Protein Function Prediction
based on Protein – Protein Interaction Networks

by

Derui Wang
Bachelor of Engineer
Huazhong University of Science and Technology

Submitted in fulfilment of the requirements for the degree of

Master of Science (by research)

Deakin University
May, 2016
I am the author of the thesis entitled

**Protein Function Prediction based on Protein – Protein Interaction Networks**

submitted for the degree of

**Master of Science (by research)**

This thesis may be made available for consultation, loan and limited copying in accordance with the Copyright Act 1968.

'I certify that I am the student named below and that the information provided in the form is correct'

Derui Wang

Full Name: .................................................................................................................................

(Please Print)

Signed: .................................................................................................................................

Signature Redacted by Library

Date: .................................................................................................................................

11/05/2016
I certify the following about the thesis entitled (10 word maximum)

**Protein Function Prediction based on Protein – Protein Interaction Networks**

submitted for the degree of **Master of Science (by research)**

a. I am the creator of all or part of the whole work(s) (including content and layout) and that where reference is made to the work of others, due acknowledgment is given.

b. The work(s) are not in any way a violation or infringement of any copyright, trademark, patent, or other rights whatsoever of any person.

c. That if the work(s) have been commissioned, sponsored or supported by any organisation, I have fulfilled all of the obligations required by such contract or agreement.

d. That any material in the thesis which has been accepted for a degree or diploma by any university or institution is identified in the text.

e. All research integrity requirements have been complied with.

‘I certify that I am the student named below and that the information provided in the form is correct’

**Derui Wang**

(Please Print)

**Signed:** .......................................................... 

**Date:** 11/05/2016

---

Deakin University CRICOS Provider Code 00113B
Acknowledgements

During the two-year master degree process, I have received helps and guidance from supervisors, faculties, students and my families.

I am glad that I had a lot of supports from my supervisor, Dr Jingyu Hou during writing my first journal article and my graduation thesis. During my way to the Master degree, advices from Dr Hou helped me a lot during bettering the quality of my paper and perfecting my understanding about knowledge and trends in the research direction. I have also received helps and assistance from Professor Wanlei Zhou, Professor Yang Xiang, Dr Sheng Wen from Deakin University, Professor Leann from Melbourne University and Professor Benjamin from University of Texas South-western Medical Center Under their advices, I sorted out my way of conducting research and really understood what is of interest to me.

I want to specially thank my parents and families for their supports along the way. The spiritual and material support they have given me is one of the essential reasons why I can successfully complete my thesis. I want also thank my friends and colleagues who have enlightened me or helped me on data structure optimisation, algorithm improvement and proof reading of my papers and thesis. At last, I am glad that I could finish my master degree in Australia in which the scenic landscapes can literally distract a lot of attention from ones.

The persistence and long process behind an impacting research and the limit of research time is such a pair of paradoxes, and I believe it will continue to be a great puzzle for young researchers like me who has just stepped into the world of research. I wish I could somehow push my research to a higher level in my future academic work.

Derui Wang
Burwood campus, Deakin University
Australia
May 2016
List of Publications

Refereed Journal Article

CONTENTS

List of figures ................................................................................................................................ 4
List of tables ................................................................................................................................. 6
Abstract .......................................................................................................................................... 7
Chapter 1 Introduction ............................................................................................................. 9
  1.1 Background ........................................................................................................................... 9
    1.1.1 In silicon protein function analysis .............................................................................. 9
    1.1.2 Protein-protein interaction network ............................................................................. 10
    1.1.3 Function annotation schemes ..................................................................................... 13
    1.1.4 Heterogeneous biological data incorporated in prediction ...................................... 15
  1.2 Problems and objectives .............................................................................................. 21
    1.2.1 Problems .................................................................................................................... 21
    1.2.2 Objectives .................................................................................................................. 22
  1.3 Significance of the research ........................................................................................ 24
  1.4 Thesis organisation ........................................................................................................ 24
Chapter 2 Preliminaries ........................................................................................................ 26
  2.1 Protein function prediction approaches based on static PPIN ........................................ 26
    2.1.1 Algorithms based on network structure information .................................................. 27
    2.1.2 Algorithms based on semantic information ................................................................. 35
    2.1.3 Algorithms based on multiple genomic data sources .................................................. 39
  2.2 Applications of dynamic PPIN based approaches .......................................................... 49
    2.2.1 DPPIN construction .................................................................................................... 49
    2.2.2 Protein complex detection ......................................................................................... 51
    2.2.3 Critical protein identification ...................................................................................... 54
  2.3 PPIN datasets used in this research ................................................................................ 55
  2.4 Validation and evaluation ............................................................................................... 56
    2.4.1 Evaluation Metrics ...................................................................................................... 56
2.4.2 Validation Methods ........................................................................................... 58
2.5 Summary ................................................................................................................... 59

Chapter 3 Iterative protein function prediction method ........................................... 61

3.1 Introduction .............................................................................................................. 61
3.2 Iterative model ........................................................................................................ 63
  3.2.1 Unannotated protein searching ................................................................... 64
  3.2.2 Protein function initialisation ...................................................................... 64
  3.2.3 Iterative function updating ........................................................................ 65
  3.2.4 Prediction result selecting .......................................................................... 66
  3.2.5 Function Prediction Algorithms .................................................................. 67
3.3 Results and discussion .......................................................................................... 70
  3.3.1 Datasets ........................................................................................................... 70
  3.3.2 Performance Metrics .................................................................................... 72
  3.3.3 Prediction Performance ............................................................................... 72
  3.3.4 Sensitivity Analysis over iteration times ..................................................... 78
  3.3.5 Robustness of the iterative approach ......................................................... 79
3.4 Conclusions .............................................................................................................. 81

Chapter 4 Contemporary protein function prediction approach ........................... 83

4.1 Introduction .............................................................................................................. 83
  4.1.1 Background .................................................................................................... 83
  4.1.2 Challenges and problems ............................................................................ 85
  4.1.3 Objective definition ...................................................................................... 86
4.2 Methods .................................................................................................................. 87
  4.2.1 Data processing and network construction .................................................. 87
  4.2.2 Protein Function Prediction ........................................................................ 90
4.3 Results ..................................................................................................................... 95
  4.3.1 Datasets .......................................................................................................... 95
  4.3.2 Results Evaluation ........................................................................................ 98
4.4 Conclusion .............................................................................................................. 104
Chapter 5 Implementations ................................................................. 106

5.1 Genetic Tools ................................................................................. 106
5.2 Implementation of the INC/IWNC methods .............................. 109
5.3 Gene Ontology tools ................................................................. 109

Chapter 6 Discussion and conclusion ............................................. 112

6.1 Discussion ..................................................................................... 112
6.2 Conclusion ................................................................................... 113
  6.2.1 Review of main findings ......................................................... 113
  6.2.2 Future work and recommendations ................................. 114

Bibliography ..................................................................................... 116

Appendix .......................................................................................... 126
List of figures

Figure 1.1.2.1. The classical yeast two-hybrid system.
Figure 1.1.2.2. Part of the yeast protein-protein interaction network.
Figure 1.1.3.1.1. DAG that represents a subset of terms and relationships in GO.
Figure 1.1.3.2.1. Part of FunCat annotation map.
Figure 1.1.4.1. Heterogeneous biological data used for protein function prediction.
Fig. 1.2.1.1. The proportions of unannotated proteins in model species.
Figure 2.1.3.1.1. A sample of GO INDEX.
Figure 2.1.3.2.1. Bi-graph cluster.
Figure 2.1.3.2.2. Illustrates two strategies for random walk.
Figure 2.2.2.1. Two types of hubs.
Figure. 3.2.1. Diagram of the proposed iterative approach.
Figure. 3.3.3.1. Comparisons on datasets containing certain levels of unannotated proteins.
Figure. 3.3.3.2. Comparisons on datasets containing random levels of unannotated proteins.
Figure. 3.3.3.3. P-R curves of iterative and non-iterative algorithms.
Figure. 3.3.4.1. Sensitivity Analysis of INC on datasets that consist of different proportions of unannotated proteins.
Figure. 3.3.5.1. Numbers of target proteins that have identical predicted functions from multiple independent predictions.
Figure 4.2.1.1. The heat map based on part of the averaged gene expression matrix.
Figure 4.2.1.2. The expression profiles of YBR272C, YIL085C, YPL210C, YJL093C, YML111W and YLR291C from yeast.
Figure 4.2.2.1. The transformation from PPIN to PFA.
Figure 4.2.2.2. The PFA graphs of YKL108W from the 10th transient PPIN and YKL092C from the 9th transient PPIN.
Figure 4.3.2.1.1. The overall recall.
Figure 4.3.2.1.2. The precision comparison of the predicted contemporary functions.
Figure 4.3.2.2.1. The Precisions of prediction results of Protein set a.
Figure 4.3.2.2.2. The Recalls of prediction results of Protein set a.
Figure 4.3.2.2.3. The F-values of prediction results of Protein set a.
Figure 5.1.1. The software structure of Genetic Tools.
Figure 5.1.2. The interfaces of Genetic Tools.
Figure 5.2.1. Implemented INC and IWNC in Genetic Tools.
Figure 5.3.1. Searching parent terms of a GO term.
Figure 5.3.2. Checking namespace of a GO term.
Figure 5.3.3. GO terms similarity calculation.
List of tables

Table 1.1.4.1. Databased of different biological data types.
Table 2.1.1. Representative PPI protein function prediction algorithms.
Table 2.1.2.1.1. Three classic measurements for GO term semantic similarity.
Table 3.3.5.1. Prediction results of YHL011C.
Table 4.2.2.2.1. PFA pruning procedure.
Table 4.3.1.1.1. The summary of training datasets.
Table 4.3.1.1.2. The features of training datasets.
Table 4.3.2.1.1. Cell cycle phases in physiological time.
Table 4.3.2.1.2. Selected father GO terms.
Abstract

Proteins play important roles in biological processes, and it is one of the main research areas in the post-genome era to annotate functions of unknown proteins. However, annotating protein functions through biological experiments is a challenging work due to the high cost. Nowadays, computational approaches are widely used in this area as they can predict functions of proteins more efficiently and effectively compared to biological experiment. On the other hand, with the rapidly expanding biological data volume, network tools are widely applied for biological data analysis.

Protein-protein interaction networks constructed by high throughput technologies provide the opportunities for predicting protein functions. A lot of heuristic methods and algorithms based on machine learning have been applied on PPI networks to predict functions of unannotated proteins over recent decades. However, existing algorithms usually only employ annotated proteins in prediction process, which results in limited prediction performance. Moreover, current algorithms are usually one-off predictions which have not considered the dynamics of protein interaction. Crucially, dynamically changing transient functions of proteins are difficult to be revealed by existing approaches and algorithms.

To tackle problems above, firstly, the research presented here proposed one iterative approach which utilises unannotated proteins and their interactions in prediction. Additionally, a software demonstration was developed for the iterative method and related bioinformatics application. Secondly, a method for dynamic contemporary protein function prediction was developed in this research. The second contemporary function prediction method incorporated heterogeneous data for the purpose of prediction. This method utilised gene expression data to investigate protein interactions and functions at specific time intervals. Therefore, this method introduced a possible solution for understanding protein activities during a short period of time. By predicting time-course
protein functions, this method elaborated the prediction of protein annotation to a more delicate level.

According to experimental results, the first iterative approach maximally improved the prediction precision and recall of representative protein function prediction algorithms by 50% on MIPS and BioGrid datasets. Importantly, the iterative approach initially proposes an idea that iteratively incorporates the interaction information of unannotated proteins into the protein function prediction and can be applied on existing prediction algorithms to improve prediction performance. On the other hand, the contemporary function prediction algorithm achieved satisfying performance both on contemporary function prediction and overall protein function prediction.
Chapter 1

Introduction

In this chapter, at first, the relevant background to this thesis is introduced in Section 1.1. Then we describe problems in the research area and define research objectives in Section 1.2. We then conclude the significance of this research in Section 1.3. Finally, Section 1.4 provides an outline about the structure of this thesis.

1.1 Background

Proteins are crucial biological molecules that play important roles in the expression of genes and assembly of other molecules. Also they form the basic structure of organisms while play vital roles in metabolism. Identifying protein functions is a crucial task in molecule biology. Understanding protein functions is necessary for studying biological processes. However, identification of protein functions through biological experiment is expensive and inefficient in many circumstances. Therefore, protein function prediction by computational approaches is becoming an increasingly important way to study functions of unknown proteins.

1.1.1 In silico protein function analysis

In early stage of in silico protein function prediction methods, the most representative method is based on alignment of amino acid/nucleotide or sequence motif searching. The idea behind using alignment to predict functions of proteins is that usually proteins have similar amino acid sequence tend to show same functions in biological pathway. Basic Local Alignment Search Tool (BLAST, Altschul et al., 1990) is a widely used tool for sequences searching and alignment. However, sequence alignment methods lack of
efficiency since it is difficult to determine sequence similarity between proteins. Moreover, it fails to consider diverse functions of protein (Chua and Wong, 2009).

1.1.2 Protein-protein interaction network

Biological networks are one of the primary methods for translating raw biological data into knowledge that could be interpreted. There are several kinds of biological networks such as gene regulation network (Davidson et al., 2005), protein-interaction network (Schwikowski et al., 2000) and disease network (Kwang et al., 2007) being used to analyse associations between biological molecules and phenotypes. Benefiting from application of high-throughput technology, data of protein interactions could be discovered much more rapidly than ever before. Subsequently, high-throughput technology is used for generating large-scale data of protein interactions which could be used to construct a protein-protein interaction network stands for proteins and their interaction (Aebersold et al., 2003). Accordingly, Protein-Protein Interaction networks (PPIN) were constructed for proteome research. A PPI network or protein-interaction network (PIN) is a biological network that state associations between proteins.

PPIN is constructed from interaction data generated by high throughput technology. The typical method of protein interaction detection is yeast two hybrid (Ito et al., 2001). The Y2H technique allows detection of interacting proteins in living yeast cells. Interaction between two proteins, called bait and prey, activates reporter genes that enable growth on specific media or a colour reaction (Anna et al., 2009). The comprehensive two-hybrid screening system is described in Figure 1.1.2.1.
Figure 1.1.2.1. The classical yeast two-hybrid system. (A) The protein of interest X, is fused to the DNA binding domain (DBD), a construct called bait. The potential interacting protein Y is fused to the activation domain (AD) and is called prey. (B) The bait, i.e. the DBD-X fusion protein, binds the upstream activator sequence (UAS) of the promoter. The interaction of bait with prey, i.e. the AD-Y fusion protein, recruits the AD and thus reconstitutes a functional transcription factor, leading to further recruitment of RNA polymerase II and subsequent transcription of a reporter gene. (Figure and legend come from the paper authored by Anna et al., 2009)

During translating interaction data into network model, proteins become nodes in network while pairwise interactions become edges in the graph. If there are multiple pieces of evidence bearing on the same pairwise interaction they are combined into a single link (Letovsky et al., 2003). A PPI network is usually represented by an undirected graph, in which a node corresponds to a protein and an edge corresponds to the interaction between a pair of proteins. Usually, the PPI network is illustrated as an undirected graph $G = (V, E)$, where $V$ represents the vertices and $E$ represents the set of edges in network (i.e. Figure 1.1.2.2: part of the yeast protein-protein interaction network generated from STRING database (http://string-db.org)).
During analysis towards functions of proteins in yeast (Saccharomyces cerevisiae) PPI network, proteins that have same function and cellular location tend to be clustered together, with 63% of the interactions occurring between proteins with a common functional assignment and 76% occurring between proteins found in the same subcellular compartment (Schwikowski et al., 2000). Followed by further research conducted by Titz et al., it was found that 70–80% of proteins share at least one function with its interacting partner (Titz et al., 2004). At the meantime, about 70% proteins in yeast PPI network share functions with their level-1 proteins (the directly interacting partners with the protein in network) and level-2 proteins (the proteins directly interact with level-1 proteins) (Chua et al., 2006). As the consequence, a lot of mathematical and computational methods have been applied on the protein-protein interaction network for predicting functions of unannotated proteins.

Current prediction algorithms from PPI network utilize PPI information, semantic information and related genomic information in prediction. However, current algorithms remove unannotated (unknown) proteins from the PPI data, which leads to absence of protein interaction information. Moreover, the research in protein interactions in living cells (Chua et al., 2009) shows that proteins interact with each other, rather than working
alone, to perform their functions in various biological processes. As a simulation of natural protein interacting process, it should be more rational to consider that functions of interacting proteins are decided mutually by both proteins in a bi-direction way. However, current algorithms and approaches treat the prediction process as one-off process, which means functions of unknown protein are raised from its interacting proteins, and the predicted results will not in return affect those interacting partners.

1.1.3 Function annotation schemes

Function annotation means discovering biological functions of a protein and depicting these functions by certain vocabularies. Meanwhile, the ultimate purpose of predicting protein functions is to assign annotation terms to unannotated protein by using information of annotated proteins in PPIN. Proteins play crucial roles in biological system, to illustrate what functions a protein has, and several annotation strategies have been proposed. In the majority of cases, annotation is written as scientific natural language. However, it is difficult for computer to unbiasedly understand in machine process. Ontologies offer a mechanism by which knowledge can be represented in a form capable of such processing (Lord et al., 2003).

1.1.3.1 Gene Ontology

Currently, one of the most important ontologies within the bioinformatics community is Gene Ontology (GO) (The Gene Ontology Consortium, 2001). GO is a rapidly growing collection of over 40,000 biological phrases, representing terms or concepts, held within a directed acyclic graph (DAG) (Figure 1.1.3.1.1). GO comprises three orthogonal taxonomies (three namespaces), which hold terms that describe the attributes of molecular function, biological process and cellular component for a gene product.
Figure 1.1.3.1.1, DAG that represents a subset of terms and relationships in GO. I (is a), P (part of) and R (relate to) are three kinds of relationships in GO (http://geneontology.org).

The GO project has developed three structured, controlled vocabularies (ontologies) that describe gene products in terms of their associated biological processes, cellular components and molecular functions in a species-independent manner. The structure of GO can be described in terms of a graph, where each GO term is a node, and the relationships between the terms are edges between the nodes. GO is loosely hierarchical, with “child” terms being more specialized than their “parent” terms, but unlike a strict hierarchy, a term may have more than one parent term (Gene Ontology Consortium, 2001). GO annotation has been a powerful standard for computational protein function analysis. In protein function prediction area, usually GO is used to define function similarity.

1.1.3.2 FunCat annotation scheme

Another important annotation strategy for protein function is Function Category (FunCat)(http://www.helmholtz-muenchen.de/en/ibis/resourcesservices/services/functcat-the-functional-catalogue). It is also a hierarchical annotation scheme. “The FunCat annotation scheme consists of 28 main categories (or branches) that cover general
features like cellular transport, metabolism and protein activity regulation. Each of the main functional branches is organized as a hierarchical, tree like structure” (Andreas et al., 2004). Main functional categories of FunCat contain Metabolism, Information pathways, Transport, Perception and response to stimuli, Developmental processes, Localization. A two-digit number represents function vocabulary. Annotation for particular protein function is hierarchical combinations of these numbers, which are separated by dot (i.e. “10.01.05.03.05”). Experimentally uncharacterized proteins are annotated by “98” (classification not yet clear-cut) or “99” (unclassified proteins).

FunCat annotation has a simpler format for programming and analysis on computer than GO. However, GO defines a more robust system of ontology and more precise vocabularies to describe specific biological functions.

1.1.4 Heterogeneous biological data incorporated in prediction

Incorporating heterogeneous biological data along with PPIN to predict protein function could decrease the false positive/false negative rate of the PPIN data. Since different types of biological data are biologically associated with each other, heterogeneous data are
capable for being incorporated with PPIN data in protein function prediction. The evidences of protein interaction observed from heterogeneous data further strengthen the credibility of interactions in PPIN, which improves the quality of the original PPIN data. The associated heterogeneous biological data types are illustrated in the diagram below:

![Heterogeneous biological data used for protein function prediction.](image)

In the following subsections, we introduced the basic concepts of these data types and their associations with protein function prediction. A list of databases resources for each type of data is presented as follow:

<table>
<thead>
<tr>
<th>Data type</th>
<th>Database</th>
<th>Resource link</th>
</tr>
</thead>
<tbody>
<tr>
<td>Protein domain</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Protein complex</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Protein sequences</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Protein Structure</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DNA Sequence</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Protein interactions</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Gene Expression Micro-array</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>DNA sequence</strong></td>
<td><strong>Gene expression</strong></td>
<td><strong>Protein Structure</strong></td>
</tr>
<tr>
<td>---------------------------</td>
<td>-------------------------</td>
<td>--------------------------------</td>
</tr>
<tr>
<td>Ensembl</td>
<td>Gene Expression Omnibus (GEO)</td>
<td>Protein Data Bank (PDB)</td>
</tr>
<tr>
<td>(Hubbard et al., 2002)</td>
<td>(Edgar et al., 2002)</td>
<td>(Helen Berman et al., 2003)</td>
</tr>
<tr>
<td>GenBank</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(Benson D. et al. 2008)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>European Nucleotide Archive (ENA)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(Cochrane, G. et al. 2007)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DNA Data Bank of Japan (DDBJ)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(Tateno Y. et al. 2002)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*Table 1.1.4.1. Databased of different biological data types.*
Protein/DNA sequence data

Protein sequence data contains the amino acid sequences of proteins. Since proteins function mainly through binding, the arrangements of amino acids derive different protein sequences. Protein sequences fold into unique three-dimensional structures. Protein sequence information was dominantly used in the early stage of protein function analysis. The sequences of proteins are commonly stored as plain text format in databases.

DNA sequences are the arrangements of nucleotides (i.e. A, C, T and G) on a DNA molecule. Proteins are coded by special regions (protein coding sequences) on DNA molecules, the protein coding sequences thus largely decide the sequences of proteins. DNA sequence are usually stored as structured text format.

Two proteins that have similar sequences are considered functionally similar; thus similar sequence motifs which are short and unique sequences have been used as a feature for similar functioning protein identification (David et al., 2007). Blast (Altschul et al., 1990) is the most widely employed sequence alignment tool based on heuristic algorithms. However, a general problem with sequence motifs is that short sequence matches typically have low statistical significance and the false-positive rate can be high.

Protein domain data

A protein domain is a stable units of protein tertiary structure that could fold autonomously that can evolve, function, and exist independently of the rest of the protein chain (Bork, 1991). A protein may contain multiple protein domains.

Protein domains are classified by their functions. Through protein structure observation, domains of a protein can be identified. Then function of the protein can be inferred from
the domain information. However, domain with unknown functions can also be used to verify the credibility of protein interactions in order to improve function prediction performance. Protein domain was incorporated with PPI data to predict protein functions (Peng et al., 2014).

**Protein complex data**

Protein complexes are groups of proteins which are physically interacting to complete crucial biological functions. Protein complexes have been manually curated from biological experiments to infer the functional connections among single proteins (i.e. CYC2008, Pu et al., 2008). Protein complex information are usually stored as text format in databases.

Protein complexes generally correspond to dense subgraphs in the PPI network. Thus, proteins in a given complex are highly interactive with each other (Bader et al., 2003, Pržulj et al., 2004). Predicting protein function hence is transformed to the question that finding to which is the complex the unannotated protein belongs. Clustering methods are representative methods for complex identification in PPIN.

**Protein Structure data**

Protein structure refers to the three-dimension arrangement of monomer amino acids. There are four levels of structures which are primary structure, secondary structure, tertiary structure and quaternary structure. Primary structure is the sequence of amino acids in the polypeptide chain. Secondary structure stands for regular local sub-structures on the actual polypeptide backbone chain. Tertiary structure refers to the three-dimensional structure of protein molecules. Finally, quaternary structure is the three-dimensional structure and physical combination of a multi-subunit protein. Protein structure data are stored as graphic structure or text such as unit cell dimensions and
angles for x-ray crystallography determined structures in databases.

Protein structure largely infers protein function since the biological function of protein molecules is determined by their 3D shape (Lopez et al, 2007). Proteins having the same biological functions resemble each other on structures. Thus, through recorded protein structure data, functions of particular proteins can be revealed.

**Gene expression data**

Gene expression data are usually from micro-array experiments or RNA sequencing. A microarray is typically a glass slide on to which DNA molecules are fixed in an orderly manner at specific locations called spots. A microarray may contain thousands of spots and each spot may contain a few million copies of identical DNA molecules that uniquely correspond to a gene (M. Madan, 2011). RNA sequencing (RNA-seq) is a relatively new analysis method which can measure quantified level of gene expression. RNA-seq measures gene expression level via measuring corresponding mRNA concentration level (Zhong et al., 2009). The quantified gene expression level could be measured by counting the RNA sequencing reads per gene.

Integrating expression data with other external information, for example evolutionary conservation of proteins, have been used to predict interacting proteins, protein complexes, and protein function. Several works (Jansen et al., 2000, Ge et al., 2001) have shown that genes with similar expression profiles are more likely to encode proteins that interact. Additionally, expression correlation coefficient has been regularly used to assess the credibility of protein interactions (Deng et al., 2003, Chao et al., 2012, Hindol et al., 2014).
1.2 Problems and objectives

We explain the problems in current protein function prediction methods and outline our objectives in this section.

1.2.1 Problems

Currently, there are large proportions of unannotated proteins in the PPIN of each species (Figure 1.2.1 (Roded et al., 2006)). However, a large proportion of prediction approaches that do not rely merely on PPIN topological information are sensitive towards unannotated proteins in the prediction domain (i.e. sub-network that consists of proteins required to serve as the information sources for prediction algorithms or to participate in the prediction processes) since unannotated proteins, as well as their corresponding interaction information, do not participate in the prediction process. In other words, a lot of interaction information has been discarded when predicting functions, which affects...
the prediction performance significantly, especially for a high quality PPI dataset. This also means algorithms that utilise annotation information (e.g. semantic information based methods or some classification algorithms) are incapable of predicting functions of a protein if most of its neighbour proteins are unannotated.

On the other hand, the research in protein interactions in living cells discovered that proteins interact with each other, rather than working alone, to perform their functions in various biological processes (Chua et al., 2009). Thus as a simulation of natural protein interacting process, it should be a more rational prediction model that functions of interacting proteins are decided mutually by both proteins in a bi-direction way. An iterative prediction approach based on iteratively updating protein similarities between annotated neighbourhood proteins and the prediction target was proposed (Chi and Hou, 2011) to address this problem. However, most of the current prediction algorithms do not reflect this feature of protein interactions, and the prediction process was considered as a one-off process. That means the prediction is mono-directed from annotated proteins to un-annotated proteins, and once the functions of un-annotated protein have been predicted, the prediction is finished.

Finally, proteins perform different functions during different times. However current computational prediction methods focus on predicting general functions that may be performed by proteins rather than addressing specific stage and time at which the functions occur.

1.2.2 Objectives

In this research, we focus on improving the performance of current static PPIN based prediction methods and discovering of novel protein function analysis methods based on DPPIN. We outline the issues that will be addressed in this research as solving the following problems:
The large proportions of unannotated proteins in different species decrease the prediction accuracy of protein functions. Annotation rates of the gene functions in representative species are varied. However, in most of the critical datasets such as human genome dataset, gene functions are far from well-studied. Even in the most studied genome, the Yeast genome, there are still about 25% of genes remaining unannotated. Since protein function prediction heavily depends on annotated protein to infer functions of unannotated protein, a lack of annotation in the PPIN could lead to incorrect prediction.

Protein interactions are dynamic biological processes. However, current protein function prediction algorithms are mostly one-off predictions which have not considered the dynamic features of the protein interactions. One-off prediction means after functions of a protein has been predicted, they do not change anymore. In another words, the prediction results do not return its feedback to the PPI network and its neighbour proteins. The impact of the prediction result towards neighbour proteins is ignored. It could be viewed as a senator failed to give feedback to his/her voters after winning a selection, which makes it difficult to exam whether voters had made the right choice. Considering that interactions between proteins are not one-direction process but bi-direction, using prediction result as feedbacks to neighbours in PPI network and iteratively change the prediction result is a potential way to address this problem.

Proteins perform their functions at certain stages of the cell cycle instead of the entire cell cycle. Though some of the function annotation terms themselves describe particular biological processes that occur at certain stage of cell cycle, current protein function prediction methods are not able to generate detailed function information on time scale. However, identifying occurring functions at certain time intervals would be of interest to a series of biological research such as drug target
identification and biological reaction environment identification.

1.3 Significance of the research

At first, the outcome of this research helps to drop the economic cost for protein function identification. Proteins are crucial biological molecules that play important roles in the expression of genes and assembly of other molecules. They also form the basic structure of organisms and act as vital parts in metabolism. Identifying protein functions is therefore a crucial task in molecule biology as it is necessary for studying biological processes. Furthermore, protein function identification is indispensable process in drug design and bio-engineering. However, the cost of identifying protein functions through biological experiments could be relatively high, and laboratory experiments could be inefficient in many circumstances. Computational approaches provide pre-analysis of potential protein functions, which provides evidence or chances for efficiently conducting experiments in laboratory.

Secondly, this research improved the protein function prediction performance, and provide an insight to protein function prediction on a more detailed level. Based on this research, existing protein function analysis methods could be further improved when the annotations of a PPIN are sparse. Moreover, this research proposed an approach for predicting protein functions at particular time. This research explored the further possibilities of computational analysis of protein function and would probably lead to a series of further related researches.

1.4 Thesis organisation

This section gives the overall structure of this thesis. This thesis consists of totally six chapters. The purposes of chapters are listed as follows: Chapter 2 provides a survey on
existing PPIN based protein function prediction methods. Representative methods and algorithms are presented and catalogued. Chapter 3 proposes an iterative prediction method proposed during this research. This method reduces the negative effects caused by unannotated proteins in PPINs during function prediction. Furthermore, this method can be applied on existing algorithms to improve their performances. Chapter 4 proposes a method for predicting contemporary protein functions. This method incorporates heterogeneous data to construct dynamic PPIN (DPPIN) and predict temporal functions performed by certain proteins. Chapter 5 introduces a GUI tool implemented in this research. The proposed iterative protein function prediction approach is embedded in the tool. Additionally, this GUI tool provides several Gene Ontology applications. Chapter 6 discusses the trends in bioinformatics research about proteomic data, and analyses potential improvements that could be made in our future work to upgrade the performance of the proposed approaches. This chapter then gives a conclusion about the outcomes from this research.
Chapter 2

Preliminaries

This chapter provides an extensive literature review of the state-of-art methods, data pre-processing and the trends in this research. At first, Section 2.1 specifically depicts representative approaches and algorithms developed for protein function prediction based on static PPIN. Secondly, Section 2.2 reviews existing DPPIN applications including critical protein identification and protein complex detection. This section exhibits some established DPPIN construction techniques as well as existing related to DPPIN. Furthermore, Section 2.3 gives an insight about the pre-processing of biological data that were used in PPIN/DPPIN construction and protein function prediction. Finally, in Section 2.4, we discuss validation methods and evaluation criteria that are wildly used to examine the performance of protein function prediction methods.

2.1 Protein function prediction approaches based on static PPIN

Usually, static PPIN based methods for protein function prediction can be divided into two categories, which are direct methods and module-assisted methods (Roded et al. 2006). A variety of new algorithms have emerged since then, many algorithms utilize information from function annotation terms, which also be known as semantic information, together with PPI network structure information, while some algorithms combine multiple sources of biological information to predict function. To indicate the trend of bringing connections among biological data into protein function analysis, in this review, algorithms will be divided into three categories based on data types used in protein interaction credibility assessment. Algorithms based on network structure information, algorithm based on semantic information and algorithms based on multiple genomic data sources. Basically,
differences between these three categories could be viewed as variation in dimension of information sources that used to assess credibility of interactions in PPI network. Within each category, algorithms are further divided into direct method and module assistant method. Direct method is based on the observation that neighbour proteins share similar functions. Direct method could be viewed as vote process done by annotated neighbour proteins to select appropriate functions for unknown protein. Module assistant methods are another kind of representative method in protein function prediction. It attempts to first identify coherent groups of proteins and then assign functions to all the proteins in each group based on function of the group which is also known as functional module. The following sections provide detailed introductions to some representative approaches based on static PPIN.

<table>
<thead>
<tr>
<th>Method Type</th>
<th>Direct Method</th>
<th>Module assistant method</th>
</tr>
</thead>
<tbody>
<tr>
<td>Network Structure Information Based Algorithms</td>
<td>NC, Chi-square, FS-Weight, PRODISTIN, Graph search, MRF, CC</td>
<td>MCODE, DPClus, MCL, CFINDER, SL, PEREIRA, OCG, SEEDY, GAS</td>
</tr>
<tr>
<td>Semantic Information Based Algorithms</td>
<td>CIA, DPA</td>
<td>SWEMODE, RNSC, STM</td>
</tr>
<tr>
<td>Multiple Genomic Data Sources Based Algorithms</td>
<td>GeneFAS, DCS/DSCP</td>
<td>SAMBA, GFA, DMSP, MS-KNN, JCA</td>
</tr>
</tbody>
</table>

Table 2.1.1. Representative PPI protein function prediction algorithms.

2.1.1 Algorithms based on network structure information

This kind of methods and algorithms only use information generated from PPI networks to assess weight (credibility) of each interaction. Also, most early methods could be
categorized into this kind of algorithm. In this section, several algorithms based on direct methods and module assistant methods that use protein interaction data to predict functions of unknown proteins are introduced.

2.1.1.1 Direct method

Typical methods consist of foundational Neighbour Counting (NC) algorithm (Schwikowski et al., 2000) and following enhanced algorithms such as Chi-square algorithm (Hishigaki et al., 2001), Function Similarity weight (FS-weight) algorithm and PRODISTIN (Brun et al. 2003). Markov random field (MRF) approach initially proposed by Deng et al. (Deng et al., 2003) assign function to proteins by probability, followed by a similar approach by Letovsky et al. (Letovsky et al., 2003). Also graphic theory based algorithm (Vazquez et al., 2003, Karaoz et al., 2004) and machine learning algorithms as collective classification (Qingyao et al., 2014, Wei et al., 2014) are introduced.

Neighbour Counting Algorithm/ Chi-square Statistics

Predicting functions of unannotated proteins based on topology information is first realized by the Neighbour Counting (NC) algorithm, which was proposed by Schwikowski et al. in 2000. It predicts functions of each protein by counting the frequency of each function being found in its neighbourhood proteins. The more frequent a function showed in protein’s level-1 neighbours (directly interacting proteins), the more likely it is a function of the protein (Schwikowski et al., 2000). Let p stand for the unannotated protein to be predicted, Np is the set of level-1 proteins of p, p’ is protein in Np. δ(p’,f)=1 if function f is found in p’, otherwise δ(p’,f)=0. A function f is scored base on formula below:
Function with higher score would be chosen as prediction result. Originally, NC algorithm chooses top three functions ranked by scores as functions of unannotated protein \( p \).

Subsequently, Hishigaki et al. improved the NC algorithm by using the statistical method, Chi-square, to calculate scores of a function assigned to unannotated protein instead of simply count how many time it occurs in neighbour proteins. Assume the number of a function \( f \) being found in level-1 neighbours of unannotated protein \( p \) is \( N_f \), and \( E_f \) denotes the expectation number of \( f \) in \( n \)-neighbouring proteins expected from the distribution on the total PPI network, a function is scored using:

\[
Score_f = \frac{(N_f - E_f)^2}{E_f} \tag{2.1.1.1.2}
\]

Function with highest score will be assigned as prediction result. In circumstances that multiple functions are both having the largest score, all of them will be assigned (Hishigaki et al., 2001).

**FS-Weighted Algorithm**

Since level-2 proteins also contain significant number of functions as information source in prediction procedure, Chua et al. proposed a new definition of protein similarity named function similarity weight (FS-weight) to measure importance of each function in different proteins towards the unannotated protein, instead of treat each protein as equally important. Meanwhile they proposed a “function similarity weighted averaging” algorithm in which function similarity weight was applied (Chua et al., 2006).
The function similarity weight defines similarity between proteins based on the number of their shared proteins in PPI network. Thus similarity among proteins that share their neighbours and highly connected should be relatively higher than others. Correspondingly, during prediction process, functions in those proteins that have similarity with unannotated protein would get higher weight in function vote. Use $N_v$ and $N_u$ to denote level-1 neighbour protein sets that two proteins $u$ and $v$ have respectively, while $u$ and $v$ are neighbour of each other ($u$ is level-1 neighbour protein of $v$ vice versa). $S(u, v)$, the similarity of these two proteins is given by:

\[
S(u, v) = \frac{2|N_u \cap N_v|}{|N_u - N_v| + 2|N_u \cap N_v|} + \frac{2|N_u \cap N_v|}{|N_v - N_u| + 2|N_u \cap N_v|}
\] (2.1.1.3)

If protein $u$ is similar to protein $w$, and protein $w$ is similar to protein $v$, it stands that $u$ and $v$ have transitive functional association, and similarity between $u$ and $v$ could be depicted as Transitive FS-Weighted similarity:

\[
STR(u, v) = \max(S(u, v), \max_{w \in N_u} S(u, w) S(w, v))
\] (2.1.1.4)

where $S(u, v)$ is the FS-weight between $u$ and $v$. Since then, prediction algorithms are usually using level-1 and level-2 protein as prediction domain. Since PPI network is represented as a graph, it is rational to use graphic algorithm for revealing functions of unknown proteins in the network.

**MRF**

MRF model is relying on a Markovian assumption: the function of a protein is independent of all other proteins given the functions of its immediate neighbours. In Deng’s approach, the function annotation problem is viewed as, for a function $f$, the probability that a protein $p$ is assigned with $f$ that
occurs with frequency \( f \) is:

\[
P(f) = \log\left(\frac{\pi}{1-\pi}\right) + \alpha N_{p,1} + \beta (N_{p,1} - N_{p,0}) - N_{p,0} \quad (2.1.1.5)
\]

in which \( \pi \) is the probability of a protein having the function \( f \). \( N_{p,1} \) is p’s neighbour protein which has function \( f \). \( N_{p,0} \) is p’s neighbour which do not have function \( f \). \( \alpha \) and \( \beta \) are estimated parameters.

**Graph Search**

Vazquez et al. published a method based on PPI network of physical interactions as determined functions of proteins by minimizing the number of protein interactions among different functional categories (Vazquez et al., 2003). In another word, this method attempts to maximize \( E \) below:

\[
E = \sum_{i,j} J_{ij} \delta(\sigma_i, \sigma_j) + \sum_i h_i(\sigma_i) \quad (2.1.1.6)
\]

\( J_{ij} = 1 \) if \( i \) and \( j \) are two linked unknown proteins, 0 otherwise. \( \delta(\sigma_i, \sigma_j) \) is the discrete function that equals 1 if \( \sigma_i = \sigma_j \) and 0 otherwise. \( h_i(\sigma_i) \) is the number of annotated partners of protein \( i \) with function \( \sigma_i \). The first term in the optimization criterion concerns unannotated proteins, whereas the second term accounts for interactions between unannotated and previously annotated proteins. The optimization searching job of \( E \) is down by simulated annealing. Similar method is proposed one step behind (Karaoz et al., 2004).
Collective Classification (CC)

Collective classification is increasingly used in PPI network data for function annotation for proteins and received attention since last decade. Various CC algorithms has been proposed (McDowell et al., 2012), including iterative classification algorithm (ICA) (Neville et al., 2000), Gibbs sampling (Gibbs) (Jensen et al., 2004) and variants of the weighted-vote relational neighbour algorithm (wvRN) (Macskassy et al., 2007). Wei et al. proposed an CC algorithm along with an active learning strategy. The active learning consists of spectral clustering algorithm that divides the whole network into several clusters and graph-based centrality metrics that assign weight to each edge. Its purpose is to select most representative and informative sample candidate by active learning in order to better training classifier (Wei et al., 2014). Classification is done via Gibbs sampling.

2.1.1.2 Module assistant method

Usually, module finding is carried out through clustering proteins in PPI network. In a recent review, clustering approaches to PPI networks can be broadly categorized as topology-free and graph-based ones. Topology-free approaches use traditional clustering techniques in terms of notions of distance between proteins that do not take into account the topology of the network. Graph-based clustering approaches consider instead the topology of the network, and usually rely on specialized clustering techniques (Pizzuti et al., 2013). Well-known algorithms such as MCODE (Bader et al., 2003), DPClus (Altaf-Ul-Amin et al., 2006), CFINDER (Adamcsek et al., 2006), MCL (Enright et al, 2002), and OCG (Becker et al., 2012) they cluster nodes into high-density clusters. Pereira (Pereira et al., 2004) clusters edges instead of nodes. Following are several representative clustering methods make use of network structure information.
**MCL**

MCL (Enright et al., 2002; Van Dongen, 2008) is a widely used graphic clustering method. It combines concepts of random walk (Lovasz, 1996) and Markov chain. Random walk assumes an object walks on the undirected graph $G = (V, E)$ from node to node along edges in the graph. When the random walk starts from a node $V_a \in V$ which has degree of $k$ (the node is connected with $k$ nodes), the object could randomly chooses a way (means an edge) to depart. Thus for each edge the probability of being chosen is $1/k$ (if the edge is un-weighted), which means after certain steps of random walk, the paths and nodes that the object walked through would be a highly connected sub-graph that $V_a$ belongs to. Markov chain is a statistical case that a variable is relying on its states in previous stage. In random walk, probability of walking to a node only depends on the states of previous walk (probabilities for next time step only depends on current probability). By using an adjacent matrix that element $a_{ij}$ stands for probability of walking to node $j$ from node $i$, it is obvious that after powering the matrix with itself $N$ times until the matrix do not change anymore, then the structure of clusters is emerging. As MCL applied on PPI network, elements of the adjacent matrix (Markov matrix) are weighted by similarities between proteins (i.e. element $a_{ij}$ is similarity between protein $i$ and $j$). The protein similarity is compute by comparing protein sequence similarity using BLAST.

**MCODE**

The "Molecular Complex Detection" (MCODE) algorithm is a cluster algorithm that detects dense and connected regions by weighting nodes on the basis of their local neighbourhood density (Bader et al., 2003). MCODE algorithm consists of three main steps, which are vertex weighting,
Molecular Complex Prediction and Post-Processing (optional). In vertex weighting stage, it applies concept of \( k \)-core, which is a sub-graph in which each vertex has a degree of at least \( k \). The highest \( k \)-core of a graph is the most densely connected sub-graph. MCODE finds out highest \( k \)-core graph for each vertex within its level-1 neighbours, then uses \( k \)-number of \( k \)-core graph and density of \( k \)-core to weight the protein. After each protein is weighted, the clustering begins. Node with the highest weight will be regarded as seed; neighbour vertexes whose weight exceeds certain threshold will be joined into the cluster. This process stops once no more vertices can be added to the complex based on the given threshold and is repeated for the next highest unseen weighted vertex in the network. Clusters containing no 2-core are filtered. After this stage is completed, we have two options in final stage, “fluff” which enlarge the cluster, and “haircut” which shrinks the cluster. For fluff, a “fluff” parameter between 0.0 and 1.0 is given. Other vertexes could be added into cluster if they have not yet been seen and if their neighbourhood density is higher than the given fluff parameter. If “haircut”, then those leaf vertexes (vertex singly connected to cluster) are removed. Also, “fluff” and “haircut” could be used together in order that applying “fluff” first, then applying “haircut”.

**Pereira**

Pereira is a link clustering method that clusters interactions in PPI network instead of clustering proteins to detect function modules. Given an input PPI network \( G(V', E') \), this approach builds the corresponding line graph \( L(V, E) \). In particular, a vertex of \( L \) represents an edge in PPI network, and two vertices are adjacent in \( L \) if and only if their corresponding interactions in PPI network share a common node (Pereira et al., 2004). Then MCL is applied to generate function modules from the graph.
2.1.2 Algorithms based on semantic information

Semantic information refers to annotation information used in prediction algorithms. Lord et al. were the first to apply GO-based semantic similarity to compare gene products (Lord et al., 2003). Essence of algorithms that employs semantic information is in which relationship between terms in annotation system are brought into prediction process. Usually, similarities between proteins and functions are defined by this relationship. In another words, interaction credibility is assessed further by semantic dimension than just by structure of PPI network. Main GO semantic similarity definition consists of information content (IC) based definition and GO graph based definition. The representative semantic based methods are introduced in the following subsections.

2.1.2.1 Direct method

Most direct methods are heuristic methods using semantic similarity between function annotation terms of proteins to determine the potential functions of unannotated proteins. The function from an annotated protein which has higher similarity with the unannotated protein will get higher weight or score when assigned to unannotated proteins as its potential function.

Information content based semantic similarity measurement

Three classic measurements for semantic term similarity were opted for evaluation in Catia’s paper. These measures use information content (IC) as metric. Information content of a term t is calculated by:

\[ IC(t) = -\log p(t) \] (2.1.2.1.1)
in which $p(t)$ is the usage frequency of term $t$ in corpus. To calculate it, first we count for each term the number of distinct proteins annotated to it or one of its descendant terms, and then divide that number by the total number of annotations within the corresponding GO namespace \((\text{Catia et al., 2008})\). In the table below are three classic IC based semantic similarity measurements.

<table>
<thead>
<tr>
<th>Author/Year</th>
<th>Measurement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Resnik's measurement ((\text{Resnik et al., 1999}))</td>
<td>$\text{Sim}_{\text{Res}}(t_1, t_2) = IC(t_A)$</td>
</tr>
<tr>
<td>Lin's measurement ((\text{Lin et al., 1998}))</td>
<td>$\text{Sim}_{\text{Lin}}(t_1, t_2) = \frac{2 \times IC(t_A)}{IC(t_1) + IC(t_2)}$</td>
</tr>
<tr>
<td>Jiang and Conrath's Similarity measurement ((\text{Jiang et al., 1998}))</td>
<td>$\text{Sim}_{\text{JAC}}(t_1, t_2) = 1 + \frac{IC(t_A) - \frac{IC(t_1) + IC(t_2)}{2}}{2}$</td>
</tr>
</tbody>
</table>

Note: $t_1$ and $t_2$ are two GO terms. $t_A$ is the ancestor of $t_1$ and $t_2$ with the highest IC value.

\( Table 2.1.2.1.1. \) Three classic measurements for GO term semantic similarity.

There are other definitions of semantics such as graph-based \(\text{simUL}\) \((\text{Gentleman et al., 2005})\) and \(\text{simGIC}\) \((\text{Catia et al., 2008})\) measures. However, most of assessment works agreed that Resnik’s measure is one of the best semantic similarity measures \((\text{Pietro et al., 2011})\).

\( \text{CIA} \)

A Cosine Iterative Algorithm (CIA) that employs semantic information for
predicting and iteratively changes similarity values between proteins based on the previous predicted functions was proposed (Xiaoxiao et al., 2011). This method uses GO annotation terms to compute similarity of two functions as following:

\[
f_{\text{sim}}(f, f') = \frac{\vec{f} \cdot \vec{f}'}{\| \vec{f} \| \| \vec{f}' \|} \tag{2.1.2.1.2}
\]

in which \(\vec{f}\) and \(\vec{f}'\) are two respective functions represented by vector form. \(\| \vec{f} \|\) and \(\| \vec{f}' \|\) are the norm of two vectors, respectively. GO annotation is used to construct the vector. As discussed in Chapter 1, GO annotation is a hierarchical annotation method that uses a directed acyclic graph to depict relationships between each GO term. A GO term may have multiple parent GO terms (ancestors), if a function is annotated by a GO term. It is also annotated by the ancestors of the GO term. Therefore, the vector element values at the index positions that correspond to these ancestors are set to 1, otherwise set to 0. If the number of terms in GO is \(t\), the dimension of each function vector \(f\) is then \(t\). It could be observed that \(f_{\text{sim}}\) get a higher value if two functions have higher proportion of GO annotation terms in common. Then protein similarity is defined as:

\[
\text{Sim}(p, p') = \frac{1}{\max(m, n)} \sum_{f \in F(p)} \sum_{f' \in F(p')} \delta(f', f') \tag{2.1.2.1.3}
\]

in which \(F(p)\) and \(F(p')\) are set of functions in protein \(p\) and \(p'\) respectively. And \(m\) is the size of \(F(p)\) (number of functions in \(F(p)\)) while \(n\) is size of \(F(p')\). After function similarity and protein similarity are defined, score of a function \(f\) in protein \(p\) is:
$Score(p,f) = \sum_{p' \in N(p)} [Sim(p,p') \times (\sum_{f' \in F(p')} fSim(f,f') \times log\frac{N}{n_{f'}})]$ (2.1.2.1.4)

where $N$ is the number of all proteins in the PPI dataset and $n_{f'}$ is the number of proteins in the dataset that have the function $f'$. The statistical factor $log\frac{N}{n_{f'}}$ is introduced to increase impact of abundant functions in whole network since when a function occupies larger proportion of function distribution in the network; it has higher probability of being a function in unannotated protein than other functions.

2.1.2.2 Module assistant method

Semantic information is used to build edge weights and then clustering methods could be applied on weighted PPI network. For illustration SWEMODE, which is evolved from MCODE is introduced:

SWEMODE

Semantic Weights for Module Elucidation (SWEMODE) algorithm uses semantic similarity between GO terms to calculate similarity of proteins that are annotated by these terms (Zelmina et al., 2006). The definition of semantic similarity between two GO terms $t_1$ and $t_2$ is consistent with definition of Lin’s. (Lin, 1998) as following:

$$Sim(t_i, t_j) = \frac{2 \ln P_{ms}(t_i, t_j)}{\ln P(t_i) + \ln P(t_j)}$$ (2.1.2.2.1)

where $P$ is the probability of a term being sampled. It is calculated by counting the number of times the term or any of its descendants occur in
SGD database annotations, divided by the total number of GO term annotations in SGD. \( P_{ms}(t_i, t_j) \) is the probability of the minimum subsume of \( t_i \) and \( t_j \), which is defined as lowest probability found among the parent terms shared by \( t_i \) and \( t_j \).

Then similarity of two proteins is calculated as the mean value of semantic similarity between GO annotation terms from each protein:

\[
Psim(P_1, P_2) = \frac{1}{m \times n} \sum_{t_i \in P_1, t_j \in P_2} Sim(t_i, t_j)
\]  

SWEMODE contains several steps. At the beginning, each node \( i \) is assigned weight. There are two types of weight strategies based on weighted cluster coefficient. First one is named \( dens(\text{cc}^w) \). It is product of each node’s weighted clustering coefficient with the connectivity of the node. The second one is called \( core(\text{cc}^w) \), which is the product of the weighted core-clustering coefficient and the connectivity of the node in the immediate neighbourhood of \( i \) that has the highest weighted clustering coefficient.

Following, SWEMODE chooses a seed that has the highest weight, and adds neighbourhood whose weights no less than the node weight percentage (NWP). Then a new seed (node with the second highest weight) and repeat this process iteratively. It is similar to MCODE. After all clusters have been formed, function of proteins can be elucidated by other known proteins in the same cluster (module).

### 2.1.3 Algorithms based on multiple genomic data sources

Genomic data are diverse and have connections with each other. As protein interaction
information in PPI network is just one representative data source among all kinds of biological data. Such an approach for function prediction was first introduced by Marcotte et al. \textit{(Marcotte et al., 1999)} in a different context \textit{(Roded et al., 2006)}. Currently, a variety of protein function prediction methods attempt to combine multiple data sources with PPI for predicting functions. Three art-of-the-state algorithms are introduced in this section. First Joint Clustering Algorithms integrates protein sequence information and protein interaction information, while second algorithm, MS-KNN, combines sequence, gene expression and protein interaction data. The third algorithm, DCS/DSCP Algorithm, uses protein domain, protein complex and protein interaction information mutually. Meanwhile, Joint Clustering Algorithm and MS-KNN use module assistant method (cluster based) for mining data and DCS/DSCP uses direct method to predict function.

**2.1.3.1 Direct method**

Combining multiple genomic data is a trend for protein function prediction \textit{(Chua et al., 2012)}. Direct methods based on multiple genomic data use related data to give out more credible interactions in PPI networks, and apply traditional direct methods on polished PPI network. Here are three representative approaches:

**GeneFAS**

Joshi et al. presented an approach that integrated protein complexes, genetic interactions, and microarray gene expression profiles as data sources for validating relationships between proteins in PPI network \textit{(Joshi et al., 2004)}. Meanwhile GO terms in the biological process namespace are used to generate GO INDEX, which is a hierarchical index of terms (Figure 2.1.3.1.1).
Based on GO INDEX, the a-priori probability is calculated for each type of high-throughput data mentioned above to estimate whether two proteins are sharing similar functions. The a-priori probability is then used to select functions for unannotated protein. A-priori probabilities are computed by comparing the pairs of genes in high-throughput data, where both the genes have annotated functions, and by simultaneously comparing the level of similarity in functions that the two genes share in terms of the GO INDEX.

The a-priori probability for the predicted protein sharing a function with one of its interacting partners are $P_1$, $P_2$, $P_3$, $P_4$ respectively, which are described below. They are assumed as independent probabilities.

- $P_1$: a-priori probability from genetic interactions
- $P_2$: a-priori probability from physical interactions
- $P_3$: a-priori probability from complex interactions
- $P_4$: a-priori probability from microarray gene expression

Then the score for a function $f$ is in unannotated protein $p$ is:

$$Score(p, f) = 1 - e^{\log(1-P_1)(1-P_2)(1-P_3)(1-P_4)}$$ (2.1.3.1.1)

Functions with higher score will be assigned to unannotated protein.
DSC/DSCP Algorithm

Two approaches that combine proteins’ domain information and protein complex information mutually to define similarity between proteins was proposed by Peng et al in 2014. Domains are some compactly structured components of a protein that can evolve, function, and exist independently of the rest of the protein chain. The vast varieties of protein functions can be derived from the different combinations and cooperation of protein domains (Bashton et al., 2007). And protein complex usually refers to group of proteins that show similar functions. First protein similarity definition in this work is domain combination similarity (DCS), which combines the domain compositions of both proteins and their neighbours. Given a PPI network, let $SN = \{p_1, p_2, \ldots, p_n\}$ represent a set of all n proteins in the PPI network and NP denotes a set of neighbor proteins of protein p with p itself included. Let $DT(p)$ denote a set of domain types in protein p. Given a protein set $S = \{p_s1, p_s2, \ldots, p_sn\}$:

$$DT(p) = \bigcup_{i=s_1, s_2, \ldots, s_l} DT(p_i) \quad (2.1.3.1.2)$$

Let $DC(p)$ be a set of distinct domain types in the neighbor proteins of p:

$$DC(p) = DT(N_p) \quad (2.1.3.1.3)$$

Let M denote the number of domain types in the whole network, and let a and b represent the number of domain types in the neighbours of proteins $p_a$ and $p_b$, respectively. Let s denote the number of common domain types in their neighbours. Then there are equations:

$$M = |DT(SN)| \quad (2.1.3.1.4)$$
The domain context similarity \( f_{\text{cont}} \) can be defined as:

\[
f_{\text{cont}}(p_a, p_b) = -\log \left( \frac{\binom{M}{s} \binom{M-s}{a-s} \binom{M-a}{b-s}}{\binom{M}{a} \binom{M}{b}} \right)
\]

(2.1.3.1.8)

in which \( \binom{M}{s} \) denotes combination number.

Another is domain combination similarity in context of protein complexes (DSCP), which extends the protein functional similarity definition of DCS by combining the domain compositions of both proteins and the complexes including them. Similarly, it could be represented as:

\[
f_{\text{comp}}(p_a, p_b) = -\log \left( \frac{\binom{M}{s} \binom{M-s}{a-s} \binom{M-a}{b-s}}{\binom{M}{a} \binom{M}{b}} \right)
\]

(2.1.3.1.9)

where \( a' \) and \( b' \) denote the number of domain types inside protein \( p_a \) and protein \( p_b \) respectively, and \( s' \) denotes the number of common domain types of \( p_a \) and \( p_b \). Finally, similarity between \( p_a \) and \( p_b \) is computed as:

\[
f_{\text{sim}}(p_a, p_b) = \alpha f_{\text{cont}}(p_a, p_b) + (1 - \alpha) f_{\text{comp}}(p_a, p_b)
\]

(2.1.3.1.10)

where \( \alpha \) is a weight coefficient. With this definition, functions of unannotated protein are assigned as protein with which has the highest similarity (Wei et al., 2014).
2.1.3.2 Module assistant method

Multiple genomic data based algorithms mainly consist of multiple data sources based clustering algorithms and Joint network based approach. Following are some widely used method and state-of-the-art methods.

**SAMBA**

Statistical-Algorithmic Method for Bi-cluster Analysis (SAMBA) was firstly applied on gene expression data clustering (Tanay *et al.*, 2002), SAMBA models the genomic data as a bi-partite graph, where nodes on one side represent genes and that on the other side represent different properties derived from the genomic data. A framework is developed by Tanay et. al. to link yeast data consisting of gene expression, protein interactions, growth
phenotype data, and transcription factor binding. SAMBA was performed to reveal the modular organization of the yeast system (Tanay, et. al., 2004). All genomic information is mapped as a weighted bi-partite graph G. Nodes on one side of G represent genes, and nodes on the other side represent properties of genes or proteins encoded by them (Figure 3.3.2). An edge with weight w between a property node v and a gene node g represents an assertion that gene g has property v with probability proportional to w. For protein interaction data, a property represents the presence of an interaction with a specific protein. A gene expression experiment is represented by several properties corresponding to different expression level ranges. A TF property represents the binding of the gene promoter by the TF. SAMBA is used to search statistically significant sub-graphs (heavy sub-graph) in bi-partite graph. Statistically significant refers to high value edge weight sum in sub-graph (such as bold lines in figure 3.3.2). Thus genes/proteins in one heavy sub-graph turn to have same functions.

**Joint Clustering Algorithm (JCA)**

Recently, based on the Markov random walk, Wang et al., proposed an approach that uses PPI network and homology information between proteins across two distinctive networks to predict protein function (Yijie et al., 2014). Two PPI networks were merged together to a new network for random walk. More than just put two networks together, Wang et al. use BLAST computing sequence similarity between each protein across two PPI network, and add all-to-all edges between proteins across network based on their sequence similarity. Thus the edges in merged PPI network could be divided to two types, which are internal edges within original network and external edges among vertexes from different networks. Let $I_1$ and $I_2$ be adjacent matrixes of each original PPI network, network No.1 and network No.2,
respectively. After integrating two PPI networks, adjacent matrix of the internal edges in new network can be written as:

\[ I = \begin{bmatrix} I_1 & 0 \\ 0 & I_2 \end{bmatrix} \]  \hspace{1cm} (2.1.3.2.1)

Similarly, matrix of the external edges built based on sequence similarity between proteins from two different networks in new network can be written as:

\[ E = \begin{bmatrix} 0 & E_{12} \\ E_{12}^T & 0 \end{bmatrix} \]  \hspace{1cm} (2.1.3.2.2)

As \( E_{12} \) is the adjacent matrix contains external edges linked proteins from network No.1 and network No.2. Next, a novel Markov random walk strategy is applied on new network. Random walk is conducted on internal edges and external edges in an alternative order. And it could start from passing an internal edge then through an external edge and repeat in the order (Order A), or, in versa order (Order B) (illustrated in figure 2.1.3.2.2).

![Figure 2.1.3.2.2. Illustrates two strategies for random walk (from original paper).](image)

Blue nodes and red nodes stand for proteins in two different PPI networks.
Dashed lines are external edges while solid lines are internal edges.

Transition matrix of each order A and B, $P_A$ and $P_B$, could then be defined based on adjacent matrix. The final transfer matrix $P$ is a linear combination of $P_A$ and $P_B$, and it is believed that probability of taking each kind of order are the same, thus $P$ should be:

$$ P = \frac{1}{2} P_A + \frac{1}{2} P_B $$

(2.1.3.2.3)

Thus Markov random walk clustering could be down on joint network. It could be found that this method actually reassigns possibility of transition between vertexes by adding another path between each two vertexes in addition to protein-protein interaction. In this paper, both networks from same species and networks from different species are merged into one for test. Prediction results predicted from the joint network outperform results that predicted from single PPI network. Similar approach based on collective classification on joint networks had also been published by Wei et al. (Wei et al., 2012).

**MS-KNN Algorithm**

Liang et al. proposed a multi-sources k-nearest neighbour (k-NN) algorithm which using protein sequence, PPI and gene expression data for measuring protein similarity. The k-nearest neighbour classifier is a basic machine learning classification algorithm. It finds K “nearest” neighbours of the data waiting to be classified, and then classify the data according to these k neighbours. The distance between data could be measured in different ways according to specific application areas. In Liang’s paper, the distance is measured by protein similarity. Thus classifying process is turned into
computing score of functions in k most similar (nearest) proteins, which is:

\[
Score(p, f) = \sum_{p' \in Nkp} Sim(p, p') \delta(p', f) \quad (2.1.3.2.4)
\]

\(Nkp\) represents the k-nearest neighbour of protein \(p\). \(Sim(p, p')\) is similarity between \(p\) and \(p'\). The similarity value is comprised with three parts, sequence similarity, PPI data similarity and similarity on expression data of genes that code corresponding proteins, corresponds to the following formula:

\[
Sim(p, p') = \frac{1}{3} Simseq(p, p') + \frac{1}{3} Simppi(p, p') + \frac{1}{3} \sum_{j=1}^{l} Simexp(p, p') \quad (2.1.3.2.5)
\]

in which \(Simseq(p, p')\) is protein sequence similarity, the similarity score was calculated as sequence percent identity divided by 100. \(Simppi(p, p')\) is similarity calculated by PPI data, the similarity was set to 1 if two proteins interacted and 0 otherwise. And \(\sum_{j=1}^{l} Simexp(p, p')\) is microarray data similarity, \(j\) is the \(j\)-th microarray data while there are totally \(l\) microarray data. Pearson correlation coefficient, a popular method for measuring the similarity between gene expressions, was used as the similarity score between two proteins (Liang et al., 2013).

It could be concluded from above algorithms that the advantages of employing multiple data sources into protein function prediction in addition to using PPI data could improve interactions credibility and low down network noise in certain extent. Proper manners for combining heterogeneous biological data is crucial for algorithms mentioned in this section. Biological thoughts are critical for handling the issue about combining data.
2.2 Applications of dynamic PPIN based approaches

This section introduced the common method of DPPIN construction and established applications of DPPIN. We have observed the absence of DPPIN based novel protein function prediction method. Hence, this section reviewed related applications of DPPIN to give a sight about foundations of DPPIN and reasoning of DPPIN based methods. Section 2.2.1 describes widely adopted DDPIN construction procedure. Section 2.2.2 and Section 2.2.3 review some leading researches about DPPIN.

2.2.1 DPPIN construction

Typical DPPIN construction methods are based on static PPI data and time-course gene expression data. The main difference of these methods is the determination criterion of actively expressing genes at each time interval. Expression threshold has been defined in various forms as the determination criteria for active genes. Thus follows are the review of widely applied DPPIN construction methods.

PPI network/dynamic gene co-expression network integration

Nan et al. (Nan et al., 2012) combined static PPIN and gene co-expression networks to construct DPPINs. A series of gene co-expression networks \( G_1, G_2, \ldots, G_T \) were built, where \( T \) is the \( T^{th} \) gene expression array from time-course expression micro-array data. The weights of edges in the gene co-expression networks were computed using the absolute Pearson correlation coefficient (PCC) of genes’ expression profiles over time. Then the networks were represented by adjacent matrices (i.e. correlation matrix). There was a cutoff threshold used to remove edges that have low PCC values. The cutoff threshold was based on the average correlation similarity from
The constructed gene co-expression networks and static PPIN were used to validate existences of interactions in each DPPIN. Specifically, gene co-expression networks integrated with the PPIN by the rule that if one interaction exists at both the PPIN and the \( G_i \), this interaction would be added to the \( D_i \). Otherwise, there is no interaction between the protein pair at the \( i \)th timestamp.

**Static expression threshold based construction**

Active threshold was firstly used to filter genes for DPPIN construction (Xiwei et al, 2011). 0.7 was chosen as the threshold to filter the gene expression profiles after the analysis on the number of selected periodic genes and the number of biologically significant functional modules predicted from the DPPINs under different threshold values.

**Active expression threshold based construction**

Active threshold was proposed to eliminate the bias of gene expression intensity (Jianxin et al, 2013). The definition of active threshold is:

\[
AcScore(p) = thr_1(p) \times F(p) + thr_2 \times (1 - F(p)) \quad (2.2.1.1)
\]

where:

\[
F(p) = \frac{1}{(1 + \sigma(p))} \quad (2.2.1.2)
\]
Here \( \text{thr}_1(p) \) is the mean of the gene expression of protein \( p \), which is also denoted as \( \mu(p) \), \( \text{thr}_2(p) = \mu(p) \times \sigma(p) \), where \( \sigma(p) \) is the standard deviation of the gene expression of protein \( p \).

### 2.2.2 Protein complex detection

DPPIN has been applied on searching protein complex. Two surveys about DPPIN based protein complex identification had been published (Sriganesh et al., 2012, Bolin et al., 2013). Proteins in a complex tend to function as a unity. From the perspective of PPIN, proteins in a complex should have stable interactions. Thus the DPPINs extracted from different times provide information about the status of protein interactions over time. In another word, stably interacting proteins tend to be in a complex. In this section, we review adopted DPPIN based applications on protein complex detection.

**Network hub analysis**

Han et al. proposed the concepts of date hub and party hub based on their research on yeast gene expression under different times (Han et al., 2004). Protein interactions are extracted based on the intensity of yeast messenger RNA expression profiling data from different cell cycle stages. Hubs in PPIN are proteins that have high topological degree and perform important functions over cell cycles. The date hub and party hub compose two types of hub in the PPIN.
Date hubs are proteins that have same neighbours at different times while party hubs are proteins that have different interacting partners at different times (figure 2.2.2.1). As the proposed conclusion, date hubs organise the proteome, connecting biological processes or modules to each other, whereas party hubs function inside modules.

**DPPIN and MCL-Caw based protein complex analysis**

Sriganesh et al. incorporated MCL-Caw method (Srihari et al., 2010) and DPPINs to detect protein complex. MCL-Caw clusters the PPI network solely on topological information to identify dense sub networks, which outputs its predicted complexes. MCL-Caw consists of six steps which are: 1). Clustering the PPI network using MCL hierarchically. 2). Categorizing proteins as cores within clusters. 3). Filtering noisy clusters. 4). Recruiting proteins as
attachments into clusters. 5). Extracting out complexes from clusters. 6).
Ranking the predicted complexes.

Then in the next step, cell cycle phase (G1, S, G2, M) were assigned to each protein based on the phase in which it showed peak expression. A protein was labelled as “dynamic” if it expresses only in one phase. Else if it expressed maximum in more than one phase, it was labelled ‘static’. Proteins which were unique for complexes (i.e. “cores”) and proteins shared by different complexes (i.e. “attachments”) were compared. The research then examined the hypothesis that constitutive expression should be more enriched in attachments compared to cores in complexes. Let $\lambda_s(\chi)$ denote the number of static proteins in set $\chi$, and $\lambda_d(\chi)$ denote the number of dynamic proteins in $\chi$. The enrichment $E$ for static (dynamic) proteins among attachments and cores in the set of complexes $\mathcal{C}$ are defined as follows. For a complex $\mathcal{C} \in \mathcal{C}$ the enrichment in the attachments, $Attach(\mathcal{C})$ is:

$$E_s(Attach(\mathcal{C})) = \frac{\lambda_s(Attach(\mathcal{C}))}{|\lambda_s(\mathcal{C})|}$$  \hspace{1cm} (2.2.1.1)$$

$$E_d(Attach(\mathcal{C})) = \frac{\lambda_d(Attach(\mathcal{C}))}{|\lambda_d(\mathcal{C})|}$$  \hspace{1cm} (2.2.2.2)

The relative enrichment $RE(Attach(\mathcal{C}))$ of static to dynamic proteins in the attachments in $\mathcal{C}$ is:

$$RE(Attach(\mathcal{C})) = \frac{E_s(Attach(\mathcal{C}))}{E_d(Attach(\mathcal{C}))}$$  \hspace{1cm} (2.2.3.3)

The enrichment and relative enrichment for cores $RE(Core(\mathcal{C}))$ is defined in a similar way. As the statistical results of $RE(Attach(\mathcal{C}))$ and
calculation showed, the constitutive expression is more frequent in proteins that shared by complexes.

2.2.3 Critical protein identification

msiDBN

msiDBN \((Yuan \ et\ al., \ 2014)\) is a deep belief network based method for critical protein identification. msiDBN utilised interaction information of DPPINs for the purpose of deep belief network training. DPPINs in this research were constructed based on active expression threshold. Then for each pair of interacting proteins at time \(t\), the Pearson correlation coefficient of their expression value at \(t - 1\), \(t\) and \(t + 1\) was computed as the weight of the interaction. The definition of critical protein is based on the intuition that most proteins have similar behaviour patterns across the time courses while the most critical proteins that are responsible for the progression of the yeast cell cycle exhibit different expression levels and more importantly they engage in different interactions with contemporary neighbours.

Next, multisource Integrated Deep Belief Network was employed to identify critical proteins. Restricted Boltzmann machines (RBMs). The RBM defines an energy function between the visible and hidden layer variables:

\[
E(V, h) = h^T W_v + d^T h + b^T v \tag{2.2.3.1}
\]

where \(h\) and \(v\) are row vectors in \(H\) and \(V\), respectively. \(b\) and \(d\) are the bias to the visible layer and hidden layer, and \(W\) is the weights between two layers. Then the distribution of RBM with a normalization factor \(Z\) is
\[ P(V, h) = \frac{1}{Z} E(V, h) \]  \hspace{1cm} (2.2.3.2)

The conditional distributions of hidden layer variables are then sampled and computed based on the following definitions:

\[ P(h_{2}^{(t)} | h) = \text{sigmoid}(c + \sum h_{1}W_{3i}^{(t)}) \]  \hspace{1cm} (2.2.3.3)

\[ P(h|h_{2}^{(1)}, h_{2}^{(2)}, ..., h_{2}^{(t)}) = \text{sigmoid}(d_{t} + \sum \sum h_{2i}^{(t)}W_{3i}^{(t)}) \]  \hspace{1cm} (2.2.3.4)

msiDBN repeatedly samples and computes based on equations above until parameters have reached convergence. The relative standard error of the network reconstruction error (root mean square error) was used to judge which proteins are critical proteins.

### 2.3 PPIN datasets used in this research

The PPI dataset of Saccharomyces cerevisiae (Yeast) were used in this research. The depositary of PPI data used in this research were DIP ([Salwinski et al., 2004. http://dip.mbi.ucla.edu/dip]), MIPS ([Mewes et al., 2002. ftp://ftp.mips.gsf.de/fungi/yeast/PPI]) and BioGrid data from the Saccharomyces Genome Database (SGD, Cherry et al., 2012 http://www.yeastgenome.org/). DIP and MIPS datasets are manually curated and only contain experimentally determined physical and genetic interactions. Thus DIP and MIPS datasets can be used as golden standards during predicted function examination. BioGRID dataset contains interactions form experiments and computational predictions. Thus BioGRID data has higher false positive rates that DIP and MIPS data. We validated our methods on these datasets with different qualities to evaluate the robustness. PPI data downloaded from databases were pre-processed to eliminate duplicated interactions and self-loop interactions.
2.4 Validation and Evaluation

This part introduces validation methods of prediction results. At first, evaluation metrics of prediction results are described in Section 2.4.1. Secondly, Section 2.4.2 presents a series of widely adopted validation methods in the research area.

2.4.1 Evaluation Metrics

2.4.1.2 Statistical Evaluation Metrics

There are several metrics being chosen as criteria during evaluating algorithm performance. First of all, precision is the standard for measuring accuracy of algorithms. Precision states the proportion of correctly predicted functions in whole set of predicted functions. The parameter Recall (i.e. sensitivity) measures the capability of discovering functions. Recall is compute as proportion of revealed functions from true functions of a protein after prediction, while F-value is a parameter assessing performance of algorithm from both precision and recall mutually. Let \( N_t \) be the total number of true functions (generated from laboratory) in protein \( P \), and given the prediction result which contains \( N \) functions, in which \( N_c \) is the number of correctly predicted functions. Then Precision, Recall and F-value are defined as:

\[
\text{Precision} = \frac{N_c}{N} \quad (2.4.1.2.1)
\]

\[
\text{Recall} = \frac{N_c}{N_t} \quad (2.4.1.2.2)
\]

\[
\text{F-Value} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (2.4.1.2.3)
\]
Statistical evaluation metrics such as misclassification rate and non-classification rate are widely adopted in machine learning area. The misclassification rate is divided into false positive rate and false negative rate. False positive (FP) rate states the ratio of incorrectly classified samples in total samples, and false negative rate stands for the ratio of samples that are not classified due to the limitations of classifiers. Complementarily, there are also true positive (TP) rate and true negative (TN) rate.

Given a set of predicted functions $F$, TP measures how many predicted functions are the true functions in $F$. FP measures number of wrong predicted result in $F$. FN stands for number of missed predicted functions from true functions of the protein. Their relationships with precision and recall are:

$$
\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{2.4.1.2.4}
$$

$$
\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{2.4.1.2.5}
$$

2.4.1.3 Multi-label learning algorithm metrics

Additionally, commonly used criteria for multi-label learning algorithm (such as collective classification) evaluation are: average precision, coverage, and one-error (Schapire et al., 2000). Given a multi-label dataset $D = \{(x_i, y_i) | 1 \leq i \leq m\}$, where $x_i \in X$ is a predicted function label and $y_i \subseteq Y$ are the true function labels of $x_i$, the evaluation measures are defined using the following two outputs provided by multi-label learnings: $s(x_i, l)$ returns a real-value that indicates the confidence for the class label $l$ to be a proper label of $x_i$; $\text{rank}(x_i, l)$ returns the ranks of class label $l$ derived from $s(x_i, l)$. Coverage evaluates number of functions that counted from top of the list of labels (functions) until cover all the true annotation of a protein:
Coverage(f) = \frac{1}{m} \sum_{i=1}^{m} \max \text{ rank}(x_i, l) - 1 \quad (2.4.1.3.1)

One-error evaluates how many times the top ranked function label is not a true function of the protein. Let \( H(x_i) \) be the top ranked function label of \( x_i \) being predicted:

\[
\text{One-error}(H) = \frac{1}{m} \sum_{i=1}^{m} \mathbb{1}[H(x_i) \not\in y_i] 
\]

(2.4.1.3.2)

Average precision evaluates the average fraction of function labels ranked above a particular true function label in rank list.

\[
\text{Avgprecision}(f) = \frac{1}{m} \sum_{i=1}^{m} \frac{1}{y_i} \sum_{l \in y_i} \frac{|P_l|}{\text{rank}(x_i, l)} 
\]

(2.4.1.3.3)

\[
P_l = \{ l' \in y_i | \text{rank}(x_i, l') \leq \text{rank}(x_i, l) \} 
\]

(2.4.1.3.4)

The smaller the value of coverage ranking loss and one-error are, the better the performance is. As for average precision, a higher value states a better performance.

### 2.4.2 Validation Methods

The Leave One-Out Cross-Validation (LOOCV) technique is widely applied in a lot of experiments aiming at testing efficiency and accuracy of prediction. Leave One-Out Cross-Validation works in following way: an annotated protein is viewed as unannotated protein, and is predicted by prediction algorithm that needs to be evaluated. Then we compare the predicted functions with true functions of the protein. Repeat this procedure on each annotated protein in the dataset, and then the performance of prediction methods could be statistically studied.
2.5 Summary

Plenty of algorithms and approaches have been applied on biological data such as protein sequences, protein-protein interaction and gene expression generated from experiments in laboratory, aiming at using computational way to find out functions of unannotated proteins. This chapter provides a detailed literature review on representative PPIN based protein function prediction approaches, methods and algorithms. Both methods based on static PPIN and methods based on Dynamic PPIN are introduced.

It is becoming prevailing to integrate data that have biological connections in protein function prediction work. Mainly, data about protein/gene sequence, protein structure, and protein complex and gene expression are incorporated with PPI network for analysis of novel protein functions. On the contrary, PPI information is helpful in revealing gene function since protein is coded by particular gene ORF, interaction among proteins implies latent association between corresponding genes.

For future research, one of the future trends of protein function prediction would be focusing on hierarchical predicting. Conventional annotation formats like Gene Ontology (GO) annotate protein functions hierarchically, therefore give out an accurate prediction about which category of functions should a protein has or an accurate ancestor term than a vague descendant term is more useful. Biological experiments for discovering novel function of unknown proteins could apply computational approaches to locate similar proteins and design experiments in a more targeted way. Furthermore, PPI network is also a useful tool for research in gene expression or connection between diseases. Several attempts have been made to using PPI network in discovering novel breast cancer-related MiRNAs (Chia-Hsien et al., 2013), classifying subtype of cancer (Ashish, et al., 2014) and pre-defining disease associations (Sun et al., 2014).

Main limit of using PPI network to predict protein functions is that PPI is a relatively
abstract network tool, which usually fails to consider relationships among protein in specific biological context. Furthermore, there are several false positive interactions in PPI dataset while some false negative interactions are not included in the PPI dataset, which brings noise data into PPI network. How to eliminate effects from noise data is another lethal problem for predicting protein functions with PPI network. Assigning credibility to interactions by incorporating relevant biological data from multiple sources is a possible way to tackle this problem.

Dynamic PPINs based algorithms are less studied than static PPIN based algorithms. The reasons are concluded as:

- Some static PPIN based algorithms can be applied on dense DPPINs. Since DPPINs are a series of static networks, developed protein function prediction algorithms are compatible with DPPINs.

- The construction of DPPIN largely relies on the quality of heterogeneous data. The combining of different biological data could further increase the chance of introducing noisy data.

- There are no corresponding annotation schemes that provide structured descriptions about the stage/time that a protein performs certain functions.

In despite of the above issues, it is a major trend in the future that function annotation scheme of genes and proteins shall contain more specific conditions which provide information about the biological environment and the prerequisite conditions for a gene or protein to perform its functions.
Chapter 3

Iterative protein function prediction method

In this chapter, we propose an iterative approach that utilises unannotated proteins and their interactions in prediction. We conducted experiments to evaluate the performance and robustness of the proposed iterative approach. The iterative approach maximally improved the prediction performance by 50%-80% when there was a high proportion of unannotated neighbourhood protein in the network. The iterative approach also showed robustness in various types of protein interaction network. Importantly, our iterative approach initially proposes an idea that iteratively incorporates the interaction information of unannotated proteins into the protein function prediction and can be applied on existing prediction algorithms to improve prediction performance.

3.1 Introduction

Static PPI based protein function prediction were categorised as direct methods and module assistant methods (Roded et al., 2006). For direct methods, at early stage, Neighbour Counting (NC) algorithm (Schwikowski et al. 2000) and Chi-square algorithm (Hishigaki et al. 2001) were proposed. These two simple unsupervised methods rely on statistical strategy to select the most frequent functions around unannotated proteins. Subsequently, Function Similarity Weighted (FSW) algorithm (Chua et al. 2006) which utilises network topology to assign weights to protein interactions was proposed. The Markov random field (MRF) approach proposed by Deng et al. (Deng et al. 2003) exploits function distributions in the PPIN to estimate parameters for function classification. Other machine learning algorithms, such as collective classification algorithm (Wu et al. 2014; Xiong et al. 2014), apply existing clustering methods on PPINs to cluster proteins, and then protein function classifiers were derived based on function information in the
clusters. There was also graphic theory based algorithms (Vazquez et al. 2003; Karaoz et al. 2004) and Function Flow method (Nebieva et al. 2005) which propagates annotations in PPINs to assign functions to unannotated proteins. As a trend of integrating multiple data sources for prediction, algorithms that incorporate semantic information of function annotations to predict protein functions were also extensively studied, such as the algorithms CIA (Chi et al. 2011) and DPA (Hou et al., 2013). Furthermore, DCS/DCSP (Peng et al. 2014) integrates protein complexes and protein domain information along with PPIN to compute similarity between proteins, and then assign functions to unannotated proteins based on protein similarity.

In module-assisted methods, discovering module is usually conducted by clustering similar proteins into a group. Then unannotated proteins are annotated with the functions prevailing in the modules (i.e. groups) to which they belong. In a recent review (Pizzuti et al. 2014), graph-based clustering approaches were broadly categorized into five types: 1) local neighbourhood density search, such as MCODE (Bader et al. 2003), DPClus (Altaf-Ul-Amin et al. 2006), and CFINDER (Adamcsek et al. 2006). Approaches in this category cluster nodes in PPINs based on density areas in PPINs; 2) cost-based local search, such as RNSC (King et al. 2004) and OCG (Becker et al. 2012), which performs local search to obtain optimum clustering in an effective way; 3) Flow simulation, such as MCL (Enright et al. 2002). It employs random walk strategy on PPIN to build a Markov chain of transition matrixes, in order to discover clusters from transition matrix once it reaches convergence; 4) Link clustering, such as PEREIRA (Pereira et al. 2004) and AHN (Ahn et al., 2010). These methods cluster edges instead of nodes of the network; 5) Population-based stochastic search, e.g. GA-PPI (Pizzuti et al. 2013), which uses genetic algorithm to generate clusters. Recently, some classic clustering methods were applied on integrated networks to discover functional modules in PPINs, for instance, SAMBA (Tanay et al. 2002), MS-KNN (Liang et al. 2013) and JCA (Yijie et al. 2014). These algorithms take advantages of associations among different types of biological data. Protein sequence, protein domain information and gene expression information are combined with PPINs to
construct comprehensive networks in order to get better clustering performance.

Methods for integrating heterogeneous biological data were also studied, e.g. IWA (Chua et al. 2007), to optimise the strategy for combining multiple data sources. Data sources are assigned weights based on their reliabilities in function, and then networks built by each type of data are combined together as a weighed network that serves as a tool for prediction.

### 3.2 Iterative model

Our iterative model is introduced in this section. Before prediction, we firstly need to find out unannotated proteins in the prediction domain. Then for each unannotated protein, its function will be initialised respectively by a prediction algorithm (e.g. Neighbour counting (NC) algorithm or Function similarity weighted (FSW) algorithm in our case). After every unannotated protein in the prediction domain is initially annotated, predicted functions in the prediction domain are then iteratively updated until every unannotated protein gets its predicted functions that do not change anymore. The approach is divided into four steps: unannotated protein searching, protein function initialisation, iterative function updating, and prediction result selection, which are illustrated in Figure 3.2.1. The pseudo code of our iterative model is provided in the appendix.
3.2.1 Unannotated protein searching

In this first step, with a PPI network $G(V,E)$, and the GO annotation file, unannotated proteins in the prediction domain (e.g. in NC algorithm, it’s the directly interacting partners of protein to be predicted) around the main prediction target protein, $P_x$, are found out from the PPI network. Main prediction target $P_x$ and other unannotated proteins are all stored in set $\psi$ as the target proteins.

3.2.2 Protein function initialisation

After finding out unannotated target proteins in the prediction domain of the main prediction target, each unannotated target protein’s functions will be initialised. Suppose
there are totally $n$ proteins (including the main prediction target protein, $P_2$) in set $\psi$.
For the function initialisation of the $i$-th protein $P_i$ ($1 \leq i \leq n$) in $\psi$, other unannotated proteins, $\varphi$ ($\varphi = \psi - \{P_i\}$) and interactions associated with them are removed from the PPI network during the initialisation. Then, functions of the unannotated target protein $P_i$ are initially predicted using an existing protein function prediction algorithm. We then similarly initialise functions for all $n$ proteins in $\psi$ until functions of all $n$ proteins have been initially predicted. For an unannotated target protein which has no annotated neighbours, its function is initialised as null (unannotated). The function initialisation starts from a randomly selected protein in $\psi$ and in a random order in our approach. This initialisation is regarded as the first round of prediction (the 1st iteration).

\subsection{3.2.3 Iterative function updating}

After function initialisation, the target proteins with initialised non-null functions then join back into the prediction domain as annotated proteins at the beginning of each iteration. Therefore, functions of these proteins can be re-predicted and updated since the protein function distribution in the prediction domain is changed.

Details of the iterative function updating are as follows: 1) At the beginning of the $t$-th iteration ($t \in [1, M], t = 1$ for the protein function initialisation), a protein $P_i$ ($1 \leq i \leq n$) is randomly selected from the set $\psi$, then a protein function prediction algorithm is applied to re-predict its functions. 2) Repeat step 1) operation for all $P_i$ in set $\psi$ one by one, until functions of each $P_i$ have been re-predicted and updated. Then the current $t$-th round of prediction ends, and the $(t + 1)$-th prediction starts. 3) If re-predicted functions of $P_i$ after the $(t + 1)$-th round of prediction are different from predicted functions after $t$-th round of prediction, then its functions are updated as the new re-predicted functions, otherwise keep the functions of $P_i$ unchanged. 4) Repeat step 2) and 3) until functions in each protein reach a steady status after the $M$-th iteration (i.e. functions of every protein in $\psi$ do not change any more or repeat in a pattern), then the iterative function updating
ends. If the predicted functions of $P_t$ repeat in a certain pattern after the $M$-th round of prediction, and have a repeat span of $T$ (i.e. prediction results from the $(t + T)$-th prediction are the same as the predicted functions from the $t$-th prediction when $t \geq M$), then we randomly choose one group of functions from these $T$ groups of predicted functions as the prediction result of $P_t$. Meanwhile we set $M$ as the total number of iteration before reaching a stable status.

In each round of iteration, the target proteins for which the function prediction is conducted are selected in a random order. The robustness of the iterative function updating towards different orders of proteins being predicted is evaluated in the experiment section.

### 3.2.4 Prediction result selecting

After each round of iteration, the predicted functions of each target protein are recorded. When the iteration ends, for each protein, its final predicted functions are selected statistically from all its predicted functions generated from different rounds of iterations. Let the set of whole function annotations in the PPI network be $F$, and $F = \{f^1, f^2, f^3, \cdots, f^K\}$, where $f^1$ to $f^K$ are function annotations and $K$ is the number of function annotation terms in the PPI network. For an unannotated protein $P_x$ (the main prediction target), predicted functions after $t$-th step of iteration are recorded as $F_{t \ P_x} = [f_{x,t}^1, f_{x,t}^2, \cdots, f_{x,t}^K]^T$, $f_{x,t}^j = 1$ if the predicted functions of protein $P_x$ from the $t$-th iteration contains function $f^j$ ($f^j \in F$), otherwise $f_{x,t}^j = 0$. $F_{1 \ P_x}$ records the initial functions of $P_x$. When the iteration reaches the stable status after the $M$-th iteration, we can have a matrix $AF_{p \ x}$ that records predicted functions of $P_x$ generated from all steps of the iteration:
\[ AF_{P_x} = [F_1_{P_x}, F_2_{P_x}, F_3_{P_x}, \ldots, F_M_{P_x}] \]  

(3.2.4.1)

\( F_t_{P_x} \) is the vector containing predicted functions of protein \( P_x \) after the \( t \)-th iteration.

Then we have:

\[ C_{P_x} = AF_{P_x} \times I \]

(3.2.4.2)

\[ I = [1,1,\ldots,1]^T, \quad C_{P_x} = [C_{P_x}^{f_1}, C_{P_x}^{f_2}, C_{P_x}^{f_3}, \ldots, C_{P_x}^{f_K}]^T, \quad \text{and} \quad C_{P_x}^{f_j} = \sum_{t=1}^{M} f_j^t \quad (1 \leq j \leq K). \]

(3.2.4.3)

The final predicted functions of the unannotated protein \( P_x \) are selected as follows:

\[ \text{Predicted function} = \{ f^j | f^j \in F, C_{P_x}^{f_j} \geq \alpha \ast \max_{1 \leq j \leq K} (C_{P_x}^{f_j}) \} \quad \alpha \in [0,1] \]

(3.2.4.4)

In our experiment, we set the parameter \( \alpha = 0.5 \). Actually the predicted functions are selected based on the frequency of their occurrences in the whole iteration process. It can be seen that the one-off algorithms are the special case of this iterative model with the total number of iteration \( M = 1 \).

### 3.2.5 Function Prediction Algorithms

Here we present three PPI based function prediction algorithms that can be used as the base/reference prediction algorithms in our iterative prediction model. These three algorithms are neighbour counting (NC) algorithm, function similarity weighted (FSW) algorithm, and function flow algorithm. The neighbour counting (NC) algorithm was proposed by Schwikowski et al. in 2000. It predicts functions of each protein by counting
the frequency of each function being found in its neighbourhood proteins. The more frequent a function shows in the protein's level-1 neighbours (directly interacting proteins), the more likely it is a function of the protein. Let \( p \) stand for the unannotated protein to be predicted, \( N_p \) is the set of level-1 proteins of \( p \), \( p' \) is a protein in \( N_p \).

\[
\delta(p', f) = 1 \text{ if function } f \text{ is found in } p', \text{ otherwise } \delta(p', f) = 0.
\]

A function \( f \) is scored based on the formula below:

\[
Score(p, f) = \sum_{p' \in N_p} \delta(p', f)
\]  

A function with a higher score would be chosen as the prediction result. Originally, the NC algorithm chooses the top three functions as the functions of the unannotated protein \( p \).

The Function similarity weighted (FSW) algorithm defines a similarity between proteins based on the number of their commonly interacting proteins in a PPI network, then assigns weights to proteins based on their similarity with the prediction target. Thus a similarity among proteins that share more common neighbours should be relatively higher than others. Functions in those proteins that have higher similarities with the unannotated protein would get higher weights in function voting. Actually, let \( N_u \) and \( N_v \) denote level-1 neighbour protein sets of two proteins \( u \) and \( v \), the similarity of these two proteins, \( S(u, v) \), is given by:

\[
S(u, v) = \frac{2|N_u \cap N_v|}{|N_u - N_v| + 2|N_u \cap N_v|} + \frac{2|N_u \cap N_v|}{|N_v - N_u| + 2|N_u \cap N_v|}
\]  

If protein \( u \) is similar to protein \( w \), and protein \( w \) is similar to protein \( v \), then \( u \) and \( v \) have transitive functional association, and similarity between \( u \) and \( v \) could be depicted as a transitive FSW similarity:

\[
S_{TR}(u, v) = \max(S(u, v), \max_{w \in N_u} S(u, w)S(w, v))
\]
$S(u,v)$ is the FSW between $u$ and $v$. The score of a function $f$ in level-1 proteins and level-2 proteins of the unannotated prediction target $u$ is defined as:

$$Score(u,f) = \sum_{v \in N_u} [S_{TR}(u,v)\delta(v,f) + \sum_{w \in N_u} S(u,w)\delta(w,f)] \quad (3.2.5.4)$$

$\delta(v,f)$ and $\delta(w,f)$ are indicator functions with $\delta(p,f) = 1$ if protein $p$ has function $f$, otherwise $\delta(p,f) = 0$. The predicted functions are those with higher scores. The prediction domain of the NC algorithm is defined as the unannotated protein and its level-1 neighbour proteins, while for the FSW algorithm it is the unannotated protein and its level-1 and level-2 neighbour proteins.

Function flow (Nebieva et al. 2005) was used as another reference algorithm. Functions propagate in the network as flows. And annotated proteins are regarded as sources of functional flows (a reservoir). An iterative algorithm using discrete time is used to simulate the spread of functional flows in neighbourhood proteins. After $d$ iteration, the functional score of a protein corresponds to the total amount of functional flows that the protein has received during the iteration process.

For each protein $u$ in the network, a variable $R^f_t(u)$ is defined corresponding to the amount in reservoirs for function $f$ that node $u$ has at time $t$. At time 0, there is only reservoir of function $f$ at node $u$:

$$R^f_0(u) = \begin{cases} \infty, & \text{if } u \text{ is annotated with } f, \\ 0, & \text{Otherwise.} \end{cases} \quad (3.2.5.5)$$

At each time step, functional flows proceeding downhill from node $u$ to $v$ satisfy the capacity constraints:
In the formula, $w_{u,v}$ is the weight of edge $(u, v)$. $w_{u,v} = 1$ in un-weighted PPIN. And $E$ is the set of edges in network. Reservoir of each protein is recomputed as there are functional flows entering and leaving the nodes.

$$R^f_t(u) = R^f_{t-1}(u) + \sum_{v:(u,v)\in E}(g^f_t(v,u) - g^f_t(u,v))$$

Finally, the functional score for protein $u$ and function $f$ over $d$ iterations is calculated as the total amount of flow that has entered the node. Functions with the highest score are selected as the predicted functions:

$$Score(u, f) = \sum_{t=1}^{d} \sum_{v:(u,v)\in E} g^f_t(v,u)$$

### 3.3 Results and discussion

#### 3.3.1 Datasets

It is noticed that PPIN usually contains noisy information due to a lot of false positive interactions generated from high throughput technologies. In our experiments, evaluation of the algorithm performance on the networks with different qualities was conducted. Totally four datasets were employed for the experiments. We employed Function Category (FunCat) annotation (Andreas et al. 2004) in our experiments. There are hierarchical annotations of 6,167 proteins in the dataset. For the purposes of prediction performance evaluation, unannotated proteins in the original PPINs were removed.

The first dataset was the Munich Information Centre for Protein Sequences (MIPS) Yeast
PPI network from FTP server of MIPS (MIPS: ftp://ftparchive.ensembl.orgfungi/yeast/PPI/). After removing 250 unannotated proteins, the dataset contains 4,554 proteins and 15,456 interactions. The MIPS dataset comprises physical interaction data including interactions collected from small-scale experiments and some core data generated from high throughput technologies (Ito et al. 2000-2001), and it is believed to be highly reliable. After downloading the raw data, we filtered redundant and self-interactions in the network. For the purpose of evaluation, we removed unannotated proteins from the network. The MIPS dataset after pre-processing had 4,273 proteins and 12,735 interactions in it.

Another dataset was the BioGrid Yeast PPI network downloaded from Saccharomyces Genome Database (SGD: http://www.yeastgenome.org/). The BioGrid PPI network contains physical interactions and genetic interactions generated by high throughput technologies, thus the dataset contains a high proportion of false positive interactions. We filtered redundant and self-interactions in the network. And for the purpose of evaluation, we removed unannotated proteins from the network as well. To make its network scale comparable with the scale of the MIPS network, we randomly selected a subnetwork of BioGrid network. After filtering out 216 unannotated proteins, the final dataset contained 4,249 proteins and 10,000 interactions for our experiment.

The third dataset was the intersection of MIPS yeast network and BioGrid network (named as BioGrid-MIPS network). This dataset consists of 3,025 distinct annotated proteins and 8,405 interactions.

We also combined gene co-expression data of yeast from COXPRESdb (Obayashi et al., 2011. COXPRESdb: http://coxpresdb.jp/) into the original PPI networks to further test the performance of the iterative model on the cases where the original PPI network structure is modified due to the incorporation of related genomic information. Data from COXPRESdb contains 4,461 expression information files generated from 3,819 gene chips,
each file represents a gene and the corresponding co-expressed genes are listed as the file content. Co-expressed genes had already been processed and ranked from high to low based on the extent of co-expression. We constructed a gene co-expression network by adding an edge between a gene and its first ranked co-expressed gene. After a gene co-expression network was constructed, we modified the MIPS PPI network by adding edges between proteins that had the corresponding edges in the gene co-expression network. i.e., we merged the co-expression network with MIPS network to form a new dataset.

### 3.3.2 Performance Metrics

We measured the performance of prediction in terms of precision and recall. Since there is a trade-off between precision and recall, we also used F-value to measure the performance of the prediction results. Actually, let $N_t$ be the total number of true functions (generated from laboratory) in protein $P$, $N$ be the number of predicted functions and $N_c$ be the number of correctly predicted functions. Then Precision, Recall and F-value are defined as follows:

\[
\text{Precision} = \frac{N_c}{N}, \quad \text{Recall} = \frac{N_c}{N_t} \tag{3.3.2.1}
\]

\[
\text{F-Value} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{3.3.2.2}
\]

Usually, higher precision, recall and F-value indicate a better performance of an algorithm. Also we used precision-recall curves to compare precision at different levels of recall.

### 3.3.3 Prediction Performance

Firstly, we evaluated the performance of the iterative model on proteins that have certain
levels of unannotated neighbours. We compared the precision, the recall and the F-value of the iteration algorithms and the non-iteration algorithms (also known as reference algorithms) on four datasets of which are the MIPS PPI network, the BioGrid PPI network, the intersection network of MIPS and BioGrid PPI network, and the Merged network of MIPS PPI network and COXPRESdb gene co-expression network.

We used leave-one-out cross (LOOC) validation method in the evaluation. For each of the above four datasets, we randomly selected nine groups of proteins as main prediction targets (i.e. the proteins for which we supposed their functions were unknown, thus we could apply prediction algorithms on them). Each group contained 100 randomly selected main prediction target proteins. From group 1 to group 9, in each dataset, 10% to 90% of proteins in the prediction domain of each target protein were randomly set as unannotated proteins. Then we evaluated our iterative model and reference algorithms on these 36 protein groups (3,600 main prediction targets with different percentage of unannotated proteins in their prediction domain) respectively. The average precisions, the average recalls and the average F-values of each group were recorded for comparison.

In order to evaluate the performance of the proposed iterative model, we applied the iterative model on Neighbour Counting (INC) algorithm and then compared INC with original NC algorithm. Furthermore, we compared our iterative approach with the Function Similarity Weighted (FSW) algorithm and Function Flow (FF) algorithm which also utilise unannotated proteins and corresponding interactions in prediction processes. To achieve comparability among iterative and non-iterative approaches, we applied the idea from FSW to weigh protein pairs but excluded those un-annotated proteins from the neighbour in each step of the iterative neighbour counting method. Thus the topological structure of the neighbour and the predicted functions of the un-annotated protein were updated iteratively. We named this iterative model Iterative Weighted Neighbour Counting (IWNC). To make the results comparable, we used unified prediction domain (level 1 and level 2 neighbours around unannotated nodes) for IWNC, FSW and FF.
We compared INC algorithm and NC algorithm on constructed four datasets. Then we compared IWNC algorithm, FSW algorithm and FF method in a similar manner. Comparisons of prediction results are presented as follows:

At first, we compared results from different algorithms on the same dataset. INC was compared with NC, while IWNC was compared with FSW and Function Flow. The above experiment results showed that our iterative algorithms (i.e. INC and IWNC) outperformed the reference algorithms in terms of Precision, Recall and F-value in most cases. Especially, when the percentage of unannotated proteins in the prediction domain
increased, precision, recall and F-value of NC algorithms showed rapid decrease, while the iterative model INC kept relatively remarkable prediction performance. Comparison between IWNC and FSW shows that IWNC generated better prediction results in most cases, even the unannotated proteins did not participate in computing of protein similarities during the prediction process. The similar conclusion could be drawn from the comparison between IWNC algorithm and Function Flow method. Furthermore, the iterative approach kept robust F-values to the variation of unannotated protein numbers in the prediction domain and the gene co-expression data information.

Secondly, we compared results on different datasets. It could be observed that the algorithms generated better prediction results from the dataset that was the intersection of the MIPS and BioGrid datasets. That is because both MIPS and BioGrid datasets endorse the protein interactions in this dataset, they therefore have a higher credibility. Correspondingly, the PPINs intersection dataset has less noisy information, which increased the prediction performance. From the results generated from BioGrid dataset, we observed that IWNC, FSW and Function Flow method did not perform as good as they did on other datasets. This was possibly caused by the quality of selected BioGrid datasets. Since we sparsified the dataset to make its scale smaller than the original BioGrid network, it had a greater possibility that the noisy information in the dataset could affect the prediction results. However, results from MIPS datasets, intersection datasets of MIPS and BioGrid and MIPS data contain gene expression information showed that the IWNC algorithm performed better than the FS-Weighted algorithm and the Function Flow method in most cases.

After evaluating the iterative model performance and robustness toward the increasing percentages of unannotated proteins in the prediction domain, we then evaluated the overall performance of the iterative model on the prediction domains with random proportions of unannotated proteins, which simulates the situation in real world PPINs. We chose dataset A: 10 groups of proteins from the MIPS dataset, and dataset B: 10
groups of proteins from the BioGrid dataset. Each group contains 50 proteins that have random proportions of unannotated neighbours. Then we tested performance of INC and NC algorithm, IWNC, FSW and FF algorithm on dataset A and B.

For dataset A, we recorded the precision, recall and F-value of each predicted protein \( P_i \) as \( Precision_i \) and \( Recall_i \) and \( F\text{-value}_i \). We then calculated average precision and average recall values of each group of proteins as:

\[
\text{Avg}(\text{Precision})_g^A = \frac{\sum_{i=1}^{50} \text{Precision}_i}{50}, \quad g = 1,2,3 \ldots 10.
\]

\[
\text{Avg}(\text{Recall})_g^A = \frac{\sum_{i=1}^{50} \text{Recall}_i}{50}, \quad g = 1,2,3 \ldots 10.
\]

\[
\text{Avg}(F\text{-value})_g^A = \frac{\sum_{i=1}^{50} F\text{-value}_i}{50}, \quad g = 1,2,3 \ldots 10.
\]

in which \( g \) is the g-th group of proteins in dataset A. So we obtained 10 pairs of precision-recall values (i.e. from \( (\text{Avg}(\text{Precision})_1^A, \text{Avg}(\text{Recall})_1^A) \) to \( (\text{Avg}(\text{Precision})_{10}^A, \text{Avg}(\text{Recall})_{10}^A) \) ) for each algorithm. Firstly, we plotted comparisons of \( \text{Avg}(\text{Precision})_g^A, \text{Avg}(\text{Recall})_g^A \) and \( \text{Avg}(F\text{-value})_g^A \) of each algorithm. Then for each algorithm we plotted its precision-recall curve which consists of 10 pairs of precision-recall values as mentioned. The same procedures were conducted on dataset B. Finally, we present the comparisons of precision, recall, F-value and precision-recall curves of iterative approach and reference algorithms. Comparisons of INC and NC algorithm on two datasets are illustrated as follow:
Comparisons on datasets containing random levels of unannotated proteins

(Results were from datasets consisting of random proportion of unannotated proteins. Precisions, Recalls and F-values from one dataset are placed in the same column).

Comparisons of P-R Curves:

Fig. 3.3.3.2 Comparisons on datasets containing random levels of unannotated proteins (Results were from datasets consisting of random proportion of unannotated proteins. Precisions, Recalls and F-values from one dataset are placed in the same column).

Fig. 3.3.3.3 P-R curves of iterative and non-iterative algorithms.
It is shown from the above overall performance evaluation results that our iterative model surpassed reference algorithms in terms of precision, recall and F-value on both datasets.

### 3.3.4 Sensitivity Analysis over iteration times

When the iteration times increase, the importance of the initial prediction may decrease. Thus a sensitivity analysis over iteration times before achieving a stable status was carried out to show the prediction performance with different times of iteration.

In order to reveal trends of sensitivity variation under datasets that consist of different rates of unannotated proteins, we chose three sets of proteins from MIPS network. The first set contained 100 proteins each of which has 10% unannotated neighbours. Another 100 proteins in the second dataset had random proportion of unannotated neighbours. In the third dataset, there are 100 proteins each of which has 90% unannotated neighbours. We then applied iterative neighbour counting (INC) to predict functions of each protein. We recorded the times of iteration on each dataset and the corresponding true positive rate (i.e. recall) of the predicted results (see Figure 3.3.4.1).

![Sensitivity Analysis over Iteration Steps](image)

*Fig. 3.3.4.1. Sensitivity Analysis of INC on datasets that consist of different proportions of unannotated proteins.*
It can be observed that the iteration needs more steps to reach the stable status when the unannotated proteins are of a higher proportion. Statistically, in the same dataset, the predicted functions had higher true positive rates when the prediction went through more iteration steps.

### 3.3.5 Robustness of the iterative approach

As indicated in our algorithm, for each round of iteration, unannotated proteins were predicted in a random order. And we randomly chose predicted results for those target proteins whose predicted functions repeated in a pattern after certain rounds of iteration. Thus another issue we were concerned about is whether the final predicted functions will be changed from different runs of prediction (i.e. the robustness of the iterative model). We ran our algorithm on the MIPS network to evaluate the robustness of our iterative model to different prediction orders of the target proteins. During each run, the prediction started from different target proteins at the beginning of each iteration, and different runs also had different orders of prediction. We then analysed the prediction results from different runs. For instance, we set 50% of protein YHL011C’s neighbourhood proteins as unannotated. FunCat annotations of the predicted functions in different iteration orders from independent runs are listed in table 3.3.5.1:

<table>
<thead>
<tr>
<th>Run No.1</th>
<th>Run No.2</th>
<th>Run No.3</th>
</tr>
</thead>
<tbody>
<tr>
<td>YNL189W</td>
<td>YOL061W</td>
<td>YHL011C</td>
</tr>
<tr>
<td>&gt;YOL061W</td>
<td>&gt;YHL011C</td>
<td>&gt;YNL189W</td>
</tr>
<tr>
<td>&gt;YHL011C</td>
<td>&gt;YNL189W</td>
<td>&gt;YOL061W</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Run No.1</th>
<th>Run No.2</th>
<th>Run No.3</th>
</tr>
</thead>
<tbody>
<tr>
<td>YNL189W</td>
<td>YHL011C</td>
<td>YNL189W</td>
</tr>
<tr>
<td>&gt;YHL011C</td>
<td>&gt;YNL189W</td>
<td>&gt;YHL011C</td>
</tr>
<tr>
<td>&gt;YOL061W</td>
<td>&gt;YOL061W</td>
<td>&gt;YOL061W</td>
</tr>
</tbody>
</table>
The 3rd iteration

<table>
<thead>
<tr>
<th>Protein</th>
<th>1st iteration</th>
<th>2nd iteration</th>
<th>3rd iteration</th>
</tr>
</thead>
<tbody>
<tr>
<td>YHL011C</td>
<td>&gt;YNL189W</td>
<td>&gt;YHL011C</td>
<td>&gt;YOL061W</td>
</tr>
<tr>
<td>YOL061W</td>
<td>&gt;YOL061W</td>
<td>&gt;YNL189W</td>
<td>&gt;YHL011C</td>
</tr>
<tr>
<td>YNL189W</td>
<td>&gt;YNL189W</td>
<td>&gt;YHL011C</td>
<td>&gt;YOL061W</td>
</tr>
</tbody>
</table>

stable status

<table>
<thead>
<tr>
<th>Protein</th>
<th>Predicted Functions</th>
<th>Predicted Functions</th>
<th>Predicted Functions</th>
</tr>
</thead>
<tbody>
<tr>
<td>YHL011C</td>
<td>01.04</td>
<td>01.04</td>
<td>01.04</td>
</tr>
<tr>
<td>YOL061W</td>
<td>01.03.01.03</td>
<td>01.03.01.03</td>
<td>01.03.01.03</td>
</tr>
<tr>
<td>YNL189W</td>
<td>02.07</td>
<td>02.07</td>
<td>02.07</td>
</tr>
</tbody>
</table>

Table 3.3.5.1. Prediction results of YHL011C. (It can be observed that in three different prediction runs, orders of proteins prediction (for example in the 1st iteration of run No.1, it starts prediction from YNL189W, then YOL061W, finally YHL011C) in each run are different. Predicted functions reached stable status after the 3rd iteration in all three runs, and different runs achieved the identical predicted functions at last). Furthermore, we randomly chose 100 proteins that have unannotated neighbours to evaluate the robustness of our iterative model. The evaluation result of iterative FSW algorithm is presented in Figure 3.3.5.1 below:

Fig. 3.3.5.1. Numbers of target proteins that had identical predicted functions from multiple independent predictions. (Numbers of the proteins that had identical predicted functions from multiple independent predictions and the proteins that had different predicted functions from multiple independent predictions were presented in this chart. Orders of proteins being predicted were set to be different in each run.
This figure shows that 82 out of 100 proteins have identical predicted results from different predicting orders, which demonstrated the robustness of our iterative model. Actually, since the $(t + 1)$-th round of prediction only use prediction results from the $t$-th iteration, thus the prediction order in each round of iteration has no influence on the final prediction results. The changes in results are mainly caused by randomly selecting function groups as prediction results after the iteration repeats in a pattern. The above experimental results also showed the iterative approach can reach stable status very fast. Actually, predicted functions are selected from the functions of annotated proteins in the prediction domain. The iterative model enlarges the prediction domain of a prediction target protein. Functions in the prediction domain are ranked and selected as the predicted results at the initialisation stage. When the iteration starts, due to the incorporation of those unannotated proteins with initially predicted functions into the PPI network, the prediction algorithms take the corresponding interactions and new functions into consideration. So functions are to be re-predicted and re-ranked. With the following iterations, however, the network structure and highly ranked candidate functions remain stable, while changes to the prediction results happen on those lowly ranked functions that endorse each other. On the other hand, the number of function is finite in the prediction domain. Therefore, predicted results will finally reach a stable status after several iterations.

### 3.4 Conclusions

This paper proposes a new iterative model on the PPI network for protein function prediction. Protein interaction information of unannotated proteins in the prediction domain was utilized in the prediction process to improve the prediction results. This model could be applied on existing protein function prediction algorithms and could enhance their prediction performance. In this paper, we employed the NC algorithm, the
FSW algorithm and the Function Flow method in our experiments as reference algorithms. Our iterative algorithms improved the prediction performance of the NC algorithms, the FSW algorithm and the Function Flow method in most cases. Also, most prediction results were independent from the orders of proteins being predicted in the iteration process. Since the predicted functions could reach the stable status usually after no more than 3 iterations, the time cost of the iterative model mainly depended on the time cost of the chosen prediction algorithms.

Our iterative approach simultaneously recorded functions of all unannotated proteins in prediction domain. So our iterative approach is capable of predicting functions of all unannotated proteins that share the same prediction domain. After iteration, a matrix in which columns indicates functions that have been predicted in each iteration can be generated for every target protein, respectively. Interpreting these generated matrix and relationships between matrices, filtering unannotated proteins taking part into prediction and assigning weights to different iteration steps are questions to be studied. Applying the iterative model on Gene Ontology (The Gene Ontology Consortium, 2000) semantic information based and genomic data based prediction algorithm is our future work. Also exploiting datasets that are integrated from diverse biological data to increase performance of the iterative approach is going to be conducted in the near future.
Chapter 4

Contemporary protein function prediction approach

In this chapter, we introduce our DPPIN based time-course protein function analysis methods. Section 4.1 firstly introduces current applications of DPPIN and existing challenges and problems, and then determines the objectives of the research. Section 4.2 introduces the entire pipeline of our approach. Section 4.3 presents the evaluations of each method in our approach. Finally, Section 4.4 and Section 4.5 provide discussion and conclusion about this research.

4.1 Introduction

4.1.1 Background

Protein is considered as a basic component in biological process. Computational protein function prediction methods are crucial for biology research since the cost of identifying protein functions through biological experiments could be relatively high, and laboratory experiments could be inefficient in many circumstances. Currently, there still are large proportions of unannotated proteins of each species (as figure 1.2.1 depicted). Therefore, manual experiments are heavily challenged in terms of efficiency. Furthermore, since the rapid speed of genomic data generation, in-vivo experiments are not capable to satisfy the need of annotating new gene and gene products.

Protein-protein interaction network derived from high-throughput technology is recognised as an informative tool for discovering protein function. In the past decade, a lot of graph-based algorithms have been developed to utilize protein associations in PPINs that are regarded as static networks. An early survey about these prediction algorithms had been complete by Roded et al. (Roded et al., 2006). Also some recent
surveys had been presented (Clara et al., 2014, Kire et al., 2014).

Most of current PPIN based protein prediction methods are built on analysis of static PPINs. Graph based clustering algorithm such as MCL and KNN. Recently, some classic clustering methods were applied on integrated networks to discover functional modules in PPINs, for instance, SAMBA (Tanay et al. 2002), MS-KNN (Liang et al. 2013) and JCA (Wang et al. 2014).

Though PPIN based protein function prediction triumphs over high-cost biological experiments in some extent, there are still gathering dark clouds that cast shadow on the application of PPINs. At first, due to the limit of high-throughput technology, the current PPINs contain large proportions of false positive interactions, and a lot of truly existing interactions are excluded in generated PPINs. It therefore brings noise information into function prediction processes. Furthermore, original PPINs are statics networks that contain interactions from different evidence sources. In another word, the interactions in a PPIN are collected from different biochemical conditions or different stages of cell processes. Thus PPINs should be dynamic networks instead of static networks.

Dynamic PPINs (DPPINs) are a series of transient static PPINs extracted from different stage of a cell cycle. Time-course gene expression data provided by DNA micro-array technology offer the opportunities to extract temporal PPINs at certain time intervals based on a static PPIN. Gene expression data is becoming a dominating solution to the construction of DPPINs. Compared to static PPINs, DPPINs are more precisely illustrating the interactions of proteins at certain stages of cell cycle. Moreover, the associations among DPPINs from different times provide further possibilities for computational analysis of protein function.

DPPIN is currently mainly used for protein complex detection and crucial protein identification. Chen et al. applied DPPIN in protein complex identification (Bolin Chen et
The protein complexes and functional modules were identified based on topology structure of the DPPINs. Protein permanently interacting with neighbouring proteins across different time intervals is considered to be a party hub that works with its interacting proteins as a protein complex. Instead, when a protein has temporal interactions with its neighbouring proteins in the DPPINs, the protein along with its neighbouring proteins are considered as in a function module. Later, Zhang et al. proposed a deep belief network based critical protein identification method (Yuan et al., 2014).

### 4.1.2 Challenges and problems

There are still vital problems to tackle before PPIN based protein functions prediction can reach a verdict of remarkable accuracy. At first, it is a common sense that protein interactions in PPIN do not always exist during the whole cell cycle. Thus a PPIN should be considered as a dynamic network that changes over time. Consequently, it is important to decide whether an interaction exists at certain time. As a promising solution to this problem, the dynamic PPINs (DPPINs) could be constructed by incorporating gene expression information with protein interaction information and DPPINs are capable to reflect temporal interactions at different moments. The extent of expression level of genes that code proteins and the correlation of expression level between genes that code a pair of interacting proteins are regarded as the state-of-the-art criteria for DPPIN construction.

However, from the perspective of function annotation, proteins often have incomplete annotations since laboratory experiments could not promise the perfection of discovered functions for proteins. This leads to incomplete prediction results and sensitivity (i.e. recall). Furthermore, current function annotations of proteins are integrations of all functions that have been detected. In another word, it lacks structured time-course information for in silicon analysis (i.e. we do not know the actually temporal functions...
that occur at a certain time if we have not read the specific descriptions of each function term). It therefore becomes a drawback that there is a gap between association of the annotations and the temporal interaction information.

4.1.3 Objective definition

The major problem to be addressed is how to identify the sets of functions performing by annotated proteins at particular time intervals (i.e. contemporary functions). A gene and the protein coded by it are considered being active at time $t$ if the expression of this gene at $t$ is at a high level. Furthermore, an interaction is considered as existing at time if the corresponding pair of interacting proteins in static PPIN are both active at time $t$. A series of temporal DPPINs therefore could be extracted from the static PPIN. However, since the annotation terms do not directly carry structured time information with them, it is vital to match functions performed at certain times to corresponding DPPIN.

The second objective is to appropriately predict protein functions based on sparse data. Since large proportions of interactions are excluded in transient PPIN, the DPPINs are usually sparser than static PPINs. Sparse data highly restrained the performance of some existing protein function prediction algorithms. Therefore, we need proper method to obtain relatively satisfying results.

By studying the approach to identify protein functions during certain interval based on DPPIN, it could significantly benefit research and technology in areas such as bioengineering and drug design. In this research, we present a novel ontology based method to tackle the above problems. The paper is organised as follows. In Section 4.2, we introduce our method. In Section 4.3, we present experimental results on performance of our method. We discuss defects of this method and the potential solutions in Section 4.4. Finally, in Section 4.5, conclusions about the method and our future work are presented.
4.2 Methods

4.2.1 Data processing and network construction

Time-course gene expression data are engaged to static PPI networks in order to construct dynamic PPINs that vary over time. A static PPI network is usually denoted by an undirected graph $G = (V, E)$ in which $G$ and $E$ represent nodes (i.e. proteins) and edges (i.e. interactions) in $G$, respectively.

The time-course gene expression data numerically records the expression levels of $N$ genes across $T$ time intervals, thus it is a $N \times T$ matrix $M_{GE}$. Each column of $M_{GE}$ depicts the expression level of genes at a certain interval. $M_{GE}$ usually contains expression records from multiple cell life cycles. Suppose there are totally $C$ life cycles, and gene expression levels of $T_C$ time intervals are recorded within each life cycle, we have $T = C \times T_C$. In order to reduce noise in expression data, we used the mean value of $C$ expression values that come from the same time interval in different cycles as the final
expression value of a gene. So a $N \times T_C$ matrix $M_{GE_{\text{Mean}}}$ can be generated as the final gene expression matrix.

### 4.2.1.1 Protein Activeness

Since a protein can be coded by several genes, we further compressed the gene expression matrix to a $N_p \times T_C$ protein activeness matrix $M_P$ that each element in the matrix is the activeness of a protein at a time interval. If a protein $P$ is coded by multiple genes (e.g. $g_1, g_2, \cdots, g_k$), the activeness of $P$ is simply represented as $A_P = \sum_{i \in [1,k]} G_i$, where $G_i$ is the expression value of gene $g_i$. Then we set expression level thresholds to judge if proteins are active or not. According to the discussion made by Jianxin et al. (Jianxin et al., 2013), we applied $3\sigma$ principle to assign each protein $P$ an active threshold at time interval $t$:

$$\text{Threshold}_{(P,t)} = W(t) \times (E(\psi_P) + 3\sigma(\psi_P)(1 - F(\psi_P))) \quad (4.2.1.1.1)$$

where $\psi_P$ is the activeness value of protein $P$. $E(\psi_P)$ and $\sigma(\psi_P)$ are the expectation and standard deviation of the activeness values over interval $1$ to interval $T_C$ of protein $P$ respectively. $F(\psi_P, t)$ is a weight function defined as follow:

$$F(\psi_P) = \frac{1}{1+\sigma^2(\psi_P)} \quad (4.2.1.1.2)$$

$W$ is a linear parameter used to balance different environments during measuring gene expression levels. Some actually active proteins that have low activeness values could still be included into DPPINs. It is defined as:

$$W(t) = T_C \times \frac{E_{t_{\text{Mean}}}}{\sum_{t \in [1,T_C]} E_{t_{\text{Mean}}}} \quad (4.2.1.1.3)$$
where $E_{t,\text{Mean}}$ is the mean value of activeness levels of $N_p$ proteins at interval $t$.

4.2.1.2 Weighted dynamic protein interaction construction

Interactions in DPPIN are constructed based on the static PPI and the activeness of interacting proteins at a certain time interval. The dynamics of PPINs can be represented by the dynamics of interactions over time. A series of temporal PPINs can be constructed to reflect the dynamics of interactions at different time intervals. Since interacting proteins are in their active form, suppose there are $P_1$ and $P_2$, which are a pair of interacting proteins in the static PPIN $G$. If both of their activeness values surpasses the thresholds at time interval $t$, the interaction $(P_1, P_2)$ exists in the temporal PPIN of time $t$. Thus we have built totally $T_c$ temporal PPINs based on active proteins at each time.
It was studied that a pair of interacting proteins tend to have more conserved co-expression than random ones \((N. Bhardwaj et al, 2005)\), and co-expression pattern over time is usually represented by co-expression coefficient. Thereby we assigned weights to the interactions in the DPPINs using co-expression coefficients of the expression values (i.e. activeness values) between the pairs of interacting proteins. We used Pearson correlation coefficient (PCC) of the activeness values that cover a period of time as the co-expression coefficient. For interaction \((P_1, P_2)\) at time \(t\), its weight \(w_{PCC}\) (the co-expression extent of \(P_1\) and \(P_2\)) is measured by the Pearson correlation coefficient of \(\{\psi((P_1,t-1), (P_1,t), (P_1,t+1))\}\) and \(\{\psi((P_2,t-1), (P_2,t), (P_2,t+1))\}\). In this way, a series of weighted temporal PPINs were constructed.

### 4.2.2 Protein Function Prediction

In this section, we introduce our proposed method for time-course function prediction. The prediction domain (i.e. the proteins whose function annotations are used for prediction) is defined as level one proteins (i.e. directly interacting protein) and level two proteins (i.e. proteins directly interacting with level one proteins) of a prediction target protein \(p_x\).

#### 4.2.2.1 PPI-Function association graph transformation

In this section, we introduced how to transform a PPIN to a PPI-Function Association (PFA) graph. Since annotated functions of a protein may not all exist at the same time, we need to find out the currently most possible existing functions. We therefore built PFA graph of sub networks from a transient PPIN to assist the identification of the contemporary functions of proteins.
Figure 4.2.2.1. The transformation from PPIN to PFA

We only use the sub network of the prediction domain of $P_x$ for transformation since most functions of a protein can be found from its directly interacting neighbours and indirectly interacting proteins (i.e. level-1 and level-2 proteins) (Chua et al., 2006). Suppose we want to analyse current functions of $P_x$ in the $t$th DPPIN. In the prediction domain of $P_x$ in the $t$th transient PPIN, there is a protein $P_1$ which has functions $\{f_1, f_2, \ldots, f_k\}$. Suppose there is another protein $P_2$ which has functions $\{f'_1, f'_2, \ldots, f'_l\}$. $P_1$ and $P_2$ are interacting in the transient PPIN. Then the edge of $(P_1, P_2)$ can be transferred to edges in PPI-Function association graph. We have:

$$(P_1, P_2) \xrightarrow{\text{Transform}} \{(P_1, f_1), (P_2, f'_1), (P_1, f_2), (P_2, f'_2), \ldots, (P_1, f_k), (P_2, f'_l)\} \quad (4.2.2.1.1)$$

Same transformation method has been applied on other edges in the PPIN. Finally, the PPIN sub network was transformed to a PPI-Function association graph.
4.2.2.2 Data pruning and core function identification

Based on the Gene ontology relations, we pruned the PPI-Function association graph to find core functions that have the highest possibilities to be current functions. Inferring functions that a protein performs at a certain time could sometimes be more interesting than identifying the group of functions that this protein has had. However, this task is difficult to tackle since the insufficient structured time-related information in current function annotation databases. We employed the relationships of GO terms to infer possible occurring functions as core functions in a transient PPIN.

Gene ontology relations are used for graph pruning. Current gene function annotation system does not provide the time information of GO annotations. However, the relationships of function terms are stated by highly structured words (e.g. "is_a", "part_of" and "regulates"). On the other hand, the GO relationships clearly infer the dependency and co-existence of function terms. For example, a “part_of” relation between function $f_1$...
and $f_2$ indicates that $f_2$ is a necessarily part of $f_1$, wherever $f_2$ exists, it is as part of $f_1$, and the presence of the $f_2$ implies the presence of $f_1$. Thus based on the GO relationships, we pruned the PFA graph to find potentially co-existing functions.

We used "is_a" and "part_of" relations in the pruning. It should be noticed that GO relations are transitive:

1) if $A$ is_a $B$, $B$ is_a $C$, then $A$ is_a $C$.
2) Similarly, if $A$ part_of $B$, $B$ part_of $C$, then $A$ part_of $C$.
3) Also, if $A$ is_a $B$, $B$ part_of $C$, then $A$ part_of $C$.

Hence if two GO terms has path which is comprised of “is_a” and “part_of”, the edges between nodes which have corresponding GO terms in the PFA should also be retained. The pruning steps are presented in the following table:

```
For each edge $(n_1, n_2)$ in PFA:
    If GO term in $n_1$ is_a GO term in $n_2$, or GO term in $n_1$ is part_of GO term in $n_2$:
        Keep $(n_1, n_2)$;
    If GO term in $n_1$ has path to GO term in $n_2$ in Ontology:
        Keep $(n_1, n_2)$;
    Else:
        Remove $(n_1, n_2)$ from PFA
```

Table 4.2.2.2.1. PFA pruning procedure.

The pruning serves as a method to find out functionally related annotations among proteins. Since protein perform function through binding with another protein, the pruning actually build links between functions that are probably performing by pairs of interacting proteins. After pruning, the largest community in PFA is considered as the community consisting of core functions. The core functions are most likely the current functions being performed by proteins. Next, the function annotations are mapped back
to corresponding proteins as their contemporary annotations of current time interval.

### 4.2.2.3 Function prediction based on SVC

For each protein $P_i$, the annotations $\{f_1, \cdots, f_k\}$ of $P_i$ are the labels of $P_i$. Predicting protein function of $P_x$ is thus a multi-label classification problem. The functions of the prediction target $P_x$ were predicted based on linear support vector machine classifier (linear SVC) \((\text{Alex and Bernhard, 2004})\).

Let $X_i \in \{X_1, \ldots, X_n\}$ be the set of proteins and let $Y_i$ be the set of labels of $X_i$, hence $Y_i = \{y_1, \cdots, y_k\}$ is the set of functions labelled to protein $X_i$. Each $y_i$ is a binary indicator: $y_i \in \{0, 1\}$ in which $y_i = 0$ means $X_i$ is not labelled with $y_i$, and $y_i = 1$ means $X_i$ is labelled with $y_i$. Given training data $X_i$ and corresponding labels $y_i$, linear SVC solves the problem:

$$
\min_{\omega, b, \varepsilon} \frac{1}{2} \omega^T \omega + C \sum_{i=1}^{n} \varepsilon_i \tag{4.2.2.3.1}
$$

Here $C > 0$ is a trade-off parameter, the minimisation problem is subject to:

$$
y_i(\omega^T \phi(X_i) + b) \geq 1 - \varepsilon_i, \varepsilon_i \geq 0, i = 1, \cdots, n \tag{4.2.2.3.2}
$$

Here $\phi$ is a function to transform $X_i$ to a higher dimensional feature space. For a protein sample $P_s$, we chose the weight of the function $f_i$ in the prediction domain of $P_s$ the as feature. There are two weight calculation strategies which are $Weight_\delta$ and $Weight_{pCC}$ defined. The weights are calculated as:

$$
Weight_\delta(f_i) = \sum_{s=1}^{n} \delta(f_i, P_{sn}) \tag{4.2.2.3.3}
$$
where $P_{sn}$ is the protein in the prediction domain of $P_s$. $f_i$ is the function annotation in the prediction domain. $\delta$ is an indicator function. $\delta(f_i, P_{sn}) = 1$ if protein $P_{sn}$ has label $f_i$. Otherwise $\delta(f_i, P_{sn}) = 0$. Similarly, we have:

$$
Weight_{pcc}(f_i) = \sum_{s_n=1}^{n} w_{pcc}(f_i, P_{sn})
$$

(4.2.2.3.4)

where $w_{pcc}$ is based on the Pearson Correlation Coefficient (PCC) of interactions in DPPIN, which has been introduced in Section 4.2.1.2.

$$
\begin{align*}
    w_{pcc}(f_i, P_{sn}) &= \begin{cases} 
        \text{PCC}(P_s, P_{sn}), & \text{if } P_{sn} \text{ is directly interacting with } P_s, \\
        \text{Max}(\text{PCC}(P_s, P_u) * \text{PCC}(P_u, P_{sn})), & \text{if } P_{sn} \text{ interacts with } P_s \text{ through } P_u
    \end{cases}
\end{align*}
$$

(4.2.2.3.5)

For a linear SVC, the training data is used to learn the weight vector $\omega$. As a multi-label classification problem, we independently trained one binary classifier for each label $y_i$ in $Y_i$.

Based on annotated proteins and their annotations, we constructed two training datasets that will be introduced in Section 4.3.1.1. After training the classifiers, we applied trained classifiers to classify prediction target proteins $P_x$ to function labels. We used the contemporary annotations from Section 4.2.2.2 as features of $P_x$.

### 4.3 Results

#### 4.3.1 Datasets

This research utilised PPI data, gene expression micro-array data and gene function annotation data. We employed yeast protein interaction datasets from the Database of Interacting Proteins (DIP, [http://dip.doe-mbi.ucla.edu/dip/](http://dip.doe-mbi.ucla.edu/dip/)) to construct the static PPIN.
The DIP dataset consists of interaction records of proteins that have amino acid chains binding based on experimentally identification. After downloading the raw network which contains 4,627 proteins and 22,874 interactions, we removed redundant and self-interactions from the network. After processing, there are 4,627 proteins and 21,750 interactions in the DIP dataset.

In order to construct dynamic PPINs, dataset from GSE3431 was used in this research. The dataset contains gene expression profiles of yeast genes in three successive cell cycles. There are gene expression profiles of totally 6,304 genes contained in this dataset. For each gene, there are expression values from 36 time intervals in which consist of 12 time intervals from each of the three cell cycles, representatively.

The gene/protein function annotations data used in this research is the Yeast GO annotation data downloaded from SGD database (http://www.yeastgenome.org/). The data contains annotation terms in 7,794 Molecular Function, 14,114 terms in Biological Process and 12,796 terms in Cellular Component. The ontology data is the 2014-05-27 version downloaded from Gene Ontology website (http://geneontology.org/). We used annotations in Biological Process (BP) namespace in our research.

### 4.3.1.1 Training data

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Number of samples</th>
<th>Number of features</th>
<th>Label</th>
</tr>
</thead>
<tbody>
<tr>
<td>Training dataset 1</td>
<td>4,627</td>
<td>2,557</td>
<td>Annotation of sample protein ( P_z ) in BP namespace</td>
</tr>
<tr>
<td>Training dataset 2a,2b</td>
<td></td>
<td></td>
<td>Protein numbers in the prediction domain of ( P_z )</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Function numbers in the prediction domain of ( P_z )</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Annotation of sample protein ( P_z ) in BP namespace</td>
</tr>
</tbody>
</table>

Table 4.3.1.1.1. The summary of training datasets.
We used two training datasets for the classifiers. First Dataset 1 was all 4,627 proteins in the DIP PPI and corresponding GO annotation in biological process namespace were used for training the classifier. For each protein $P_i$, the numbers of functions in level-1 and level-2 neighbouring proteins of $P_i$ are counted as features of $P_i$. There are totally 2,557 GO annotations, thus this training dataset contains 2,557 features.

Secondly, for each prediction target $P_x$, the training Dataset 2a and Dataset 2b only consisted of the proteins in the prediction domain of $P_x$. In this case, each protein had an individually trained classifier. Similar with Dataset 1, features in Dataset 2a were weighted by $Weight_\delta$. In contrast with the first training dataset, features in Dataset 2b was weighted by $Weight_{PCC}$ defined in Section 4.2.2.3. The feature selection procedures are introduced in Table 4.3.1.1.2.

**Training dataset 1 feature: the number of a BP GO annotation in the DIP PPIN.**

For each sample $P_s$ in DIP PPIN:

For feature $e_k$ in features:

For function in level-1 and level-2 protein of $P_s$:

$feature_k = \text{Counted number of function if } function = feature_k$

**Training dataset 2a feature: the number of a BP GO annotation in the prediction domains of neighbouring proteins of the testing protein $P_x$.**

For each testing protein $P_x$ from the DPPIN at the $t^{th}$ interval:

Weighted each $P_s$ by $w^s_{PCC}$ for each sample $P_s$ in the prediction domain of $P_x$

For each testing protein $P_x$ from the DPPIN at the $t^{th}$ interval:

For function in functions of $P_s$:

if function = feature$_k$:

$Weighted\_feature_k += 1$

**Training dataset 2b feature: the weighted number of a BP GO annotation in the prediction domains of neighbouring proteins of the testing protein $P_x$.**

For each testing protein $P_x$ from the DPPIN at the $t^{th}$ interval:

Weighted each $P_s$ by $w^s_{PCC}$ for each sample $P_s$ in the prediction domain of $P_x$

For each testing protein $P_x$ from the DPPIN at the $t^{th}$ interval:

For function in functions of $P_s$:

if function = feature$_k$:

$Weighted\_feature_k += w^s_{PCC}$
4.3.1.2 Data for prediction

Leave one out cross validation (LOOCV) was adopted for the purpose of validation. Based on LOOCV, we treated an annotated protein as unannotated protein $P_x$ (i.e. prediction target) and then applied prediction approach on $P_x$. The predicted functions of $P_x$ were compared with the ground-truth functions of $P_x$ to evaluate the prediction performance.

First testing dataset *Protein set a* contained 30 randomly chosen prediction targets that presented in different DPPINs from the cell cycle. For each protein $P_x$, its functions in each DPPIN were predicted individually, then the prediction results from different DPPINs (i.e. different time intervals) were merged together as the total predicted annotations of $P_i$. We then compared the predicted functions of $P_i$ with its ground truth annotations in terms of precision, recall and F-value.

Secondly, in order to examine whether predicted contemporary functions of $P_i$ are correct, we selected 49 prediction targets that have functions relating to Mitosis phase of the yeast cell cycle as prediction targets to test our method. The detailed evaluation method is introduced in Section 4.3.2.1.

4.3.2 Results Evaluation

The overall precision of the prediction results was evaluated using leave-one-out-cross-validation methods. Furthermore, we validated predicted contemporary functions based on particular GO terms whose time of presence are known.
4.3.2.1 Precision of contemporary function prediction

We choose a function set consists of GO terms whose occurring time can be identified to evaluate whether the predicted contemporary functions are actually matched with the functions that happen during corresponding cell-cycle. As we know, the Gene Ontology system has a hierarchical structure in which descendant terms are usually more specific than their ancestors. In the Biological Process namespace, higher level GO terms describe general biological processes which may present across the entire cell cycle. Thus for the purpose of validation, we chose GO terms which are specific enough to instruct the time when they occur as test standard.

GO terms which are related to Mitosis phase were extracted from the ontology to build the marking function dataset. Cell cycle contains four phases that are sequentially G1, S, G2 and M. Mitosis (M phase) is a special and dramatic phase in cell cycle (Cooper et al., 2000). The cell cycle in S. Cerevisiae is an alternation between two self-maintaining stable steady states (G1 and S/G2/M). The Start transition carries a cell from G1 to S/G2/M, and the Finish transition from M back to G1 (Nasmyth, 1996, Tyson et al., 1995, Tyson et al., 2001). In this research, we extracted GO terms that describe biological processes occurring during the Mitosis stage as annotations used for validation.

At first, in order to choose DPPINs which are located in the intervals corresponding to M phase, we adopted the corresponding high-resolution cell cycle timing data of S. Cerevisiae (Maga et al., 2007). The downloaded cell cycle phases in physiological time records are as follows:

<table>
<thead>
<tr>
<th>Phase</th>
<th>Beginning</th>
<th>End</th>
</tr>
</thead>
<tbody>
<tr>
<td>G1, Pre-replicative late</td>
<td>10</td>
<td>38</td>
</tr>
<tr>
<td>G1/S</td>
<td>38</td>
<td>55</td>
</tr>
<tr>
<td>S</td>
<td>55</td>
<td>89</td>
</tr>
</tbody>
</table>
According to the time duration proportions of G2/M, M and M/G1, we selected DPPINs in the 6th and 7th time interval as M phases related DPPINs (i.e. Strictly M phase related DPPINs). The DPPINs in the 5th, 6th, 7th and 8th time intervals are selected as G2/M, M and M/G1 phases related DPPINs (i.e. Non-strictly M phase related DPPINs).

On the other hand, we located “father GO terms” which are located nearest to the ontology root (i.e. GO:0008150, biological_process) and related to M Phase based on Ontology graph search. The defined father GO terms suffice the features that: 1) The father GO term is related to M phase; 2) The ancestor of the father GO term is not related to M phase or is related to other phases other than M phase. We choose three representative father GO terms as shown in table below:

<table>
<thead>
<tr>
<th>GO ID</th>
<th>Name</th>
<th>Ancestor</th>
</tr>
</thead>
<tbody>
<tr>
<td>GO:0000279</td>
<td>M phase</td>
<td>GO:0022403 cell cycle phase</td>
</tr>
<tr>
<td>GO:0000910</td>
<td>Cytokinesis</td>
<td>GO:0022402 cell cycle process</td>
</tr>
<tr>
<td>GO:0000280</td>
<td>Nuclear division</td>
<td>GO:0048285 organelle fission</td>
</tr>
</tbody>
</table>

Secondly, we extracted the descendant terms of selected father GO terms according to the hierarchical structure of GO. All the father terms and their descendants formed a list of marking functions. These marking functions appear during M phase which happens at the end of the cell cycle. We then randomly chose 49 proteins that contain the marking functions and exist in the 6th and 7th DPPIN as the statistical population.
We validated predicted contemporary functions of proteins from the 6th and 7th DPPINs. We compared the number of proteins that predicted to have M phase functions against the statistical population (the 49 proteins) as the overall prediction recall. Then we compared the predicted M phase functions of each protein with the true M phase functions of this protein as single precision. To avoid bias, the predicted functions that are higher-level terms of true M phase functions are regarded as correct prediction. The mean of the single precisions was calculated as the average precision.

*Figure 4.3.2.1.1. The overall recall.*

It can be discovered that results from classifier trained by Dataset 2a and Dataset 2b have similar recall and both are higher than recall by Dataset 1.
It could be observed that three classifiers had similar average precision. The size of each marker was associated with the semantic similarity between the predicted function set and the true function set. We adopted Wang’s similarity definition (Wang et al., 2007) in the calculation. The larger the marker is, the higher semantic similarity the predicted functions and true functions have. Generally, the predictions having higher precisions are illustrated in the figure by larger markers, which indicates the prediction results have high semantic correlation with actual functions.

**4.3.2.2 Overall prediction precision**

In addition to contemporary function validation, we also evaluated the overall prediction performance by merging predicted functions of a protein from different transient PPINs. We then compared the total annotation with the ground truth function. The Precision, Recall and F-value of predicted functions of proteins in Protein set a are illustrated in figures below:
Figure 4.3.2.2.1. The Precisions of prediction results of Protein set a. The precision of each protein is plotted. The average precision of prediction results is plotted as the line in the figure. It can be observed that classifier trained by Dataset 2a has the highest prediction precision, then is the classifier trained by Dataset 2b. Classifier trained by Dataset 1 has the worst performance.

Figure 4.3.2.2.2. The Recalls of prediction results of Protein set a. The recall of each protein is plotted. The average recall of prediction results is plotted as the line in the figure. Different with the precision, it can be observed that classifier trained by Dataset 2b has the highest prediction precision, then is the classifier trained by Dataset 2a. Classifier trained by Dataset 1 still has the worst performance.
The classifier trained by *Dataset 2a* outperformed other classifiers in terms of average precision. However, classifier trained by *Dataset 2b*, which employed PCC values of protein interactions as weight, had the best performance in terms of average recall. The PCC value of interactions contains noise since the PCC is calculated only based on three adjacent time intervals. On the other hand, the quality of the gene expression micro-array also affects the calculation of PCC.

**4.4 Conclusion**

In this paper we presented a method to predict contemporary protein functions based on transient DPPIN. We designed a gene ontology based method to find active functions at certain time intervals. We extracted M phase related functions annotations as benchmarks to validate predicted contemporary functions from DPPINs that locate in the M phase intervals. We also evaluated the precision, recall and F-Value of predicted functions.
composed of functions that are predicted from DPPINs across the whole cell cycle.

There are two main factors that would affect the measured prediction performance of contemporary functions. Firstly, the original annotation dataset was filtered to a much sparser benchmark dataset that only consists of specific GO terms indicating functions within the M phase. However, there could be functions other than chosen ones existing at the interval. Therefore, the incomplete benchmark dataset could give rise to bias in prediction results evaluation. Consequently, the evaluation results might be below the true performance. Secondly, the chosen M phase intervals could overlap with other phases. It hence increased the incompleteness of the benchmark dataset.

As the future work, the evaluation method will be improved to avoid bias. It is promising to incorporate knowledge about cell cycle pathway into contemporary function determination and prediction results evaluation. Furthermore, a proper kernel for the support vector classifier will be sought to improve the prediction performance.
Chapter 5

Implementations

During this research, we implemented a demonstration of a software tool named Genetic Tools. The tool currently consists of protein function analysis module and Gene Ontology semantic analysis module. The protein function module contains our iterative methods INC and IWNC. This tool will be updated to embed more relevant algorithms. The tool can be downloaded from:

https://www.dropbox.com/sh/aba6sy5pu4ksuec/AADHK4LazN9XWSwbwy0K9ScKa?oref=f=e.

We present the basic features of the tool in the following sections.

5.1 Genetic Tools

Genetic Tools is an implementation of our iterative approach based on python tkinter toolkit and currently it is still a demonstration version. This tool has a clear GUI for user to perform protein function analysis or GO annotation analysis. The designed software structure chart is illustrated as below:
Currently, this tool works based on formatted data stored in local database. Connections to different online databases will be added in future. Currently it has protein interaction data of *S. Cerevisiae* for the purpose of demonstration. Data of other important species and other biological data will be linked in.

Screen shots of the homepage (agreement page), species option page and the tool list page under each species category are as follows:
Fig 5.1.2. The interfaces of Genetic Tools. Starting page (Graph from: www.coursebuffet.com), List of species (Currently only S. Cerevisiae available) and List of tools available after choosing species.

For each species, users can choose analysis they want to conduct from the tool list. Currently there are protein function analysis tools and GO semantic analysis tools available.
5.2 Implementation of the INC/IWNC methods

IWNC and INC are embedded in Genetics Tools. Given the Ensembl gene ID as input, user can view the predicted functions and the sub network consists of prediction target and its neighbouring proteins in the PPI network.

![Implemented INC and IWNC in Genetic Tools](image)

**Fig 5.2.1. Implemented INC and IWNC in Genetic Tools.**

5.3 Gene Ontology tools

Several GO annotation related tools are implemented in Genetic Tools. Given a GO term, users are able to locate the namespace of the term, find the parent annotation terms and calculate semantic similarity between GO terms.
Fig 5.3.1. Searching parent terms of a GO term.

Fig 5.3.2. Checking namespace of a GO term.
The semantic similarity definition adopted in Genetic Tools was proposed by Wang et al. (Wang et al. 2007). Users need to make sure the GO terms to be calculated are in the same name space, and these two terms need to be separated by a comma.
Chapter 6

Discussion and conclusion

This chapter provides an overall discussion and conclusion about accomplished research and future work. Section 6.1 discusses the existing deep challenges and future trends of computational protein function prediction. Section 6.2 concludes the pros and cons of our developed approaches, and then makes recommendations and description for our future work.

6.1 Discussion

Boundaries of researches with regard to heterogeneous biological data are increasingly blurred since the integration of biological data. The proper manner of using data is critical for improving prediction performance. In contrast with other types of data, biological data contain noise information and have pool structure for traditional algorithms to interpret. Nevertheless, the latent but important information in tremendous volume of unorganised biological data requires to be excavated by efficient and effective computational methods.

The future revolution of protein function prediction may come from two aspects. Firstly, the annotation strategy of protein function tends to be more specific and better structured, which will convert more biological data into computable information to boost the performance of computational function analysis. On the other hand, computational methods are enabled to participate in more areas of functional genomics. Secondly, along with the development of novel sequencing, spectrum and imaging devices, more types of data will be accessible.
6.2 Conclusion

The research presented in this thesis consists of two major parts. The first part is the proposed iterative protein function prediction method that increases the performance of protein function prediction when there are large proportions of unannotated protein in PPIN. Secondly, the proposed contemporary function prediction method escalates computational prediction of protein function to a more specific level. Two protein function prediction methods were proposed to computationally analyse protein functions during this research. This section summarises the issues addressed by this research, then provides details for our future work.

6.2.1 Review of main findings

The proposed methods in this research have addressed the unannotated protein issue, the one-off prediction issue and the transient function prediction issue. The two methods we proposed (Chapter 3 and Chapter 4). These methods are concluded below.

- The iterative method introduced in Chapter 3 improves performance of prediction when there are high proportions of unannotated proteins in PPINs. The method developed an iterative framework that repeatedly updates and predicts function of proteins in PPIN. The final predicted functions are chosen based on their statistical significance based on multiple iterations.

  The evaluation demonstrated that the iterative method significantly improved the performance of existing algorithms. The method had higher sensitivity when the times of iteration increased. On the other hand, the iterative method is robust to varied levels of unannotated proteins, iteration times and different prediction sequences.

- The contemporary function prediction algorithm predicts functions that are
contemporarily performed by proteins. The method introduced PFA graph based on GO term associations to build links between function and time. Linear SVC was adopted to classify function labels.

The method successfully predicted contemporary functions of the Mitosis phase using transient PPINs located around M phase interval. Three sets of training data were used to train classifiers. It was discovered that training data constructed by protein samples in the neighbouring area of the prediction target could achieve better performance. The overall prediction results across all time intervals achieved satisfying performance.

6.2.2 Future work and recommendations

Although the proposed methods addressed issues presented in Chapter 1, there are several improvements could be made in our future work. These include:

- Iteration weight: In the iterative model, iterations could be weighted to address the importance of each iteration. Hence the final calculation of function presences in different iterations can be further specified.

- Data fusion: The incorporation of different biological data in the research can follow certain framework of data integration (e.g. Chua et al, 2007). Thus the noise information and bias information can be diminished in certain extent.

- Classification method and parameter choice: Different classification methods have been applied for protein function prediction. However, the quality of the training data and the parameter settings of the classification model can be further improved by adopting data pre-processing methods and choosing parameters that have better correlation with prediction data.
Hierarchical prediction: The GO ontology has a hierarchical structure in which the leave terms are specific and the root terms are more general. Protein function prediction thus can exhibit different prediction confidences based on the level of the predicted function terms in the ontology tree. To be more precise and informative for biologist, the prediction can output the most confident levels of predicted functions in the GO structure.

There is significant potential lying in computational proteomic data analysis since tremendous biological data are generated not like ever since. However, the noisy data quality and the limited volume of certain types of data present challenges. Proper data processing techniques and prediction models with specialised parameters are the possible solutions.
Bibliography


Ge, H., Liu, Z., Church, G. M., and Vidal, M., “Correlation between transcriptome and interactome mapping data from Saccharomyces cerevisiae”. Nat. Genet. 29, 482-486,


Hindol Rakshit, Nitin Rathi, Debjani Roy, “Construction and Analysis of the Protein-Protein Interaction Networks Based on Gene Expression Profiles of Parkinson's Disease”. PLOS one, Vol. 9, Issue 8, August 2014.


M. Madan Babu, “Introduction to microarray data analysis”. Computational Genomics (Ed: R. Grant), Horizon Press, U.K.


Salwinski L, Miller CS, Smith AJ, Pettit FK, Bowie JU, Eisenberg D. “The Database of


Appendix

Pseudo code of the iterative model

Description: The pseudo code of our iterative approach. The code mainly consists of four parts, preliminary, protein function initialization, iterative function updating and prediction results selection.

Preliminary:
Input PPI data \( G(V, E) \), annotation file;
Construct the set of main target protein \( P_x \) and other unannotated target proteins in prediction domain as \( \Phi \);
\( P_l \in \Phi \);
\( t = \) iteration step;
\( FP_{x,t} = \) Set of functions of \( P_x \) from \( t \)-th iteration;
\( FP_{t,l} = \) record of predicted function of \( P_l \) from \( t \)-th iteration;
\( NP_{t,l} = \) Set of annotated proteins in prediction domain of \( P_l \);
\( F_{p,t} = \) Set of functions in \( \Phi \) after the \( t \) - th iteration, \( F_{p,t} = \bigcup FP_{t,l} \);
\( F \) = Set of all functions in the PPI network;
\( \text{Frequency}(function, protein) = \) Times of a function being assigned to protein \( \in \Phi \);
Prediction result of \( P_x = \emptyset \);

Protein function initialization
For function in \( F \):
\( \text{Frequency}(function, P_x) = 0 \);
End for
For \( P_l \) in \( \Phi \):
\( FP_{t,l} = \) Predicted by Neighbour Counting/FS weighted based on \( F_{p,0} \) and \( NP_{t,l} \);
For function in \( FP_{x,t} \):
\( \text{Frequency}(function, P_x) += 1 \)
End for
Update \( NP_{t,l} \) for every \( P_l \);
Update \( F_{p,0} \) to \( F_{p,1} \) using \( FP_{t,l} \);

Iterative function updating:
While not stable:
For \( P_l \) in \( \Phi \), in the \( t \) - th iteration:
\( FP_{t,l} = \) Predicted by NC/FS weighted based on \( F_{p,t-1} \) and \( NP_{t,l} \);
For function in \( FP_{x,t} \):
\( \text{Frequency}(function, P_x) += 1 \);
Update \( F_{p,t-1} \) to \( F_{p,t} \) using \( FP_{t,l} \);
Update \( NP_{t,l} \) for every \( P_l \);
\( t += 1 \);
End While after reaching stable status after \( M \)-th iteration

Prediction results selection:
For function in \( F \):
If \( \text{Frequency}(function, P_x) \geq \alpha \cdot \max_{f \in \bigcup_{i < t} FP_{x,i}} \text{Frequency}(f, P_x) \):
Prediction result of \( P_x \) is \( function \);
Return Prediction result of \( P_x \) and end for