IDENTIFICATION OF FUNCTIONALLY ORTHOLOGOUS PROTEIN GROUPS IN DIFFERENT SPECIES BASED ON PROTEIN NETWORK ALIGNMENT

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

ÖMER NEBİL YAVEROĞLU

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
COMPUTER ENGINEERING

SEPTEMBER 2010

Approval of the thesis:

IDENTIFICATION OF FUNCTIONALLY ORTHOLOGOUS PROTEIN GROUPS IN DIFFERENT SPECIES BASED ON PROTEIN NETWORK ALIGNMENT

submitted by **ÖMER NEBİL YAVEROĞLU** in partial fulfillment of the requirements for the degree of **Master of Science in Computer Engineering Department, Middle East Technical University** by,

Prof. Dr. Canan Özgen _____
Dean, Graduate School of **Natural and Applied Sciences**

Prof. Dr. Adnan Yazıcı _____
Head of Department, **Computer Engineering**

Asst. Prof. Dr. Tolga Can _____
Supervisor, **Computer Engineering Department, METU**

Examining Committee Members:

Prof. Dr. Göktürk Üçoluk _____
Computer Engineering Department, METU

Asst. Prof. Dr. Tolga Can _____
Computer Engineering Department, METU

Prof. Dr. Gerhard Wilhelm Weber _____
Institute Of Applied Mathematics, METU

Asst. Prof. Dr. Yeşim Aydın Son _____
Informatics Institude, METU

Asst. Prof. Dr. Sinan Kalkan _____
Computer Engineering Department, METU

Date: _____

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name: ÖMER NEBİL YAVEROĞLU

Signature :

ABSTRACT

IDENTIFICATION OF FUNCTIONALLY ORTHOLOGOUS PROTEIN GROUPS IN DIFFERENT SPECIES BASED ON PROTEIN NETWORK ALIGNMENT

Yaveroğlu, Ömer Nebil
M.Sc., Department of Computer Engineering
Supervisor : Asst. Prof. Dr. Tolga Can

SEPTEMBER 2010, 61 pages

In this study, an algorithm named ClustOrth is proposed for determining and matching functionally orthologous protein clusters in different species. The algorithm requires protein interaction networks of the organisms to be compared and GO terms of the proteins in these interaction networks as prior information. After determining the functionally related protein groups using the Repeated Random Walks algorithm, the method maps the identified protein groups according to the similarity metric defined. In order to evaluate the similarities of protein groups, graph theoretical information is used together with the context information about the proteins. The clusters are aligned using GO-Term-based protein similarity measures defined in previous studies. These alignments are used to evaluate cluster similarities by defining a cluster similarity metric from protein similarities. The top scoring cluster alignments are considered as orthologous. Several data sources providing orthology information have shown that the defined cluster similarity metric can be used to make inferences about the orthological relevance of protein groups. Comparison with a protein orthology prediction algorithm named ISORANK also showed that the ClustOrth algorithm is successful in determining orthologies between proteins. However, the cluster similarity metric is too strict and many cluster matches are not able to produce high scores for this metric. For this reason, the

number of predictions performed is low. This problem can be overcomed with the introduction of different sources of information related to proteins in the clusters for the evaluation of the clusters. The ClustOrth algorithm also outperformed the NetworkBLAST algorithm which aims to find orthologous protein clusters using protein sequence information directly for determining orthologies. It can be concluded that this study is one of the leading studies addressing the protein cluster matching problem for identifying orthologous functional modules of protein interaction networks computationally.

Keywords: Orthology Detection, Network Alignments, GO Terms, Graph Matching Algorithms, Protein Networks

ÖZ

FARKLI TÜRLERDE BULUNAN FONKSİYONEL OLARAK ORTOLOG OLAN PROTEİN GRUPLARININ PROTEİN AĞLARININ HİZALANMASINA DAYALI OLARAK BELİRLENMESİ

Yaveroğlu, Ömer Nebil
Yüksek Lisans, Bilgisayar Mühendisliği Bölümü
Tez Yöneticisi : Y. Doç. Dr. Tolga Can

EYLÜL 2010, 61 sayfa

Bu çalışmada, farklı türlerde bulunan fonksiyonel açıdan ortolojik (orthologous) olan protein kümelerinin belirlenmesi ve eşleştirilmesi için ClustOrth isminde bir algoritma önerilmiştir. Bu algoritma karşılaştırılacak organizmaların protein etkileşim ağlarına ve bu etkileşim ağlarında bulunan proteinlerin GO terimlerinine öncü bilgi olarak ihtiyaç duymaktadır. Tekrarlanan Yürüyüş Algoritması ile fonksiyonel olarak ilişkili protein gruplarının belirlenmesinden sonra yöntem, belirlenen protein gruplarını tanımlanan bir benzerlik ölçütüne bağlı olarak birbirine eşler. Protein gruplarının benzerliklerinin değerlendirilmesi için çizge (graph) teorisi tabanlı bilgi proteinlerin içerik bilgisi ile birlikte kullanılmıştır. Kümeler daha önceki çalışmalarda tanımlanmış olan GO terimi tabanlı protein benzerlik ölçüleri kullanılarak hizalanmıştır. Bu hizalamalar küme benzerliklerini değerlendirmek için protein benzerliklerinden küme benzerlik ölçüsü tanımlayarak kullanılmıştır. En yüksek skoru üreten hizalamalar ortolojik olarak kabul edilmiştir. Ortoloji bilgisi sağlayan çeşitli veri kaynakları tanımlanan küme benzerlik ölçüsünün protein gruplarının ortolojik bağlarının tahmin edilmesinde kullanılabileceğini göstermiştir. Protein ortolojisi tahmin etme yöntemi olan ISORANK isimli algoritma ile yapılan karşılaştırmalar, ClustOrth'un proteinler arasındaki ortolojilerin belirlenmesinde

başarılı olduğunu göstermiştir. Fakat küme benzerliği ölçüsü çok katıdır ve birçok küme eşleştirmesi bu ölçüt için yüksek skorlar üretememektedir. Bu nedenden ötürü yapılan tahminler düşük sayıdadır. Bu problem kümelerin değerlendirilmesi için kümelerdeki proteinlerle ilgili farklı bilgi kaynaklarının eklenmesi ile aşılabilir. ClustOrth aynı zamanda ortolojileri tanımlamak için protein dizi bilgisini kullanarak ortolojik protein kümelerinin bulunmasını hedefleyen NetworkBLAST algoritmasından daha iyi çalışmıştır. Özetle bu çalışma protein etkileşim ağlarının ortolojik fonksiyonel modüllerinin berimsel olarak bulunması için protein kümelerini eşlemeye çalışan öncü çalışmalardan biridir.

Anahtar Kelimeler: Ortoloji Belirleme, Ağ Hizalamaları, GO Terimleri, Çizge Eşleme Algoritmaları, Protein Ağları

To everyone who introduces a meaning to my life...

ACKNOWLEDGMENTS

I would like to thank Asst. Prof. Dr. Tolga Can for all the help he provided throughout this study. Without his great ideas, comments, corrections and patience; there was no way for me to complete this thesis. I also would like to thank him for introducing me the area of Bioinformatics. With this great area of study, I can now make use of my interest in biology while working on the area of computer science. One of the reasons for me to decide on working as an academician is my appreciation on his work and his helpful, considerate and caring personality. Thanks for being one of the role models in my life.

My family deserves the greatest appreciation for making me who I am. Without the risks they have taken, it would not be possible for me to be in the place I am right now. Thanks for supporting me while I am following my dreams about my life.

My dearest friends; Hande Çelikkanat, Selma Süloğlu, Nilgün Dağ , Sinan Kalkan and Burçin Sapaz (There is no order of you in my heart but ladies first:). I do not know how these two years would be without your friendship but I am sure that it would not be the best two years of my life. Thanks for your friendship, your care, your help and your everything. I will never forget the great times we had together. The insight I got from you about life, being an academician and being a friend is of great value.

I would like to thank everybody who taught anything to me. From primary school to college, every instructor who provided me a piece of information about life are of great importance to me. Knowledge is power and I feel very strong with the things I learned from you.

Finally I would like to thank everyone who believe in me.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

List of Algorithms

CHAPTER 1

INTRODUCTION

Proteins are the basic building blocks of the cellular processes. All the activities occuring within a cell are performed with the interactions of proteins. Proteins are gene products that are produced as a result of the process called central dogma. The information coded on a DNA sequence is first transcripted on a messenger RNA (mRNA). The transcripted mRNA leaves the nucleus and transfers the coded information to ribosomes that are located in cytoplasm. When the mRNA's bind to ribosomes translation event starts and the proteins are synthesized according to the encoded information. This process is summarized in Figure 1.1.

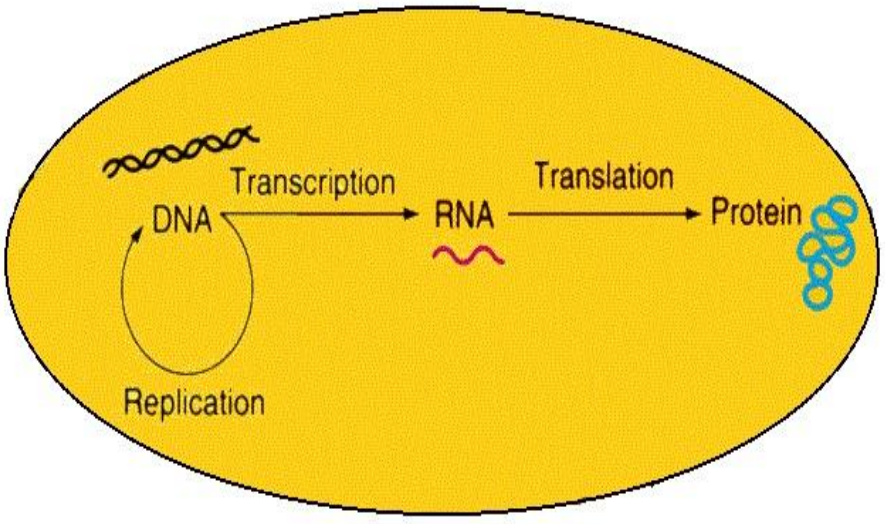

Figure 1.1: The visual description of the Central Dogma process (taken from Griffiths *et al.*, 1996)

Two genes or gene products that are descendants of a common ancestral DNA sequence are called homologous. Homology may occur either by speciation or duplication events. Genes

or gene products in different species that evolved from a common ancestor are called orthologous. Orthology occurs as a result of speciation event. Orthologous genes are functionally related. They perform similar functions in different species. On the other hand, genes or gene products within a genome that are descendants of a common ancestral gene are called paralogs. Paralogy occurs as a result of duplication events. Paralog genes or gene products do not have the same function since duplication events occur for evolving new functions. However the functions they perform may be related to each other.

Detection of functionally orthologous protein groups is an area that attracted many researchers in the last years. Identification of such related groups of proteins is important for determining the functional modules playing a role in a cell. Comparison of protein groups in different species have many application areas in genetics, disease research and drug discovery. Nowadays, a newly discovered drug is tested on mice (*Mus Musculus*) or chimpanzees (*Pan Troglodytes*) prior to testing on humans (*Homo Sapiens*). The studies determining the similar cellular functions between different species made these tests possible and reliable. Identifying similar protein groups allows the scientists to predict the function and behaviour of a group of proteins within a cellular process. By determining the evolutionarily and functionally related protein groups, it is possible to understand the function of a protein in more detail by considering previous studies performed on similar protein groups in different species.

Graph theory includes various algorithms that can be applied in the area of biological networks. Many biological network problems can be reduced to graph partitioning and graph alignment problems. For the problem of identifying functionally orthologous protein groups within species, protein network alignments are applied using different similarity metrics. These alignments are performed either globally or locally. Global network alignments are usually used for finding the best overall alignment between the protein networks of two species. Because of the duplication and speciation events, this type of alignment is quite difficult to perform. On the other hand, local network alignment strategy is used to compare two subsets of proteins from the considered proteomes.

This chapter is divided into two sections for the sake of clarity. In the first section, an overall summary of the studies performed in this area is provided. The related tools and databases available are also described in this section. The studies are grouped depending on the scope they are related. In the second section, our contributions are summarized. The problems

attacked are defined and an overall summary of the steps of the solution is given.

1.1 Studies and Tools in the Literature

1.1.1 Studies on Protein Interaction Networks

Detection of the orthologous groups of proteins is an important research task since predictions on the behaviour of a protein can be made with this information. Orthology information represent direct relation in the tree of evolution. By detecting orthologous groups of proteins, functionally related protein groups can be determined. By mapping these orthologous proteins within different species, several structural or functional information can be discovered without any experimental effort. This computational mapping of orthologous protein pairs can even be used for drug discovery.

Current research in the area of proteomics is mainly focused on the prediction of protein function using protein sequences and protein interaction information [5, 9, 17, 19, 20, 21, 23, 36, 37, 38, 43, 45]. Orthology information about proteins is also commonly used while predicting function. But, there are not any studies which try to predict the orthologous groups of proteins by using the GO terms of a protein.

The Gene Ontology (GO) terms are used to define several aspects of proteins in a standardized way [14]. GO annotation is the *de facto* standard used for evaluating studies performed on proteins. GO ontologies can be classified into three groups depending on the aspects they define; namely, the molecular function of a protein, the biological process that the protein takes part in and the cellular component the protein is located. The GO terms are defined in a hierarchical manner. In the hierarchy of the GO terms, the terms closer to the roots provide a general description while the terms closer to leaves are more specific. In order to be able to use GO terms as a similarity measure between two proteins, a formulation of a similarity needs to be defined. In the study performed by Wu *et al.* [45], a normalized distance metric is provided for evaluating the similarity of two proteins by using the GO terms associated with the proteins. This metric uses the hierarchical tree of GO terms in order to find the amount of similarity between two proteins. It consists of three parameters. The first parameter measures the distance between the most recent common ancestor of two GO terms and the root of the GO term tree. The second term calculates the maximum distance from the considered GO

terms to their descendant leaves. The last term measures the shortest distance between the two terms. Combining these parameters and normalizing them with the use of the depth of the GO Term tree, a similarity metric between two GO terms is constructed. For the similarity of two proteins, the maximum similarity is taken into account after computing this similarity measure for each GO term that a protein is related to. This approach is a simple and elegant way to compute the similarities of two GO terms. This similarity metric is used in the series of methods applied in this study in order to evaluate the similarity of two proteins. The details of the scoring algorithm can be found in Section 2.2.3.1.

In many of the studies, conserved protein interactions are found by orthology. However, in the study of Bandyopadhyay *et al.* [5], they aim to use conserved protein interactions in functional orthology prediction. The main idea of their study is "A protein and its functional orthologs are likely to interact with proteins in their respective networks that are themselves functional orthologs.". The algorithm applied in that study consists of three main steps. As a starting point, the orthologous clusters of proteins are determined [29]. After generating the orthologous protein clusters, these clusters are aligned by the application of a global alignment algorithm on the whole proteomes of the two considered species, namely *Saccharomyces Cerevisiae* and *Drosophila Melanogaster*. Then a conservation index is calculated with respect to the degrees of the matched proteins and the conserved interactions they have. This index is used to generate a probabilistic model that makes inferences on whether the proteins are orthologous or not. After training this probabilistic model with a subset of the whole data, inferences are made on the orthology status of proteins. The conservation index defined in that study is used as a parameter of the similarity metric used for evaluating the similarities of two clusters.

In [38, 37], a pairwise global alignment method named ISORANK is proposed. With this method, it is possible to align the protein networks of two species and find functionally orthologous protein pairs. The method uses the information about the protein sequences along with the network of protein interactions. The intuition behind the algorithm is similar to Google's PageRank Algorithm. By performing random walks on the protein interaction networks of two species, the similarities between each possible pair of proteins is determined. By ranking each possible pair of proteins depending on the conservation of the interactions with their neighbors, a similarity matrix of proteins is constructed. This similarity matrix is used to determine the most similar protein pairs and align them. By matching and eliminating

most similar proteins one by one, a global alignment of the protein interaction networks of two species is constructed. After performing the alignment between the species *S. Cerevisiae* and *D. Melanogaster*, evidence from the Inparanoid database showed that the functionally orthologous proteins are matched as a result of the performed alignment. The algorithm is claimed to be error tolerant and this is validated by removing two edges from the network data depending on a predefined probability and introducing two new edges which do not exist in the original network. Preserving the node degrees in the network this way, tests on error tolerance of the method could be performed. The results show that the method is error tolerant with 0.2 percent points of accuracy reduction as a result of the randomization performed with a probability value of 0.5. Presented in RECOMB'07, it is one of the most cited studies in Google Scholar on the problem of protein network alignment. This method is used as a benchmark to compare the performance of the algorithm proposed in this thesis because of this wide acceptance of the proposed method.

PathBLAST [19, 20] is another widely accepted method for the problem of protein interaction network alignment. It is designed to find a match between a given pathway and a subject protein interaction network. By the use of the method, it is possible to perform functional annotation on a group of proteins. This method uses sequence similarity in order to evaluate the similarities of two protein interaction networks. The method allows gaps in the alignments which provides flexibility on performing matches. The tool is accessible via web. The web based software can align a group of proteins to protein interaction networks of a variety of well-known organisms.

Another study using protein interaction networks and protein sequence similarities in order to find conserved patterns of protein interaction in multiple species is the study performed by Sharan *et al.* [36]. The aim of the study intersects with our study since both of the studies aims to find clusters of proteins that are conserved during the speciation event of different species. Their algorithm, named NetworkBLAST, first performs global alignment on the protein interaction networks provided. This alignment is performed using the protein sequence similarities of the proteins in the provided networks. Then this global alignment is used to determine the seeds representing conserved subnetworks. Using these seed nodes, the conserved subnetworks are expanded with the use of a probabilistic model. Experiments on the developed method are performed on the protein interaction networks of three different species, namely *Saccharomyces Cerevisiae*, *Caenorhabditis Elegans*, and *Drosophila Melanogaster*.

It is claimed that protein functions and protein interactions can be predicted with the application of the method. The achieved results are tested using two-hybrid analysis and validated by cross validation. Since NetworkBLAST aims to find orthology information about protein clusters just like our proposed algorithm does, it is possible to compare the results of the two studies during the evaluation of the developed algorithm.

In the study of Brun *et al.* [9], a method for the functional classification of proteins is proposed. In this method, the protein interaction and protein sequence information are used in order to construct a hierarchical tree of proteins. In this tree, the proteins are positioned such that functionally similar proteins are close to each other. This hierarchical tree used for clustering and determining the functional classes of uncharacterized proteins. These classes are determined by sequence alignment analysis and robustness measurements.

The study perfomed by Hirsh *et al.* [17] tries to define the conserved protein complexes by considering the possible evolutionary changes. As a result of the gene duplication events and changes in linkage dynamics of protein interactions, two conserved protein complexes may appear differently in two species. By generating a probabilistic model that the conserved protein complexes between different species fit, they try to model evaluationary generation of protein complexes.

Similar to the study of Hirst *et al.* [17], Koyutürk *et al.* [21] have also proposed a solution to find conserved protein groups considering the evolutionary changes. Using the duplication and divergence models, they try to extend the idea of protein sequence alignment to protein network alignment. The match, mismatch and duplication events on protein network alignment are considered as matches, mismatches and gaps in protein sequence alignment. The alignment is constructed by considering protein orthologies and evaluated using the evoluationary events.

Letovsky *et al.* [23] tries to resolve a different problem using protein interaction networks. The aim of their algorithm is to predict function of proteins. They predict the GO terms of unlabeled proteins by considering the GO terms of the labeled neighboring proteins. The method is based on the local density enrichment. It is assumed that unlabeled proteins are more likely to have GO terms similar to the GO terms of the proteins they interact. By using Markov Random Fields to iterate over the protein interaction network, they perform this prediction.

The survey written by Watson *et al.* [43] describes a list of approaches for predicting the functions of proteins. The methods used for function prediction are grouped on two main groups, the sequence based methods and structure based methods. The paper provides an overall view of these approaches. It is stated that usually several approaches are combined in studies to get more accurate prediction results.

Graph partitioning is also applied on protein interaction networks in order to determine the functional modules of proteins. The main idea in these approaches is that functional modules are in fact strongly connected subgraphs of the protein interaction networks. One of the most elegant and efficient solutions on this problem was the one proposed by Macropol *et al.* [25]. This algorithm, named Repeated Random Walk (RRW), begins a number of random walks starting from each node in the graph. While iterating over the graph nodes, the weights of the edges related with the node are used to determine the probabilities of the possible following states. These states also include a restart probability for which the random walker turns back to the starting node. At each arrival to a node, the probability of using the node is updated. This iterative algorithm is applied for each node in the graph until a convergence of node probability values occurs. After the probability values of the nodes are determined this way, clusters of similar probability nodes are extracted from the graph. Several experiments show that the method performs better than a benchmark clustering method named Markov Clustering Algorithm by means of both precision and accuracy.

The study performed by Chen *et al.* [11] proposes another solution to functional module detection. By using the betweenness-based partitioning algorithm, groups of proteins that form a functional module are determined. The application of the proposed method on *Saccharomyces Cerevisiaee* showed that known protein complexes in literature are successfully identified by the method. A relatively older study performed by Pereira-Leal *et al.* [30] also performs unsupervised clustering on the protein interaction of *Saccharomyces Cerevisiae* to find functional modules. By applying an unsupervised clustering algorithm named TribeMCL and using the confidence values of protein interactions as the edge weights of the interaction network, they identified 1046 functional modules from the *Saccharomyces Cerevisiae* protein interaction network. The tool named PRODISTIN which is developed by Brun *et al.* [9] finds the functional modules of protein interactions by using hierarchical clustering. Using the functional similarities and protein sequence similarities, a hierarchical tree of protein similarities are formed. The hierarhical tree is used to classify the proteins and determine the

functional modules. Their method is able to classify 11% of the proteins of the proteome of *Saccharomyces Cerevisiae*. The study performed by Milenkovic *et al.* [27] finds the functional groups of proteins by considering the vector of graphlet degrees. Trying to fit the local protein interactions into a random graph, they define the functional modules in protein interaction networks.

Although there are many studies trying to find the functional modules of protein interaction, a different study performed by Milenkovic *et al.* [28] tries to fit the real protein interaction networks into random graph models. The study is named GraphCRUNCH and provides a variety of global network measures for use while fitting the real world network into a randomized graph model. A web based tool implementing the described method is available. Parallel programming is used to compute the results. So it is possible to analyze and model biological networks in a fast manner. As a future work of the study, identifying the model of protein interaction network will allow the alignment of protein interaction networks. Another study trying to fit a geometric graph to a protein-protein interaction graph is performed by Higham *et al.* [16]. This main contribution of this study is that they prove that protein interaction networks have a geometric structure. Although in these two studies suggesting that metabolic networks can be modelled with random networks, the study performed by Jeong *et al.* [18] and Tanaka *et al.* [41] suggest that these networks should be modelled using scale free networks. Naturally metabolic networks are dominated by a few highly connected nodes called hubs. These hubs link rest of the network which are less strongly connected. This structure of metabolic networks is similar to World-Wide Web.

1.1.2 Studies about Algorithms in Graph Theory

Graph theoretic algorithms are commonly used for extracting information from biological networks. Graph clustering, graph partitioning and graph alignment are the most common problems that have use with biological networks. It is possible to determine strongly connected protein interactions of a protein network using graph partitioning and clustering algorithms. Similar regions of protein interaction networks are determined by the use of graph alignment algorithms. In order to find a suitable solution for these problems, several algorithms in graph theory are considered. In this part of the text, the main focus will be on graph clustering and partitioning algorithms since they are applicable for solving many biological network prob-

lems. Some of these algorithms which are applied in a biological context are summarized in the previous section. In this section, the algorithms that are not applied on protein interaction networks are summarized.

Among numerous studies in the area of graph theory, two surveys are useful for getting an overall view of the solutions for graph theoretic problems. The survey performed by Brandes *et al.* [8] provides a list of the indices for graph clustering such as coverage, performance, intra-cluster and inter-cluster conductance. The survey compares three solutions to graph clustering problem which uses these indices. These compared solutions are Markov Clustering, Iterative Conductance Cutting and Geometric MST Clustering. As a result of the performance comparison they performed, they claim that the results produced by Markov Clustering are well but they may include some trivial clusters. On the other hand, they claim that the algorithm performs slower that the other algorithms. On the other hand, iterative conductance cutting performs faster but the authors suspect that the intra-cluster index indice used to perform the clustering does not measure the quality of the clustering appropriately.They conclude their survey by saying that Geometric MST clustering performs best among the compared algorithms.

The survey written by Schaeffer *et al.* [34] provides information about graph clustering with a great level of detail. After giving the basic definitions in graph theory, they define the measures of graphs that can be used in graph clustering. Using these definitions they define the global clustering techniques such as iterative or online computation of global clustering, hierarchical clustering, divisive global clustering, agglomerative global clustering. They also define local clustering methods used for local searches in graphs. Methods for comparing the performances of these algorithms are also provided. The text is concluded by listing a number of application areas of graph clustering. Graph clustering has usages in data transformations, information networks, database systems and analysis of biological and social networks.

Another survey on graph clustering is given by Anders *et al.* [3]. While proposing a new unsupervised clustering algorithm named Hierarchical Parameter-free Graph Clustering, with the literature survey performed, they provide an overall picture of the currently used graph neighborhood definitions. In order to model the local to global neighborhoods of their graph, several neighborhood relations in graphs are considered such as Nearest Neighborhood, Minimum Spanning Tree, Relative Neighborhood, Gabriel Graph, Dealunay Triangulation. All

these neighboring strategies are tested with the developed algorithm and compared as a conclusion of their study.

After getting an overall view of the techniques applied for graph clustering problems, several studies on graph clustering and graph partitioning are considered. An optimization on semi-supervised kernel based graph clustering approaches is suggested by Kulis *et al.* [22] in 2009. Their clustering method tries to perform graph clustering on an image dataset using the Hidden Markov Random Fields. The main advantage of the method is the ability of clustering both the vector-based and cluster-based data. They concluded their study claiming that the semi-supervised nature of the algorithm can be automatized by integration of a machine learning strategy to the algorithm. This way the required prior information can be replaced by the learned information. Although the results are promising, the method seems to suit for use in image processing applications.

In the study performed by Günter *et al.* [15], a new graph clustering technique similar to one of the benchmark unsupervised clustering techniques named Self Organizing Maps is proposed. Providing the details of the algorithm, they have also defined and compared several cluster validation indices in literature. An example application of the developed algorithm is performed on character classification problem. Another study performed by Biemann *et al.* [6] proposes another randomized algorithm for graph clustering. This algorithm named *Chinese Whispers* is similar to RRW algorithm by means of the neighboring effects of nodes but it is a lot simpler when compared to RRW. The method can be applied in various areas but the target application area in the performed study is Natural Language Processing in the paper written. A kernel based, divide and conquer graph clustering algorithm is proposed by Dhillon *et al.* [12]. The algorithm first coarses the initial input graph into as small clusters as possible. Then during the refining phase of the algorithm, the similar clusters at the end of the coarsening phase are combined to get larger clusters. This multilevel clustering technique is shown to perform faster on large graphs when compared to spectral methods. The method is applied on the Internet Movie Database (IMDB) and promising results have been achieved. A study performed by Roxborough *et al.* [32] suggests a solution based on ratio cut method used in circuit partitioning. The method is rather old when compared to other clustering strategies and for that reason can not be considered as a strong clustering method. A similarly old clustering technique developed by Edachery *et al.* [13] suggests clustering graphs using distance-k cliques. A distance-k clique is defined as “A subset V’ of the node set V of a graph

G = (V,E) is defined to be a Distance-k Clique if every pair of nodes in V' is connected in G by a path of length at most k." in the text. Trying to convert the graph partitioning problem into distance-k clique problem, the authors of the study develop a graph clustering algorithm. Although it is possible to consider this solution for use in some graph clustering problems, the solution is not flexible since it can only find distance-k cliques. Without prior information about what the *k* value should be, the method is not applicable. Another study using cliques for graph clustering is performed by Brandenburg *et al.* [7]. They define a graph as a cycle of cliques and they try to partition a given graph into a number of cliques by breaking this definition of cycle into pieces. The problem of determining a cycle of cliques is proven to be NP-Complete. The method they proposed may be successful in determination of the cliques in a graph. But the aim in graph partitioning is not always determination of cliques. On the other hand, this approach would result with a number of single nodes. For these reasons, the method should not be considered as a strong clustering technique. In 2002, Luo *et al.* [24] have proposed another graph clustering method. This method tries to perform graph clustering by using the graph-spectral features. The clustering is performed by applying multi-dimensional scaling on the eigenvectors constructed by using graph-spectral features as eigenvalues. The performance of the method is evaluated by applying on sequences of image data. Finally Rizzi *et al.* [31] proposed a genetic algorithm based solution for graph clustering problem. This algorithm required Euclidean space and a fitness function together with the graph to be clustered. The algorithm generates a hierarhical tree of connected subgraphs generated from the whole graph and evaluates the clusters depending on the defined fitness function until the best possible clustering is achieved.

Apart from these methods, Sablowski *et al.* [33] developed a tool to cluster graphs with nodes less than 3000. This tool applies the Basic-ISODATA algorithm for clustering. Euclidean distances between nodes are considered and the graph is divided into a number of clusters which is provided by the user prior to the execution.

A relatively different study performed by Bunke *et al.* [10] tries to produce a representation for clusters of graphs. Despite of the success of the method in embedding the structural information into the cluster and removing noise from this information, the method is computationally expensive. The computational cost of the method is tried to be overcomed by proposing an approximation to the original method. Even with the loss of information caused by the approximation method, it is possible to use the representation successfully while performing

graph clustering.

1.1.3 Databases for the Extraction of Biological Information

When starting a study in the area of bioinformatics, one of the most challenging tasks is finding a reliable and complete dataset to work on. Depending on the scope of the study, there exists several different database options that can provide the datasets to work on. Among this variety of database options, it is difficult to decide on the database that meet the requirements of the study being worked on. Also another problem in choosing the correct database is that the annotation used in the database should be one of the standardized annotations that is commonly used in literature. Otherwise finding information about the elements included in the database becomes difficult and this causes problems especially during the validation of the performed study.

Since the scope of this study is based on protein interaction networks, a suitable database that include protein interactions should be determined. In the survey written by Xenarious *et al.* [46], an overall view of the databases that includes protein interaction information are given. Apart from describing the most commonly used protein interaction databases, various types of information are given about the construction of the databases such as how the interaction information is extracted, how various types of protein interactions are encoded and why confidence levels of interactions are needed. Providing short descriptions of the databases such as BIND, MIPS, PROTEOME, PRONET, CURAGEN and PIM, the study provided us insights during the database selection process.

Among the various options of protein interaction databases, three databases attracted our attention most. Database of interacting proteins (DIP) [47] is one of the most frequently used databases for extracting protein interaction information since it is built in 2001. Currently this database holds 70411 protein interactions between 22630 proteins of 274 species. It is frequently used in studies on protein interaction networks. The proteins are identified by DIP accession number. However it is also possible to reach the SWISS-PROT, GenBank and PIR id's of proteins through this database.

Another frequently used protein interaction database is the Biomolecular Interaction Network Database (BIND) [4]. This database has a more general definition of interactions. It does

not only cover protein interactions but also interactions between small molecules and nucleic acids. It is also possible to describe chemical reaction, photochemical activation and conformational changes using this database. Currently the BIND database has been upgraded under the name Biomolecular Object Network Databank (BOND) with a group of tools to query and process the data inside. The database can be accessed and processed through programs constructed on SOAP architecture. The database includes information about 188517 interactions for the moment.

STRING [26] is another good option for getting protein interaction data. It is first developed in 2003 and it is frequently used in studies on protein interaction networks. Enseml Protein ID's are used to annotate the proteins in the interactions. One of the major advantages of the database is that it provides not only experimentally proved protein interactions but also interactions predicted by several computational methods along with their confidence values. The interactions in the database include physical interactions and functional associations. As of August 2010, the database includes information about 2590259 proteins of 630 organisms. One of the main advantages of this database is that it is frequently updated to include the latest experimental and computational information. Another advantage of the database is the ease of access to different annotations of proteins with the search tool provided in the project web site. Although there exists several other alternatives such as BioGRID [39], STRING fulfill the requirements of our study. It is a realiable, up-to-date and large database that can provide the interaction information we need to use.

After deciding on the database to be used, a new source of data is required to succesfully apply the developed algorithm on the extracted dataset. The GO terms of the proteins extracted from STRING database are required to be determined since the developed algorithm makes use of this information in order to find the similarities between protein pairs. There are easy-to-use web based tools for extracting GO terms of a protein. "Clone/Gene ID Converter" [1] and "Babelomics" [2] are the most popular tools used for this aim. While searching all the GO terms associated with the protein, they can also determine annotations of the proteins in different standards. For example, it is possible to find the SWISS-PROT annotation of a protein given the ENSEMBL gene or protein id. These annotation conversion tools are frequently used while working with datasets from different sources. Although "Clone/Gene ID Converter" and "Babelomics" do not have any advantages over each other my means of the quality of the data returned, "Babelomics" provide a wider range of organisms to be queried.

”Clone/Gene ID Converter“ provides support for only *Mus Musculus*, *Rattus Norvegicus* and *Homo Sapiens* , ”Babelomics“ provide support for 11 different species also covering the species supported by ”Clone/Gene ID Converter“. The annotation types that they cover differs for less popular annotation standards. But they all cover the main annotation standards.

During the evaluation of the results achieved by the application of the developed algorithm, orthology information about the proteins are required. STRING database [26] provides an orthology ontology for which the distances between the orthological terms are defined. This orthological terms and distances are taken from the COG database [42]. Developed in 2001, COG database provides an ontology for defining the phylogenetic lineages between proteins. It is possible to find out the orthological closeness of two proteins with the usage of this database. But there is also another database alternative called Inparanoid [29] which provides information about the orthologies of proteins pairs. This database is constructed by a method named Inparanoid which tries to find the orthological protein pairs of different protein interaction networks. Although many orthology detection studies use Inparanoid database to validate their results, the database does not have a built-in ontology for defining orthological groups of proteins. They just define the orthological distance between two proteins. This strategy of orthology determination does not provide the distance of two proteins that are not defined in the database. Usage of an ontology is certainly superior on understanding the distances between proteins when compared to such an approach.

Another database that helps for validating the results of the study is the 'Interolog/Regulog Database' developed by Yale Gerstein Lab [48]. The name interolog stands for conserved protein interactions between two ortholog protein pairs. The database keeps information about orthologous protein interactions that are conserved among different species. By looking for protein sequence similarity and determining ortholog proteins, a combined score of sequence similarity is produced. This combined sequence similarity score is used to determine the conserved protein interactions between species. The application of this method resulted with a list of orthologous protein interactions, namely interologs. This interolog list is available for use as an online database. In this database all the predictions performed are not provided but the top scoring 1% of the predictions are included. So the predictions in the database have high confidence values. This database is proved to be reliable by applying two hybrid experiments on the 45 predicted interologs. The two hybrid tests confirmed that the predictions on interologs are correct. The validity of these results are proved by showing the statistical

significance of these results with the computation of the P value. By another case study performed on Ste5-MAPK complex, they have predicted five of the six subunits in yeast based on only one MAP kinase in worm. With all these validation strategies, interolog is shown to be a powerful and realiable method for determining conserved protein interactions between species. The method has a different aspect which determines the conserved protein-DNA interations named Regulogs. But this part of their study is out of the scope of our study.

1.1.4 Tools for Visualizing Protein Networks

Although visualization of protein interaction networks are out of the scope of this study, visualization tools are used to view the results of the performed alignments. The survey written by Sutherman *et al.* [40] mentions the main problems in biological network visualization and provides a list of tools that can be used for this purpose. The main problems in visualizing the protein interaction networks is the number of nodes and edges that should be rendered. The illustration should be simple and understandable while grouping the similar groups of proteins close to each other. Avoiding the overlaps of nodes and edges is a difficult task alone. When the criteria of keeping similar proteins together is added, the problem becomes a lot more complex. Also since the dataset to be visualized is too large to be understood all at once, there exists a need for querying and filtering the rendered data. Several software tools developed for this purpose are discussed in the survey. The advantages and disadvantages of the tools named Pathway Studio, Cytoscape, Osprey, Patika, VisANT, ProViz, and BiologicalNetworks/PathSys are discussed helping the users to choose among them. Also several network layout strategies are introduced to the users such as circular, hierarchical, force-directed and simulated annealing.

Among the tools introduced in the survey [40], CytoScape [35] seems to meet the needs of this study. Cytoscape is one of the most frequently used software tools for the purpose of biological network visualization. Since it is an open-source software and it is possible to extend the functionalities of the tool by implementing plugins, it is well accepted by the bioinformatics community. Cytoscape is capable of rendering huge protein interaction networks. It allows the usage of several network layout strategies. It also provides filtering functions for the ease of processing of the rendered graph. All these functionalities make Cytoscape a good choice for the visualization of alignments performed in this study.

However for getting an overall view of all the clusters formed during the implementation of the algorithm, Cytoscape was not fast enough to generate the visual images of all the clusters formed. For automatic generation of the cluster images, another tool named yFiles [44] is used. yFiles is a Java-based API for visualization and automatic layout of graph structures. Applying the features of this tool on the clusters formed during the application of our algorithm, it was possible to compare the resulting cluster matches visually.

1.2 The Scope and Contribution of the Thesis

In this study, we have performed the implementations of the explained methods using Java. Java is a practical programming language in bioinformatics studies. Since it is not dependent on a specific operating system or platform, it enables the usage of the produced executables on any platform. In order to compute the cluster matches, many data searching and retrieval operations are required. The solution used to retrive the required data as fast as possible during the execution of the program is creating an organized database and keeping all the data in this database. Since a database organizes and indexes the data inside, searching and retrieving data is fast and easy. For this purpose, MySQL is used as the database server. Because of the full support provided online and the easy integration with Java, it is determined to be used in the implementations of the methods. The JDK version used for the implementation is JDK 5 and the MySQL version used is 5.1. The implementations are completed in Windows 7 enviroment.

During the implementation, as a first step, functionally related groups of proteins are determined using only the protein interaction networks taken from the STRING database [26]. For this step of the solution, the Repeated Random Walks Algorithm [25] is applied directly without any major changes. The second step of the algorithm compares formed protein groups of different species and finds functionally similar groups of proteins between different species. By using the protein interaction networks and GO Annotations of proteins, our solution not only determines the functional modules of proteins but also relates them between two species.

Studies until today were focused on determining orthologous protein pairs. There are not many studies trying to match protein clusters of different species and trying to detect orthologically related protein groups. The study performed by Singh *et al.* [38, 37] tries to solve a

similar problem with ClustOrth. But their solution relates protein orthologies in the different species not protein cluster orthologies. On the other hand, our algorithm is able to relate both single proteins and clusters of proteins. The NetworkBLAST algorithm [36] has the same purpose as our study. However, the followed approaches of the two studies differ. NetworkBLAST first performs a global alignment over the protein interaction networks. However our solution first defines the clusters in the protein interaction networks and then it tries to find the orthologous relations between these clusters. Furthermore, our results show that the proposed algorithm in this thesis outperforms NetworkBLAST.

Most of the studies related to the protein interaction network alignment use sequence based similarity metrics during their alignment processes. To our knowledge, no protein network alignment algorithm uses the GO Annotations of the protein as a distance metric. There are a small number of studies for comparing the similarities of two GO terms. The solution proposed in this thesis suggests using GO terms of proteins as a distance metric for aligning protein interaction networks.

Similar methods in the literature mostly align and compare the two well-known organisms, namely *Saccharomyces Cerevisiae* and *Drosophila Melanogaster*. These two organisms have been used as benchmark datasets in almost all the studies performed in the area. Because of the computational complexity introduced by the highly evolved organisms, to our knowledge, there are not any computational studies working on *Mus Musculus* and *Homo Sapiens*. These two organisms have many similarities since they are close to each other in the evolutionary tree and they should be compared computationally. Another contribution that this study provides is the comparison of these organisms. *Mus Musculus* and *Homo Sapiens* protein interaction networks are selected as the benchmark datasets of the study. So, our final results suggest similar clusters of proteins from these two organisms.

CHAPTER 2

MATERIALS AND METHODS

In this chapter, the details of the methods applied to find out a mapping of functionally orthologous groups of proteins in different species are explained in detail. The functionally orthologous groups of proteins are found by using the protein interaction networks of the two species and GO Annotations of the considered proteins. These data had to be extracted and organized from several databases. After extracting and preparing the datasets for use, a series of methods are applied in order to find the functionally related protein groups and mapping these functionally related groups between species to find common biological processes of two species.

2.1 Construction of the Dataset

There are two major information types used in this study, namely the protein interaction networks and the GO Terms of the proteins in these networks. Although the method is applicable for finding the functionally orthologous groups of proteins between any species, we prefered to apply and test our method on *Mus Musculus* and *Homo Sapiens*. The reason for us to choose these organisms is that they contain more protein interactions and more functional complexity when compared to other organisms. They can be considered as the most evolved organisms for which the genome is fully sequenced. Also many sources prove that many functionally similar cellular processes exist between these organisms. These two organisms are close to each other in the evolutionary tree. These similarities may result with biologically more meaningful results if the method works well. Although these two organisms are quite popular in literature, we could not find a computational method that is applied on these two organisms to find the biologically related functional processes of these organisms. The

reason for this situation is most probably related to the increase in computational complexity with respect to the number of proteins in the provided protein interaction networks. Highly evolved mammalian protein interaction networks are difficult to be handled by computational methods especially when protein sequence similarity is used as the similarity metric between proteins. This is the reason for the use of relatively simpler proteomes for the evaluation of computational studies in the area.

The STRING Database [26] is used to extract the protein interaction networks of the organisms *Mus Musculus* and *Homo Sapiens*. Although there exists many databases such as DIP [47], BIND [4], BioGRID [39] that we can extract interaction information; we prefer to use the STRING database. STRING database is becoming quite popular especially in the recent studies. The statistics they provide at the homepage of the database show that it is frequently used and many studies are performed using this database. It is easy to search a specific protein and its interacting partners with the use of the search engine provided in the database web site. The answer to the query is returned visually in an easily understandable way. Any information about the queried protein can be reached using the list of references provided with the search results. On the other hand, the dataset can be downloaded as a whole in a text file. This downloadable text file is organized in an easy to parse way. The most important property of the database is it is updated twice a year introducing newly discovered interactions. This provides the opportunity to work with the most recent data. All these advantages lead us to use this database for finding the protein interaction networks we need.

In this database, interactions in the networks are provided with different levels of confidence values depending on the reliability of the method that suggested the interaction. The interactions with a confidence value less than 400 are defined as low confidence in the database. In this study, low confidence interactions are not considered in order to avoid false positive interactions. For that reason, the interactions with a confidence value lower than 400 are eliminated from the protein networks used. Although it is possible to eliminate some of the methods used for predicting the interactions included in the database, no constraints related to interaction prediction method are included while selecting the interacting protein pairs.

The second type of information to be used with our method was the GO Annotations of the proteins in the protein interaction networks. The GO Terms are products of a huge project Gene Ontology Project [14]. The aim of this project is to standardize the functions and prop-

erties of genes and gene products. With the use of GO Terms, it is possible to define the cellular component that a gene product is active, the biological process that the protein has a role and the molecular function that gene product has. In other words it is possible to define the role, molecular and cellular properties of a protein in a standardized way by this annotation strategy. Usage of this information allowed us to determine similarities between proteins by considering different aspects. In order to find the GO Annotations of the proteins in the *Mus Musculus* and *Homo Sapiens* protein interaction networks, a web based tool named "Clone/Gene ID Converter" [1] is used. The proteins in STRING database are annotated with Ensembl Gene Annotation. Unfortunately this annotation model is not the one used in the ontology database files provided in the official web page of Gene Ontology Project. This tool works as a web based search tool for mapping different annotation methods of proteins and genes. It also searches and lists the associated GO Terms of a list of proteins annotated in any standard annotation model. With the use of this tool, it was possible for us to determine all the GO Terms associated with the proteins in the used protein interaction networks.

After processing the data extracted from the above mentioned databases and tools, the following dataset files are prepared for use with the designed method:

- The list of protein interactions with their confidence values for both the *Mus Musculus* and *Homo Sapiens* protein interaction networks
- The list of proteins in the protein interaction networks together with their corresponding GO Term lists

With the use of these two types of information, we managed to find functional protein groups and perform a mapping between the protein groups of *Mus Musculus* and *Homo Sapiens*.

2.2 Functional Orthology Mapping

After getting the dataset ready as described in the previous section, a series of methods are applied in order to discover the functionally related orthologous groups of proteins between two species. For this purpose, whole protein interaction graph of species is divided into subgraphs of strongly connected nodes. Repeated Random Walks Algorithm [25] is applied on the protein interaction networks of both species for generating these strongly connected

subgraphs. After this process, we aimed to find a mapping between these subgraphs of the two species. In order to find a mapping, the subgraphs are first considered for their similarity by means of their graph theoretic properties. An elimination on matches that are not likely to be related is performed this way. After the elimation, the GO terms of the proteins are used to have a semantically meaningful cluster match. By considering the GO terms of the proteins, the possible cluster matches are scored and a sorted list of cluster matches is produced by means of this scoring scheme. These main steps of the algorithm are illustrated in Figure 2.1. The details of this process is explained in the following subsections.

2.2.1 Extraction of Strongly Connected Protein Groups

The first step of our method is based on determining strongly connected nodes of the whole protein interaction network. There exist several studies showing that strongly connected proteins are similar by means of their function. So it is possible to determine functional modules from the whole protein interaction networks by considering the graph theoretic properties of the network.

Although there exist many different algorithms for detecting strongly connected subgraphs in networks, most of them are similar by means of using the Google's PageRank Algorithm as the main idea. Among the various choices, a recent study performed by Macropol *et al.* [25] was superior compared to other studies. With the parametric nature of the method, their solution provides the flexibility to determine the maximum and minimum cluster sizes, overlap thresholds of the subgraphs formed and many algorithm dependent parameters.

The algorithm starts by some repeated walks from each of the nodes in the network. The walks are traced with a probability relative to the edge weights of the graph. A random walker starting from a node determines which node to go next by considering the relative weights of the edges. Although this determination process is performed randomly, the relative edge weights and random start probabilities play a curricial role. After performing the walk for sometime, the nodes of the network gets some importance values by means of the number of times they are visited. These importance values determine the strongly connected nodes of the network.

With the application of this method on our *Mus Musculus* and *Homo Sapiens* protein interac-

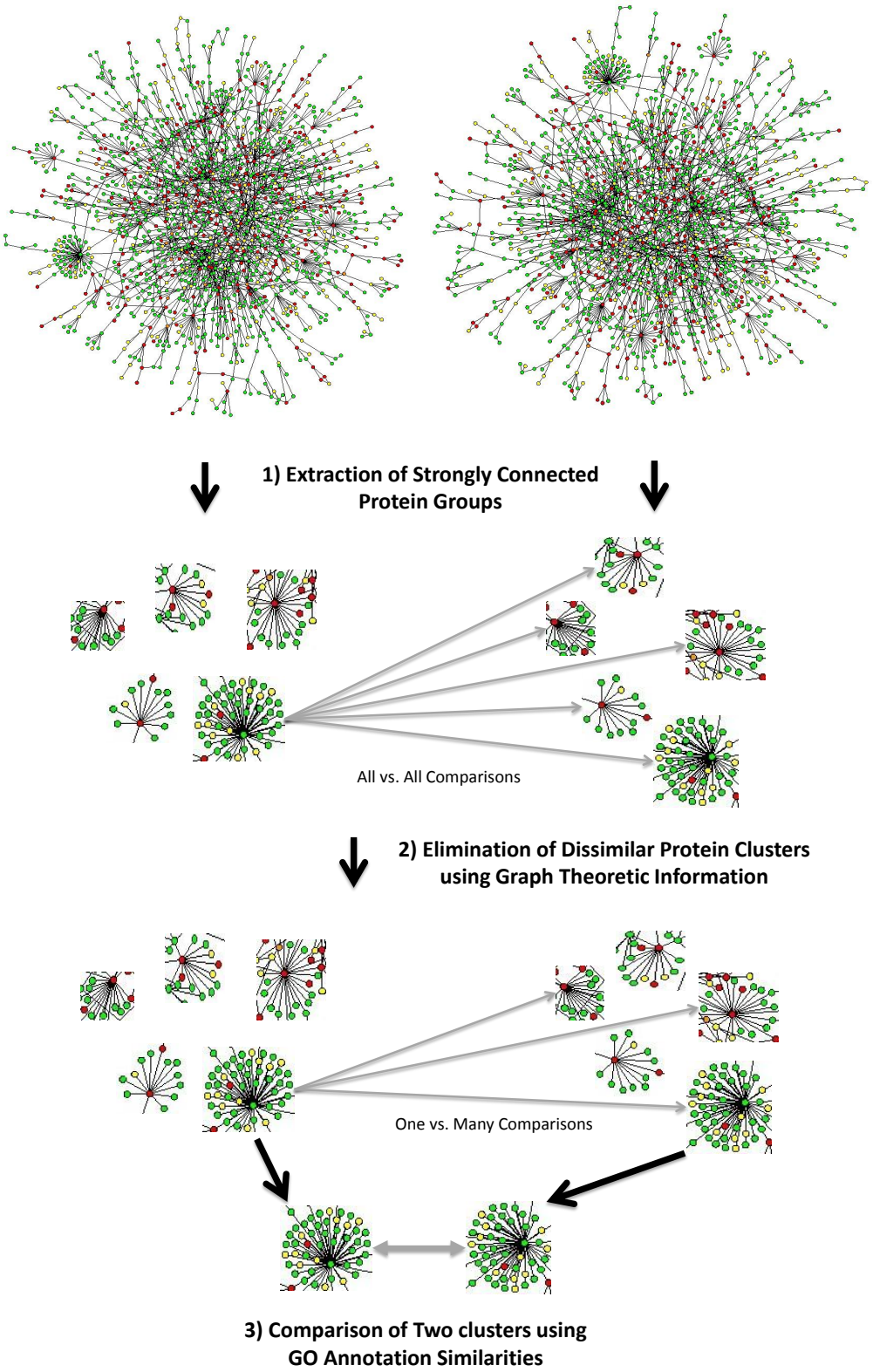

Figure 2.1: The illustration of the main steps of the algorithm

tion networks extracted from the STRING Database, we determined the functional modules in these networks.The algorithm is applied with a minimum cluster size of 5 nodes and maximum cluster size of 30 nodes. These sizes are determined by considering the biologically meaningful functional modules in the literature. In fact, some outlier groups are ignored with this selection but the results cover most of the well-known functional modules. Applying this algorithm seperately on the protein interaction networks of *Mus Musculus* and *Homo Sapiens*, 1065 clusters from *Mus Musculus* network and 1346 clusters from *Homo Sapiens* network are formed. The next step of our method is relating these 1065 clusters of *Mus Musculus* proteins with 1346 clusters of *Homo Sapiens*.

2.2.2 Elimination of Dissimilar Protein Cluster Mappings Using Graph Theoretic Information

The computational cost of considering 1065 clusters for similarity with 1346 clusters is high. When performed without any elimination on the number of clusters to be considered, 1433490 pairs of clusters should be compared by means of graph theoretic similarity and similarity by means of GO Terms. In order to reduce this complexity and the number of comparisons required, the pairs of clusters that are not likely to be matched are eliminated by a simple graph theoretic approach. In this elimination, the number of nodes and edges in the clusters are considered.

Although the number of proteins and protein interactions in a functional module can increase or decrease as a result of duplication and speciation events, it is nearly impossible that a cluster with 5 nodes is functionally orthologous with a cluster of size 30. Similarly a cluster which is losely connected with interactions is not expected to be functionally related with a cluster which is strongly connected. Using this logic, an elimination on the number of possible cluster matches is performed on the constructed clusters.

The first criteria used to eliminate the clusters is the number of nodes that the two clusters have. Assume the number of nodes in cluster from *Mus Musculus* is n_1 and the number of nodes in cluster from *Homo Sapiens* is n_2. The elimination criteria accepts or rejects the

possible matches according to the criteria defined in Equation 2.1.

$$criteria1(n_1, n_2) = \begin{cases} accept & \text{for } 0.75n_1 \leq n_2 \leq 1.25n_1 \\ reject & \text{otherwise} \end{cases} \tag{2.1}$$

In other words, the clusters with similar node counts are taken into account for a possible match. As the node count of the clusters increase, more number of changes are allowed since the occurence of duplication events is more likely. We have taken the tolerance range relative to the number of nodes in the cluster with this formulation. We allowed a change in number of nodes in the cluster with a 0.25 fraction of increase or decrease. We have determined this fraction constant by considering several clusters of different sizes. The constant satisfied our expectations for clusters of size from 5 to 30.

If a match passes the first criteria, it is tested with the second criteria. The second criteria considers the relative number of edges of the two clusters. This criteria tries to compare the compactnesses of the two clusters. Assume the number of edges of the cluster from *Mus Musculus* network is e_1 and the number of edges of the cluster from *Homo Sapiens* network is e_2. This criteria eliminates the cluster matches that do not satisfy the Equation 2.6.

$$minEdgeCount_i = n_i - 1 \tag{2.2}$$

$$maxEdgeCount_i = \frac{n_i \times (n_i - 1)}{2} \tag{2.3}$$

$$range_i = maxEdgeCount_i - minEdgeCount_i \tag{2.4}$$

$$comp_i = \frac{e_i - minEdgeCount_i}{range_i} \tag{2.5}$$

$$criteria2(n_1, n_2, e_1, e_2) = \begin{cases} accept & \text{for } 0.75comp_2 \leq comp_1 \leq 1.25comp_2 \\ reject & \text{otherwise} \end{cases} \tag{2.6}$$

The compactness annotated with $comp_i$ in the equations is a value between 0 and 1 representing how strongly a cluster's nodes are connected. The compactness value in Equation 2.5 is calculated by evaluating the position of the number of edges in the range determined by the maximum number of edges and minimum number edges that a cluster has with regard to the number of nodes it has. In Equation 2.6, the compactness value is allowed to change with a fraction of 0.25 percent. This constant value is selected by considering not to allow a match between a loosely connected cluster with a strongly connected one.

For each possible pair of cluster matches between *Mus Musculus* and *Homo Sapiens* these two tests are applied. A list of possible matches for each of the *Homo Sapiens* clusters is produced by applying these two tests. After this elimination, the possible matches are considered for functional relevance by adding the GO Terms information into the matching algorithm.

2.2.3 Mapping the Clusters of Proteins Depending on GO Annotation Similarity

It is possible to define the properties of a protein by means of several aspects such as molecular function, biological process and subcellular location. For this reason, a protein may be associated with one or more GO Terms each defining a different aspect or property of the protein.

In order to use the GO Terms of proteins for mapping clusters, a way to define the similarity of two proteins by means of GO Terms was required. On the other hand, since the problem to be solved is not just performing protein matching but cluster matching, a method to use this similarity information in cluster matching was required.

2.2.3.1 Defining the Similarity of Two Proteins Using Associated GO Terms

All GO Terms defined by GO Consortium are hierarchically related with each other. This hierarchical structure can be considered as a tree. As this tree is traced from the roots to the leaves, the GO Terms are more specialized and they define a more specified function or location. GO Terms can define the biological process, the molecular function and the cellular location of a protein. When these three ontologies are considered, the tree defining the relations between the GO Terms is as a connected tree with three different roots. These root terms are 'GO:0008150' for 'biological process' ontology, 'GO:0005623' for cellular location ontology and 'GO:0003674' for molecular function. Although there exists some GO Terms that are not connected to this tree of GO Term relations, they can be ignored since the reason for disconnectedness is that they do not provide any information about the protein. They stand for unknown information. For example, 'GO:0000004' stands for 'biological process unknown'. Similarly, 'GO:0008372' means 'cellular component unknown' and 'GO:0005941' means 'unlocalized'. These non-informative terms associated with the proteins used are eliminated before considering them for similarity.

The study performed by Wu *et al.* [45] provides a simple and elegant solution for evaluating the similarity of protein pairs by using the GO Terms of the proteins. This proposed solution is used for evaluating the similarity of two proteins in our study. Considering the similarities of proteins between clusters, a cluster similarity measure which is described in the next section is introduced. But at this point, the method proposed by Wu *et al.* is described in detail since it is directly applied for computing the similarities of two proteins.

In fact Wu *et al.* introduces a similarity metric for two GO Terms. Their approach for comparing two GO terms not only considers the distances between the terms but also the specificity of the GO Terms considered. By considering all vs. all similarities of GO terms associated with the two proteins, they define the similarity of the two proteins as the highest scoring match for the GO Terms of the two proteins.

While considering two GO terms, the most recent common ancestor (MRCA) in the GO term hierarchy is found as the first step. Next, three different distances are computed using the locations of the nodes and the MRCA in the GO term tree. Then, these three distances are used as parameters to compute the similarities of two GO terms.

A path is defined to be the collection of nodes which are traced to reach from a GO Term to the root of the GO Tree. The distance between two GO Terms is defined to be the minimum number of nodes to be traced to reach from one GO Term to the other. With these definitions of path and distance for GO Terms, the three parameters of the GO term similarity is defined as in Equations 2.7, 2.8 and 2.9.

The first parameter α defines the maximum distance of the most recent common ancestor of the two GO terms to the roots of the tree. It is defined as in Equation 2.7. This parameter measures how specific MRCA of the two terms is.

$$\alpha = \max_{path_m \in Paths(term_i), path_n \in Paths(term_j)} \left\{ \begin{array}{c} \text{the number of common terms} \\ \text{between } path_m \text{ and } path_n \end{array} \right\} - 1 \qquad (2.7)$$

The second parameter β measures how relatively general $term_i$ and $term_j$ are in the GO hierarchy. It is defined in Equation 2.8. In Equation 2.8, U = {all leaf nodes descending from $term_i$} and V = {all leaf nodes descending from $term_j$}. Simply the value of this parameter is equal to distance between the least specified term and its closest leaf. The function named dist is the distance between two GO terms. It is equal to the minimum number of nodes required

to reach from one term to another.

$$\beta = \max\{\min_{u \in U}\{\text{dist}(term_i\ ,\ \text{u})\}, \min_{v \in V}\{\text{dist}(term_j\ ,\ \text{v})\}\} \tag{2.8}$$

The third parameter γ is the shortest distance between the two terms. Computation of this parameter is described in Equation 2.9. It is used for evaluating the local distances between two terms relative to the maximum depth of the GO tree.

$$\gamma = \text{dist}(\text{MRCA}\ ,\ term_i) + \text{dist}(\text{MRCA}\ ,\ term_j) \tag{2.9}$$

The illustration of these parameters can be found in Figure 2.2. All these parameters are put into Equation 2.10 in order to get a similarity measure for two GO Terms. This normalized similarity measure considers the distances between the nodes and the specificity of the nodes. It returns a normalized value between 0 and 1. 0 means that the common ancestor of the two GO terms are actually the root of the tree. So they are totally dissimilar. Similarly a score of 1 means that two GO terms are same and also quite specific that they are leaf nodes in the GO term tree. For that reason, this metric not only considers the local similarities of the GO terms but it also measures the amount of information they provide.

$$\text{RSS}(term_i\ ,\ term_j) = \frac{maxDepth^{GO}}{maxDepth^{GO} + \gamma} \times \frac{\alpha}{\alpha + \beta} \tag{2.10}$$

For using this similarity metric for determining the similarity of two proteins, the authors have computed all vs. all GO similarities for the GO Terms associated with the two proteins. The maximum similarity score of all vs. all comparison is used as the similarity of the two proteins. This is formulated as in Equation 2.11 for proteins P and Q.

$$RSS^{GO}(\text{P,Q}) = \max_{u \in \text{terms(P)}, v \in \text{terms(Q)}}\{\text{RSS(u,v)}\} \tag{2.11}$$

Although this method has been applied for computing the similarities of protein pairs in our study, a difference exist between the proposed method and our application of the method. The authors of the paper use the 'part-of' relations as well as 'is-a' relations in the GO terms tree while applying this method. For the implementation completed for this study, only 'is-a' relations are taken into account while constructing the GO Term tree. The number of 'part-of' relations are small in size and they are not significant. We also think that 'is-a' and 'part-of' relations have completely different semantic meanings. They should not be used in the same way while computing this similarity score. For this reason in the implementation performed in the scope of this study, the 'part-of' relations are ignored.

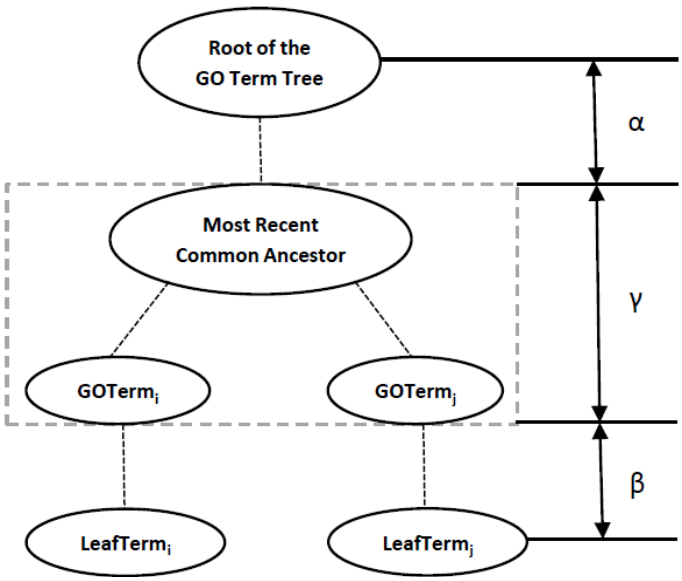

Figure 2.2: The visual description of the parameters used in evaluating the similarity of two GO Terms. This figure is adapted from the study of Wu *et al.* [45]

2.2.3.2 Using the GO Terms Similarity for Cluster Matching

After defining the scoring metric for the similarity of two proteins, the next step is matching similar clusters of proteins. The similarity score of two clusters are defined relative to the similarities of the proteins forming the cluster. For defining the similarities of two clusters, there are three simple and commonly used methods when a similarity metric for comparing the nodes of the clusters is defined. These three commonly used methods are based on taking the shortest distance, the longest distance and the average distance between all the nodes of the two clusters. For the protein cluster matching, it is possible to define the similarity of two clusters by these three methods. However, the problem of computing the similarity of two clusters can be considered as a maximum weight bipartite matching problem. There exist several linear time algorithms solving the maximum weight bipartite matching problem in the literature. In order to compute the similarities of two clusters, an alignment has to be made between the clusters. The algorithms for maximum weight bipartitie matching problem try to maximize the total score achieved by the alignment of the nodes. So, the similarity score of two clusters is the sum of protein similarities when optimal alignment is performed between the two clusters.

Although applying a linear time algorithm for solving maximum weight bipartite matching

can be applied succesfully to define the metric for the similarity of two clusters, the study performed by Singh *et al.* [37] showed that it is in fact unnecessary. They experienced that applying a greedy heuristic algorithm performs better when used as the similarity metric between two clusters. This algorithm first computes all vs all similarity scores of the nodes. Then, it iterates by first finding and removing the nodes with maximum similarity scores and computing the following maximum similar nodes. This process goes on until all nodes of one of the clusters is matched to the other cluster. This greedy strategy does not guarantee total maximal score. But it guarantees to find and match the nodes with maximum similarity. This greedy algorithm is summarized in Algorithm 1.

As recommended in the study of Singh *et al.* [37], we have computed the similarities between two clusters using this alignment strategy. In Algorithm 1, the function named computeProteinSimilarity returns a similarity value for two proteins depending on the associated GO terms. This value is computed as described in the previous section. Since the computed scores are used for comparing the similarities of different clusters, a normalization depending on the number of nodes used to get the total score is performed. For that reason, each invocation of this method returns a value between 0 and 1. A value of 0 means that the two clusters are totally different and a value of 1 means that the two clusters are the same. This value is called *goSimilarityScore* in the rest of this text.

Although the score produced by GO term similarities is important for determining the similarities of two protein clusters, the matched clusters should also be evaluated according to graph theoretic similarities. The conservation of the graph theoretic properties are also an important parameter defining the similarity of two clusters. For computing the amount of conservation between two clusters, conservation index defined in Equation 2.12 is used. Given an alignment between two graphs, this conservation index defines a value of similarity depending on the number of edges conserved between the two graphs. This value is between 0 and 1, 0 meaning there are no conserved edges between the two graphs. A value of 1 means that all the edges of the clusters are conserved. So, having a value close to 1 shows the two clusters are similar. This conservation index is defined in the study performed by Bandyopadhyay *et al.* [5].

$$conservationIndex = \frac{2 \times \text{Number of Conserved Edges}}{\text{Number of Edges in Graph 1} + \text{Number of Edges in Graph 2}} \quad (2.12)$$

After computing the *goSimilarityScore* and *conservationIndex*, these two similarity evaluation

Algorithm 1 The algorithm for computing the similarities of two clusters using GO terms

```
similarities[][];
alignedProteins[];
U = {Proteins in cluster 1}
V = {Proteins in cluster 2}
{Compute all vs. all score between proteins}
for all u_i ∈ U do
    for all v_i ∈ V do
        similarities[i][j] = computeProteinSimilarity(u_i , u_j);
    end for
end for
{Make the alignment}
repeatTimes = max {sizeof(U) , sizeof(V)}
for i = 1 to repeatTimes do
    max_i = findMaximumSimilar_i(similarities);
    max_j = findMaximumSimilar_j(similarities);
    addAligned(alignedProteins , max_i , max_j);
    for all u_i ∈ U do
        similarities[i][max_j] = 0;
    end for
    for all v_j ∈ V do
        similarities[max_i][j] = 0;
    end for
end for
{Compute and return normalized alignment score}
score = 0;
for all a_i ∈ alignedProteins do
    score += computeProteinSimilarity(a_i);
end for
{Normalize computed score and return}
normalizedScore = score / repeatTimes;
return normalizedScore;
```

values should be combined to give a total score. This total score should be normalized since this value will be used to compare each possible cluster pairs for similarity. Keeping this purpose in mind, the *totalScore* is defined as in Equation 2.13. Multiplication of the two terms produces a value between 0 and 1 again. It was also possible to take the average of the *goSimilarityScore* and *conservationIndex* values as the *totalScore*. But an exponential relation would be more meaningful since the two parameters are both very important. Two clusters with high GO term similarity should not be considered as a match if their graph structures are not conserved as a result of the alignment performed for computing this GO similarity score. Similarly, two clusters should not be considered as similar when their GO terms are not similar but their graph structure are similar. By multiplication, these mismatches between scoring parameters are penalized strictly. Averaging can tolerate a very low similarity of one of the parameters if the other is high. This situation is avoided with the definition of *totalScore* as a multiplication.

$$totalScore = goSimilarityScore \times conservationIndex \tag{2.13}$$

The *totalScore* is used to evaluate and compare the similarities while matching two clusters. The algorithm developed in this study does not force the each formed clusters to be matched with another cluster. Instead it returns a list of similar clusters sorted from the most similar clusters to least. This approach allows a cluster to be matched with more than one cluster. This can be considered as an advantage since most studies try to perform one to one matching. This enforcement to perform one-to-one matching is a fallacy. A group of proteins may be specialized for more than one function in a different species. For that reason, there may exist several matches for a cluster of proteins in a different species.

CHAPTER 3

RESULTS

The series of methods described in Chapter 2 are implemented in Java and tested on the datasets of protein interactions taken from STRING database [26]. ClustOrth is able to match all the clusters formed from the datasets containing all known protein interactions for *Mus Musculus* and *Homo Sapiens* organisms. It takes about 650 minutes in other words 10 hours to complete cluster matching process. The method can be defined as memory efficient since it can complete the computations required for cluster matching with at most 2 GBs of memory. When the size of the protein interaction datasets are considered, it can be said that the algorithm is efficient by means of memory and time.

There are not many studies on finding and matching functionally orthologous groups of proteins in literature. Most studies are based on determining orthologous protein pairs using protein interaction and protein sequence information. For this reason, it was difficult to verify whether the algorithm is performing well. For the purpose of validating the results, several approaches have been followed. First, matched clusters are compared depending on the orthologous groups of proteins in the clusters. STRING database provides an ontology for defining the orthologous similarities of proteins. With the use of this orthology information of proteins, the orthological similarities of the matched clusters are evaluated. Since ClustOrth performs a mapping between proteins during the alignment of clusters, it can also perform orthology mapping by means of proteins. By making this modification, ClustOrth has been compared with another algorithm named ISORANK developed by Singh *et al.* [38]. Another effort on understanding the performance of ClustOrth is spent by comparing the method with another protein cluster orthology prediction algorithm named NetworkBLAST [36]. The error tolerance of our method is also evaluated by introducing some false positive interactions into the dataset and evaluating the resulting cluster matches by means of orthology relevance

of the proteins. Finally the computational complexity of the algorithm is briefly discussed.

3.1 Orthological Relevance of Mapped Protein Groups

STRING database [26] provides not only a list of protein interactions but also a number of protein features together with the interactions. It is possible to get information about the protein sequence, the actions of proteins in a cell, relations of these proteins between different species and the orthological relevance of proteins. Among these provided information, orthological information is an important piece of information to validate the cluster matching results. The information about the orthological class of the proteins are provided with reference to COG database [42]. COG terms represent strong phylogenetic lineages and are extracted using the protein sequece similarities. For each protein defined in the STRING database, the corresponding orthological term in the COG database is provided. Using these terms, it is possible to validate whether the clusters matched with ClustOrth are orthologically relevant or not.

However, a straightforward validation of the clusters is not possible. Since the COG term scores only define the orthological similarities of two proteins, the scoring should be extended to evaluate protein clusters. The scores between two COG terms are defined with a value between the 0 - 1000 range. The higher the value of the score, the more similar the two COG terms. The mapping between the proteins and the COG terms can be considered as a one-to-one relation when a small number of exceptions are ignored. For this reason, it is quite easy to define the orthological distance between two proteins. It is basically the distance value defined between the COG terms of the two terms. In order to evaluate the orthological similarities of the matched protein groups using the protein pair similarity scores, an algorithm to extend the scores of protein pairs to protein cluster match scores is required.

The algorithm used for producing similarity scores of two clusters are defined using the maximum weight bipartite matching problem. After computing the all-vs-all similarity scores for the proteins of the matched clusters, the constructed distance matrix is used to perform an alignment of the proteins in order to give maximum total weight. Getting an alignment for proteins of two clusters, the similarity metric is defined as the average score of the aligned protein pairs. So all the scores achieved by the alignment is summed and divided by the number of aligned proteins. This procedure is summarized in Algorithm 2. In this algorithm,

the function named *computeOrthologicSimilarity* returns the COG term similarity score of two proteins. The other function named *alignMaximumWeightBipartite* performs maximum weight bipartite matching on the two clusters and returns a list of protein pairs that are aligned with each other. The performed alignment with this function is guaranteed to return the maximum total score.

Algorithm 2 The algorithm for computing the score to validate the orthological similarities of two clusters

```
similarities[][];
alignedProteins[];
U = {Proteins in cluster 1}
V = {Proteins in cluster 2}
{Compute all vs. all score between proteins}
for all u_i ∈ U do
    for all v_i ∈ V do
        similarities[i][j] = computeOrthologicSimilarity(u_i , u_j);
    end for
end for
{Make the alignment}
alignedProteins = alignMaximumWeightBipartite(U , V);
{Compute and return normalized alignment score}
score = 0;
for all a_i ∈ alignedProteins do
    score += computeProteinSimilarity(a_i);
end for
{Normalize computed score and return}
normalizedScore = score / repeatTimes;
return normalizedScore;
```

There seems to be a contradiction between the alignment method used for the evaluation of the cluster matches and the alignment method used during the computation of the cluster similarities while performing cluster matching. For matching the clusters, the maximum scoring protein pairs are eliminated one by one but during the evaluation of the similarities of cluster matches maximum weight bipartite matching is applied. The choice on the cluster alignment

algorithms are not made this way by coincidence. While performing cluster matching, the main aim was finding the most similar protein pairs and extending this similarity to form clusters. Using such a method is claimed to be perform better in the study performed by Singh *et al.* [37]. The idea of performing such an alignment is also more meaningful because of the nature of the evolution of proteins. In protein interaction networks, some proteins are connected with many other proteins and these protein are called hub proteins. These hub proteins take part in the main biological processes. Because of this reason, they are more likely to be conserved during evolution. Determining these proteins and their interacting partners, it is possible to determine main biological processes in a cell. Since these proteins are conserved more during evolution, performing an alignment based on maximum score matches is more meaningful than performing maximum weight bipartite matching. Pairwise similarities of proteins are more important than getting an overall high score in this respect. However while validating the cluster matches, maximum weight bipartite matching is more meaningful to apply since the main aim is not determining the maximum similar protein pairs but determining the orthological similarities of the clusters. Maximum weight bipartite matching return the maximum similarity that can be achived for two clusters. So it gives an overall score of orthological relevance of two clusters.

The developed cluster construction and mapping method returns a sorted list of cluster matches together with their match scores. Computing an orthological similarity score with Algorithm 2, the prediction performance of the method is tested. A cluster match is accepted to be valid if a match score over 0.75 is achieved with the above defined computation. In Figures 3.1 and 3.2, the results of this evaluation of formed cluster matches from the organism *Mus Musculus* and *Homo Sapiens* is provided. It is also possible to see a down sampled list of values forming these charts in Table 3.1. Figure 3.1 shows the relation between the different cutoff match scores and the accuracy of cluster matching for those cutoff values. The accuracy is defined as in Equation 3.1.

$$accuracy = \frac{\text{number of validated cluster matches}}{\text{total number of cluster matches performed}} \tag{3.1}$$

For a defined cutoff value, all the cluster matches having more than or equal to match scores are evaluated by means of orthology. If the evaluation score of the clusters are over 0.75, the match is accepted as orthologically relevant. Computing the accuracy as defined in Equation 3.1 it is possible to define the threshold to be used further. As can be seen from Figure

3.1, a cutoff threshold of 0.85 results with predictions over 80% accuracy. The accuracy value increases exponentially as the cutoff threshold increase. This shows the validity of the scoring metric developed for performing cluster matching. However as can be seen from Figure 3.2, the number of predictions for cutoff values over 0.8 are extremely low. For a cutoff value of 0.9, only 6 clusters can be matched but all these matches are validated. Similarly for a cutoff value of 0.8, only 50 of the cluster matches are validated for 86 performed matches. The matches are not informative for cutoff values below 0.76.

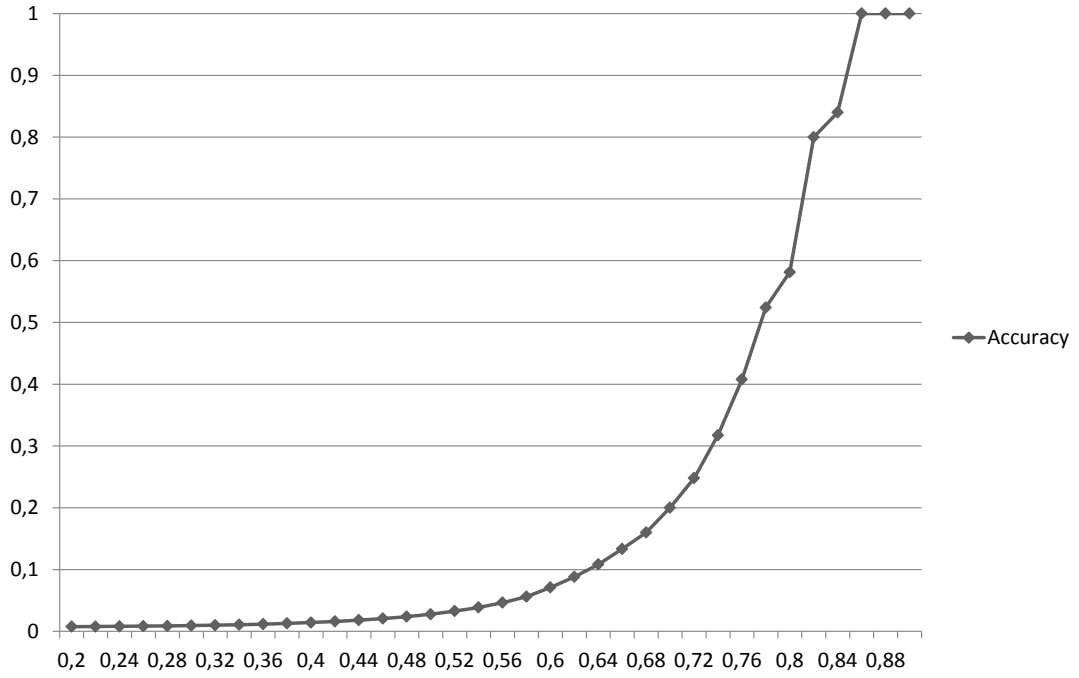

Figure 3.1: Chart representing the orthologically related cluster match accuracies with respect to different cutoff values of cluster similarity. The horizontal axis represents different cluster similarity cutoff values used. The values are computed depending on the protein interaction networks and GO similarities as defined in Section 2.2.3.2. The vertical axis stands for the cluster match accuracies for the defined cutoff value by means of cluster orthology.

The validation strategy applied here is informative for showing the validity of the scoring mechanism constructed for evaluating the scoring metric defined for cluster comparison. It is shown that high similarity scores of clusters result with more accurate orthology predictions. Although the number of clusters matched with high scores are low, the performed matches are significant. For a cutoff value of 0.85, it is possible to perform orthology prediction with 100% accuracy of orthological relevance for *Mus Musculus* and *Homo Sapiens* organisms.

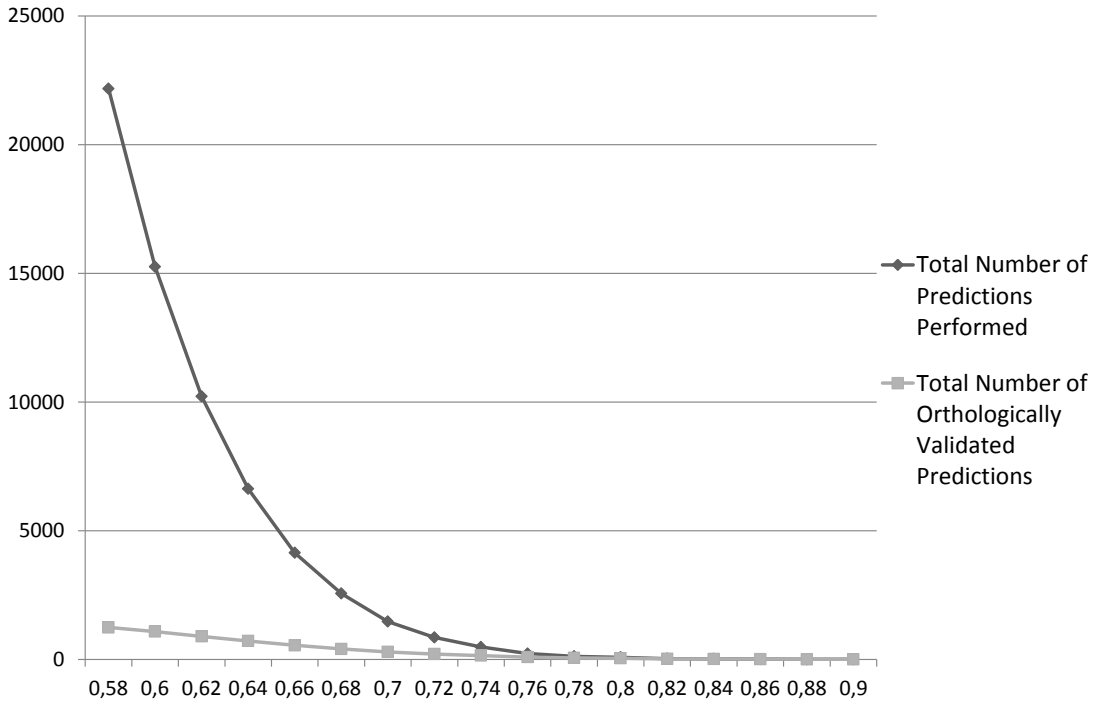

Figure 3.2: Chart representing the number of cluster matches performed and the number of validated matches among these. The horizontal axis represents different cluster similarity cutoff values used. The values are computed depending on the protein interaction networks and GO similarities as defined in Section 2.2.3.2. The vertical axis stands for the number of cluster matches defined for the cutoff value

Table 3.1: The table of values used for forming the charts in Figures 3.1 and 3.2. The columns of the table represents the achieved results for different cutoff values of cluster similarity. Information about the number of predictions performed, the number of validated predictions among the performed predictions and the accuracy of the predictions for cluster similarity value over the defined cutoff value can be achieved from these columns.

Cutoff Value	**0,66**	**0,70**	**0,74**	**0,78**	**0,82**	**0,86**	**0,90**
Number of Validated Predictions	553	295	154	65	28	14	6
Number of Performed Predictions	4144	1473	485	124	35	14	6
Accuracy	0,133	0,200	0,317	0,524	0,8	1	1

3.2 Comparison with ISORANK Algorithm Used in Protein Orthology Mapping

The validity of the scoring metric used for performing cluster matching is shown in Section 3.1. However there is still need for comparing the performance of the method with another so-

lution in literature in order to understand how well the algorithm works. Unfortunately there are not many studies trying to find the orthologically relevant functional modules between protein interaction networks of different species. Nevertheless there exists methods trying to find orthologically related proteins between different species. One of the most successful studies trying to predict protein orthologies is ISORANK [38]. By providing the protein networks and bit scores of all vs. all BLAST alignments of protein sequences of the organisms, ISORANK is able to determine the orthologically related proteins of the provided protein interaction networks. Because of the mentioned success of the method among other solutions, ISORANK is selected for comparing the performance of the method developed.

In spite of the inconsistency of the outputs produced by ISORANK and our method, it is possible to compare the performances of the two algorithms. The algorithm developed in this thesis can be modified to list orthologically relevant protein pairs. As described in Section 2.2.3.2, an alignment is performed for the proteins while clusters are being matched. This alignment can be used to infer protein orthology. For the top scoring cluster matches, the performed alignments are used to infer protein orthology. Protein similarity scores and cluster match scores are used together to perform protein orthology prediction. Taking clusters having a cluster similarity over a threshold value and considering the proteins having similarity score of 1 for these clusters, predictions of protein orthologies are performed. Applying different threshold values for cluster matches, a number of protein orthologies are predicted for comparison with the results produced by ISORANK.

Applying ISORANK on *Mus Musculus* and *Homo Sapiens* datasets extracted from STRING database was not possible because of the computational complexity of the method. There are two bottlenecks of ISORANK. The algorithm requires all vs. all bit scores of BLAST alignments for every protein in the compared organisms. For 14071 *Mus Musculus* and 15857 proteins of *Homo Sapiens* BLAST alignments has to be made for (14071 * 14071 + 15857 * 15857 + 14071 * 15857) / 2 = 336280669 protein pairs. Performing this many BLAST alignments without any heuristics can take months. Although this problem can be overcomed by applying heuristics, the algorithm is not fast enough to produce a results in a reasonable time. According to the definitions provided by the authors, the number of repetitions required for the data to converge is around 30. However during the experiments we have performed, only 3 repetitions could be completed after applying the method for 70 hours. This means that for the datasets of *Mus Musculus* and *Homo Sapiens*, the computation for determining

orthologous protein pairs takes around a month. However our method was able to complete the cluster matching for these datasets in around 10 hours.

Because of this reason, we needed a smaller dataset to compare the performances of the two methods. The benchmark dataset used for evaluating the performance of ISORANK belongs to organisms *Saccharomyces Cerevisiae* and *Drosophila Melanogaster*. As the first effort to perform a comparison between the methods, the dataset provided together with the ISORANK executable is used. However the dataset did not include confidence values for protein interactions. This lack of information resulted with poor clusters after the application of RRW. However even with the missing information, RRW algorithm should have found cliques in the provided networks. The RRW algorithm could not find such clusters but rather loosely connected ones. This result made us think the possibility that the provided dataset is not complete. However this effort helped us to understand the weakness of our approach. The interaction networks should be complete and confidence values should be defined between protein interactions to get reasonable results.

The second effort spent on the *Saccharomyces Cerevisiae* and *Drosophila Melanogaster* organisms was based on using STRING database to get the protein interaction networks of these organisms. After extracting the protein interactions of the two organisms from the database, all vs. all BLAST scores of the proteins in the networks are computed with a heuristic. In this heuristic the pairs of proteins that have low bit scores are ignored. If the BLAST E-value of the two proteins are over 40, the pair is ignored and no alignments are performed on these pairs. An E-value of 40 means that the similarity found between the pair is not significant. So they can be ignored. When the total number of BLAST alignments are considered, the alignments are performed only as many times as the sum of the number of *Saccharomyces Cerevisiae* proteins and the number of *Drosophila Melanogaster* proteins. After getting the bit scores of all vs. all BLAST alignments, the ISORANK method is applied on the constructed dataset with the default parameters defined in the paper of the study.

A change in the GO annotation extraction process was needed since “Clone/Gene ID Converter” [1] does not provide support for the organisms *Saccharomyces Cerevisiae* and *Drosophila Melanogaster*. Instead Babelomics ID-Converter tool [2] is used to extract the GO Annotation terms for these organisms. Babelomics is a web based project with different solutions produced for gene and protein identification related studies. Among the range of tools

they provide, ID-Converter tool enables conversion of different gene and protein annotations. This tool is also capable of listing the GO terms of a protein. When compared to "Clone/Gene ID Converter" [1], it provides a wider range of organisms to search for. They both cover the well-known annotations such as Ensembl, Uniprot, Swissprot. But for annotations that are not popular as much as these well-known models, their coverage change.

After applying our method with the modification of extracting protein orthologies, we were able to get lists of orthologous proteins for different threshold values of cluster similarity. On the other hand, ISORANK produced a list of ortholog proteins together with the confidence values of the orthologies. In order to compare the results produced by the two methods, the following procedure is used. For different cutoff values for cluster similarity, the performed protein orthology predictions are extracted. For the number of orthology predictions performed for the defined cutoff value, same number of top scoring protein orthologies produced by ISORANK are taken. These two lists of protein orthologies are compared by using Interolog Database [48]. Interolog is a term used for defining protein interactions that are conserved orthologically between species. This database provides a list of protein interaction pairs that are orthologous. The method used to construct this database uses protein sequences of interacting proteins to get a combined score of sequence similarities. Comparing these similarities for existing interactions, inferences of interolog interactions are made. 45 results produced by the interolog extraction method are validated by two hybrid tests. The statistical significance of these results are proved by using a hypergeometric model and computing the P value for these results. Another case study performed on a well-known protein complex named Ste5-MAPK showed that 5 of the 6 known subunits were successfully predicted. One final point to mention is that only top 1% of the found interolog information included in the database in order to avoid false positives. With all these work, the interolog database can be accepted as a reliable source of interaction orthology.

Interolog database is used in a flexiable manner to evaluate the outputs of the two algorithms. The proteins in orthologous interactions are considered as related. For example, given that A - B and A' - B' are interologs, we have considered A - A' , A - B' , B - A' and B - B' as ortholog proteins. By checking the existence of the protein orthology prediction performed by the two algorithms in the database in the defined manner, we have tried to evaluate how successfully the two methods identify protein orthology. The comparison of the results produced by ISORANK and our method are summarized in Figure 3.3 and Figure 3.4. Also a

downsampled list of achieved values that form these charts are provided in Table 3.2. The accuracy of the orthology prediction between proteins are calculated as the fraction of protein pairs existing the Interolog database over the total number of predictions performed for the defined threshold value. The comparison of these accuracy values with respect to the different cluster similarity threshold values can be found in Figure 3.3. As can be seen from Figure 3.3, ISORANK has more accurate results compared to our method for low values of cluster similarity cutoff. However our method outperforms the method ISORANK for cluster similarities over 90%. This implies that our scoring metric is in fact better than the scoring metric used in ISORANK for determining the orthologies. But it is so strict that only a small number of clusters can pass this threshold test. Apart from accuracy comparison, the number of predictions performed at different threshold values and the amount of predictions that are valid are represented in Figure 3.4. Both methods can not make highly accurate orthology mappings. The number of confident predictions are low for both of the methods. But the mappings performed by ISORANK does not seem to be correlated with the confidence values returned. Our scoring metric directly effects the correctness of orthology predictions.

Table 3.2: The table of values used for forming the charts in Figures 3.3 and 3.4. The columns of the table represents the achieved results for different cutoff values of cluster similarity. The number of predictions performed for different cutoff values are used to define number of best scoring orthology predictions to be compared from the results of ISORANK. Information about the number of performed predictions, the number of validated predictions and the accuracy of the predictions for cluster similarity over the defined cutoff value can be achieved from these columns. Both results for ISORANK and our algorithm are provided in this table.

Cutoff Value	**0,66**	**0,70**	**0,74**	**0,78**	**0,82**	**0,86**	**0,90**
Number of Performed Predictions	715	368	244	181	152	119	28
Number of Valid Prediction of our method	48	40	35	33	33	30	18
Accuracy of our algorithm	0,067	0,108	0,143	0,182	0,217	0,252	0,642
Number of Validated ISORANK Predictions	184	109	79	59	50	40	11
Accuracy of ISORANK Predictions	0,257	0,296	0,323	0,325	0,328	0,336	0,392

The validation could also be performed by using the COG terms defined in the COG database as performed in Section 3.1. But the COG terms provided in STRING database did not seem to cover all the proteins of the organisms *Saccharomyces Cerevisiae* and *Drosophila Melanogaster*. For many of the proteins in these databases, the associated COG terms could

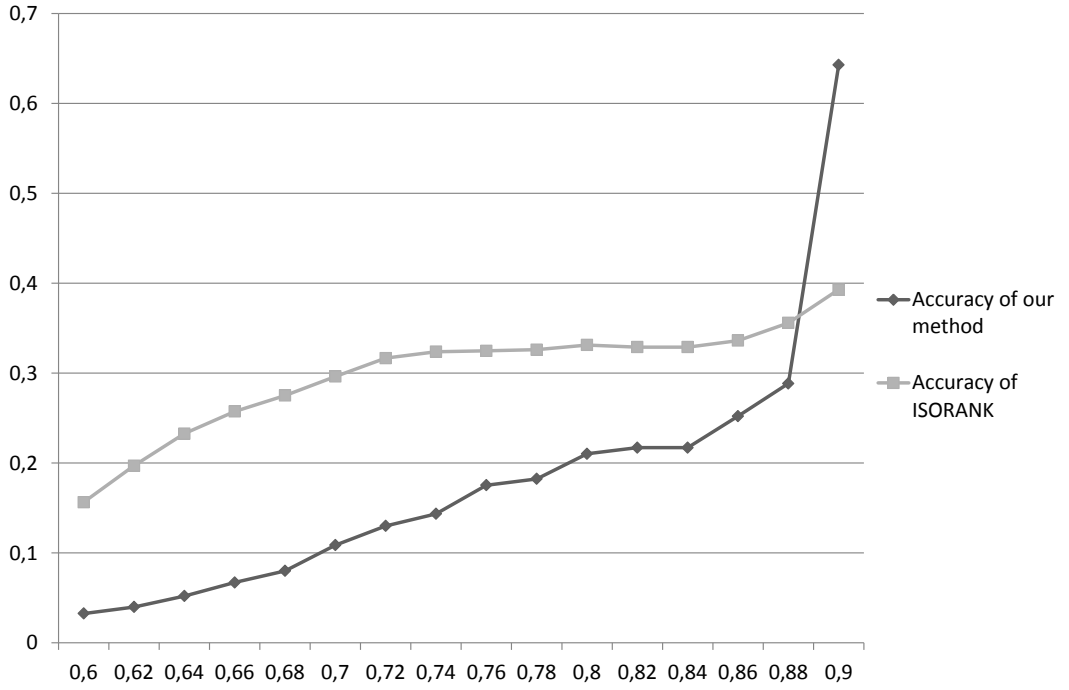

Figure 3.3: Chart representing the protein orthology prediction accuracy comparison of ClustOrth and ISORANK algorithms. The horizontal axis represents different cluster similarity cutoff values used. The values are computed depending on the protein interaction networks and GO similarities as defined in Section 2.2.3.2. The vertical axis stands for the protein orthology prediction accuracies for the defined cluster similarity cutoff values. The Interolog database is used to validate protein orthologies.

not be determined. For this reason, Interolog database is used to validate and compare the results achieved by the two algorithms. Another advantage of using the Interolog database is that the performed orthology predictions are validated from another source of orthology data. These are the reasons for the different data sources used in this part of the evaluation of the results.

It should be emphasized that ClustOrth does not aim to perform orthology prediction between protein pairs. The aim of this study is to define functionally orthologous modules in protein interaction networks of different species. When considered with this manner, it is not expected to get higher accuracies of protein orthology prediction compared to a method that aims to find orthologous protein pairs. Since ISORANK uses protein sequence information directly, it is expected to produce better results since evolution can directly be observed from protein sequence information. However the ISORANK method is computationally complex

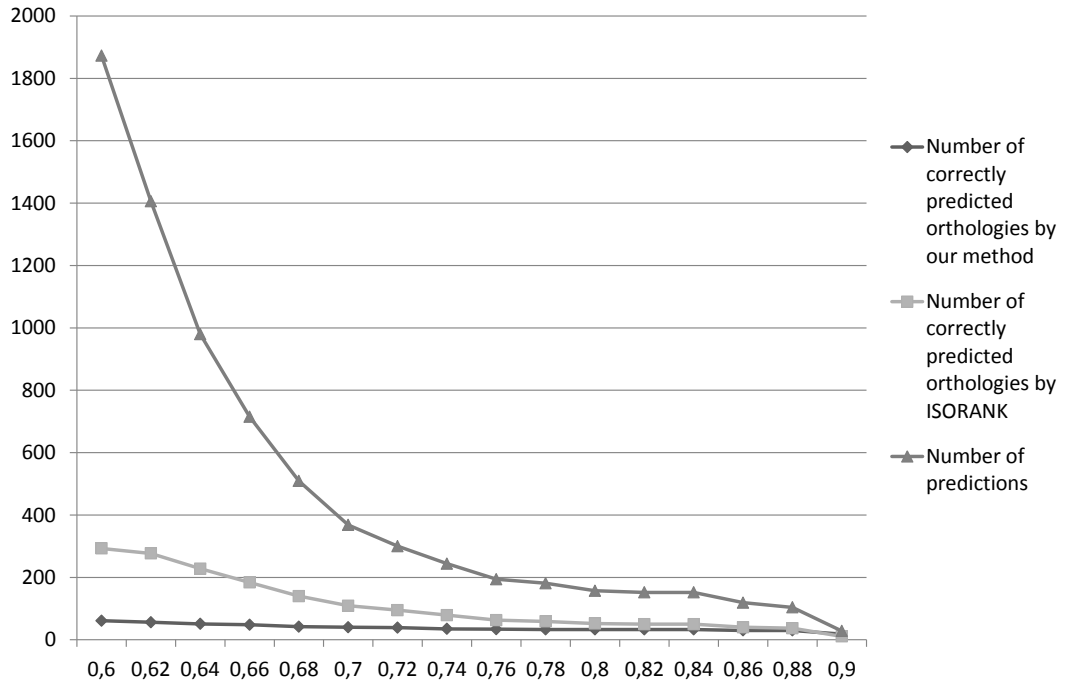

Figure 3.4: Chart comparing the number of correctly predicted protein orthologies by ClustOrth and ISORANK algorithms. The horizontal axis represents different cluster similarity cutoff values used. The values are computed depending on the protein interaction networks and GO similarities as defined in Section 2.2.3.2. The vertical axis stands for the number of predicted orthologies for the defined cluster similarity cutoff values. The Interolog database is used to validate protein orthologies.

since it tries to align many protein sequences using BLAST. ClustOrth can predict orthology information in a simpler manner using GO terms of the proteins. Although it does not seem to be as accurate as ISORANK, it can produce a result for larger protein interaction networks such as *Mus Musculus* and *Homo Sapiens*.

3.3 Comparison with NetworkBLAST Algorithm Used in Orthological Mapping of Protein Clusters

One of the benchmark methods used in protein cluster determination and orthological mapping is the NetworkBLAST algorithm [36]. This method is developed as an extension to protein interaction network comparison tool named PathBLAST [20]. PathBLAST performs local alignment of a group of interacting proteins on protein interaction networks. How-

ever it does not have the capability of comparing two protein interaction networks globally. NetworkBLAST is an extension to PathBLAST algorithm which aims to compare protein interaction networks as a whole and determine conserved protein groups in these networks.

Basically NetworkBLAST performs global alignment on the provided protein interaction networks using protein sequence similarity and protein interaction information. Protein sequence similarity is determined using the E-values for BLAST alignments of all-vs-all protein pairs. Then this alignment is used by NetworkBLAST to determine the seed nodes of the aligned graphs which represent conserved protein groups. These seed nodes are expanded using a probabilistic model to form the final protein clusters. The tests are performed on the datasets extracted from the Database of Interacting Proteins (DIP) [47] during the evaluation of the performance of NetworkBLAST algorithm. With the tests performed by applying the method on the protein interaction networks of three organisms, it is claimed that 71 conserved subnetworks of these three organisms could be found. The application of the method results with two output files. One of these output files provide the list of aligned nodes and complexes and the other provides the complete alignment graph of the protein interaction networks used to discover these conserved protein complexes.

The method is applied on the datasets extracted from STRING database [26] for comparing the performances of NetworkBLAST and our algorithm. *Drosophila Melanogaster - Saccharomyces Cerevisiae* and *Mus Musculus - Homo Sapiens* organism pairs are used to test the performances of the algorithms. Running the implementation provided by the authors of NetworkBLAST with the default parameters defined in the user manual of the implementation, no clusters could be identified by NetworkBLAST for the two organism pairs. We have run the implementation with various values as the parameters in order to understand whether these unsuccessful results occured as a result of incorrect parameter settings. Also the provided datasets are considered for errors according to the descriptions provided with the user manual of the implementation. But for none of these efforts, NetworkBLAST could determine conserved protein complexes between species. However NetworkBLAST was able to parse the input files correctly and perform global alignment on the provided protein interaction networks. This shows that the algorithm is not succesful in determining the protein interaction clusters for protein interaction data collected from different data sources. If it is not our mistake, the algorithm cannot evaluate large, real-world protein interaction networks. However our algorithm was able to determine 25 protein cluster matches for which 21 is validated to

be orthologous for the datasets of *Mus Musculus* and *Homo Sapiens* for a cluster cutoff value of 0.86. Similarly for the organisms of *Drosophila Melanogaster* and *Saccharomyces Cerevisiae*, 28 predictions could be performed for a cluster cutoff value of 0.9 for which 18 of them are validated. This shows that our algorithm is a lot more flexiable in determining the conserved clusters of proteins in protein interaction networks of different species.

Although no cluster orthology predictions could be performed by NetworkBLAST, the global alignment performed during the application of the algorithm is produced as output. The experiments on *Drosophila Melanogaster - Saccharomyces Cerevisiae* and *Mus Musculus - Homo Sapiens* protein interaction networks showed that the performed alignments can be used for determining protein orthology. Matched protein pairs of the performed alignments are evaluated for orthological relevance using the COG terms of proteins. The COG validation experiments for the protein interaction networks of *Drosophila Melanogaster* and *Saccharomyces Cerevisiae* showed that among 1806 protein matches 979 are orthologically relevant. Similarly for the organisms of *Mus Musculus* and *Homo Sapiens*, 1051252 protein matches are orthologically relevant among 1101452 alignment matches. The global alignment is performed correctly according to these results. As long as the provided BLAST E-values are correct, this was an expected result since protein sequence similarity is directly considered for evaluating protein orthology. The relevance of the performed alignments show that there are not any problems during the computation of the provided all-vs-all BLAST values. On the other hand these results show that the input is provided in the correct format for running the implementation of the algorithm. For all these reasons, protein orthology relevance results show that NetworkBLAST is not as strong as our method for the determination and matching of protein clusters.

We have not considered using their benchmark dataset with our method since our previous experience during the tests of ISORANK method showed that the datasets used to validate specific implementations are not complete most of the time. Reducing the size of the provided sample input, the aim of evaluating the sample dataset faster are among the reasons for the incompleteness of these provided datasets. Our algorithm cannot perform correctly with these incomplete datasets since it runs the Repeated Random Walks algorithm [25] for constructing the protein clusters. For this reason the tests applied to evaluate the performance of NetworkBLAST are performed using the datasets extracted from the STRING database.

The reason for our method to perform better in cluster determination and matching compared to NetworkBLAST is the difference between the alignment strategies. NetworkBLAST performs global alignment as the first step. On the other hand, the first step of our algorithm is dividing the protein interaction network into strongly connected subnetworks. The alignment is performed on these subnetworks constructed after the graph partitioning. This approach increases the computational cost of the solution. This computational cost can easily be observed when the running times of the two methods are compared. However this computational cost comes with the flexibility in determining orthological clusters. Even with the most loosely connected protein interaction network, the developed method is able to determine a number of cluster candidates and then evaluate the relevance of these candidates. NetworkBLAST is a lot more strict when considered from this perspective and it cannot produce any cluster matches if there are no obviously orthologous clusters in the provided networks.

3.4 The Error Tolerance of the Method

The effect of false positive protein interactions are considered to analyze the tolerance of the method to errors. For this purpose, false positive interactions are included into the datasets to introduce some errors. While introducing the errors, the degrees of the nodes should be protected as much as possible. For this purpose the protein interaction networks are randomized depending on a probability value. Two protein interactions in the dataset, namely A-B and C-D, are removed from the dataset and new interactions, A-D and B-C, are introduced as false positives. This randomization process is illustrated in Figure 3.5. The probability value determines the number of times this process is repeated. It is defined relative to the number of protein interactions in the dataset. During our experiments we have used 0.05 for the probability of randomization. So 10 of 100 interactions are randomized with our application of the method. The randomized interactions may result with an interaction that already exists in the database. But these repeated interaction pairs are eliminated during the application of the methods defined for orthology extraction. The reason for choosing such a large randomization probability is to find the validity of the defined scoring metric. If the results are unsuccessful because of the randomization performed, this shows that the interaction data used to predict orthologies is an important part of the prediction process. Keeping the probability value high, the interaction data is defined to be nonrealistic but the degrees and connectivity of the nodes

in the real dataset are tried to be protected as much as possible.

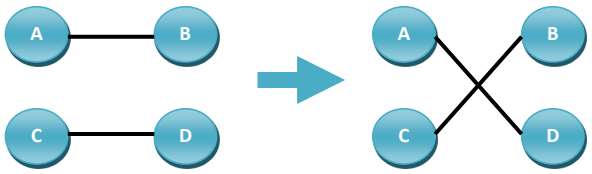

Figure 3.5: The illustration of the interaction randomization process

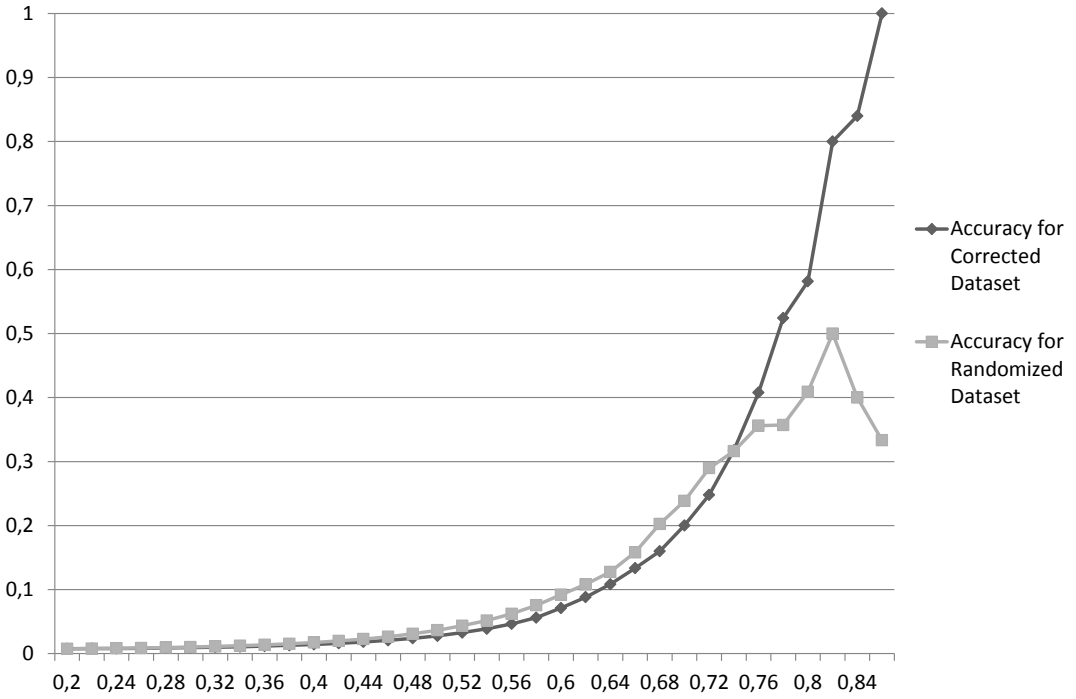

Figure 3.6: Chart comparing the accuracy changes for the original and randomized datasets. The horizontal axis represents different cutoff values of cluster similarity used to accept or reject predictions. The vertical axis stands for accuracy values of predictions achieved for different cutoff values.

The series of methods developed for cluster orthology detection are applied just in the same way as described in Chapter 2. The results of the application of the methods are illustrated in Figures 3.6 and 3.7. Some of the result values used for the construction of these charts are provided in Table 3.3. As can be seen from Figure 3.6, the cluster similarity metric becomes invalid for the randomized interactions. The accuracy is relatively low for high values of

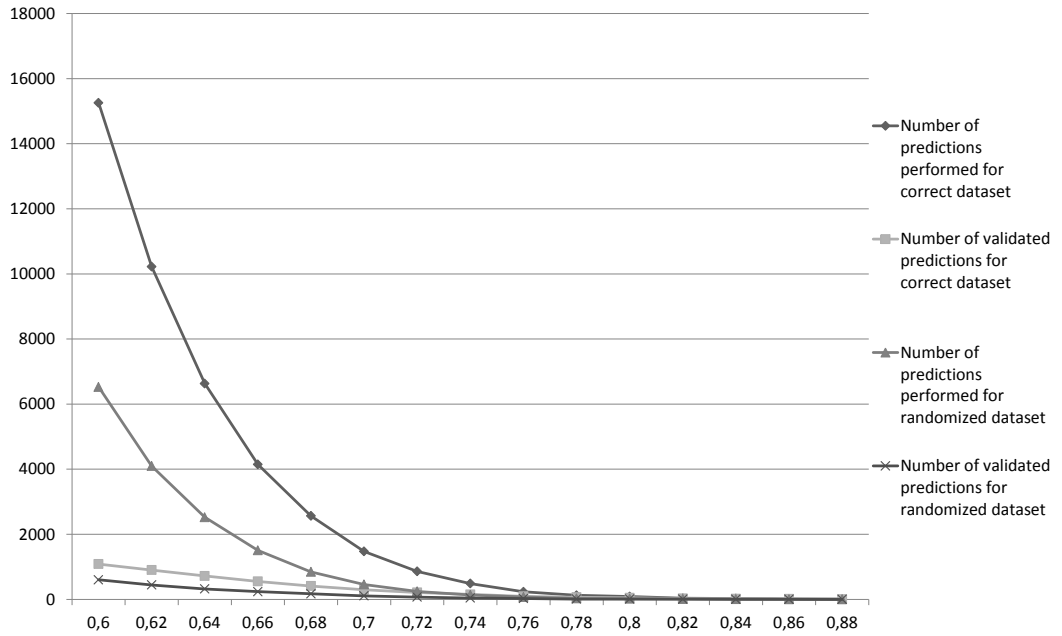

Figure 3.7: Chart representing the number of predictions performed for the original and randomized datasets. The horizontal axis represents different cutoff values of cluster similarity used to accept or reject predictions. The vertical axis stands for the number of predictions performed for different cutoff values. It is also possible to see the number of validated orthology predictions in this chart.

cluster match similarity. There are no cluster matches with similarity value over 0.9. All the orthologically relevant clusters are defined to be along the range of 0.6 to 0.7. Also as can be seen from Figure 3.7, the number of cluster match predictions increase drastically. More orthologically relevant protein groups can be determined between the cluster similarity range of 0.6 and 0.7. However this shows that cluster similarity metric is not successful in determining the orthological protein groups any more. So the cluster similarity metric becomes irrelevant with the orthological terms of proteins in the clusters. The reason for the decrease in cluster similarity is related to the first phase of the algorithm. Since the interaction data is randomized, the protein clusters are not preserved. So the protein interaction data does not include functional modules of interactions. This shows that the protein interaction data has an important role in defining the similarities of the cluster. The most critical part of the algorithm is the formation of the protein clusters. Errors in protein interaction data prevents the formation of functionally related clusters. However the experiment performed here is over exagerated. With a small number of false positive protein interactions, the methods developed here is expected to work correctly.

Table 3.3: The table of values used for forming the charts in Figures 3.6 and 3.7. The columns of the table represents the achieved results for different cutoff values of cluster similarity on original and randomized datasets. Information about the number of performed predictions, the number of validated predictions and the accuracy of the predictions for cluster similarity over the defined cutoff value can be achieved for both of the applied datasets from these columns.

Cutoff Value	**0.60**	**0.64**	**0.68**	**0,72**	**0,76**	**0,80**	**0,84**
Number of predictions performed on original dataset	15255	6631	2563	859	233	86	25
Number of validated predictions performed on original dataset	1085	410	1568	213	95	50	21
Accuracy of the predictions on original dataset	0.0711	0.1084	0.1599	0.2479	0.4077	0.5813	0.84
Number of predictions performed on randomized dataset	6527	2524	844	245	73	22	5
Number of validated predictions on randomized dataset	601	322	171	71	26	9	2
Accuracy of the predictions on randomized dataset	0.0920	0.1275	0.2026	0.2897	0.3561	0.4090	0.4

It can be concluded that the method developed to determine orthologous protein clusters are not error tolerant during the formation of functionally related protein clusters. But a small number of false positive interactions can be tolerated for large clusters. On the other hand, the results of randomization showed that the algorithm and distance metrics developed are in fact valid for determining the orthologies correctly. The randomization process destroy the functional modules inside the protein interaction network. It is no suprise that the method can not determine functional modules. This result shows that the method avoids false positive predictions. If there exist a strongly connection group of proteins, it determines these protein groups.

3.5 Computational Complexity of the Method

The developed algorithm consists of three main steps. The first step determines the strongly connected proteins in the proteomes of different species. The second step is the elimination of dissimilar cluster pairs using graph theoretic properties of the formed clusters. Finally the cluster similarities are evaluated by using the GO Annotations of the proteins. The computational complexity of the implemented algorithm can be evaluated as a sum of these three main steps.

In the first step of the algorithm, the clusters are determined with the application of Repeated Random Walks Algorithm (RRW) developed by Macropol *et al.* [25]. It is claimed that the computational complexity of the RRW algorithm is $O(w \times |V^2|)$ where V is the number of nodes in the protein interaction network and w is the number of iterations until the convergence of the values. Highest cost is introduced by the number of nodes in this complexity model. The number of proteins is relatively too large when compared to other parameters.

In the second phase of the algorithm, two types of information are used, namely the number of nodes in clusters and the number of edges between the proteins of the clusters. The computational complexity of making the decision of either to eliminate the cluster pair or not is $O(1)$ if these two types of information are known. However the node and edge counts should also be extracted from the protein interaction networks. Calculation of these values can be performed with a computational complexity of $O((V + E) \times TC)$ where V is the total number of nodes in the proteomes, E is the total number of edges in the proteomes and TC is the total number of clusters formed by RRW. The total cost of the second stage of the algorithm becomes $O((V + E) \times TC)$ when these two complexities are multiplied.

The final step of the algorithm evaluates the cluster similarities for reasonable cluster matches. For evaluating the computational complexity of this phase of the algorithm, a bottom-up approach will be used. The computational complexity of evaluating the similarities between two GO terms is defined to be $O(GOV + GOE)$ since the GO term similarity evaluation algorithm can be reduced to longest path problem. In this complexity formula, GOV stands for the number of GO Terms and GOE stands for the number of edges between the GO terms in the GO Hierarchy Tree. However the maximum depth of the GO Hierarchy Tree is not high and the longest path search is performed using the ancestor and descendant relationships. So

the complexity of evaluating the similarities of two GO terms are directly proportional to the depth of the GO Hierarchy Tree. Since the maximum depth of the GO Hierarchy Tree is about 30, the similarity evaluation of two GO Terms can be accepted to be $O(1)$. The similarity of two proteins are computed by all-vs-all comparison of the associated GO Terms of two proteins. So the computational complexity of evaluating the similarities of two proteins is $O(GOC_1 \times GOC_2)$ where GOC_1 stands for the number of GO Terms associated with the first protein and GOC_2 stands for the second protein. The cluster similarity is evaluated by aligning the proteins in the clusters using the protein similarity metric. The number of nodes in clusters are more or less the same for the compared cluster pairs. For this reason, we assume the number of proteins the two clusters are equal. The computational complexity for finding clusters similarity is $O(C^2 \times (COG_1 \times COG_2) + C)$ in this situation where C stands for the number of nodes in a cluster. This computational complexity can be further simplified to $O(C^2 \times (COG_1 \times COG_2))$. Finally the clusters should be compared to each other according to the elimination performed in the second step of the algorithm. If the number of clusters matches that passed the elimination in the second step is AC, then the overall complexity of the third step of the algorithm is $O(AC \times (C^2 \times (COG_1 \times COG_2)))$.

With these complexities of the phases of the algorithm, the complexity of the whole algorithm is defined to be $O((w \times |V^2|) + ((V + E) \times TC) + (AC \times (C^2 \times (COG_1 \times COG_2))))$ where the parameters in this definition are given in the construction of these complexities. With these complexity definitions, the most dominant steps of the algorithm seems to be the cluster construction phase and the cluster alignment phase. The given complexity definition can be summarized as the algorithm is exponentially increasing with respect to the number of nodes in the protein interaction networks and the number of nodes in the constructed clusters.

CHAPTER 4

DISCUSSION AND FUTURE WORK

In this study, an exponential algorithm for predicting orthologically relevant protein clusters in different species is proposed. This algorithm uses protein interaction information together with the GO annotations of proteins to infer the functionally related protein groups and perform matching between these protein groups of different species. Protein orthology prediction problem is addressed by several studies in literature. Most of these studies aim to find orthology information about the proteins with the use of protein sequence information together with protein interaction information. However there are not many studies aiming to find orthology between clusters of functionally related proteins. In this study such a method is proposed without using protein sequence information.

The proposed algorithm consists of three main steps. The first step is to determine clusters of proteins that are strongly connected. In this step, Repeated Random Walks Algorithm [25] is applied with the use of the protein interactions extracted from STRING database [26]. The second step of the algorithm is eliminating the possible cluster matches that are not related by considering the graph theoretic properties of the clusters. With this step of the algorithm, the cluster matches that are quite dissimilar are elimated. This way the number of computations required to be performed in the third step of the algorithm is reduced. In the third step of the algorithm, the final cluster matching is performed. A cluster similarity metric including biological information through the use of GO terms is defined for comparing clusters of proteins of different species. The result of applying these three steps of the algorithm is a list of cluster matches sorted according to their cluster similarity.

The experiments performed on the organisms *Mus Musculus* and *Homo Sapiens* showed that the cluster similarity defined to match clusters can succesfully determine the orthologies.

Cluster matches with a similarity score over 0.85 is shown to be orthologically relevant. For the validation of the cluster matches, COG terms [42] of the proteins are used to evaluate the orthology of the clusters. COG database provides an ontology determining the orthological distances between proteins. The evaluation performed with the COG terms show that the cluster matches having similarity score over 0.86 are 100% orthologically relevant. For a cluster similarity of 0.8, the accuracy of orthology reduces to 60%. These results can be used as a proof for the validity of the developed scoring metric.

The performance of the method is also compared with a protein orthology prediction method called ISORANK [38]. ClustOrth aims to find orthologically related protein clusters. However a small modification allows prediction of protein orthologies. The network alignment performed on the clusters is used to extract orthologous proteins for the high scoring cluster matches. The protein orthology predictions are evaluated using Interolog Database [48]. The results of our method are not as accurate as predictions performed by ISORANK for low values of cluster similarity. But for clusters similarities over 0.9, the accuracy of the predictions are higher that the results of ISORANK. It should be noted that ISORANK is computationally more complex that the method developed in this study. All vs. all protein similarities should be calculated using BLAST algorithm in order to apply ISORANK. This process is time consuming for large protein interaction networks. On the other hand, even with the similarity scores supplied the main ISORANK algorithm is not able to produce results for large protein networks. It was not possible to get orthology matches for the organisms *Mus Musculus* and *Homo Sapiens*. For this reason, the comparison tests are performed on the *Saccharomyces Cerevisiae* and *Drosophila Melanogaster* organisms. This comparison test showed the relevance of our cluster similarity metric with orthology. These tests also validated our results on different organisms and with different orthology information sources. Although less number of predictions can be done compared to ISORANK, the accuracy of the predictions are higher than predictions performed by ISORANK for high cluster similarity values.

There does not exist many computational methods trying to perform orthological protein cluster matching. The method named NetworkBLAST [36] is one of the benchmark solutions on this problem in literature. When the performance of ClustOrth is compared with NetworkBLAST, it is a lot more flexiable in determining clusters of proteins in protein networks. It does not force any constraints on the evaluation of protein cluster orthologies but scores the possible cluster matches of protein networks. This makes the method more tolerant to miss-

ing information in protein interaction networks. In the previous validation efforts, this score is proved to be valid in protein cluster orthology determination. Experiments on two different organism pairs show that our method is more successful in determining protein complexes and finding the orthological relevance between these complexes. However NetworkBLAST is computationally less expensive than our method. The number of network alignments required to compare constructed protein clusters increase the computational cost of our method. It can also be said that NetworkBLAST is more successful in determining orthologous protein pairs. Direct usage of protein sequence similarities increase the accuracy of protein orthology predictions. These predictions are extracted from protein network alignments performed during the application of the method.

As a final validation strategy, error tolerance of the method is tested. The protein interactions of the *Mus Musculus* and *Homo Sapiens* organisms are randomized and the methods are applied in the same way as applied on the correct data. The accuracy results decrease for high cluster similarity values drastically. However the number of predictions performed increased since the functionally related protein clusters are distorted. The functionally related modules in protein interaction networks should be protected for the method to function properly. However the method can be accepted error tolerant for small mistakes in the interaction data which does not affect the functional modules.

During the experiments performed to validate the method, several weaknesses of the algorithm are determined. The first weakness of the algorithm is the construction phase of the protein clusters. The provided protein interaction network should be complete in order to be partitioned correctly with the Repeated Random Walks algorithm. Otherwise the method cannot find the clusters of functionally related proteins successfully. Likewise providing the confidence levels of the protein interactions is highly recommended. Otherwise the Repeated Random Walks algorithm consider each defined edge equally and this results with clusters that are not strongly related. This weakness can be overcomed by using a reliable source of protein interaction data such as STRING, DIP or a similar well-known database.

Another weakness of the method is that the number of orthologous protein clusters are small in number. This value can be increased with the inclusion of other measures of orthological similarity to increase the similarity scores between clusters. Currently used scoring metric is too strict to accept clusters as orthologous if there is any lack of information on any of the

proteins in the clusters. Introduction of a different type of data may increase the confidence values for this type of the clusters.

Although we define orthological cluster similarities in this study, we do not define a cutoff value for determining whether two clusters are orthologous or not. With some learning algorithms, the cutoff value for the defined similarity metric can be learned with respect to the processed dataset. This way, the algorithm can also perform direct orthology predictions. This learning process should be related with a protein orthology database in order to perform accurate predictions. On the other hand, instead of using some constant values in the cluster elimination phase of the algorithm, learning algorithms may be further introduced to learn the most suitable values for these constants. Another different solution may be graph embedding. We think that graph embedding may simplify the implementation and reduce the complexity of the method. However the memory requirements of the solution may be higher than current implementation.

Also by improving the implementation of the algorithm, developing a web server for the online usage of this method can be considered as another future work of this study. Given the protein interaction files representing protein interaction networks, the web based implementation can produce lists of protein clusters constructed from the provided protein interaction files together with the matches between these clusters. The GO Annotations of the proteins in the protein interaction networks can be extracted online with the use of web services. The main challenge in implementing this web based solution is the response time of the algorithm. A method for returning the results after a couple of hours should be found. An email can be sent when the computation of cluster matches are completed. Also another problem may be related to the server load. The server load would be too high for a couple of computations performed at the same time. So a powerful server should be used together with a succesful task scheduler. The implementation can also be parallelized in order to get the results faster. The cluster similarity evaluation phase of the developed algorithm can be distributed over a number of processors and all clusters can be evaluated all together in a short time.

On the other hand, the orthological accuracies achieved for clusters with similarity values over 0.85 are expectionally high. This is a strong evidence of the correctness of the scoring metric defined. To our knowledge there are not many previous studies trying to determine and match functionally related protein groups. The developed algorithm is as good as the

current solutions in literature. With the developed algorithm, it is possible determine and match protein groups orthologically. With an improvement on the similarity metrics used in the algorithm, we believe that the method can address many more orthologically relevant protein clusters correctly.

REFERENCES

[1] Andreu Alibés, Patricio Yankilevich, Andrés Cañada, Ramón Diaz-Uriarte "IDconverter and IDClight: Conversion and annotation of gene and protein IDs" *BMC Bioinformatics*, 2007. 8:9

[2] Fátima Al-Shahrour, Pablo Mínguez, Joaquín Tárraga, David Montaner, Eva Alloza, Juan M. Vaquerizas, Lucia Conde, Christian Blaschke, Javier Vera and Juaquín Dopazo, "BABELOMICS: a systems biology perspective in the functional annotation of genome-scale experiments", *Nucleic Acids Research* , 2006. 34, W472-W476.

[3] Karl-Heinrich Anders, "A Hierarchical Graph-Clustering Approach to find Groups of Objects", *ICA Commission on map generalization, 5th workshop on progress in automated map generalization*.

[4] Gary D. Bader, Ian Donaldson, Cheryl Wolting, B. F. Francis Ouellette, Tony Pawson and Christopher W. V. Hogue, "BIND - The Biomolecular Interaction Network Database", *Nucleic Acids Research* 2001. Vol. 29, No. 1 242-245

[5] Sourav Bandyopadhyay, Roded Sharan and Trey Ideker, "Systematic Identification of functional orthologs based on protein network comparison", *Genome Res.* 2006. 16: 428-435

[6] Chris Biemann, "Chinese Whispers - an Efficient Graph Clustering Algorithm and its Application to Natural Language Processing", *Workshop on TextGraphs at HLT-NAACL 2006* 2006. pages 73-80

[7] F. J. Brandenburg, "Graph clustering I: Cycles of cliques", *Graph Drawing* 1997. Volume 1353/1997 Pages 158-168

[8] Ulrik Brandes, Marco Gaertler and Dorothea Wagner, "Experiments on Graph Clustering Algorithms", Algorithms - ESA 2003. vol. 2832, pp. 568-579

[9] Christine Brun, François Chevenet, David Martin, Jérõme Wojcik, Alain Guénoche and Bernard Jacq, "Functional classification of proteins for the prediction of cellular function from a protein-protein interaction network", *Genome Biol.* 2004. 5(1): R6

[10] Horst Bunke, P. Foggia, C. Guidobaldi and M. Vento, "Graph Clustering Using the Weighted Minimum Common Supergraph", *IAPR Workshop GbRPR* 2003. pp.235-246

[11] Jingchun Chen and Bo Yuan, "Detecting functional modules in the yeast protein-protein interaction network", *Bioinformatics* 2006. 22(18):2283-2290

[12] Inderjit Dhillon, Yuqiang Guan and Brian Kulis, "A fast Kernel-based Multilevel Algorithm for Graph Clustering", *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining* 2005. pages 629-634

[13] Jubin Edachery , Arunabha Sen and Franz J. Brandenburg, “Graph Clustering Using Distance-k Cliques” , *Graph Drawing* 1999. Volume 1731/1999, pages 98-106

[14] The Gene Ontology Consortium, “Gene ontology: tool for the unification of biology” *Nat. Genet.* May 2000. 25(1):25-9.

[15] Simon Günter and Horst Burke, “Validation indices for graph clustering”, *Pattern Recognition Letters* 2003. 24, 1107-1113

[16] Desmond J. Higham, Marija Raajski and Nataša Pržulj, “Fitting a geometric graph to a proteinâprotein interaction network”, *Bioinformatics* 2008. 24(8):1093-1099

[17] Eitan Hirsh and Roded Sharan, “Identification of conserved protein complexes based on a model of protein network evolution”, *Bioinformatics* 2007. 23(2):e170-e176

[18] H. Jeong, B. Tombor, R. Albert, Z. N. Oltvai and A.L. Barabasi, “The large-scale organization of metabolic networks” , *Nature* 2000. 407, 651-654

[19] Brian P. Kelley, Roded Sharan, Richard M. Karp, Taylor Sittler, David E. Root, Brent R. Stockwell and Trey Ideker, “Conserved pathways within bacteria and yeast as revealed by global protein network alignment”, *PNAS* 2003. vol. 100 no. 20 11394-11399

[20] Brian P. Kelley, Bingbing Yuan, Fran Lewitter, Roded Sharan, Brent R. Stockwell and Trey Ideker, “PathBLAST: a tool for alignment of protein interaction networks”, *Nucleic Acids Research* 2004. 32(Web Server Issue):W83-W88

[21] Mehmet Koyutürk, Yohan Kim, Umut Topkara,Shankar Subramaniam, Wojciech Szpankowski and Ananth Grama, “Pairwise Alignment of Protein Interaction Networks”, *Journal of Computational Biology* 2006. 13(2): 182-199

[22] Brian Kulis , Sugato Basu, Inderjit Dhillon and Raymond Mooney, “Semi-supervised graph clustering: a kernel approach”, *Machine Learning* 2009. Volume 74, Number 1, 1-22

[23] Stanley Letovsky and Simon Kasif, “Predicting protein function from protein/protein interaction data: a probabilistic approach” , *Bioinformatics* 2003. Vol. 19 Suppl. 1, pages i197-i204

[24] Bin Luo, Richard C. Wilson and Edwin R. Hancock, “Spectral Feature Vectors for Graph Clustering”, *Structural, Syntactic, and Statistical Pattern Recognition* 2009. Volume 2396/2009, Pages 423-454

[25] Kathy Macropol, Tolga Can and Ambuj K Singh, “RRW: repeated random walks on genome-scale protein networks for local cluster discovery”, *BMC Bioinformatics* 2009. 10:283

[26] Christian von Mering, Martijn Huynen, Daniel Jaeggi1, Steffen Schmidt, Peer Bork and Berend Snel, “STRING: a database of predicted functional associations between proteins”, *Nucleic Acids Research* 2003. Vol. 31, No. 1 258-261

[27] Tijana Milenković and Nataša Pržulj,“Uncovering Biological Network Function via Graphlet Degree Signatures”, *Cancer Inform.* 2008. 6: 257-273

[28] Tijana Milenković , Jason Lai and Nataša Pržulj, “GraphCrunch: a tool for large network analyses”, *BMC Bioinformatics* 2008. 9:70

[29] Kevin P. O'Brien, Maido Remm and Erik L.L. Sonnhammer, "Inparanoid: A Comprehensive Database of Eukaryotic Orthologs" , *Nucleic Acids Research* 2005. 33:D476-D480

[30] Jose B. Pereira-Leal, Anton J. Enright and Christos A. Ouzounis, "Detection of functional modules from protein interaction networks", *Proteins: Structure, Function, and Bioinformatics* 2003. Volume 54 Issue 1, Pages 49 - 57

[31] Stefano Rizzi, "Genetic operators for hierarchical graph clustering" *Pattern Recognition Letters* 1998. Pages 1293-1300

[32] Tom Roxborough and Arunabha Sen, "Graph clustering using multiway ratio cut (Software demonstration)", *Graph Drawing* 1997. Volume 1353/1997, pages 291-296

[33] Reinhard Sablowski and Arne Frick, "Automatic Graph Clustering", *Proceedings of the Symposium on Graph Drawing table of contents* 1996. Pages: 395 - 400

[34] Satu Elisa Schaeffer, "Graph clustering", *Computer Science Review* 2007. Volume 1, Issue 1, Pages 27-64

[35] Paul Shannon, Andrew Markiel, Owen Ozier, Nitin S. Baliga1, Jonathan T. Wang, Daniel Ramage, Nada Amin, Benno Schwikowski and Trey Ideker,"Cytoscape: A Software Environment for Integrated Models of Biomolecular Interaction Networks", *Genome Res.* 2003. 13: 2498-2504

[36] Roded Sharan, Silpa Suthram, Ryan M. Kelley, Tanja Kuhn, Scott McCuine, Peter Uetz, Taylor Sittler, Richard M. Karp and Trey Ideker, "Conserved patterns of protein interaction in multiple species", *PNAS* 2005. vol. 102 no. 6 1974-1979

[37] Rohit Singh, Jinbo Xu and Bonnie Berger, "Pairwise Global Alignment of Protein Interaction Networks by Matching Neighborhood Topology" , *RECOMB2007* 2007., LNBI 4453, pp. 16-31

[38] Rohit Singh, Jinbo Xu and Bonnie Berger, "Global alignment of multiple protein interaction networks with application to functional orthology detection" , *PNAS* 2008. vol. 105 no. 35 12763-12768

[39] Chris Stark, Bobby-Joe Breitkreutz, Teresa Reguly, Lorrie Boucher, Ashton Breitkreutz and Mike Tyers, "BioGRID: a general repository for interaction datasets", *Nucleic Acids Research*, 2006. Vol. 34, Database issue D535-D539

[40] Matthew Suderman and Michael Hallett, "Tools for visually exploring biological networks", *Bioinformatics* 2007. 23(20):2651-2659

[41] Reiko Tanaka, "Scale-Rich Metabolic Networks", *Phys. Rev. Lett.* 2005. 94, 168101

[42] Roman L. Tatusov, Darren A. Natale, Igor V. Garkavtsev, Tatiana A. Tatusova, Uma T. Shankavaram, Bachoti S. Rao, Boris Kiryutin, Michael Y. Galperin, Natalie D. Fedorova and Eugene V. Koonin, "The COG database: new developments in phylogenetic classification of proteins from complete genomes" , Nucleic Acids Research, 2001. Vol. 29, No. 1 22-28

[43] James D Watson, Roman A Laskowski and Janet M Thornton, "Predicting protein function from sequence and structural data", *Current Opinion in Structural Biology* 2005. Volume 15, Issue 3, Pages 275-284

[44] Roland Wiese, Markus Eiglsperger and Michael Kaufman, “yFiles: Visualization and Automatic Layout of Graphs”, “Lecture Notes in Computer Science”, 2002. Volume 2265/2002, 588-590.

[45] Xiaomei Wu, Lei Zhu, Jie Guo, Da-Yong Zhang and Kui Lin,“Prediction of yeast protein-protein interaction network: insights from the Gene Ontology and annotations”, *Nucleic Acids Research* 2006. 34(7):2137-2150.

[46] Ioannis Xenarios and David Eisenberg, “Protein interaction databases”, *Current Opinion in Biotechnology* 2001. Volume 12, Issue 4, Pages 334-339

[47] Ioannis Xenarios, Lukasz Salwinski, Xiaoqun Joyce Duan, Patrick Higney, Sul-Min Kim and David Eisenberg, “DIP, the Database of Interacting Proteins: a research tool for studying cellular networks of protein interactions”, *Nucleic Acids Research* 2002. Vol. 30, No. 1 303-305

[48] Haiyuan Yu, Nicholas M Luscombe, Hao Xin Lu, Xiaowei Zhu, Jing-Dong J. Han, Nicolas Bertin, Sambath Chung, Marc Vidal, Mark Gerstein “Annotation transfer between genomes: protein-protein interologs and protein-DNA regulogs” *Genome Research* 2004. 14: 1107-18.

APPENDIX A

GLOSSARY OF THE TERMS

- Babelomics : A web based tool that can be used for conversion of different protein annotations
- Drosophila Melanogaster : Fruit Fly
- GO Term : Gene Ontology Term
- Homo Sapiens : Human
- Homology : Similarities between the anatomy, nucleic or amino acid sequences / structures in organisms owing to shared ancestry
- MRCA : Most recent common ancestor
- Mus Musculus : Mouse
- Orthology : Genes or gene products in different species that derive from a common ancestor
- Paralogy : Homologous genes within a single species that diverged by gene duplication
- Proteome : The entire set of proteins expressed by a genome, cell, tissue or organism
- Proteomics : The large scale studies of proteins, particularly their structures and functions
- RRW : Repeated Random Walks Algorithm
- Saccharomyces Cerevisiae : Baker's Yeast