction network," *BMC Bioinformatics*, vol. 7, no. 1, p. 199, 2006.

[54] Q. Yang and S.-H. Sze, "Path matching and graph matching in biological networks," *Journal of Computational Biology*, vol. 14, no. 1, pp. 56–67, 2007.

[55] Y. Tian, R. C. Mceachin, C. Santos, J. M. Patel, *et al.*, "SAGA: a subgraph matching tool for biological graphs," *Bioinformatics*, vol. 23, no. 2, pp. 232–239, 2007.

[56] G. W. Klau, "A new graph-based method for pairwise global network alignment," *BMC Bioinformatics*, vol. 10, no. suppl. 1, p. S59, 2009.

[57] S. Bruckner, F. Hüffner, R. M. Karp, R. Shamir, and R. Sharan, "Topology-free querying of protein interaction networks," *Journal of Computational Biology*, vol. 17, no. 3, pp. 237–252, 2010.

[58] Q. Huang, L.-Y. Wu, and X.-S. Zhang, "Corbi: a new R package for biological network alignment and querying," *BMC Systems Biology*, vol. 7, no. suppl. 2, p. S6, 2013.

[59] X. Qian, S. M. E. Sahraeian, and B.-J. Yoon, "Enhancing the accuracy of hmm-based conserved pathway prediction using global correspondence scores," *BMC Bioinformatics*, vol. 12, no. 10, p. S6, 2011.

[60] M. Girvan and M. E. Newman, "Community structure in social and biological networks," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 99, no. 12, pp. 7821–7826, 2002.

[61] M. E. Newman, "Finding community structure in networks using the eigenvectors of matrices," *Physical Review E*, vol. 74, no. 3, p. 036104, 2006.

[62] V. Spirin and L. A. Mirny, "Protein complexes and functional modules in molecular networks," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 100, no. 21, pp. 12123–12128, 2003.

[63] S. Z. Dadaneh and X. Qian, "Bayesian module identification from multiple noisy networks," *EURASIP Journal on Bioinformatics and Systems Biology*, vol. 2016, no. 1, p. 5, 2016.

[64] S. A. Cook, "The complexity of theorem-proving procedures," in *Proceedings of the third annual ACM symposium on Theory of computing*, pp. 151–158, ACM, 1971.

[65] R. Sharan, I. Ulitsky, and R. Shamir, "Network-based prediction of protein function," *Molecular Systems Biology*, vol. 3, no. 1, p. 88, 2007.

[66] R. Kannan, S. Vempala, and A. Vetta, "On clusterings: Good, bad and spectral," *Journal of the ACM (JACM)*, vol. 51, no. 3, pp. 497–515, 2004.

[67] J. Leskovec, K. J. Lang, and M. Mahoney, "Empirical comparison of algorithms for network community detection," in *Proceedings of the 19th International Conference on World Wide Web*, pp. 631–640, ACM, 2010.

[68] G. Micale, A. Pulvirenti, R. Giugno, and A. Ferro, "GASOLINE: a Greedy And Stochastic algorithm for Optimal Local multiple alignment of Interaction NEtworks," *PLoS ONE*, vol. 9, no. 6, 2014.

[69] D. Szklarczyk, A. Franceschini, M. Kuhn, M. Simonovic, A. Roth, P. Minguez, T. Doerks, M. Stark, J. Muller, P. Bork, *et al.*, "The STRING database in 2011: functional interaction networks of proteins, globally integrated and scored," *Nucleic Acids Research*, vol. 39, no. suppl. 1, pp. D561–D568, 2011.

[70] A. Ruepp, B. Brauner, I. Dunger-Kaltenbach, G. Frishman, C. Montrone, M. Stransky, B. Waegele, T. Schmidt, O. N. Doudieu, V. Stümpflen, *et al.*, "CORUM: the comprehensive resource of mammalian protein complexes," *Nucleic Acids Research*, vol. 36, no. suppl. 1, pp. D646–D650, 2008.

[71] J. M. Cherry, E. L. Hong, C. Amundsen, R. Balakrishnan, G. Binkley, E. T. Chan, K. R. Christie, M. C. Costanzo, S. S. Dwight, S. R. Engel, *et al.*, "Saccharomyces Genome Database: the genomics resource of budding yeast," *Nucleic Acids Research*, p. gkr1029, 2011.

[72] J. Hu and K. Reinert, "LocalAli: an evolutionary-based local alignment approach to identify functionally conserved modules in multiple networks," *Bioinformatics*, vol. 31, no. 3, pp. 363–372, 2015.

[73] G. Liu, L. Wong, and H. N. Chua, "Complex discovery from weighted ppi networks," *Bioinformatics*, vol. 25, no. 15, pp. 1891–1897, 2009.

[74] E. I. Boyle, S. Weng, J. Gollub, H. Jin, D. Botstein, J. M. Cherry, and G. Sherlock, "GO::TermFinder-open source software for accessing Gene Ontology information and finding significantly enriched Gene Ontology terms associated with a list of genes," *Bioinformatics*, vol. 20, no. 18, pp. 3710–3715, 2004.

[75] G. O. Consortium *et al.*, "Gene ontology consortium: Going forward," *Nucleic Acids Research*, vol. 43, no. D1, pp. D1049–D1056, 2015.

[76] M. Ashburner, C. A. Ball, J. A. Blake, D. Botstein, H. Butler, J. M. Cherry, A. P. Davis, K. Dolinski, S. S. Dwight, J. T. Eppig, *et al.*, "Gene Ontology: tool for the unification of biology," *Nature Genetics*, vol. 25, no. 1, pp. 25–29, 2000.

[77] H. Jeong, X. Qian, and B.-J. Yoon, "Effective comparative analysis of protein-protein interaction networks by measuring the steady-state network flow using a markov model," *BMC Bioinformatics*, vol. 17, no. 13, p. 395, 2016.

[78] T. M. Cover and J. A. Thomas, *Elements of information theory*. John Wiley & Sons, 2012.

[79] L. Vincent and P. Soille, "Watersheds in digital spaces: an efficient algorithm based on immersion simulations," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 6, pp. 583–598, 1991.

[80] D. Gleich, "gaimc: graph algorithms in matlab code," *Matlab Toolbox. Matlab*, 2009.

[81] Y.-K. Shih and S. Parthasarathy, "Identifying functional modules in interaction networks through overlapping markov clustering," *Bioinformatics*, vol. 28, no. 18, pp. i473–i479, 2012.

[82] R. Andersen, F. Chung, and K. Lang, "Local graph partitioning using pagerank vectors," in *Foundations of Computer Science, 2006. FOCS'06. 47th Annual IEEE Symposium on*, pp. 475–486, IEEE, 2006.

[83] Y. Wang and X. Qian, "Functional module identification in protein interaction networks by interaction patterns," *Bioinformatics*, p. btt569, 2013.

[84] T. Nepusz, H. Yu, and A. Paccanaro, "Detecting overlapping protein complexes in protein-protein interaction networks," *Nature Methods*, vol. 9, no. 5, pp. 471–472, 2012.

[85] D. Szklarczyk, A. Franceschini, S. Wyder, K. Forslund, D. Heller, J. Huerta-Cepas, M. Simonovic, A. Roth, A. Santos, K. P. Tsafou, *et al.*, "STRING v10: protein–protein interaction networks, integrated over the tree of life," *Nucleic Acids Research*, p. gku1003, 2014.

[86] D. V. Veres, D. M. Gyurkó, B. Thaler, K. Z. Szalay, D. Fazekas, T. Korcsmáros, and P. Csermely, "ComPPI: a cellular compartment-specific database for protein–protein interaction network analysis," *Nucleic Acids Research*, vol. 43, no. D1, pp. D485–D493, 2015.

[87] A. Patil, K. Nakai, and H. Nakamura, "HitPredict: a database of quality assessed protein–protein interactions in nine species," *Nucleic Acids Research*, vol. 39, no. suppl. 1, pp. D744–D749, 2011.

[88] P. Pagel, S. Kovac, M. Oesterheld, B. Brauner, I. Dunger-Kaltenbach, G. Frishman, C. Montrone, P. Mark, V. Stümpflen, H.-W. Mewes, *et al.*, "The MIPS mammalian protein–protein interaction database," *Bioinformatics*, vol. 21, no. 6, pp. 832–834, 2005.

[89] S. Pu, J. Wong, B. Turner, E. Cho, and S. J. Wodak, "Up-to-date catalogues of yeast protein complexes," *Nucleic Acids Research*, vol. 37, no. 3, pp. 825–831, 2009.

[90] J. Yu, S. Pacifico, G. Liu, and R. L. Finley, "DroID: the drosophila interactions database, a comprehensive resource for annotated gene and protein interactions," *BMC Genomics*, vol. 9, no. 1, p. 461, 2008.

[91] R. Drysdale, "FlyBase: a database for the Drosophila research community," *Drosophila: Methods and Protocols*, pp. 45–59, 2008.

[92] N. Wiwatwattana and A. Kumar, "Organelle DB: a cross-species database of protein localization and function," *Nucleic Acids Research*, vol. 33, no. suppl. 1, pp. D598–D604, 2005.

[93] L. Stein, P. Sternberg, R. Durbin, J. Thierry-Mieg, and J. Spieth, "WormBase: network access to the genome and biology of Caenorhabditis elegans," *Nucleic Acids Research*, vol. 29, no. 1, pp. 82–86, 2001.

[94] U. Consortium *et al.*, "UniProt: a hub for protein information," *Nucleic Acids Research*, p. gku989, 2014.

[95] H. M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T. N. Bhat, H. Weissig, I. N. Shindyalov, and P. E. Bourne, "The protein data bank," *Nucleic Acids Research*, vol. 28, no. 1, pp. 235–242, 2000.

[96] H. Berman, K. Henrick, and H. Nakamura, "Announcing the worldwide protein data bank," *Nature Structural & Molecular Biology*, vol. 10, no. 12, pp. 980–980, 2003.

[97] R. D. Finn, P. Coggill, R. Y. Eberhardt, S. R. Eddy, J. Mistry, A. L. Mitchell, S. C. Potter, M. Punta, M. Qureshi, A. Sangrador-Vegas, *et al.*, "The Pfam protein families database: towards a more sustainable future," *Nucleic Acids Research*, vol. 44, no. D1, pp. D279–D285, 2016.

[98] M. J. Gabanyi, P. D. Adams, K. Arnold, L. Bordoli, L. G. Carter, J. Flippen-Andersen, L. Gifford, J. Haas, A. Kouranov, W. A. McLaughlin, *et al.*, "The structural biology knowledgebase: a portal to protein structures, sequences, functions, and methods," *Journal of Structural and Functional Genomics*, vol. 12, no. 2, pp. 45–54, 2011.

APPENDIX A

LIST OF DATABASES FOR COMPARATIVE NETWORK ANALYSIS

Databases for protein-protein interactions for various species are listed in Table A.1, and the list of databases for known biological complexes are provided in Table A.2. Additionally, databases for protein sequence and homology are listed in Table A.3

Name of database	Link
BioGrid [43]	https://thebiogrid.org
ComPPI [86]	http://comppi.linkgroup.hu
DIP [44]	http://dip.doe-mbi.ucla.edu/dip/Main.cgi
HPRD [45]	http://www.hprd.org
HitPredict [87]	http://hintdb.hgc.jp/htp/index.html
IsoBase [42]	http://cb.csail.mit.edu/cb/mna/isobase/
IntAct [47]	http://www.ebi.ac.uk/intact/
MINT [46]	http://mint.bio.uniroma2.it
MIPS [88]	http://mips.helmholtz-muenchen.de/proj/ppi/
STRING [69]	https://string-db.org

Table A.1: List of available databases for PPI network analysis.

Database for known biological complexes	Link	Target Species
CORUM [70]	http://mips.helmholtz-muenchen.de/corum/	Human, Mouse, and Rat
CYC2008 [89]	http://wodaklab.org/cyc2008/	Yeast (S. cerevisiae)
DroID [90]	http://www.droidb.org/Index.jsp	Fly (Drosophila)
FlyBase [91]	http://flybase.org	Fly (Drosophila)
Organelle DB [92]	http://labs.mcdb.lsa.umich.edu/organelledb/index.php	138 organisms
SGD [71]	http://www.yeastgenome.org	Yeast (S. cerevisiae)
WormBase [93]	http://www.wormbase.org	Worm (C. elegans)

Table A.2: Databases for known biological complexes.

Name of database	Link
UnitProt [94]	http://www.uniprot.org
PDB [95]	http://www.rcsb.org/pdb/home/home.do
wwPDB [96]	http://pfam.xfam.org
Pfam [97]	http://xfam.org
SBKB [98]	http://sbkb.org

Table A.3: Databases for proteins.

APPENDIX B

SOFTWARE AVAILABILITY

Table B.1 lists online links for the proposed algorithms.

Proposed algorithms	Link
SEQUOIA [18]	http://www.ece.tamu.edu/ bjyoon/SEQUOIA/
CUFID-align [77]	http://www.ece.tamu.edu/ bjyoon/CUFID/
CUFID-query	http://www.ece.tamu.edu/ bjyoon/CUFID/

Table B.1: List of softwares proposed in this dissertation.