ORTHOLOGOUS PAIR TRANSFER AND HYBRID BAYES METHODS TO PREDICT THE PROTEIN-PROTEIN INTERACTION NETWORK OF THE *ANOPHELES GAMBIAE* MOSQUITOES

By

Qiuxiang Li

SUBMITTED IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY
AT
IMPERIAL COLLEGE LONDON
SOUTH KENSINGTON, LONDON SW7 2AZ
2008

© Copyright by Qiuxiang Li, 2008

IMPERIAL COLLEGE LONDON
DIVISION OF
CELL & MOLECULAR BIOLOGY

The undersigned hereby certify that they have read and recommend to the Faculty of Natural Sciences for acceptance a thesis entitled "**Orthologous pair transfer and hybrid Bayes methods to predict the protein-protein interaction network of the *Anopheles gambiae* mosquitoes**" by **Qiuxiang Li** in partial fulfillment of the requirements for the degree of **Doctor of Philosophy**.

Dated: 2008

Research Supervisors: _____
Professor Andrea Crisanti

_____
Professor Stephen H. Muggleton

Examining Committee: _____
Dr. Michael Tristem

_____
Dr. Tania Dottorini

IMPERIAL COLLEGE LONDON

Date: **2008**

Author: **Qiuxiang Li**

Title: **Orthologous pair transfer and hybrid Bayes methods to predict the protein-protein interaction network of the *Anopheles gambiae* mosquitoes**

Division: **Cell & Molecular Biology**

Degree: **Ph.D.** Convocation: **March** Year: **2008**

Permission is herewith granted to Imperial College London to circulate and to have copied for non-commercial purposes, at its discretion, the above title upon the request of individuals or institutions.

Signature of Author

THE AUTHOR RESERVES OTHER PUBLICATION RIGHTS, AND NEITHER THE THESIS NOR EXTENSIVE EXTRACTS FROM IT MAY BE PRINTED OR OTHERWISE REPRODUCED WITHOUT THE AUTHOR'S WRITTEN PERMISSION.

THE AUTHOR ATTESTS THAT PERMISSION HAS BEEN OBTAINED FOR THE USE OF ANY COPYRIGHTED MATERIAL APPEARING IN THIS THESIS (OTHER THAN BRIEF EXCERPTS REQUIRING ONLY PROPER ACKNOWLEDGEMENT IN SCHOLARLY WRITING) AND THAT ALL SUCH USE IS CLEARLY ACKNOWLEDGED.

Table of Contents

Table of Contents **iv**

List of Tables **vii**

List of Figures **ix**

Abstract **xii**

Acknowledgements **xiv**

1 Introduction **1**
1.1 Motivations 1
1.1.1 Malaria Disease 1
1.1.2 Control Strategies 3
1.2 Sex Determination in *Drosophila melanogaster* 5
1.3 Sex Determination in *Anopheles gambiae* 6
1.4 Comparisons 6
1.5 Project Objectives 7
1.6 Project outlines 8
1.7 Organization of the Thesis 13

2 Background **14**
2.1 Protein-protein interaction 14
2.2 Domain-domain interaction 17
2.2.1 Orthologous clusters 18
2.2.2 Phylogenetic tree 25
2.2.3 Comparison of Orthologous cluster method and Phylogenetic tree technique 28

3 Related Work **34**
3.1 Attribute-value Based Learning and Limitations 34
3.2 Inductive Logic Programming 36
3.2.1 Introduction 36
3.2.2 Applications of Inductive Logic Programming 39
3.2.3 Inverse Entailment and PROGOL 41
3.3 Analogical Reasoning 53
3.3.1 Introduction 54
3.3.2 Applications of Analogical Reasoning 56
3.4 Data Set 57
3.5 Data Representation 57
3.6 Feature Selection 58
3.6.1 Global and Local Attributes 58
3.6.2 Combining the Global, Local and Relational Attributes 61
3.6.3 Feature Calculation 65
3.7 Background Knowledge 72
3.8 Experiments 73
3.8.1 Example of the Execution of PROGOL 73
3.8.2 Parameters Selection 77
3.8.3 Preliminary Test 78
3.9 Discussion 79
3.10 Summary 82

4 Materials and Methods **90**
4.1 Data 93
4.2 Feature Extraction 93
4.3 Building protein-protein interaction map with orthologous proteins transfer method 95
4.3.1 The Inparanoid database 96
4.3.2 The algorithm 96
4.4 Hybrid Bayes method 106
4.4.1 Introduction 106
4.4.2 Hybrid Bayes method - from protein-protein interaction to domain-domain interaction 109
4.4.3 Markov chain Monte Carlo 118
4.4.4 Hybrid Bayes method - from domain-domain interaction to protein-protein interaction 119
4.4.5 Details of the algorithm and domain detection 120

5 Results **125**
5.1 Orthologous protein transfer method 125
5.2 Hybrid domain-domain interaction method 129
5.3 The first voting machine 134
5.4 The second voting machine 137
5.5 Comparing with randomly generated network 140
5.6 *A. gambiae* protein-protein interaction prediction 151
5.6.1 Orthologous transfer method 151
5.6.2 Hybrid method and the first voting machine 151

6 Discussion and conclusions **162**
6.1 Discussions 162
6.1.1 Applying the super-domain concept 165
6.1.2 The domain interaction profile pairs method 166
6.1.3 Boosting the individual dataset 167
6.1.4 Validation of potential protein interactions 168
6.2 Conclusions 168

Bibliography **170**

List of Tables

2.1 **Male double sex protein sequences used to construct the phylogenetic tree** . . . 32
2.2 **Female double sex protein sequences used to construct the phylogenetic tree** . . . 33

3.1 **A list of different ILP systems** . . . 37
3.2 **PROGOL's simple set covering algorithm** . . . 47
3.3 **PROGOL's algorithm for searching the subsumption lattice.** 48
3.4 **Definitions to outline PROGOL's complexity** . . . 49
3.5 **Amino Acid Attributes and the Division of the Amino Acids Into Three Groups for Each Attribute** . . . 60
3.6 **Feature vectors and dimensions** . . . 62
3.7 **Background knowledge: Global Attributes 1** . . . 84
3.8 **Background knowledge: Global Attributes 2** . . . 85
3.9 **Background knowledge: Global Attributes 3** . . . 86
3.10 **Background knowledge: Global Attributes 4** . . . 87
3.11 **Background knowledge: Relational Attributes** . . . 88
3.12 **Positive Examples Covered by Rules: Inverse Frequency** . . . 89

4.1 **Proteins and domains contained** . . . 112

5.1 **Prediction results vs data from public databases** . . . 126
5.2 **Chi-square tests for *S. cerevisiae* protein interaction data** . . 132

5.3 **Examples of common protein interactions of *S. cerevisiae* predicted from several methods** 135
5.4 **Example 1 of common protein interactions of *Drosophila melanogaster* predicted from several methods** 136
5.5 **Example 2 of common protein interactions of *Drosophila melanogaster* predicted from several methods** 136
5.6 **Example 3 of common protein interactions of *Drosophila melanogaster* predicted from several methods** 137
5.7 **Example 4 of common protein interactions of *Drosophila melanogaster* predicted from several methods** 137
5.8 **The *A. gambiae* ATP-binding proteins identified with our experiments** 160

List of Figures

1.1 **World Malaria Situation**. Malaria is mainly endemic in tropical and subtropical regions. (white: no malaria; gray: isolated cases; orange,: malaria-endemic regions) 2
1.2 Somatic sex determination pathway of *D. melanogaster* 9
1.3 Project plan 9
1.4 Project flowchart 11

2.1 Overview of the Inparanoid algorithm. 21
2.2 Clustering of additional orthologs (in-paralogs). 22
2.3 Phylogenetic tree constructed from male dsx proteins 30
2.4 Phylogenetic tree constructed from female dsx proteins 31

3.1 An analogy problem 83

4.1 Overview of the orthologs transfer algorithm. 98
4.2 Clusters for two species from Inparanoid database 99
4.3 Known Drosophila interactions 99
4.4 A pair of interacting proteins from Drosophila 100
4.5 Highlighted clusters for two species from Inparanoid database 101
4.6 A pair of interacting proteins inferred from the above algorithm . . . 102
4.7 Protein interaction map for Anopheles 103
4.8 Sex-determination related proteins of *Drosophila melanogaster* 104
4.9 Sex-determination related proteins of *Anopheles gambiae* 105

4.10 Protein interaction pair and domains contained 113
4.11 Overview of the virtual domain algorithm. 121

5.1 Chi-square test method 130
5.2 Chi square values vs the hypothetical protein interaction space . . . 131
5.3 ROC curves with different prediction methods 139
5.4 γ=1.5, the connectivity distribution of a scale free network that follows power-law 144
5.5 γ=2.5, the connectivity distribution of a scale free network that follows power-law 145
5.6 γ=3.5, the connectivity distribution of a scale free network that follows power-law 146
5.7 γ=4.5, the connectivity distribution of a scale free network that follows power-law 147
5.8 γ=1.5, the connections as real vs. the connections as model are plotted. This is obtained by assuming all complexes having higher probability than the threshold are seen in the experiment. 149
5.9 γ=1.5, the total connectivity of a protein in the model network is plotted as a function of the real connectivity of that protein from the simulated network 150
5.10 γ=2.5, the connections as real vs. the connections as model are plotted. This is obtained by assuming all complexes having higher probability than the threshold are seen in the experiment. 152
5.11 γ=2.5, the total connectivity of a protein in the model network is plotted as a function of the real connectivity of that protein from the simulated network 153
5.12 γ=3.5, the connections as real vs. the connections as model are plotted. This is obtained by assuming all complexes having higher probability than the threshold are seen in the experiment. 154

5.13 γ=3.5, the total connectivity of a protein in the model network is plotted as a function of the real connectivity of that protein from the simulated network 155
5.14 γ=4.5, the connections as real vs. the connections as model are plotted. This is obtained by assuming all complexes having higher probability than the threshold are seen in the experiment. 156
5.15 γ=4.5, the total connectivity of a protein in the model network is plotted as a function of the real connectivity of that protein from the simulated network 157
5.16 ROC curves with different weights 159
5.17 Specificity and sensitivity of hybrid bayes for different thresholds . . 161

Abstract

Based on the published protein-protein interaction maps of five organisms and other public databases for domain-domain and protein-protein interactions, two new approaches are proposed to infer the protein-protein interaction network of the *Anopheles gambiae* (*A. gambiae*) mosquitoes.

Our main contributions are: i) Adopted an orthologous protein pair transfer method that has so far not been seen in literature ii) Proposed a new hybrid Bayes method; iii) Used voting machines at two levels of the combined classifier/predictor; iv) Used heterogeneous datasets as the training data; v) And finally, used the trained classifier to predict the protein interaction maps for *A. gambiae*, arguably one of a few least known organisms in terms of protein interaction mechanism.

With the first method, the orthologous and in-paralogous protein clusters are extracted for both species. The relations between two peer-to-peer proteins in the two species are identified so that the interactions in the *D. melanogaster* protein interaction maps are transferred to pairs of interacting proteins in *A. gambiae*.

The second strategy, namely the hybrid Bayes, is based on the domain composition of proteins, with which we utilize a probability model to build virtual domain-domain maps by integrating large-scale protein interaction data from five organisms, namely *Saccharomyces cerevisiae*, *Caenorhabditis elegans*, *Escherichia coli*, *Mus musculus* and *Drosophila melanogaster*.

For the hybrid Bayes method, once the virtual domain-domain interaction maps are constructed, we propose two ways to predict the protein-protein interaction maps.

These two methods are compared and then combined to form a voting machine to collectively decide a protein-pair's candidacy. The users could adjust the weights for different methods to flexibly control the output. Parameters are chosen through running different experiments on the training data set.

While both the orthologous cluster and hybrid Bayes methods produce encouraging results the second one predicts more protein-protein interaction than the first. Yet these two data sets share a very small fraction of common interactions. We adopt a second voting machine and calibrate the parameters with the putative protein interaction data. Those parameters for the voting machine are used to predict the protein-protein interaction maps of the *A. gambiae* and produces reasonably good results.

Acknowledgements

There are many people I wish to thank. First and foremost, I am deeply indebted to my supervisors, Prof. Andrea Crisanti and Prof. Stephen H. Muggleton, for their many suggestions, great efforts, unlimited patience and constant support. Working with them has been a continuous learning experience. They are always available to help point me in the right direction. If I were to list all of the help they have offered me I would probably still be sitting at my desk writing this section at the same time next day so I am not going to. I thank Prof. Crisanti for providing a fellowship to support my research. The ORS scholarship provided by the Universities UK is very much appreciated.

I also wish to express my gratitude to Dr. Flaminia Catteruccia for her guidance and sincere help since before starting the project.

My special thanks go to Prof. Michael J.E. Sternberg and Dr. Oliver Billker, my advisors, for their consistent encouragement and useful advices.

Finally, I wish to thank my family members for their continuous support.

Qiuxiang Li
September 28, 2007

Chapter 1

Introduction

1.1 Motivations

1.1.1 Malaria Disease

Malaria is the most significant insect-borne human tropical disease in the world. It causes high morbidity and mortality worldwide. Approximately 40% of the world's population and mainly those living in the poorest countries are at risk of malaria. The World Health Organization estimates that 300 to 500 million individuals are infected annually, resulting in 1 to 3 million deaths[1]. The disease is endemic in 100 countries of Africa, Asia and Latin America. Particulary the sufferers in Africa represent more than 90% of the total cases in the world. The majority of the patients are children under the age of 5 (Fig. 1.1[2]).

[1] http://www.who.int/mediacentre/factsheets/fs094/en/

[2] http://www.iamat.org/pdf/WorldMalariaRisk.pdf World Malaria Risk Chart, status as at March 15, 2004.

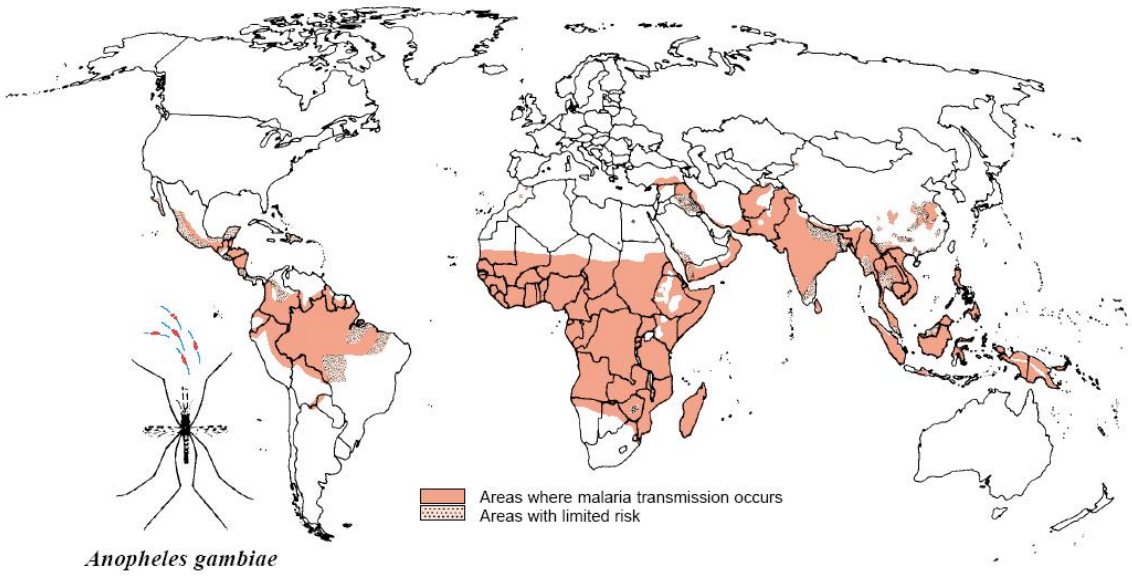


Figure 1.1: **World Malaria Situation**. Malaria is mainly endemic in tropical and subtropical regions. (white: no malaria; gray: isolated cases; orange,: malaria-endemic regions)

1.1.2 Control Strategies

So far the major Malaria control strategies include chemotherapy and/or vaccines to decrease the prevalence of the disease in the human host, or the use of insecticides to reduce the vector population. The mosquito vector represents a main target for Malaria control since the ability of a vector to survive is the single most important factor that determines the length of the infective period and therefore affects the rate of transmission of the disease. The recent global prevalence of malaria can be attributed to the development of Plasmodium resistance to anti-malarial drugs as well as vector resistance to residual insecticides. Drug pressure is implicated in resistance to anti-malarials [69]. The adoption of sub-standard or sub-therapeutic doses of the drugs further helps produce drug resistance through selection of resistant parasites, which leads to the rapid spread of multiple drug-resistant strains of P. falciparum. Although vaccines represent a promising future control strategy, the development has been very complicated due to various reasons. Among them are the rapidity of cell invasion, stage-specific immunity and variation in surface parasite antigens resulting in strain-specific immunity [204].

To date, the most successful control method used against malaria in reducing transmission are those aimed at controlling the vector population. The use of insecticides such as DDT was anticipated to eradicate malaria until mosquito resistance and behavioral changes such as avoidance tactics became widespread, preempting their future use. Moreover, insecticide-impregnated bednets have been a very effective method in preventing malaria transmission.

Most mosquito species which are vectors of human disease can be genetically transformed, including *Aedes aegypti*, *Culex quinquefasciatus*, *A. stephensi*, *A. gambiae*

and *A. albimanus*. In order to reduce the transmission of mosquito-borne diseases, biological control based on releasing genetically modified mosquitoes into natural populations has been proposed as a potential application of transformation technology. Such population replacement strategies are based on the introduction of manipulated mosquitoes carrying a transgene that blocks pathogen development/survival, thus producing a refractory phenotype.

Sterile insect technique (SIT) is another vector control strategy which is species-specific and involves the production of large numbers of sexually active, genetically sterile males by factories. The mechanism to reduce wild populations is that sterilized males could mate with wild-type virgin females, thus ensuring that no viable progeny are produced. Consequently larger numbers of males are needed than predicted by simple models in order to ensure that competitiveness is maintained [16]. One important thing is that the accidental presence of female mosquitoes must be avoided for such a strategy to be successful, as the existence of the female mosquitoes would reduce the efficiency and effectiveness of the process. The production of mass-produced males is most efficient through female-killing and genetic sorting, collectively called genetic sexing mechanisms (GSM). GSM is achieved by identifying and modifying the key steps in the genetic regulation of sex determination, in order to bring about sex-specific lethality [16]. One important GSM approach is through impairing female or improving male viability by introducing specific conditional lethal or advantageous genes.

Consequently, identification of the sex determination pathway in *A. gambiae* is important not only to obtain a greater understanding of the basic biology of the major malaria vector but as a potential application in GSM for the SIT.

1.2 Sex Determination in *Drosophila melanogaster*

Dipteran insects have developed a number of different sex-determining molecular systems. Nevertheless, the basic chain of events in these insects is the same, suggesting that different modes of sex determination have a common genetic and molecular basis [162, 148, 121, 140, 12, 104, 186].

Sex determination mechanisms in *Drosophila melanogaster* (*D. melanogaster*) have been extensively studied [12, 32, 82, 104, 121, 140, 162, 173, 174]. What has now become clear is that, in this organism, a primary genetic or environmental signal, which is different in males and females, leads to the differential expression of a key gene *sxl*. The activity of *sxl* controls, via a short cascade of subordinate genes, the activity of the sex differentiation genes that finally transform the initial sex determining molecular signal into alternative sexual phenotype [174]. In *D. melanogaster* the ratio of X-chromosomes to autosomes decides the sexual development of the organism. A ratio of 1.0 (2X:2A) induces female development, whereas both XY and X0 individuals, having a ratio of 0.5 (1X:2A), yield males [32]. The Y chromosome does not have a sex determining function however it is necessary for male fertility. Sex determination is accomplished through the complex interactions of genes that regulate a double switch involving the gene *sxl* and *dsx*. Sex-specific splice forms of *dsx* are generated by *tra* and *tra2* and they ultimately control a series of distinct downstream genes that determine the differentiation into adult female and male individuals [53]. The X dosage activation mechanisms of *sxl* and its role in sex determination seems unique to Drosophila whereas the structure and the function of *dsx* is conserved in all insects analyzed so far as well as in distantly related organisms [115]. The Drosophila sex determination pathway (Fig. 1.2) is illustrated in [174].

In *D. melanogaster*, a number of genes and signals form a complex pathway that acts in sex determination. Five genes play an essential role in this pathway. They are *sex-lethal* (*sxl*) [142, 7, 23], *transformer* (*tra*) [143, 90, 180], *transformer-2* (*tra2*) [6, 15, 24], *double-sex* (*dsx*) [7, 23, 29, 28, 37], and *intersex* (*ix*) [26, 62]. Many other genes also function in the pathway. They include *deadpan* [213], *emc* [117], *dissatisfaction* [57], *groucho* [149], *fl(2)d* [70], *hermaphrodite* (*h*er) [158, 157], *msl-2* [13], *lethal-2* [216], *daughterless* [38], *snf* [59, 169], *ovo* [83, 150], *runt* [47, 106], *sisterless-a* [52], *sisterless-b* [43, 187, 33] and *vir* [81, 172].

1.3 Sex Determination in *Anopheles gambiae*

In Anopheles mosquitoes, the vectors of human malaria, very little is known about the sex determination mechanisms. The completion of more than 90% of the genome sequence of *A. gambiae* [86, 126] has allowed us to ascertain the presence of *dsx, tra* and *tra2* in the mosquito genome thus indicating that the malaria vector may share this part of the sex determining pathway with *D. melanogaster*. A few preliminary comparisons have recently been reported in the literatures [45, 93, 97, 214]. However we are still far from elucidating the sex determination pathway in *Anopheles Gambiae* (*A. gambiae*).

1.4 Comparisons

In other fly species, like *Bombyx mori*, *Bactrocera tryoni*, *Megaselia scalaris*, and *Ceratitis capitata* [128, 171, 179, 191, 212], sex determination doesn't strictly follow

the same pathway as in *D. melanogaster*, nevertheless the research in these organisms has provided us with valuable data for reference.

There are a number of studies that compare sex determination mechanisms between *D. melanogaster* and other fly species. Those comparisons are with *Caenorhabditis* in [85]; with *Musca domestica* in [123]; and with *Ceratitis capitata* in [168, 116]. The knowledge gained through comparative studies on sex determination pathways in these organisms provides researchers with an invaluable tool for elucidating sex determination mechanisms in other organisms.

There are also quite a few comparative genome and proteome analyses of *A. gambiae* and other organisms. Among those are the comparative genomic analyses of *D. melanogaster* and *A. gambiae* [214, 19, 45, 93]. Severson et al. did a comparative genome analysis of the yellow fever mosquito *Aedes aegypti* with *D. melanogaster* and *A. gambiae* [176]. *A. gambiae* and *A. funestus* were also compared for inversions and gene order shuffling in [177].

1.5 Project Objectives

To step closer to the problem described in the previous section, this project aims to find a reliable way to infer the protein-protein interaction network of *A. gambiae*. the biologists then could use the protein-protein interaction network to find the sex determination pathway that provides SIT techniques to produce male-only mosquitoes in factory.

The research works we carrried out include:

1. The construction of the *D. melanogaster* sex determination pathway model with

Inductive Logic Programming methods;

2. The construction of the *A. gambiae* sex determination pathway from the *D. melanogaster* sex determination pathway model using Analogical Reasoning;

3. The orthologous and in-paralogous transfer method to predict protein-protein interaction network and its valuation;

4. The Bayes method to predict protein-protein interaction network and its valuation;

5. a voting machine to combine the proposed methods to collectively predict the protein-protein interaction network;

6. The practical application of the model to *A. gambiae*and validation.

1.6 Project outlines

The research was initially proposed to carry out in the following steps, this can also be seen from Figure 1.3.

1. Based on the knowledge derived from *D. melanogaster*, we plan to identify those genes which have sex-determination function in *A. gambiae*. Currently there are only several genes (*sxl, tra, tra2*, and *dsx*) that are known to have such function. The identification of these genes attributes to many biologists's hard work through the past decade. In order to identify other genes that are sex-determination related in *A. gambiae*, we have to find some non-biological ways, i.e., using machine learning methods to infer the rest genes based on

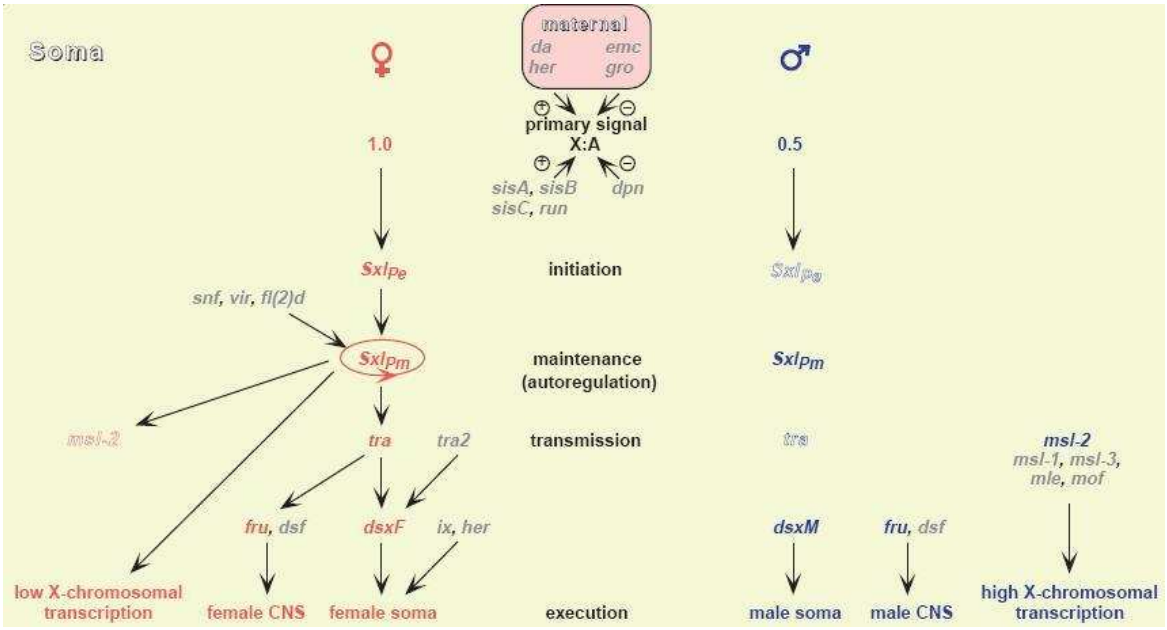


Figure 1.2: Somatic sex determination pathway of *D. melanogaster*

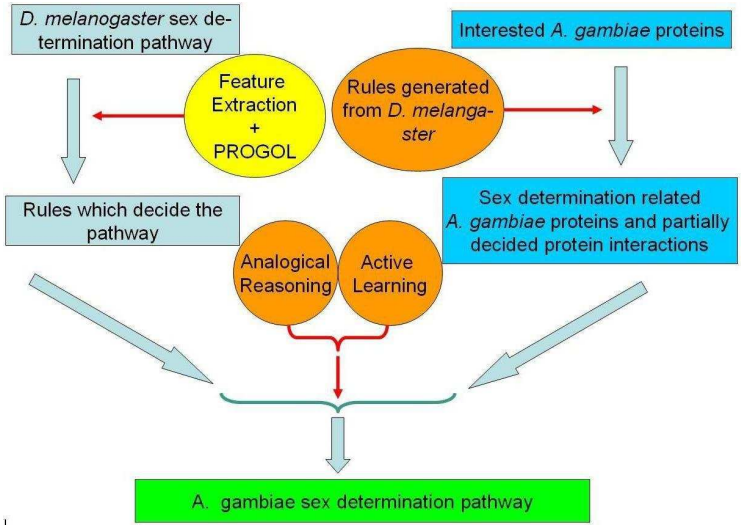


Figure 1.3: Project plan

existing knowledge to *A. gambiae*. Using an Inductive Logic Programming tool PROGOL, we try to infer a model that can map the gene sequences to the sex-determination function group. Ideally with this model (or rules in the case of PROGOL), we can accurately discriminate the sex determination related genes from a huge number of unlabelled genes. We name this as 'model1'. Next within the scope of the sex determination related genes, we plan to partition those genes with activation function apart from those with inhibition function. This partition criteria need also be learned by PROGOL to infer a model ('model2').

2. Our first plan was to directly infer the sex determination related genes in *A. gambiae* (with the known 90% whole genome sequence) using 'model1' together with use of Analogical Reasoning. Those identified sex determination related genes will be further classified by 'model2' into activation and inhibition group, also with the use of Analogical Reasoning. The *D. melanogaster* sex determination network architecture and weights (parameters) between the objects (genes) will be changed to suit the data from *A. gambiae*. Analogical Reasoning will play a key role through this step. (Figure 1.4). This is the most important step, which decides whether we can successfully identify the sex determination pathway in *A. gambiae*.

3. Unfortunately the models and rules generated by inductive logic programming were too general and hence not very useful to narrow our problem. That motivates us to tackle the problem from a different angle. Instead of trying to infer the sex-determination pathway, we decided to predict the protein-protein interaction network for *A. gambiae* and thus enable the biologists to do further selective experiments with the reduced candidates pool.

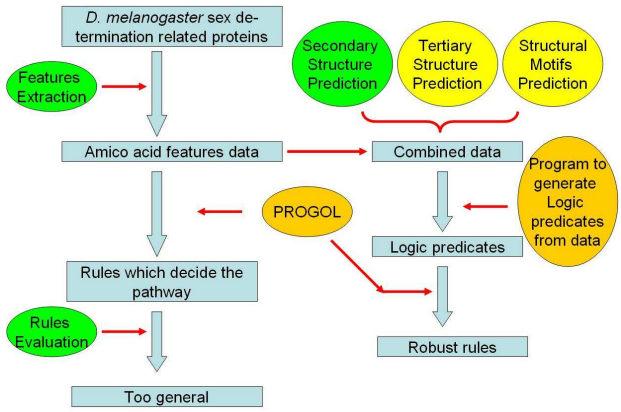


Figure 1.4: Project flowchart

4. We first proposed to use orthologous transfer method to do the protein-protein interaction prediction. By definition orthologos between 2 species have evolved from one single gene in their ancient common ancestor. Thus, orthologos are likely to have the same function in both species. Eukaryotic genes form large homologous families that cannot be classified by simple best-best hit methods. InParanoid is a fully automatic method for finding orthologs and in-paralogs between 2 species. Ortholog clusters in the InParanoid are seeded with a two-way best pairwise match, after which an algorithm for adding in-paralogs is applied. The method bypasses multiple alignments and phylogenetic trees, which can be slow and error-prone steps in classical ortholog detection. Still, it robustly detects complex orthologous relationships and assigns confidence values for in-paralogs. We later realized although this method can return good results but due to limitation of the database data, the inferred dataset are also incomplete. Phylogenetic tree, or evolutionary tree method provides an alternative way to

identify evolutionary relationships among various biological species or other entities, i.e., proteins, that are believed to have a common ancestor. It is easier to view the relationships from a tree-like structure; however, effective distance evaluation scoring methods are needed to quantitatively measure how similar two proteins are. It is also slow to construct such a tree for large dataset.

5. Thus we proposed a Bayes method to predict the protein-protein interaction network. Firstly, from some well-known species, we collect their protein-protein interaction data from database. We then try to break down to the domain level, and see what domains are included in each protein. Secondly, from the protein-protein interaction network, we go down one level and try to build a domain-domain interaction model using complex mapping relationship. Lastly, we again go up one level back to the protein level, but apply to an unknown protein interaction dataset. By now we would have inferred a protein-protein interaction network.

6. We further found different methods having different pros and cons. We then propose to combine the two methods using a voting machine, with different weighing adjustment factors. These methods are compared and considered relatively reliable to predict a protein-protein interaction network in the species we are interested in *A. gambiae*.

7. Finally, with different prediction methods, we try to infer the protein-protein interaction network in *A. gambiae*.

1.7 Organization of the Thesis

The remainder of this thesis is organized as follows.

Chapter 2 presents some background information. Chapter 3 presents some related work, that we have done with Inductive Logic Programming as our preliminary methods to probe this problem. It also describes the results of running PROGOL and gives a brief discussion. As the result from method is not ideal, we proceed to chapter 4 to propose some more effective methods.

Chapter 4 gives a detailed introduction of the methods we proposed in this project. Firstly the method of transferring annotations via orthologous and in-paralogous proteins is presented, followed by the approach to build protein-protein interaction maps with hybrid bayes method via virtual domain-domain interactions. Next we give an example showing how this algorithm works. Lastly the technique to detect domains in a protein is briefly described.

Chapter 5 describes the results obtained from two proposed methods, and proposes a voting machine to combine these two methods.

Finally chapter 6 discusses and summarizes the work and gives future work.

Chapter 2

Background

Transactions between proteins offer the mechanistic basis for much of the physiology and function of all organisms. Comprehensive analysis of the proteome of any organism is an extraordinary challenge. The development of genome-scale protein-interaction maps is a powerful first step towards addressing this challenge and provides the framework upon which a systems-biology understanding of cells and organisms can be developed.

2.1 Protein-protein interaction

Until recently, classical genetics and biochemistry were the main techniques used to investigate how organisms develop, reproduce, behave and age. But with the availability of complete genome sequences new approaches have been emerging. Complete sets of proteins 'proteomes' can be predicted from genome sequences and used to characterize protein functions globally. For example, through the large-scale identification

of physical protein-protein interactions [96], comprehensive protein interaction maps are being generated. These maps might help us understand the processes that control the biology of living organisms, such as transcript regulations, protein complexes, and sex-determination pathways.

Recent advances in high-throughput methods have led to the development of methods that use the protein interaction data generated from such techniques directly in the prediction of novel protein interactions. Again, these techniques do not rely on homology, but rather use the combination of experimental interaction data, along with specific features observed within the sequence itself in the process of predicting an interaction.

Particularly many works have been carried out focusing on the yeast protein-protein interaction networks. Attempts to explore the yeast protein interactome [92, 175, 111, 74, 210, 110, 153, 181] and analysis of protein-protein interactions [197, 56, 193, 201, 27, 144, 183, 119, 152, 98] have been reported by several groups.

Two research groups used yeast two-hybrid assays to generate 5,719 interactions between proteins of the yeast *Saccharomyces cerevisiae*. This allows study to the large-scale conserved patterns of interactions between protein domains. Using evolutionarily conserved domains defined in a protein-domain database called PFAM (http://PFAM.wustl.edu), Deng applies a Maximum Likelihood Estimation method to infer interacting domains that are consistent with the observed protein-protein

interactions [41].

Other than the work in yeast, Enright et al. show that 215 genes or proteins in the complete genomes of *Haemophilus influenzae* and *Methanococcus jannaschi* are involved in 64 unique fusion events solely based on sequence comparison [51]. Futschik et al present here a first comparative analysis of eight currently available large-scale maps with a total of over 10,000 unique proteins and 57,000 interactions included [60]. Zhong et al computationally integrated interactome data, gene expression data, phenotype data, and functional annotation data from three model organisms: *Saccharomyces cerevisiae*, *Caenorhabditis elegans*, and *Drosophila melanogaster*, and predicted genome-wide genetic interactions in C. elegans [215].

There are quite a few public databases for storing protein-protein interaction data. The most popular one is DIP (Database of Interacting Proteins) [211], a database that stores experimentally determined protein-protein interactions. DIP is intended to provide a comprehensive and integrated tool for browsing and efficiently extracting information about protein interactions and interaction pathways in biological processes. Beyond classified details of protein-protein interactions, the DIP is useful for understanding protein function and protein-protein relationships, investigating the properties of networks of interacting proteins, evaluating predictions of protein-protein interactions, and studying the evolution of protein-protein interactions. On

the pathway side, CellCircuits [118] database provides free access to molecular network models, designed to bridge the gap between databases of individual pairwise molecular interactions and databases of validated pathways.

2.2 Domain-domain interaction

The rapid developments of proteomics technology have enabled new proteins to be discovered at an unprecedented speed, which boost the research of high throughput experimental methods in detecting protein interactions and complexes. Such bottom-up, data-driven approach has resulted in data that may be uninformative or potentially problematic, requiring further validation and annotation. The *InterDom* [137] database focuses on providing supporting evidence for the detected protein interactions based on putative protein domain interactions. Adopting an integrative approach, *InterDom* extracts potential domain interactions by integrating data from multiple sources, ranging from domain fusions, protein interactions and complexes, to scientific literature.

The total number of putative domain interactions inferred in *InterDom* are 30,037. Among which, the number of putative domain interactions inferred from protein complexes (PDB) is 3,658, from protein interaction databases (BIND and DIP) is 25,741, and from domain fusion hypothesis is 2,768.

In [164], Riley et al describe domain pair exclusion analysis (DPEA), a method for

inferring domain interactions from databases of interacting proteins. DPEA features a log odds score, reflecting confidence that domains i and j interact. They analyzed 177,233 potential domain interactions underlying 26,032 protein interactions. In total, 3,005 high-confidence domain interactions were inferred from their research.

2.2.1 Orthologous clusters

There are two popular databases which store clusters of orthologous groups: Inparanoid and COGs.

Inparnoid

The Inparanoid program was developed at the Center for Genomics and Bioinformatics to address the need to identify orthologs while differentiating between inparalogs and outparalogs [163, 141]. Homologs which originate following gene duplications are called paralogs, a term in biology often mistakenly thought to apply to homologs within a genome. However, paralogy can exist between genes in different species, since gene duplication events could occur either before or after speciation. Thus, the term inparalogs indicates paralogs that appear via a gene duplication event after speciation, while outparalogs arise following a gene duplication before speciation. Since an outparalog pair ought to have a more diversified function than inparalogs, it is useful to distinguish between the two.

The Inparanoid algorithm is based on pairwise similarity scores which are calculated with the NCBI BLAST program. InParanoid detects best-best hits between sequences from two different organisms. In the first step 2 main orthologs form an orthologous group, thereafter other sequences are added to this group if they are closely related to one of the main orthologs. These members of the orthologous group are called in-paralogs. A confidence score is provided for each in-paralog member. The confidence value shows how closely related the in-paralog is to the main ortholog.

Orthologs are likely to have conserved function in both species because by definition orthologs between two species have evolved from one single gene in their ancient common ancestor. Another way to detect orthologs would be from phylogenetic trees. This has been widely used for single gene families, but these are slow and difficult to automate. Moreover, the preliminary steps like clustering genes into homologous families and creation of multiple alignments are needed. Also the topology of the phylogenetic tree is strongly dependent on choice of tree building method. Automatic clustering methods based on two-way best genome-wide matches on the other hand, have not been able to effectively distinguish in-paralogs from out-paralogs. The problem of in-paralog clustering is more important for analyzing eukaryotic genomes. Eukaryotic genes form large homologous families that cannot be classified by simple best-best hit methods. InParanoid is a fully automatic method for finding orthologs and in-paralogs between two species. Ortholog clusters in the InParanoid are seeded

with a two-way best pairwise match, after which an algorithm for adding in-paralogs is applied. The method bypasses multiple alignments and phylogenetic trees, which can be slow and error-prone steps in classical ortholog detection. Still, it robustly detects complex orthologous relationships and assigns confidence values for in-paralogs.

An Inparanoid cluster is seeded by a reciprocally bestmatching ortholog pair, around which inparalogs are gathered independently, while outparalogs are excluded. Here, seed-ortholog pair refers to the two seed members that are orthologous to each other, around which their inparalogs are clustered. Each is referred to the seed-inparalog when comparing against inparalogs in its own genome. Each member of the cluster receives an inparalog score, which reflects the relative distance to the seed-inparalog (1.0 = identical to the seed-inparalog; 0.0 = of equal distance to the seed-inparalog as the distance between the seed-ortholog pair) [163]. The confidence that the original seed-ortholog pair are true orthologs is estimated by sampling how often the pair is found as reciprocally best matches by a bootstrapping procedure. Bootstrap values were generated by counting how many times the seed-pair genes were each others best match in a sampling with replacement procedure that was applied to the original Blast alignment. In summary, an Inparanoid ortholog cluster contains a seed-ortholog pair with bootstrap confidence values, and a list of inparalogs with inparalog scores. The algorithm is illustrated in Fig. 2.1. The program requires two fasta format sequence files A and B with protein sequences. All-versus-all BLAST

search is run (Fig. 2.1, step 1) and sequence pairs with mutually best hits are detected (Fig. 2.1, step 2). Sequences from out-group species are optionally used to detect cases of selective loss of orthologs. The A-B sequence pairs are eliminated if either sequence A or sequence B scores higher to out-group sequence than they score to each other (Fig. 2.1, steps 3 and 4).

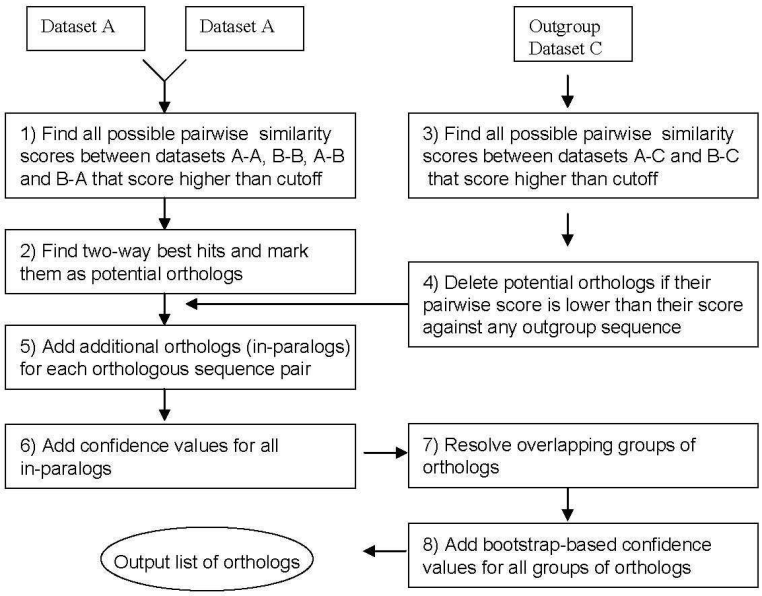


Figure 2.1: Overview of the Inparanoid algorithm.

Each circle in Fig. 2.2 represents a sequence from species A (black) or species B (red). Main orthologs (pairs with mutually best hit) are denoted C_a and C_b. Their similarity score is shown as S. The score should be thought of as reverse distance between C_a and C_b, higher score corresponding to shorter distance. The main assumption for clustering of in-paralogs is that the main ortholog is more similar to in-paralogs from the same species than to any sequence from other species. On this graph it means that all in-paralogs with score S or better to the main ortholog are inside the circle with diameter S that is drawn around the main ortholog. Sequences outside the circle are classified as out-paralogs. In-paralogs from both species A and B are clustered independently.

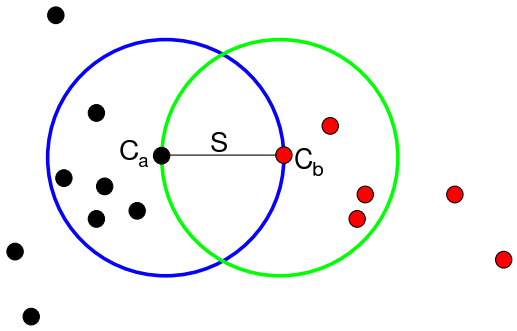


Figure 2.2: Clustering of additional orthologs (in-paralogs).

In the case that one-to-many or many-to-many types of orthology, several in-paralogs form a cluster, in which all proteins are orthologous to one or many proteins in another species. Despite that all in-paralogs are considered orthologs, some might

be very similar to the main ortholog, while others may be so dissimilar that they are almost excluded from the group. To characterize the feature of in-paralogs quantitatively, a confidence value is assigned to denote "how orthologous" a given sequence is. The score simply shows how far a given sequence is from the main ortholog of the same species on a scale between 0% and 100%. On this scale, 100% is assigned to the main ortholog and 0% is assigned to a sequence with the lowest similarity score needed to be included as in-paralog of a given group. A general formula to calculate this confidence value is:

$$\text{Confidence for } A_p = 100\%(scoreAA_p - scoreAB)/(scoreAA - scoreAB)$$

$$\text{Confidence for } B_p = 100\%(scoreBB_p - scoreAB)/(scoreBB - scoreAB)$$

where, A_p is an in-paralog from dataset A and B_p is an in-paralog from dataset B respectively. A is the main ortholog from dataset A while B is the main ortholog from dataset B; scoreAB is the similarity score between protein X and Y in bits. In the process of clustering, if several groups of orthologs are merged, they will retain their original confidence values. Hence, after merging two groups, a group of orthologs can contain more than one member with 100% confidence.

Clusters of orthologous groups

The availability of multiple, essentially complete genome sequences of prokaryotes and eukaryotes spurred both the demand and the opportunity for the construction of an evolutionary classification of genes from these genomes. Such a classification system

based on orthologous relationships between genes appears to be a natural framework for comparative genomics and should facilitate both functional annotation of genomes and large-scale evolutionary studies.

Clusters of Orthologous Groups of proteins (COGs) are described by comparing protein sequences encoded in complete genomes, representing major phylogenetic lineages [190]. Each COG consists of individual proteins or groups of paralogs from at least 3 lineages and thus corresponds to an ancient conserved domain. Recently a construction of clusters of predicted orthologs for 7 eukaryotic genomes was added to the database and named KOGs. The COG collection currently consists of 138,458 proteins, which form 4,873 COGs and comprise 75% of the 185,505 predicted proteins encoded in 66 genomes of unicellular organisms; While the eukaryotic orthologous groups (KOGs) include proteins from 7 eukaryotic genomes: three animals (the nematode *Caenorhabditis elegans*, the fruit fly *Drosophila melanogaster* and *Homo sapiens*), one plant, *Arabidopsis thaliana*, two fungi (*Saccharomyces cerevisiae* and *Schizosaccharomyces pombe*), and the intracellular microsporidian parasite *Encephalitozoon cuniculi*. The current KOG set consists of 4,852 clusters of orthologs, which include 59,838 proteins, or around 54% of the analyzed eukaryotic 110,655 gene products.

2.2.2 Phylogenetic tree

A phylogenetic tree, is a tree showing the evolutionary relationships among various biological species or other entities that are thought to have a common ancestor. In a phylogenetic tree, each node with descendants represents the most recent common ancestor of the descendants, and the edge lengths in some trees represent the time estimates.

In the area of phylogenetic inference, trees are used as visual displays that represent hypothetical, reconstructed evolutionary events. The tree in this case consists of:

- internal nodes which represent taxonomic units such as species or genes; the external nodes, those at the ends of the branches, represent living organisms.
- The lengths of the branches usually represent an elapsed time, measured in years, or the length of the branches may represent number of molecular changes (e.g. mutations) that have taken place between the two nodes. This is calculated is from the degree of differences when sequences are compared
- Sometimes, the lengths are irrelevant and the tree represents only the order of evolution.
- Finally the tree may be rooted or unrooted.

The problem in phylogenetic tree construction, is to find the tree which best describes the relationship between a set of objects. In any molecular phylogenetic reconstruction the following points need to be addressed.

- Molecular sequences
- Sequence alignment is the essential preliminary to tree reconstruction
- Converting the alignment data into a phylogenetic tree
- Assessing accuracy of a reconstructed tree
- Molecular clocks enable the time of divergence of ancestral sequences to be estimated

Phylogenetic trees are classied into three types.

Distance based methods

These methods require the definition of a distance function between objects. A tree is constructed so that the distances between any pair of objects can be mapped over the tree as accurately as possible.

Input: a set of sequences, each sequence corresponds to a species and an symmetric matrix with positive distances. Output: an edge weighted tree (rooted or unrooted) with the present day species at the leaves and the internal nodes representing ancestral species.

What we need is:

- a measure of the distance (and its variance) between species. This requires three things:
  - A model of evolution
  - Dayhoff matrices
  - Dynamic programming
- a method for constructing good initial topologies
- a method of fitting the distances between sequences to a tree topology
- a local search of this topology for better topologies

Character based methods

Under this classification, species are grouped according to the maximum number of matching characters. Instead of reducing all the individual variations between sequences to a single value (namely the distance in the distance based methods), they are treated separately and called "characters". A character is a heritable trait possessed by an organism. Characters can be derived from observable properties or can be derived from subsequences or even proteins or DNA. Characters are usually described in terms of their states. When proteins are used there are 20 possible states

per position (character), when DNA is used there are 4 states. This construction is usually done via parsimony.

Assume that the characters are positions in an amino acid sequence and that the states are the amino acids themselves. Another state called a gap is allowed, which is represented by a "-". Parsimony selects the phylogeny that minimizes the number of evolutionary changes for a data set. This minimum number of changes is the optimization criterion. Parsimony could be done at the DNA level. If it is done at the amino acid level, a weighted version of parsimony should be used, because some amino acids are known to be similar to others, i.e., can be exchanged and nothing much will happen.

Probabilistic methods

This classification is based on the likelihood (probability) of a certain tree explaining the set of objects. A tree, which somehow maximizes this likelihood, is selected.

2.2.3 Comparison of Orthologous cluster method and Phylogenetic tree technique

Phylogenetic tree can show a clear view of the relationships of many species or proteins in a big picture. While inparanoid database and COGs are obviously inferior in this regard.

One obvious disadvantage of Phylogenetic tree is it is very slow to construct for very large dataset. Determining the distances between protein/DNA sequences for a large data set can take a long time. Considering we need to build a phylogenetic tree for 30,000 protein sequences, the computer might spend a lot of time doing multi-sequence alignment. Without some parallel hardware support or algorithm improvement, the brute-force method might take forever to finish.

Another disadvantage of using Phylogenetic tree is the result is highly dependent on the seeding of the construction process and very sensitive to the sequence. To illustrate this, we give an example from our previous experiments with doublesex. doublesex (dsx) is unusual among the known sex-determination genes of *Drosophila melanogaster* in that functional homologs are found in distantly related species. In flies, dsx occupies a position near the bottom of the sex determination hierarchy. It is expressed in male- and female-specific forms and these proteins function as sex-specific transcription factors. For *Anophele gambiae*, the situation is similar. We use the male-specific form of dsx protein of *Drosophila melanogaster*, *Anophele gambiae*, together with *Bombyx mori*, *Megaselia scalaris*, *Musca domestica*, *Ceratitis capitata*, *Bactrocera tryoni*, *Bactrocera oleae*, and produced Fig. 2.4 and Fig. 2.3. We can see the slight sequence change in dsx proteins lead to significantly different order in the phylogenetic tree. The original sequences are shown in Table 2.1 and Table 2.2 respectively. The program used to generate the phylogenetic tree is Clustalw.

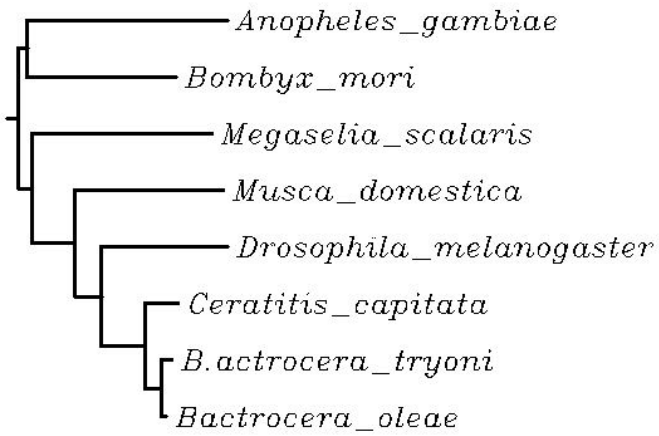


Figure 2.3: Phylogenetic tree constructed from male dsx proteins

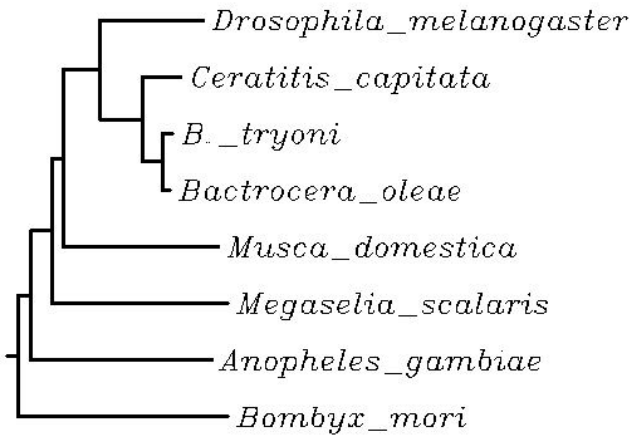


Figure 2.4: Phylogenetic tree constructed from female dsx proteins

Table 2.1: **Male double sex protein sequences used to construct the phylogenetic tree**

>Anopheles_gambiae Male; MVSQDRWAEA MSDSGYDSRT DGNGASSSCN NSLNPRTPPN CARCRNHGLK IGLKGHKRYC KYRTCHCEKC CLTAERQRVM ALQTALRRAQ TQDEQRALNE GEVPPEPVAN IHIPKLSELK DLKHNMIHNS QTRSFDCDSS TGSMASAPGT SSVPLTIHRR SPGVPHHVAE PQHLGATHSC VSPEPVNLLP DDELVKRAQW LLEKLGYPWE MMPLMYVILK SADGDVQKAH QRIDEGKRTI KTYEALVKSS LDPNSDRLTE DDEDENISVT RTNSTIRSRS SSLSRSRSCS RQAETPRADD RALNLDTKSK PSTSSSSGTG CDRDDGDCIT FDDSASVVRA THASRSATRM SRGRSRSQTK RYSQTVESTN APSRSPGPDE EPSVYKSLAE AASKMARSFI PAREPEDLHT TTHKSPERED NPSQPYEAYL ESVRRSKKSF PHKDAEGVTE SAEDCYDKEK EHRIPYSLPK STFDRLDLLK KPNGLPFPMY KYNELEANNF PLPLLLPGLE AVNRTLYTAH FPTHLLPSSL YPPVSSESTT APIFHTHFLG YQPQMQLPHV EPFYRKEQQQ QQLQQTLAEP KEQTTSSSPS NNRLTPPKGT FFYASAVENS LTAHQASIAT IH
>Drosophila_melanogaster Male; MVSEENWNSD TMSDSDMIDS KNDVCGGASS SSGSSISPRT PPNCARCRNH GLKITLKGHK RYCKFRYCTC EKCRLTADRQ RVMALQTALR RAQAQDEQRA LHMHEVPPAN PAATTLLSHH HHVAAPAHVH AHHVHAHHAH GGHHSHHGHV LHHQQAAAAA AAAPSAPASH LGGSSTAASS IHGHAHAHHV HMAAAAAASV AQHQHQSHPH SHHHHHQNHH QHPHQQPATQ TALRSPPHSD HGGSVGPATS SSGGGAPSSS NAAAATSSNG SSGGGGGGGG GSSGGGAGGG RSSGTSVITS ADHHMTTVPT PAQSLEGSCD SSSPSPSSTS GAAILPISVS VNRKNGANVP LGQDVFLDYC QKLLEKFRYP WELMPLMYVI LKDADANIEE ASRRIEEARV EINRTVAQIY YNYYTPMALV NGAPMYLTYP SIEQGRYGAH FTHLPLTQIC PPTPEPLALS RSPSSPSGPS AVHNQKPSRP GSSNGTVHSA ASPTMVTTMA TTSSTPTLSR RQRSRSATPT TPPPPPPAHS SSNGAYHHGH HLVSSTAAT
>Ceratitis_capitata Male; MVSEDNWNSD TMSDSDIHDS KADACGGASS SSGSSISPRT PPNCARCRNH GLKITLKGHK RYCKFRYCTC EKCRLTADRQ RVMALQTALR RAQAQDEQRV LQIHEVPPGV HAPAALLNHH HLHHHHHLNP NHHADCSSLA RRCHHHITTA IRSPPHAELG SGGGGLAGGI GSAITSVPVS APPPEHHMTT VPTPAQSLEG SSDTSSPSPS STSGAALPIS VVGRKPSLHP NGVHMPLAQD VFLEHCQKLL EKFRYPWEMM PLMYVILKDA GADIEEASRR IEEAKQIVNQ TISLHWMDRQ LYYNYYSSAA LVNTVPTYFP YPIAIGSNGL LTSQFSHLTA SIDRRRLEQP TLSRMPPSPS KPSRPASILS DTMSPPAT
>B.actrocera_tryoni Male; MVSEDSWNSD TIADSDMRDS KADVCGGASS SSGSSISPRT PPNCARCRNH GLKITLKGHK RYCKFRFCTC EKCRLTADRQ RVMALQTALR RAQAQDEQRV LQIHEVPPVV HGPTALLNHH HLHHHHHLNQ NHHASAAAAA AAAAAHHHIS TAIRSPPHAE HGGGNVSSSG GIAGGIGSAI TSVPGSVPPP EHHMTTVPTP AQSLEGSSDT SSPSPSSTSG AVLPISVVGR KPSLHPNGVN IPLAQDVFLE HCQKLLEKFR YPWEMMPLMY VILKDAGADI EEASRRIEEA KRIVNQTISL HWMDRQLYYN YYSSAALVNT PPTYFPYPIA IGSNGLLTSH FSHLTASMRP PSPEQPTLSR TPPSPSKPSR PGSILSETMS PPAAATNLPS SATAAAAT
>Bactrocera_oleae Male; MVSEDNWNSD TMSDSDMHDS KADVCGGASS SSGSSISPRT PPNCARCRNH GLKITLKGHK RYCKFRYCTC EKCRLTADRQ RVMALQTALR RAQAQDEQRV LQIHEVPPVV HGPTALLNHH HLHHHHHLNQ NHHASAAAAA AAAAAHHHIS TAIRSPPHAE HGGGNVSSGG NGGIAGGIGS GITSVSGSVP PPEHHMTTVP TPAQSLEGSS DTSSPSPSST SGAVLPISVV GRKPSLHPNG VNIPLAQDVF LEHCQKLLEK FRYPWEMMPL MYVILKDAGA DIEEASRRIE EAKRIVNQTI SLHWMDRQLY YNYYSSAALV NTPPTYFPYP IAIGSNGLLT SHFSHLTASI RPPSPEQPTL SRTPPSPSKP SRPGSILSET MSPPAAATSL TSSATAAAAT
>Musca_domestica Male; MVSEDSNWHS SDTMSDTDMH DSKNDICGGA SSSSGSSGTP RTKPNCARCH NHGLKIKLKG HKRYCKYRFC NCEKCRLTAD RQRVMALQTA LRRAQQQDEA RILQMHEVPP VVHPPTALLN AHHHHHHPLP HHITQQLHHH PHHPHPHLVD VSAVAAAAAA GVGVGPVPPH HIAAAAIPTI RSPPHSDHSA NGGGGGGGGG GGGGGSGSGG GGGGSAGGGS NGGGGGVGPS SSSMNGMASS SSAASSSTAP PHHTPPDHTH HHHHHHHPHP HLVSVPPTAQ SVDSSCDSSS PSPSSTSGVA VPVLVPNRKP NPEQQQNGAD MSIDLILDYC QKLIEKFGYP WEMMPLMYVI LKDAGVDIDE ASKRIEEAIQ LFKQYDSLIS IYDGHEWRSK ASLKRKAESG ARNAECDETT KRMRIEATEH LNQLTQTYYN YQRYAALPPV YWGYPSIQFG RAVWTELPNP NFAAIIPPHL AATTPDGPQS LSRRSPSPFK NSRPSSSLGS ESTTVTSLPT PGVLAAAAAA AAAAAAT
>Megaselia_scalaris Male; MVSDWQSDTM SEADCEQKGD ICGGASSSSG SSASPRTPPN CARCRNHSLK IALKGHKRYC KYRYCDCEKC RLTADRQKIM AAQTALRRAQ AQDESRPLSA GEIPATIHPA QYTLMQINSQ PYPVVHPHHH IAHNNHNHHV NQHHPHHIMH NNQPHLHHQV TAVTSSGGIS KSPVEHNPHQ ITVPTPAQSL EGSRDSSSAS PSSTSNGGAV AAPGSSAIVP VKKVGAPNGS TSTGIQKESL LDCCHRLLEQ FRFPFEMMPL MYVILKSVDD EEEASRLISE GLHITEPRLR AYRNYIALMY GITLPCYPYI PFSNLSYFGL TSNTSGPITD SPTNLSVSNN NDSNPVAIMN STPSTMISHN NTSSRGSPPP SLLPPTANRS HSPIFDLSAH RQSLQLSQED SRKEVEVNVH RFHRNDQEKL AFNRELSPDH KRLLDSQVTI NHEHEGSRKR RLESRSPSIE EQPQFLKRMY GFQPVYDLST HRPPLRSSQE ECRKEEEELN VHRFRRYAQE KLAFNGQETQ AAINHEHELK MRESRKRHHE SRSPSIDEQS QKKICLSPPV IRSDSTDVER GSP
>Bombyx_mori Male; MVSMGSWKRR VPDDCEERSE PGASSSGVPR APPNCARCRN HRLKIELKGH KRYCKYQHCT CEKCRLTADR QRVMAKQTAI RRAQAQDEAR ARALELGIQP PGMELDRPVP PVVKAPRSPM IPPSAPRSLG SASCDSVPGS PGVSPYAPPP SVPPPPTMPP LIPTPQPPVP SETLVENCHR LLEKFHYSWE MMPLVLVIMN YARSDLDEAS RKIYEGYWMM HQWRLQQYSL CYGALELSAR KDVAALCCLR DTCWRPRSRR VWCPSS

Table 2.2: **Female double sex protein sequences used to construct the phylogenetic tree**

>Anopheles_gambiae female; MVSQDRWAEA MSDSGYDSRT DGNGASSSCN NSLNPRTPPN CARCRNHGLK IGLKGHKRYC KYRTCHCEKC CLTAERQRVM ALQTALRRAQ TQDEQRALNE GEVPPEPVAN IHIPKLSELK DLKHNMIHNS QTRSFDCDSS TGSMASAPGT SSVPLTIHRR SPGVPHHVAE PQHLGATHSC VSPEPVNLLP DDELVKRAQW LLEKLGYPWE MMPLMYVILK SADGDVQKAH QRIDE GQAVV NEYSRLHNLN MFDGVELRNT TRQSG
>Drosophila_melanogaster female; MVSEENWNSD TMSDSDMIDS KNDVCGGASS SSGSSISPRT PPNCARCRNH GLKITLKGHK RYCKFRYCTC EKCRLTADRQ RVMALQTALR RAQAQDEQRA LHMHEVPPAN PAATTLLSHH HHVAAPAHVH AHHVHAHHAH GGHHSHHGHV LHHQQAAAAA AAAPSAPASH LGGSSTAASS IHGHAHAHHV HMAAAAAASV AQHQHQSHPH SHHHHHQNHH QHPHQQPATQ TALRSPPHSD HGGSVGPATS SSGGGAPSSS NAAAATSSNG SSGGGGGGGG GSSGGGAGGG RSSGTSVITS ADHHMTTVPT PAQSLEGSCD SSSPSPSSTS GAAILPISVS VNRKNGANVP LGQDVFLDYC QKLLEKFRYP WELMPLMYVI LKDADANIEE ASRRIEE GQY VVNEYSRQHN LNIYDGGELR NTTRQCG
>Ceratitis_capitata female; MVSEDNWNSD TMSDSDIHDS KADACGGASS SSGSSISPRT PPNCARCRNH GLKITLKGHK RYCKFRYCTC EKCRLTADRQ RVMALQTALR RAQAQDEQRV LQIHEVPPGV HAPAALLNHH HLHHHHHLNP NHHADCSSLA RRCHHHITTA IRSPPHAELG SGGGGLAGGI GSAITSVPVS APPPEHHMTT VPTPAQSLEG SSDTSSPSPS STSGAALPIS VVGRKPSLHP NGVHMPLAQD VFLEHCQKLL EKFRYPWETM PLMYVILKDA GADIEEASRR IEE GQHVVNE YSRQHNLNIF DGGELRSTTR QCG
>B._tryoni female; MVSEDSWNSD TIADSDMRDS KADVCGGASS SSGSSISPRT PPNCARCRNH GLKITLKGHK RYCKFRFCTC EKCRLTADRQ RVMALQTALR RAQAQDEQRV LQIHEVPPVV HGPTALLNHH HLHHHHHLNQ NHHASAAAAA AAAAAHHHIS TAIRSPPHAE HGGGNVSSSG GIAGGIGSAI TSVPGSVPPP EHHMTTVPTP AQSLEGSSDT SSPSPSSTSG AVLPISVVGR KPSLHPNGVN IPLAQDVFLE HCQKLLEKFR YPWEMMPLMY VILKDAGADI EEASRRIEE G QHVVNEYSRQ HNLNIYDGGE LRSTTRQCG
>Bactrocera_oleae female; MVSEDNWNSD TMSDSDMHDS KADVCGGASS SSGSSISPRT PPNCARCRNH GLKITLKGHK RYCKFRYCTC EKCRLTADRQ RVMALQTALR RAQAQDEQRV LQIHEVPPVV HGPTALLNHH HLHHHHHLNQ NHHASAAAAA AAAAAHHHIS TAIRSPPHAE HGGGNVSSGG NGGIAGGIGS GITSVSGSVP PPEHHMTTVP TPAQSLEGSS DTSSPSPSST SGAVLPISVV GRKPSLHPNG VNIPLAQDVF LEHCQKLLEK FRYPWEMMPL MYVILKDAGA DIEEASRRIE E GQHVVNEYS RQHNLNIYDG GELRSTTRQC G
>Musca_domestica female; MVSEDSNWHS SDTMSDTDMH DSKNDICGGA SSSSGSSGTP RTKPNCARCH NHGLKIKLKG HKRYCKYRFC NCEKCRLTAD RQRVMALQTA LRRAQQQDEA RILQMHEVPP VVHPPTALLN AHHHHHHPLP HHITQQLHHH PHHPHPHLVD ASAVAAAAAA GVGVGPVPPH HIAAAAIPTI RSPPHSDHSA NGGGGGGGGG GGGGGSGSGG GGGGSAGGGS NGGGGGVGPS SSSMNGMASS SSAASSSTAP PHHTPPDHTH HHHHHHHPHP HLVSVPPTAQ SVDSSCDSSS PSPSSTSGVA VPVLVPNRKP NPEQQQNGAD MSIDLILDYC QKLIEKFGYP WEMMPLMYVI LKDAGVDIDE ASKRIEE GQH VVNEYSRQHN LNIYDGCELR CATRQCG
>Megaselia_scalaris female; MVSDWQSDTM SEADCEQKGD ICGGASSSSG SSASPRTPPN CARCRNHSLK IALKGHKRYC KYRYCDCEKC RLTADRQKIM AAQTALRRAQ AQDESRPLSA GEIPATIHPA QYTLMQINSQ PYPVVHPHHH IAHNNHNHHV NQHHPHHIMH NNQPHLHHQV TAVTSSGGIS KSPVEHNPHQ ITVPTPAQSL EGSRDSSSAS PSSTSNGGAV AAPGSSAIVP VKKVGAPNGS TSTGIQKESL LDCCHRLLEQ FRFPFEMMPL MYVILKSVDD EEEASRLISE GQYAVNEYSR QHNLNIFDGG ELRSQSRQCG
>Bombyx_mori female; MVSMGSWKRR VPDDCEERSE PGASSSGVPR APPNCARCRN HRLKIELKGH KRYCKYQHCT CEKCRLTADR QRVMAKQTAI RRAQAQDEAR ARALELGIQP PGMELDRPVP PVVKAPRSPM IPPSAPRSLG SASCDSVPGS PGVSPYAPPP SVPPPPTMPP LIPTPQPPVP SETLVENCHR LLEKFHYSWE MMPLVLVIMN YARSDLDEAS RKIYE GKMIV DEYARKHNLN VFDGLELRNS TRQKMLEINN ISGVLSSSMK LFCE

Chapter 3

Related Work

Our research first attempted to decipher the sex-determination pathway of *A. gambiae* directly using Inductive Logic Programming and Analogical Reasoning.

3.1 Attribute-value Based Learning and Limitations

Machine learning concerns with the question of how to construct computer programs that automatically improve with experience [125]. Many successful machine learning applications have been developed in the past three decades of research, ranging from programs that can recognize a human being's face, to programs that can translate one natural language into another.

Earlier work in machine learning focused on tasks that involve the learning of classification functions from data represented by a vector of attributes and their

values, i.e., attribute-value representation [159]. For example, a system could learn to classify text documents collected from the web, represented as word frequency, document frequency and invert document frequency. Another example would be the learning of association rules that predict the products a customer would purchase based on the customer's "attributes" like salary, gender, age, career, hobbies, and the customer's prior purchasing behavior. Most of these earlier work can be summarized under the title "attribute value learning" in which the representation of an object (i.e. training example) is in the form of a vector of values (one for each attribute). One popular approach is learning with neural networks in which the attribute values are numerical. Yet, another popular approach is "symbolic machine learning" in which attribute values can be either numerical or simply a symbol.

Attribute value learning is called propositional learning since these systems find expressions equivalent to sentences in propositional logic. Although attribute value learning is very useful in most of the situations, it is not suitable in some cases if we are are facing tasks that require a more complicated representation framework than that of propositional logic. One example is data mining with multi-relational databases which requires the use of variables and predicates to capture relational knowledge embedded in the relational tables [49]. Unfortunately, none of these features is available in propositional logic.

3.2 Inductive Logic Programming

3.2.1 Introduction

As we mentioned in the previous section, Relational learning (RL) [49] concerns learning from multiple relational tables that are richly connected. The most widely studied methods for inducing relational patterns are those in inductive logic programming (ILP) [129, 107, 139]. ILP [132, 107] aims to overcome some limitations of attribute-value based learning algorithms by using Horn clause first-order logic as the representation language for the hypotheses. It stands at the intersection of machine learning and logic programming. ILP has been extensively researched over the past two decades, which produced two major approaches: 1) top-down and 2) bottom-up. The former is characterized by searching the hypothesis space in a general to specific manner [160] while the latter in a specific to general order [20].

Bottom-up approaches were pioneered by Stephen Muggleton who started the first workshop for Inductive Logic Programming in 1991, which became popular within the machine learning community in Europe. Quite a few useful ILP systems have been developed so far, which were also applied to many interesting areas. The top-down approach started by Ross Quinlan who developed the well known learning system C4.5 [159], from which the basic approach was extended to handling Horn clause logic in the system called Foil [160]. Foil was applied to many interesting

research problems like text categorization, for instance. Today, ILP has already become a well established discipline within machine learning with many different areas of useful applications, even started merging with other areas of research like probabilistic reasoning [103]. A list of important ILP systems are shown in Table 3.1.

Table 3.1: **A list of different ILP systems**

ILP System	**Author(s)**	**Time**	**Reference**	**Approaches Adopted**
MIS	Shapiro	1983	[160]	top-down, incremental, non-heuristic
CIGOL	Muggleton & Buntine	1988	[131]	bottom-up (inverting resolution), incremental, compression
FOIL	Quinlan	1990	[160]	top-down, non-incremental, information-gain
GOLEM	Muggleton & Feng	1990	[132]	bottom-up, non-incremental, compression
LINUS	Lavrac, Dzeroski & Grobelnik	1991	[108]	transformation to attribute-value learning
PROGOL	Muggleton	1995	[136]	hybrid, non-incremental, compression

ILP is the study of learning methods for data and rules that are represented in first order predicate logic. Predicate logic allows for quantified variables and relations and can represent concepts that are not expressible using examples described as feature vectors. A relational database can be easily translated into first-order logic and be used as a source of data for ILP. As an example, consider the following rules, written in Prolog syntax, that define the uncle relation:

$uncle(X, Y) :- brother(X, Z), parent(Z, Y).$ $uncle(X, Y) :- husband(X, Z), sister(Z, W), parent(W, Y).$ The goal of ILP is to infer rules of this sort given a

database of background facts and logical definitions of other relations [129, 107]. For example, an ILP system can learn the above rules for uncle (the target predicate) given a set of positive and negative examples of uncle relationships and a set of facts for the relations parent, brother, sister, and husband (the background predicates) for the members of a given extended family, see below.

Facts:

$parent(Chandler, Ryne), parent(Monica, Ryne),$

$parent(Rachel, Emma), parent(Ross, Emma), brother(Ross, Monica),$

$sister(Monica, Ross), husband(Ross, Rachel), husband(Chandler, Monica),$

$sister(Phoebe, Monica), brother(Joy, Ross), brother(Joy, Monica).$

Positive Examples:

$uncle(Ross, Ryne), uncle(Chandler, Emma),$

$uncle(Joy, Emma), uncle(Joy, Ryne).$

Negative Examples:

$uncle(Ross, Monica), uncle(chandler, Ross),$

$uncle(Monica, Rachel), uncle(Rachel, Chandler),$

$uncle(Rachel, Monica), uncle(Phoebe, Emma), uncle(Phoebe, Ryne).$

Alternatively, rules that logically define the brother and sister relations could be supplied and these relationships inferred from a more complete set of facts about only the "basic" predicates: *parent*, *spouse*, and *gender*.

3.2.2 Applications of Inductive Logic Programming

ILP in Molecular Biology

One of the interesting applications of ILP is, perhaps, in the now rapidly growing discipline called bioinformatics. In ILP, all the expressiveness is in the features that the body construction is essentially propositional and every ILP system does constructive induction. The feature construction is a discovery task.

ILP has been extensively used in many molecular biology related tasks. Among them are structure activity relationships and pharmacophore discovery [188, 58]; predicting of the mutagenicity of a set of 230 aromatic and heteroaromatic nitro compounds [185]; induction of enzyme classes from biological databases [135]; discovering of rules governing the three-dimensional topology of protein structure [194]; automated discovery of structural signatures of protein fold and function [195, 196]; uncovering structural principles describing protein fold space [35]; quantitative predictions of biological activity [184]; inferring of the metabolic pathway [9]; and robotic system to automate the scientific process [105].

Let's see two examples of applying ILP in predicting the Mutagenicity of Chemical Compounds. One background information is 42 out of the 230 compounds investigated in this project are regression unfriendly by classical regression methods.

By using ILP, Srinivasan and Muggleton derived a single rule with an accuracy of 88% estimated from a leave-one-out validation [185]. The rule generated discovers a

new fact that the presence of a five-membered aromatic carbon ring with a nitrogen atom linked by a single bond followed by a double bond indicates mutagenicity [21]. Lavrač and Flach ran LINUS on the 42 regression-unfriendly molecules, using a non-determinate background theory consisting of all 57 first order features with one utility literal concerning atoms without the bond information. Using CN2, the following rules were generated [109]:

$$mutagenic(M, false) :- not(has_atom(M, A), atom_type(A, 21)), logP(M, L), \\ L > 1.99, L < 5.64.$$

$$mutagenic(M, false) :- not(has_atom(M, A), atom_type(A, 195)), lumo(M, Lu), \\ Lu > -1.74, Lu < -0.83, logP(M, L), L > 1.81.$$

$$mutagenic(M, false) :- lumo(M, Lu), Lu > -0.77.$$

$$mutagenic(M, true) :- has_atom(M, A), atom_type(A, 21), lumo(M, Lu), Lu < -1.21.$$

$$mutagenic(M, true) :- logP(M, L), L > 5.64, L < 6.36.$$

$$mutagenic(M, true) :- lumo(M, Lu), Lu > -0.95, logP(M, L), L < 2.21.$$

Where *atom_type* denotes the type of an atom, $logP$ is log of the compound's octanol/water partition coefficient (hydrophobicity), and $lumo$ is the energy of the compound's lowest unoccupied molecular orbital. The first clause can be interpreted as "if a compound doesn't contain an atom of type 21, and the log value of hydrophobicity of this compound lies between 1.99 and 5.64, then the compound is not possible to be mutagenic." The rest five clauses can be explained in a similar way.

ILP in Other Fields

ILP has also been extensively used in other fields like Engineering, Environmental Sciences and Natural Language Processing. Among them is spatial data mining in Geographic database [156]. Dzeroski has applied ILP in detecting traffic problems [48]. There are also applications in Intrusion Detection through Behavioral Data [75]. Dolsak and Muggleton used ILP in finite element mesh design [44]. Several applications have also been reported in natural language processing [127, 39]. Van Baelen and De Raedt even used ILP in analysis and prediction of piano performances [198]

3.2.3 Inverse Entailment and PROGOL

The Theory on Inverse Entailment

Ever since induction can be treated as the inverse of deduction, inverse implication [133] has been the core problem in ILP [133]. Earlier approaches to the problem involved inverting resolution in theorem proving [131]. Nonetheless, such approaches are incomplete since inverting θ-subsumption is incomplete [155]. It has been shown by [155] that if C θ-subsumes D, then C implies D (i.e. $C \models D$). However, Plotkin also noted that C implies D does not necessarily mean that C θ-subsumes D. That simply means, if one performs generalization under θ-subsumption of a set of clauses S, it is possible that a suitable generalization is failed to be found, even if there exists a clause C such that $C \models S$.

Finally, the discovery that such a distinction between θ-subsumption and implication between a pair of clauses C and D is only relevant when C can self-resolve [133] has led to the development of the precise conditions under which a clause C implies another clause D. There are different ways of formalizing these conditions and the most graceful and widely known one is inverse entailment [136], as it is grounded in model theory. We briefly introduce the basic idea of inverse entailment. The general problem specification of ILP is that given background knowledge B and examples E find the best (or simplest) consistent hypothesis H such that

$$B \wedge H \models E \tag{3.2.1}$$

This can be rearranged to

$$B \wedge \overline{E} \models \overline{H}$$

Let $\overline{\bot}$ be the conjunction of ground literals which are true in all models of $B \wedge \overline{E}$(It exists if B and E are definite logic programs). Since $\overline{H}$ is true in every model of $B \wedge \overline{E}$, therefore we have

$$B \wedge \overline{E} \models \overline{\bot} \models \overline{H}$$

So, for all H

$$H \models \bot \tag{3.2.2}$$

Generally, only the case where H and E are single clauses is considered. It is clear that any H that satisfies equation 3.2.1 also satisfies equation 3.2.2. Thus, to

find solutions to equation 3.2.2 is sufficient. The following theorem states conditions under which a solution exists [136]:

Theorem 1. Let C and D be definite clauses and $S(D)$ be the sub-saturants of D. $C \models D$ iff one of the following conditions holds:

1. D is a tautology,
2. C θ-subsumes D,
3. C θ-subsumes $C' \in S(D)$.

The definition of sub-saturants is given in [136]. Intuitively, $S(D)$ is the set of clauses subsumed by some C such that a Herbrand model of $C \wedge \overline{D}$ does not exist (which is the condition under which $C \models D$). The third condition corresponds to the case when C can self-resolve but as it is remarked in [133] that in most real world applications this case is not significant, to find solutions to equation 3.2.2, one usually just considers the second condition.

The ILP algorithm PROGOL [136] is an implementation of the theory of inverse entailment. PROGOL searches only the subsumption lattice of the bottom clause, $\perp$, to find solutions to H. This means that PROGOL only considers hypotheses H such that

$$\square \preceq H \preceq \perp$$

where $\square$ is the so called "empty" clause which denotes the empty set.

We need to note two points here. First, a lattice is a partially ordered set in which all nonempty finite subsets have a least upper bound and a greatest lower bound. Since the relation that orders the set (of hypotheses) here is θ-subsumption ($\preceq$). Therefore, the term "subsumption lattice" is used. Next, the empty clause and the most general clause are the same thing. More exactly, the former refers to an empty set of literals, the latter has an empty set of literals in its body. Thus, they will be used interchangeably unless a distinction is necessary. The empty clause is the least upper bound of the subsumption lattice and the bottom clause is the greatest lower bound. PROGOL searches for a good clause to add to the building theory by first constructing the most specific clause (the bottom clause), a step commonly called saturation, and then searches the subsumption lattice (of clauses bounded between the empty clause and the bottom clause) in a Foil-like manner (i.e., from general to specific).

Mode Declarations

It is understandable that he bottom clause can have an infinite number of literals in its body. This means an unlimited search space and unlimited search time. In order to overcome this problem, a technique called "mode declaration" is adopted by PROGOL. A recall bound and input-output modes of the variables for a predicate are contained in "mode declaration". PROGOL also uses the variable depth bound.

Together these methods constrain the size of the bottom clause. There are two kinds of Mode declarations: head and body where the former is the mode declaration for the target predicate and the latter are mode declarations for all the predicates in the given set of background knowledge provided to PROGOL.

A mode declaration has either the form $modeh(n, atom)$ (the head mode declaration) or $modeb(n, atom)$ (a body mode declaration), where n, the recall bound, is an integer greater than zero or '*' (In fact, PROGOL substitutes an arbitrary large number, by default, 100 for "*") and atom is a ground atom. Terms in the atom are either normal or a place-marker. A normal term is either a constant or function symbol followed by a bracketed tuple of terms. A place-marker is either +type, -type, or #type, where +type is used where there is an input argument of a predicate, -type is used for an output argument, and #type is a constant.

The recall bound is the maximum number of alternative solutions for instantiating the atom used by the algorithm. A recall of '*' indicates all solutions. +type, -type, and #type correspond to input variables, output variables, and constants respectively.

A sample set of mode declarations for a grammar learning problem from Muggleton's PROGOL4.2 tutorial is given below:

$$: -modeh(1, s(+wlist, -wlist)).$$

$$: -modeb(1, prep(+wlist, -wlist)).$$

$modeh(1, s(+wlist, -wlist))$ is the (head) mode declaration because the target

concept to learn is $s(X, Y)$ where the list of words, X union Y , is a sentence accepted by the grammar. In this mode declaration, the first argument of the target predicate is declared to be an input variable of type *wlist* ("Word list") and it has a recall bound of 1. $modeb(1, prep(+wlist, -wlist))$ is a (body) mode declaration for a background predicate prep(X, Y) which recognizes if the first word in the list of words X is a preposition and Y is the rest of the list of words in X. Similarly, the first argument is declared to be an input variable of type *wlist*.

The rest of the body mode declarations for the simple grammar learning problem is given below[1]:

$$: -modeb(1, det(+wlist, -wlist)).$$

$$: -modeb(1, noun(+wlist, -wlist)).$$

$$: -modeb(1, tverb(+wlist, -wlist)).$$

$$: -modeb(1, iverb(+wlist,_w list)).$$

$$: -modeb(*, np(+wlist, -wlist)).$$

$$: -modeb(*, vp(+wlist, -wlist).$$

where each mode declaration corresponds to a distinct syntactic category.

The pair of lists X and Y in $s(X, Y)$ is also more formally called a difference list in the context of parsing. More precisely, a difference list consists of two lists: 1) an ordinary list, and 2) a pointer to the tail of the ordinary list. The intuition is that it represents the ordinary list minus the elements in the tail. In *Prolog* notation, the

[1] $http://www.doc.ic.ac.uk/\tilde{}shm/Software/PROGOL4.2/positives/gram_book1.pl$

following pairs are all representations of the same list, which has elements a, b, and c:

$[a, b, c]$ []

$[a, b, c, d, e]$ $[d, e]$

The PROGOL Algorithm

PROGOL uses a simple set covering algorithm in which each iteration performs the following steps [136]:

Table 3.2: **PROGOL's simple set covering algorithm**

Repeat step 1 to 4 until there remains no uncovered positive examples
1. Randomly chooses an example (seed example) from the set of uncovered positive examples;
2. Finds a clause (with maximal compression defined below) that generalizes the seed example chosen in step 1;
3. Adds the clause found to the building theory;
4. Positive examples covered by the clause are removed;

In order to find the clause with maximal compression, PROGOL searches the subsumption lattice with an A*-like algorithm. A simple outline of this algorithm is given in Table 3.3.

For each candidate clause s, the followings are calculated : 1) p_s = the number of positive examples covered by s, 2) n_s = number of negative examples covered by s, 3) c_s = length of the clause s - 1, 4) h_s = minimum number of further atoms to complete the clause, 5) $f_s = p_s$ - $(n_s + c_s + h_s)$.

Table 3.3: **PROGOL's algorithm for searching the subsumption lattice.**
Suppose E is the example being generalized.

1. $Open = \square$, $Closed = \emptyset$;
2. $s = best(Open)$, $Open = Open$ - s, $Closed = Closed \cup s$;
3. if $prune(s)$ goto 5;
4. $Open = (Open \cup refinements(s))$ - $Closed$;
5. if $terminated(Closed, Open)$ return $best(Closed)$;
6. if $Open = \emptyset$ return E (no generalization);
7. goto 2;

h_s is calculated by inspecting the output variables in the clause and determining whether they have been defined. For example, the clause $s(A, B)$, would have $h_s = 3$ because it requires at least three literals from $\perp$ to construct a chain of atoms connecting A to B. This is found from a static analysis of $\perp$.

f_s is a measure of how well a clause s explains all the examples with preference given to shorter clauses. The function $best(S)$ returns a clause $s \in S$ with the highest f value in S.

$prune(S)$ is true if $n_s = 0$ and $f_s > 0$. In this case, it is not worth considering refinements of s as they cannot possibly do better since any refinement will add another atom to the body of the clause and so cannot have a higher value of p than s does. It also cannot improve upon n_s as the latter is zero. $terminated(S, T)$ is true if $s = best(S)$, $n_s = 0$; $f_s > 0$ and for each t in T it is the case that $f_s \geq f_t$. In other words none of the remaining clauses nor any potential refinements of them can

possibly produce a better outcome than the current one.

This algorithm is guaranteed to terminate and to return the clause (if it exists) which has maximal compression. In the worst case, it will consider all clauses in the subsumption lattice. We will outline the complexity of PROGOL's bottom clause in the following section.

Complexity of PROGOL's Bottom Clause

The complexity of PROGOL's bottom clause is illustrated in [136]. Suppose we have the following definitions:

Table 3.4: **Definitions to outline PROGOL's complexity**

the set of mode declarations: Δ;
the cardinality of Δ: $\|\Delta\|$;
the upper boundary of the number of +type occurrences in each *modeh*: n^-;
the upper boundary of the number of -type occurrences in each *modeh*: n^+;
the upper boundary of the number of +type occurrences in each *modeb*: n^+;
the upper boundary of the number of -type occurrences in each *modeb*: n^-;
the recall of each mode $\in \Delta$: r_i;
the upper boundary of all the recalls: r;
the variable depth bound: h;
the bottom clause generated given a h: $\perp_h$.

Then the cardinality of $\perp_h$ is bounded by r, $|\Delta|$, n^-, n^+, and h:

$$| \perp_h | \leq (r \times |\Delta| \times n^- \times n^+)^{hn^+}$$

This shows that the complexity on the size of the bottom clause constructed by

PROGOL is exponential with respect to the variable depth bound h. Because the number of hypotheses in the subsumption lattice is two to the power of the number of literals in the body of the bottom clause, PROGOL searches a hypothesis space doubly exponential in complexity during execution. This may suggest that we must be very careful in choosing the parameters and writing the mode declarations while writing a PROGOL-executable program in order to keep the complexity at a reasonable level.

Using PROGOL - An Example

Recall from Equation 3.2.1 that the general knowledge-based induction problem is to solve the entailment constraint

$$Background \wedge Hypothesis \models Classifications$$

for the unknown $Hypothesis$, given the $Background$ knowledge and examples described by $Classifications$. To illustrate this, we will use the problem of learning supervisor-student relationship from examples. The descriptions of the background knowledge are quite straightforward. $staff(A)$ denotes A is an academic staff; $dept(A, B)$ says A belongs to the department of B; $student(A)$ represents A is a student. In $research(A, B)$, A is either a staff or a student, B are the research interests of A in keywords list format.

Suppose the target concept we want to learn is "supervisor". Thus the positive examples contain the true supervisor-student relationship and the negative examples

contain the false one.

Let's use a Prolog predicate $resemble(A, B, C)$ to represent the level of similarity of A and B's research interests. C might have three values: low, ave, high.

background knowledge:

$staff(andrea_crisanti)$. $staff(stephen_muggleton)$. $staff(robert_sinden)$.

$staff(mike_sternberg)$. $staff(marek_sergot)$. $staff(tim_jones)$.

$dept(andrea_crisanti, bio)$. $dept(stephen_muggleton, doc)$. $dept(robert_sinden, bio)$.

$dept(mike_sternberg, bio)$. $dept(marek_sergot, doc)$. $dept(tim_jones, chem)$.

$research(andrea_crisanti, [molecular, parasitology, malarial, parasites, protein, microarray])$.

$research(stephen_muggleton, [ilp, machine_learning, molecular_biology, nlp])$.

$research(robert_sinden, [molecular, cell, biology, malarial, mosquito, parasites, plasmodium])$.

$research(mike_sternberg, [protein_modelling, structure_ibioinformatics, docking, fold_recognition])$.

$research(marek_sergot, [legal_reasoning, temporal_reasoning, formal_theory, bioinformatics])$.

$research(tim_jones, [quantum_dots, inorganic_semiconductor, organic_semiconductor, optoelectronic])$.

$student(qiuxiang_li)$. $student(elisa_petris)$. $student(julian_gray)$. $student(hiroaki_watanabe)$.

$student(phil_carter)$. $student(david_lee)$. $student(r_sinha)$.

$dept(qiuxiang_li, bio)$. $dept(elisa_petris, bio)$. $dept(julian_gray, bio)$. $dept(hiroaki_watanabe, doc)$.

$dept(phil_carter, bio)$. $dept(david_lee, chem)$. $dept(r_sinha, bio)$.

$research(phil_carter, [structural_bioinformatics, docking])$.

$research(qiuxiang_li, [ilp, bioinformatics])$

$research(david_lee, [quantum_dots])$.

$research(elisa_petris, [mosquito, molecular, cell, biology, malarial,])$.

$research(julian_gray, [protein, microarray])$.

$research(r_sinha, [infectivity, plasmodium])$.

$research(suan_mitchel, [formal_theory])$.

$research(hiroaki_watanabe, [ilp, bayesian_network, convergence_of_logic_and_probability])$.

Examples

Positive Examples:

$supervisor(mike_sternberg, phil_carter)$. $supervisor(stephen_muggleton, hiroaki_watanabe)$.

$supervisor(stephen_muggleton, qiuxiang_li)$. $supervisor(andrea_crisanti, qiuxiang_li)$.

$supervisor(andrea_crisanti, elisa_petris)$. $supervisor(andrea_crisanti, julian_gray)$.

$supervisor(marek_sergot, suan_mitchel)$. $supervisor(tim_jones, david_lee)$.

$supervisor(robert_sinden, r_sinha)$.

Negative Examples:

$supervisor(mike_sternberg, elisa_petris)$. $supervisor(stephen_muggleton, julian_gray)$.

$supervisor(stephen_muggleton, qiuxiang_li)$. $supervisor(andrea_crisanti, hiroaki_watanabe)$.

$supervisor(andrea_crisanti, david_lee)$. $supervisor(andrea_crisanti, r_sinha)$.

$supervisor(marek_sergot, julian_gray)$. $supervisor(tim_jones, julian_gray)$.

$supervisor(robert_sinden, hiroaki_watanabe)$.

Using PROGOL, we can infer the following rules which represent the supervisor-student relationship:

$$supervisor(A, B) :- staff(A),\ student(B),\ dept(A, C),\ dept(B, C),\ research(A, D),\ research(B, E),\ resemble(D, E, ave).$$

$$supervisor(A, B) :- staff(A),\ student(B),\ dept(A, C),\ dept(B, C),\ research(A, D),\ research(B, E),\ resemble(D, E, high).$$

$$supervisor(A, B) :- staff(A),\ student(B),\ dept(A, C),\ dept(B, D),\ research(A, E),\ research(B, F),\ resemble(E, F, high).$$

The first two rules can be translated into English as "A is B's supervisor if A is an academic staff, B is a student, A and B belong to the same department, and that the level of similarity of their research interests is either 'ave' or '$high$' "; the third rule represents "A is B's supervisor if A is an academic staff, B is a student, A and B belong to different departments, and that the level of similarity of their research interests is '$high$' ".

Note that the three rules generated are based on our specific examples and data representation only, thus it might not be a general model.

3.3 Analogical Reasoning

Analogies have been extensively studied in Artificial Intelligence (AI) [167]. These are problems of the form "A is to B as C is to ?". Analogies in reasoning or learning (Analogical Reasoning) [78, 101, 124, 84, 100, 99, 134, 73, 102, 40, 207, 165, 200, 87, 80] is a heuristic-based Machine Learning approach where the solution to a problem is found by noting similarities with previously solved problems. It helps model builders apply their modelling experience to construct new models and improve their modelling knowledge through learning.

3.3.1 Introduction

In Webster's new collegiate electronic dictionary, two definitions of *analogy* are given: "inference that if two or more things agree with one another in some respects they will probably agree in others" and "correspondence in function between anatomical parts of different structure and origin." The first definition for the term *analogy* refers to an inference process where aspects of an entity or a situation that are unknown might be temporarily assigned a predicted value based on familiarity with a known entity or situation, given the observation that some aspects are shared between the entities or situations in question. However, what differentiates analogical reasoning from other forms of difference-based reasoning is the second definition. It states that two entities are analogous if there is a functional relationship or correspondence in their behavior, even if structural differences between the entities may exist. In other words, this definition tells us that entities are analogous if they have parts behaving in similar ways, although differences in appearance or orientation may exist.

We are particularly interested in the analogy, that provides insight into the solution of a target problem based on the solution, or derivation of the solution, for a source problem. Namely, the discovery of an analogical relationship between a source and target problem specification must be able to lead to a consistent mapping between the source and target solution. By nature, analogical reasoning is imprecise:

relationships that are common to two systems do not guarantee that an analogical relationship exists in any other particular attribute which is outside of the functionality directly implied by the common role behaviors. It simply tells us that an analogical reasoning may suggest that a certain deduction is true, but this may not always be the case.

The transfer of the value of an aspect A_1 from a source case s to a target case t based on the similarity of s and t with respect to another aspect A_2 is the standard form of reasoning by analogy. It should be noted that justifications are necessary for such analogical inferences. They are given by connections between the aspects A_2 and A_1. These justifications of analogical inferences have usually been represented propositionally, e.g. as determination, schemata, connections, or similarity transforms.

Example of such connections is "if two organisms are quite close in the phylogenetic tree, then they probably share the same sex determination pathway". Suppose the source case is *D. melanogaster* and the target case is *A. gambiae*. We choose *sex_determination_pathway* as an aspect and *genome_sequece_similarity* as another aspect. Based on the similarity of *D. melanogaster* and *A. gambiae* with respect to *genome_sequece_similarity*, the value of *sex_determination_pathway* could be transferred from the source case to the target case with corresponding analogical inference. That is, from $sex_determination_pathway(D.\ melanogaster)$ to $sex_determination_pathway(A.\ gambiae)$.

Another example of analogical reasoning is shown in Fig. 3.1. There are two steps in this example. In step 1, two phrases/abbreviations A and B are given as input and one calculates a set of properties, relations and "similarities" such as acronym of the nouns, rotations and reflections which take A into B and relate C to the possible answer. Step 2 forms a set of theories or transformation rules taking A into B. To solve this problem, one then attempts to generalize these theories to cover additional data (C and the answer figure and text). This should result in a subset of the admissible theories, i.e. transformation rules which take A into B and C into exactly one answer figure. Finally, one should be able to pick up the most specific theory from these admissible theories.

Eventually, one might formulate the following hypothesis as a Prolog clause.

is_to(X, Y) :- contains(X, X_Text, X_Shape),

remove_non_nouns(X_Text, U), acronymize(U, F_Text),

rectangle_to_ellipse(X_Shape, F_Shape), assembles(F_text, F_Shape, Y).

3.3.2 Applications of Analogical Reasoning

Analogical Reasoning has been extensively researched over the past two decades. Some recent applications reported are for knowledge discovery in molecular biology by Hass et al [77], for software reuse by whitehurst [203], for general Problem Solving by veloso [199], for designing physical devices by Bhatta [17] and in scientific problem

solving by Clement [31].

3.4 Data Set

The whole genome database for *D. melanogaster* and *A. gambiae* were used (from NCBI) in our study. Secondary structure information for each sequence was predicted by a software called Psipred [122].

As the first step of our project is to build a model to discriminate the sex determination related and unrelated genes, we identified 26 proteins as the positive examples for training. We randomly selected 26 proteins from the rest of the dataset as the negative examples. The corresponding genes that represent the proteins we chose are: '*da*', '*doublesex*', '*dpn*', '*dsf*', '*emc*', '*fl2d*-1', '*fl2d*-2', '*fru*', '*gro*', '*her*', '*ix*', '*mle*', '*mof*', '*msl*-1', '*msl*-2', '*msl*-3', '*otu*', '*ovo*', '*run*', '*sisa*', '*sisb*', '*sisc*', '*snf*', '*sxl*', '*tra*', '*tra*2' and '*vir*'.

3.5 Data Representation

ILP systems represent their data as logic programs. PROGOL, the ILP system used throughout this work, utilizes the formalism of Prolog. The basic syntactic structure in Prolog is a relation, also called a predicate, an example of which would be $adjacent(A, B)$, which states that the objects designated by A and B are adjacent in

the primary structure. Rules, also called clauses, have the form Head : − Body, and are interpreted as follows: "if the conditions in the Body of the clause are true then Head is a logical consequence". For example,

$$sex_determination(A) :- number_helices(A, B), instability_index(A, C),$$
$$aliphatic_value(A, D), range_r(14 =< B),$$
$$interval_l(C =< 88.140), interval_l(D =< 79.83).$$

is interpreted as: "if there exists a helix B with a length of at least 14, and the instability index of the protein sequence C is less than 88.410, and the aliphatic value of the sequence D is less than 79.83, then sequence A must have sex determination function".

We will illustrate this in the background knowledge section.

3.6 Feature Selection

3.6.1 Global and Local Attributes

Construction of feature vectors is the key to successful rules generation. For each protein sequence, feature vectors are assembled from encoded representations of tabulated residue properties including molecular weight, amino acids composition, predicted secondary structure composition, hydrophobicity, normalized van der Waals volume, polarity, polarizability, amino acid pI, theoretical pI, charge, sulfur, aromatic

class, instability index, aliphatic index, estimated extinction coefficient and grand average of hydropathicity index for each residue in sequence. The use of some of the features was motivated by previous studies of proteins [46]. We didn't use two important properties surface tension [114, 170, 178] and solvent accessibility [4, 3] as we found it is difficult to produce the data for each sequence in the dataset. We might integrate these two properties later on.

Our approach uses a combination of local and global information about amino acid sequences. A protein sequence is represented by a set of parameter vectors based on various physicochemical and structural properties of amino acids along the sequence. These parameter vectors were constructed in two steps. The sequence of the amino acids was transformed into sequences of certain physicochemical or structural properties (attributes) of residues. Twenty amino acids were divided into three groups for each of nine different amino acid attributes representing the main clusters of the amino acid indices. Thus, for each attribute, every amino acid was replaced by the index 1, 2, or 3 according to one of the three groups to which it belongs. The attributes we have used included the predicted secondary structure, in which the indices 1, 2, and 3 correspond to the helix, strand, and coil, respectively. For the other eight attributes, those of hydrophobicity, normalized van der Waals volume, polarity, and polarizability, etc the 20 amino acids were divided into three groups according to the magnitudes of their numerical values. The ranges of these

numerical values and the amino acids belonging to each group are shown in Table 3.5.

Table 3.5: **Amino Acid Attributes and the Division of the Amino Acids Into Three Groups for Each Attribute**

Property	**Group 1**	**Group 2**	**Group 3**
Secondary Structure	H (α helix)	E (β strand)	C (Coil)
Hydrophobicity	Polar R,K,E,D,Q,N	Neutral G,A,S,T,P,H,Y	Hydrophobic C,V,L,I,M,F,W
Normalized van der Waals volume	0 - 2.78 G,A,S,C,T,P,D	2.95 - 4.0 N,V,E,Q,I,L	4.43 - 8.08 M,H,K,F,R,Y,W
Polarity	4.9 - 6.2 L,I,F,W,C,M,V,Y	8.0 - 9.2 P,A,T,G,S	10.4 - 13.0 H,Q,R,K,N,E,D
Polarizability	0 - 0.108 G,A,S,D,T	0.128 - 0.186 C,P,N,V,E,Q,I,L	0.219 - 0.409 K,M,H,F,R,Y,W
Amino Acid PI	2.77 - 3.22 D,E	5.02 - 6.30 C,N,F,T,Q,Y,S,M, W,I,V,G,L,A,P	7.47 - 11.15 H,K,R
Amino Acid Charge	D, E	R, K	H
Aromatic class	F	W	Y
Sulfur	C	M	A,R,N,D,Q,E,G,H,I, L,K,F,P,S,T,W,Y,V

The three descriptors, "composition" (C), "transition" (T), and "distribution" (D), were calculated for a given attribute to describe the global percent composition of each of the three groups in a protein, the percent frequencies with which the attribute changes its index (i.e., from group X to group Y, where $X \neq Y$, $1 \leq X$, $Y \leq 3$) along the entire length of the protein sequence, and the distribution pattern of the attribute along the sequence, respectively.

Let us take the secondary structure attribute as an example. All amino acids are divided into three groups: helix, strand, and coil. The "composition" descriptor C

consists of the three numbers: the global percent compositions of helix, strand, and coil elements in the protein. The "transition" descriptor T also consists of the three numbers: the percent frequency with which: 1) a helix residue is followed by a strand residue or a strand residue by a helix residue; 2) a helix residue is followed by a coil residue or a coil residue by a helix residue; and 3) a strand residue is followed by a coil residue or a coil residue by a strand residue. The "distribution" descriptor D consists of the five numbers for each of the three groups: the fractions of the entire sequence, where the first residue of a given group is located, and where 25%, 50%, 75%, and 100% of those are contained. Thus, the complete parameter vector contains 3 (C) + 3 (T) + 5 × 3 (D) = 21 scalar components.

All together, the nine different amino acid attributes produce nine parameter vectors each containing 21 scalar components. The other parameter vectors used was the vector of some local attributes, like predicted estimation coefficient.

The composition and dimensionality of feature vectors are shown in Table 3.6. We have combined different parameter sets into one dataset so that each protein is represented by a 215-dimensional feature vector.

3.6.2 Combining the Global, Local and Relational Attributes

To use the global and local attributes only is not sufficient to represent the complexity of the molecular biology data. The ability to manipulate relational information is a

Table 3.6: **Feature vectors and dimensions**

Parameter name	**Dimension**
Molecular Weight	1
Amino acids composition	20
Predicted Secondary Structure	21
Hydrophobicity	21
Normalized van der Waals volume	21
Polarity	21
Polarizability	21
Amino Acid PI	21
Theoretical PI	1
Amino Acid Charge	21
Sulfur	21
Aromatic class	21
Instability index	1
Aliphatic index	1
Estimated extinction coefficient	1
Grand average of hydropathicity index	1

distinctive feature of ILP systems. Suppose a protein A has two α-helices $h1$, $h2$ and one β-strand $b1$. If the $h1$ is neighbored to $b1$ and $b1$ followed by $h2$ with a distance of 5 residues, we could easily represent these by the following: $neighbour(A, h1, b1)$, $distance(A, b1, h2, 5)$. The above representations are both concise and easy to understand by a human. Of course it is still possible to represent these relations with a traditional attribute value format to be used in the attribute value based learning, but one may imagine how complex it would be to write these in a traditional machine learning method.

Several relational attributes have been incorporated with the global attributes. A protein sequence and all its α-helices and β-strands ids are associated by

$has_helix(+protein, -helix_t)$ and $has_strand(+protein, -strand_t)$ respectively.

The helix id, helix starting position and helix length are represented by $helix(+helix_t, -nat, -nat)$, while the strand id, strand starting position and strand length are represented by $strand(+strand_t, -nat, -nat)$. Here $helix_t$ and $strand_t$ are α-helix id and β-strand id respectively which uniquely identifies the helix or strand in the whole dataset. The variable nat which is an integer value represents either starting position or length of an element.

Intervals of values are used by the global predicates and the values for the boundaries are learnt by the algorithm. For the predicates *distance*, we have used predefined intervals. Another measure was to number the helices and strands separately (*number_helices*, *number_strands*) to allow the learning algorithm to rely on the most conserved numbering scheme. Finally, The distribution of the length of the secondary structure was calculated for each protein sequence. If the actual value was lower or equal to the mean minus (or plus) two standard deviations the value was replaced by *very-lo* (*very-hi*), if it was lower or equal to the mean minus (or plus) one standard deviation the value was assigned *lo* (*hi*).

Take the protein sequence $'cg14476_pb'$ as an example. Suppose it has 2 helices and 1 strand; the first helix starts at position 2 and has a length of 17 and the second helix starts at position 22 and has a length of 3; the strand starts at position 48 and has a length of 2. Then we could represent the two α-helices with $'cg14476_pb_h0'$ and

$'cg14476_pb_h1'$ and the β-strand with $'cg14476_pb_s0'$. We can calculate the ending position of $h0$ with the starting position and length, which is $2 + 17 - 1 = 18$. While $h1$ starts at position 22, this means there are three residues between $h1$ and $h2$. We thus have their distance value which is 3. With the same method, we can get the distance between $h1$ and $s0$, which is -28 (the distance is negative means the second parameter is less than the first one).

The relational attributes can be represented below:

$$has_helix(cg14476_pb, cg14476_pb_h0)$$

$$has_helix(cg14476_pb, cg14476_pb_h1)$$

$$has_strand(cg14476_pb, cg14476_pb_s0)$$

$$number_helices(cg14476_pb, 2)$$

$$number_strands(cg14476_pb, 1)$$

$$helix(cg14476_pb_h0, 2, 17)$$

$$helix(cg14476_pb_h1, 22, 3)$$

$$strand(cg14476_pb_s0, 48, 2)$$

$$distance(cg14476_pb_h0, cg14476_pb_h1, 3)$$

$$distance(cg14476_pb_s0, cg14476_pb_h1, -28)$$

We experimented with this approach and found that we generated slightly more rules then use global attributes only method.

3.6.3 Feature Calculation

The calculation methods of the features of a protein sequence we chose will be described in detail in this section.

Molecular Weight: The molecular weight of the protein or peptide is shown in Daltons. We refer to the formula in [54] to calculate this value:

$$MW = (nA \times 71.07) + (nR \times 156.18) + (nN \times 114.08) + (nD \times 115.08) + (nC \times 103.10)$$
$$+(nQ \times 128.13) + (nE \times 129.11) + (nG \times 57.05) + (nH \times 137.14) + (nI \times 113.15)$$
$$+(nL \times 113.15) + (nK \times 128.17) + (nM \times 131.19) + (nF \times 147.17) + (nP \times 97.11)$$
$$+(nS \times 87.07) + (nT \times 101.10) + (nW \times 186.20) + (nY \times 163.17) + (nV \times 99.13) + 18.02$$

Amino Acid PI: Grouping of amino acids based on certain properties can be found at the following link: $http://www.bioscience.org/urllists/aminacid.htm$. However they didn't provide our required classification. We regrouped the amino acids into 3 groups based on their pI value. We are still not very sure whether this is an appropriate classification in terms of biological/physicochemical properties. The amino acids pI values can be found below:

Group	Amino Acid pI value
Group1	D 2.77 E 3.22
Group2	C 5.02 N 5.41 F 5.48 T 5.64 Q 5.65 Y 5.66 S 5.68 M 5.74 W 5.89 I 5.94 V 5.96 G 5.97 L 5.98 A 6.00 P 6.30
Group3	H 7.47 K 9.59 R 11.15

Theoretical PI: The isoelectric point is the pH at which the protein has no net charge. The net charge of a protein is calculated as the sum of the number of positively charged residues (protonated lysine, arginine, histidine), minus the number of negatively charged residues (deprotonated tyrosine, cysteine, glutamate, aspartate), plus the number of protonated amino termini, minus the number of deprotonated carboxyl termini. The net charge calculation does not take into account any electrostatic interactions within the protein that may perturb ionization [189, 18, 205]. For each amino acid of interest, the number of protonated residues is determined by the following equation:

$$N_p = N_t[H^+]/([H^+] + K_N)$$

where N_p = number of protonated residues, N_t = total number of residues of a specific amino acid, $[H^+]$ = hydrogen ion concentration, K_N = dissociation constant for the amino acid of interest that is equal to the following: 10^{-pK_N}. The isoelectric point must exist and lie between pH 1.0 and 13.0.

Charge of Amino Acid: The following amino acids are divided into three groups

according to their biochemistry properties [34]:

Group	Amino Acid(s)	Charge per amino acid @ pH 6
Group1	D, E	-
Group2	R, K	+
Group3	H	+

Aromatic Value: We chose three amino acids with aromatic Rings [89]: Phe (F), Trp (W), Tyr (Y) and put them into group 1, 2 and 3 respectively.

Sulfur Value: 2 amino acids which are sulfur-containing were chosen [89]: $Cys(C)$, $Met(M)$. We put these two into group 1 and 2 respectively. Then we put all the rest amino acids which are not sulfur-containing to the third group:

Group	Amino Acid(s)
Group1	C
Group2	M
Group3	A, R, N,D,Q,E,G,H,I,L,K,F,P,S,T,W,Y,V

Instability index : (II) statistical analysis of 12 unstable and 32 stable proteins has revealed that there are certain dipeptides, the occurence of which is significantly different in the unstable proteins compared with those in the stable ones [76]. The authors of this method have assigned a weight value of instability to each of the 400 different dipeptides (WV). Using these weight values it is possible to compute an instability index (II) which is defined as:

$$II = \frac{10}{L} \times \sum_{i=1}^{L-1} WV(x[i] \times x[i+1])$$

where: L is the length of sequence $WV(x[i] \times x[i+1])$ is the instability weight value for the dipeptide starting in position i.

A protein whose instability index is smaller than 40 is predicted as stable, a value above 40 predicts that the protein may be unstable.

Estimated extinction coefficients: It has been shown [64, 146] that it is possible to estimate the molar extinction coefficient of a protein from knowledge of its amino acid composition. From the molar extinction coefficient of tyrosine, tryptophan and cystine (cysteine residues do not absorb appreciably at wavelenghts $> 260nm$, while cystine does) at a given wavelength the extinction coefficient of a denaturated protein can be computed using the equation:

$$E(Prot) = Numb(Tyr) \times Ext(Tyr) + Numb(Trp) \times Ext(Trp)$$
$$+Numb(Cystine) \times Ext(Cystine)$$

The absorbance (optical density) can be calculated using the following formula:

$$Absorb(Prot) = E(Prot)/Molecular_weight$$

The conditions at which these equations are valid are: pH 6.5, 6.0 M guanidium hydrochloride, 0.02 M phosphate buffer at $280nm$. The values we used are:

$$Tyr(Y) = 1490, Trp(W) = 5500, Cys(C) = 125$$

Aliphatic index: A statistical analysis has shown that the aliphatic index, which is defined as the relative volume of a protein occupied by aliphatic side chains (alanine, valine, isoleucine, and leucine), of proteins of thermophilic bacteria is significantly higher than that of ordinary proteins. We chose the amino acids with aliphatic R-Groups and the aliphatic index of a protein is calculated according to the following formula [91]:

$$Aliphatic_index = X(Ala) + a \times X(Val) + b \times (X(Ile) + X(Leu))$$

where X(Ala), X(Val), X(Ile), and X(Leu) are mole percent ($100 \times molefraction$) of alanine, valine, isoleucine, and leucine. The coefficients a and b are the relative volume of valine side chain ($a = 2.9$) and of Leu/Ile side chains ($b = 3.9$) to the side chain of alanine. The aliphatic index may be regarded as a positive factor for the increase of thermostability of globular proteins.

GRAVY: The grand average of hydropathicity index indicates the solubility of the proteins: positive GRAVY (hydrophobic), negative GRAVY (hydrophilic) The GRAVY value for a peptide or protein is calculated as the sum of hydropathy values of all the amino acids, divided by the number of residues in the sequence. The amino acid scale values we used are :

Ala:	1.800	Arg:	-4.500	Asn:	-3.500	Asp:	-3.500	Cys:	2.500
Gln:	-3.500	Glu:	-3.500	Gly:	-0.400	His:	-3.200	Ile:	4.500
Leu:	3.800	Lys:	-3.900	Met:	1.900	Phe:	2.800	Pro:	-1.600
Ser:	-0.800	Thr:	-0.700	Trp:	-0.900	Tyr:	-1.300	Val:	4.200

Hydrophobicity: The Hydrophobicity groups are classified according to the methods described in [30, 61, 88, 166, 25, 36]:

Group	Hydrophobicity	Hydrophobicity group
Group1	Polar	R,K,E,D,Q,N
Group2	Neutral	G,A,S,T,P,H,Y
Group3	Hydrophobic	C,V,L,I,M,F,W

Normalized van der Waals volume: The Normalized van der Waals volume groups are classified according to the method presented in [55]:

Group	Normalized van der Waals volume	Group
Group1	0 - 2.78	G,A,S,C,T,P,D
Group2	2.95 - 4.0	N,V,E,Q,I,L
Group3	4.43 - 8.08	M,H,K,F,R,Y,W

Polarity: The Polarity groups are classified according to the method in [71]:

Group	Polarity	Group
Group1	4.9 - 6.2	L,I,F,W,C,M,V,Y
Group2	8.0 - 9.2	P,A,T,G,S
Group3	10.4 - 13.0	H,Q,R,K,N,E,D

Polarizability: The Polarizability groups are classified according to the method in [25]:

Group	Normalized van der Waals volume	Group
Group1	0 - 2.78	G,A,S,C,T,P,D
Group2	2.95 - 4.0	N,V,E,Q,I,L
Group3	4.43 - 8.08	M,H,K,F,R,Y,W

Secondary Structure Information: We tried two secondary structure prediction software: PRFO and PSIPRED. The PROF protein secondary structure prediction [145] package was developed by Ross King. For a protein sequence with 549 residues, the approximate time taken to produce a secondary structure prediction is around 170 seconds.

The PSIPRED protein structure prediction server [122] allows users to submit a protein sequence, perform a prediction of their choice and receive the results of the prediction both textually via e-mail and graphically via the web. It is freely available to non-commercial users here: $http://bioinf.cs.ucl.ac.uk/psipred/$.

We ran our sequence database (18,107 protein sequences for *D. melanogaster* and 15,212 protein sequences for *A. gambiae*) using both tools and we found PSIPRED produced significantly more β-strands than PROF did, while most of the segments of a predicted sequence by PROF consist of coils (C), with a few α-helices distributed sparsely across the sequence, and far less β-strands. Moreover, PSIPRED is generally 20%-30% faster than PROF observed from our experiments. We decided to adopt the data produced by PRIPRED as the results are more biologically meaningful.

As we have 33,319 sequences in total, one may imagine how long PSIPRED will

take to run the job. In order to keep the execution time at a reasonable level, we resample the raw sequences into a list of sub-files, each of which contains only 300 sequences. We ran our job using the workstations at Doc (texel01∼texel46, sync01∼sync20), each job on one workstation. Approximately we can collect and recombine the data after 12 hours.

Note that the database (nr) we used has 497,508 sequences and has a size of 222M. Together with all its index files, the database size is 582M. We also tried the release of nr on Apr 20 2004, which together with all its index files has a size of around 2,000M. The running time is 5 times slower than using the old database nr. Therefore we decided to use the old database nr. The latest database nr can be found here: $ftp://ftp.ncbi.nlm.nih.gov/blast/db/nr.tar.gz$.

The Multiple Sequence Alignments software we used was CLUSTAL W (1.74). It is available at this site: $ftp://ftp.ebi.ac.uk/pub/software/unix/clustalw$.

The $BLASTP ftp://ftp.ncbi.nlm.nih.gov/blast/executables/release/2.2.8/$ we used was version 2.2.8 (Jan 05 2004) release.

3.7 Background Knowledge

In preparation for a machine learning task, the user must define the predicates that can be used to construct new rules and know how they can be combined with one another. This effectively defines the space of possible rules.

The background knowledge for the protein sequence discrimination task comprises 68 predicates, see Table 3.7, Table 3.8, Table 3.9, Table 3.10 and Table 3.11 for details. They can be classified into three categories: global, local and relational. The global and local predicates include the protein length, the number of helices and the number of strands. Other machine learning techniques, such as neural network and decision trees, only use this type of representation. The relational information comprises 15 predicates. Among them are the predicate *distance*, which describes the relationship between two consecutive secondary structure elements; the predicate *helix*, etc.

3.8 Experiments

3.8.1 Example of the Execution of PROGOL

In our experiments, we use progol4.4 (tutorial available here [130]). The learning algorithm performs two main steps: (i) the construction of the most specific clause and (ii) a general to specific search to find an optimal rule. To illustrate the execution of PROGOL we use an example from the protein SCUTE ($CG3827\text{-}PA$). In the first step, PROGOL selects a positive example randomly, here represented by the protein identifier $cg3827_pa$, and derives all the relevant information for this example from the background knowledge:

CPROGOL Version 4.4

...

[Generalising $sex_determination(cg3827_pa)$.]

...

[Most-specific clause reduced by 9 literals]

[Most specific clause is]

$$sex_determination(A) :- total_length(A, B), number_helices(A, C),$$
$$number_strands(A, D), has_helix(A, E), has_strand(A, F),$$
$$percent_amino_acid(A, G, H, I, J, K, L, G, M, G, N, H, O, P, Q, I, R,$$
$$S, T, U, V, W), molecular_weight(A, X), percent_helix(A, Y),$$
$$percent_strand(A, Z), percent_coil(A, A0), percent_helix_strand(A,$$
$$V), percent_helix_coil(A, B0), percent_strand_coil(A, H),$$
$$location_dis_helix(A, C0, D0, E0, F0, G0), location_dis_strand(A,$$
$$H0, I0, J0, K0, L0), location_dis_coil(A, V, M0, N0, O0, P0),$$
$$...$$
$$instability_index(A, I4), extinction_coefficients(A, J4),$$
$$aliphatic_value(A, K4), gravy_value(A, L4), range(3 =< (D =< 3)),$$
$$range(8 =< (C =< 8)), range(345 =< (B =< 345)), range_l(B =< 345),$$
$$range_l(C =< 8), range_l(D =< 3), range_r(3 =< D), range_r(8 =< C),$$
$$...$$
$$interval_r(100.000 =< P0), interval_r(3.815e + 04 =< X).$$

$[C : -1, 49, 49, 0\ sex_determination(A).]$

$[C : -2, 49, 49, 0\ sex_determination(A) : -\ total_length(A, B).]$

$[C : -1, 49, 48, 0\ sex_determination(A) : -\ number_helices(A, B).]$

$[C : 0, 48, 45, 0\ sex_determination(A) : -\ number_helices(A, B), number_strands(A, C).]$

...

This constitutes a reservoir of information that can be used to construct new rules. In the second step, PROGOL searches the space of all possible rules, starting with the most general one, which is "everything is sex_determination_related": $[C : -1, 49, 49, 0 sex_determination(A).]$ The search uses a branch-and-bound-like algorithm guided by a measure of compression. This measure depends on the number of positive and negative examples covered as well as the length of the clause. The rule is specialized: "every protein such that the number of helices it contains is B": $[C : -1, 49, 48, 0 sex_determination(A) : -\ number_helices(A, B).]$ which leads to a new value for the compression measure. The clause is further specialized: "every protein such that the number of helices and strands are B and C respectively".
$[C : 0, 48, 45, 0 sex_determination(A) : -\ number_helices(A, B),\ number_strands(A, C).]$

At the end of the search, PROGOL has found the rule that maximizes its measure of compression:

Resource limit exceeded

[1000 explored search nodes]

f=0,p=7,n=0,h=0

[Result of search is]

[C:5,11,4,0 $sex_determination(A) :- number_helices(A, B), number_strands(A, C),$

$location_dis_aromatic_2(A, D, D, D, D, D), aliphatic_value(A, E), interval_l(E =< 65.860).$]

The final result produced by PROGOL is:

CPROGOL Version 4.4

...

$sex_determination(A) :- number_helices(A, B),$

$number_strands(A, C), location_dis_aromatic_2(A, D, D, D, D, D),$

$location_dis_aromatic_3(A, E, F, G, H, I), interval_l(E =< 10.470),$

$interval_r(82.270 =< H).$

$sex_determination(A) :- number_helices(A, B), number_strands(A, C),$

$range(5 =< (B =< 5)), range_r(10 =< C).$

$sex_determination(A) :- number_helices(A, B),$

$location_dis_aromatic_2(A, C, D, E, F, G), range_r(14 =< B),$

$interval_l(C =< 51.450), interval_r(67.190 =< D).$

$sex_determination(A) :- number_helices(A, B), range(31 =< (B =< 31)).$

$sex_determination(A) :- number_strands(A, B), location_dis_aromatic_3(A,$

$C, D, E, F, G), interval_r(99.890 =< G).$

[Total number of clauses = 18]

[Time taken 282.86s]

3.8.2 Parameters Selection

An empirical investigation into the effect of a variety of parameters was made. Three different percentages of noise were sampled: 0, 10 and 20%. The parameter noise controls the percentage of false positive examples allowed. Three inflate rates were tested: 100, 200 and 400. Three sets of negative examples each having: $\times 1$, $\times 12$ and $\times 14$ more negative examples than positives were tested. This was done for pragmatic reasons, machine learning algorithms were developed, in general, to work with an equal number of positive and negative examples. The combination: noise = 20, inflate = 200 and $\times 12$ gave the best result, in the sense that it maximizes the sum of the number of rules and minimizes the false negatives. This combination was used throughout the tests.

The maximum number of hypotheses (nodes) explored during the search was restricted to 1,000. Finally, the parameter c, which controls the maximum number of predicates in the body of a rule, was set to nine. We observed that this limit was never reached.

Performance analysis were carried out over the whole *D. melanogaster* data sets. We ran all the training and testing tasks using Doc's Linux workstation (2.0GHz

CPU, 512M RAM). The final rules were learnt on the complete data sets.

3.8.3 Preliminary Test

We have generated 5 rules using PROGOL from the training process. To test the accuracy and specificity of our rules, we scan the whole genome sequence of *D. melanogaster* with these rules. As we mentioned before, the rules inferred are too general. We could see this clearly from the fact that the 5 rules together cover 10430 proteins out of the total number of proteins (18107).

Total Hits: 10430

Distributed by rules:

[0]: 1025

[1]: 3988

[2]: 138

[3]: 3996

[4]: 1283

...

Positive Examples covered: 62

Number of Single Positive Example covered: 44

Note that some positive examples may appear more than

one time, as they satisfy more than rule generated.

Some of the positive examples satisfy more than one rule generated, some covers

no rules. The inverse frequency table 3.12 gives an overview of such a situation. It is an indicator of how well the rules generated can cover the positive examples.

3.9 Discussion

A list of vectors contains 215-dimension scalar values are used to describe each protein, as motivated by previous studies of proteins properties. It has been reported that, not all feature vectors play equal role, some seem to play more prominent role in specific aspects. For instance, the protein composition, secondary structure and hydrophobicity have been found to play more prominent role than other feature vectors in protein fold recognition. It is of interest to examine which feature vectors play more prominent role in our protein function classification (sex determination related/unrelated). Our analysis indicates that the composition, charge, polarity, hydrophobicity play more prominent role than other feature vectors. Charge and polarity are important for electrostatic interactions and hydrogen-bonding, which are two important factors for the binding of a protein with its partner. Therefore their importance in protein function is expected. The secondary structure does not appear to be a prominent feature vector as in protein fold recognition. Secondary structure is the basis for structural frame of protein fold. It is however not necessarily the essential element for the sex determination function of a protein. Therefore, it is understandable that this feature vector is not a prominent one.

Several factors may affect the classification accuracy. One is the diversity of the negative protein samples. It is obvious that only a small faction of proteins are represented as the negative examples in the training sets. This can be improved along with increasing the number of negative examples in a single running cycle or increasing the number of tests with different negative examples and merge the final rules generated. These strategies enable the generation of truly optimal support vectors. PROGOL optimization procedure and feature vector selection may also be improved.

Another factor may be the data we have used is imperfect. In an ideal situation, the induced concept descriptions will agree with the classifications of the descriptions of all concept instances. However, in practice it frequently happens that data given to the learner contains various kind of errors, either random or systematic. In such circumstances, we say that the learner has to deal with *imperfect data*.

In ILP, where relational descriptions are learned in a first-order language from examples E and background knowledge B, the following forms of imperfect data can be encountered:

1. random errors (noise):
    (a) noise in training examples E (caused by erroneous argument values and/or erroneous classification of facts as true or false),
    (b) noise in background knowledge B;

2. too sparse training example E (incompleteness) from which it is difficult to reliably detect regularities;

3. Imperfect background knowledge B (inappropriateness):

   (a) background knowledge B may contain predicates that are not relevant for the learning task,

   (b) predicates in B may be insufficient for learning (essential predicates may be missing);

4. missing argument values in training examples E (missing values).

Having realized this, we plan to reduce the noise in the background knowledge by filtering out some very long proteins or proteins whose sequences haven't been finalized (just filled by a long list of repeating amino acids produced with a random guess to fill the contigs) from training data set. We need refine the predicates in the background knowledge as well.

Due to the lack of the gene expression data, we currently conduct our experiments based on the published whole genome sequence of *D. melanogaster* and *A. gambiae*. The sequence data is static while the gene expression data is dynamic; in nature it is easy to infer the gene interaction relationships from the gene expression data. Virtually all the studies of genetic networks are based on analyzing the gene expression data. It explains why the results we have produced are not so good so far.

3.10 Summary

The identification of the sex determination pathway of *A. gambiae* directly is not a trivial task. It involves biological science and computer science, especially in the fields of genetics, artificial intelligence, and machine learning. The inductive logic programming system PROGOL has been shown to be a useful tool for predicting protein secondary structure, inferring metabolic pathway, and for solving many other molecular biology related problems. We have done experiments with PROGOL based on the chosen training examples from *D. melanogaster*. Several rules have been produced by PROGOL. Unfortunately the rules generated are too general, which is not sufficient to discriminate the sex determination related proteins from unrelated ones in *D. melanogaster*.

Improved feature vector selection and refined negative samples selection, better PROGOL data file representation and PROGOL parameters setting, the including of pruning may be used to further improve the experiment results.

In summary, the idea of using Inductive Logic Programming to infer the complex protein-protein interaction relationship does not seem to work very well. We have to seek alternative way to address this problem. Our next step will be to work on the protein-protein interaction level with different mathematical methods trying to find the rules and similarities between different organisms.

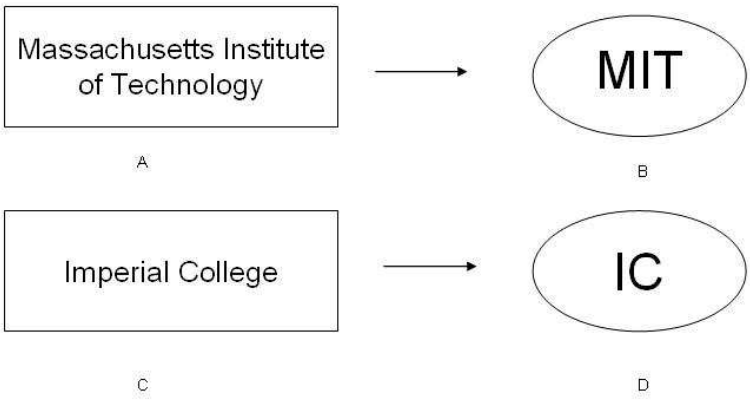


Figure 3.1: An analogy problem

Table 3.7: **Background knowledge: Global Attributes 1**

Predicates	**Interpretation**
$percent_amino_acid(P, C1, \ldots, C21)$	C1,. . . ,C21 represent the compositions of the 21 amino acids in sequence P
$molecular_weight(P, A)$	A is the molecular weight of sequence P
$theoretical_pi(P, A)$	A is the predicted isoelectric points (Theoretical PI) of sequence P
$instability_index(P, A)$	A is the instability index of sequence P
$extinction_coefficients(P, A)$	A is the extinction coefficient of sequence P
$aliphatic_value(P, A)$	A is the aliphatic value of sequence P
$gravy_value(P, A)$	A is the grand average of hydropathicity index of sequence P
$percent_hydro_com(P, C1, C2, C3)$	C1, C2 and C3 are compositions of group 1, 2 and 3 respectively in the sequence P
$percent_hydro_tran(P, T1, T2, T3)$	T1, T2 and T3 are transition percentages of group 1$\Longleftrightarrow$2, 1$\Longleftrightarrow$3, and 2$\Longleftrightarrow$3, respectively in the sequence P
$location_dis_hydro_1(P, D1, \ldots, D5)$	D1,. . . ,D5 are the locations that 0%,25%, 50%, 75%, 100% of the group 1 appear in the sequence P divided by the sequence length
$location_dis_hydro_2(P, D1, \ldots, D5)$	D1,. . . ,D5 are the locations that 0%,25%, 50%, 75%, 100% of the group 2 appear in the sequence P divided by the sequence length
$location_dis_hydro_3(P, D1, \ldots, D5)$	D1,. . . ,D5 are the locations that 0%,25%, 50%, 75%, 100% of the group 3 appear in the sequence P divided by the sequence length
$percent_vanderwaals_com(P, C1, C2, C3)$	C1, C2 and C3 are compositions of group 1, 2 and 3 respectively in the sequence P
$percent_vanderwaals_tran(P, T1, T2, T3)$	T1, T2 and T3 are transition percentages of group 1$\Longleftrightarrow$2, 1$\Longleftrightarrow$3, and 2$\Longleftrightarrow$3, respectively in the sequence P
$location_dis_vanderwaals_1(D1, \ldots, D5)$	D1,. . . ,D5 are the locations that 0%,25%, 50%, 75%, 100% of the group 1 appear in the sequence P divided by the sequence length
$location_dis_vanderwaals_2(D1, \ldots, D5)$	D1,. . . ,D5 are the locations that 0%,25%, 50%, 75%, 100% of the group 2 appear in the sequence P divided by the sequence length
$location_dis_vanderwaals_3(D1, \ldots, D5)$	D1,. . . ,D5 are the locations that 0%,25%, 50%, 75%, 100% of the group 3 appear in the sequence P divided by the sequence length

Table 3.8: **Background knowledge: Global Attributes 2**

Predicates	**Interpretation**
$percent_polarity_com(P, C1, C2, C3)$	C1, C2 and C3 are compositions of group 1, 2 and 3 respectively in the sequence P
$percent_polarity_tran(P, T1, T2, T3)$	T1, T2 and T3 are transition percentages of group 1$\Longleftrightarrow$2, 1$\Longleftrightarrow$3, and 2$\Longleftrightarrow$3, respectively in the sequence P
$location_dis_polarity_1(P, D1, \ldots, D5)$	D1,. . . ,D5 are the locations that 0%,25%, 50%, 75%, 100% of the group 1 appear in the sequence P divided by the sequence length
$location_dis_polarity_2(P, D1, \ldots, D5)$	D1,. . . ,D5 are the locations that 0%,25%, 50%, 75%, 100% of the group 2 appear in the sequence P divided by the sequence length
$location_dis_polarity_3(P, D1, \ldots, D5)$	D1,. . . ,D5 are the locations that 0%,25%, 50%, 75%, 100% of the group 3 appear in the sequence P divided by the sequence length
$percent_polarizability_com(P, C1, C2, C3)$	C1, C2 and C3 are compositions of group 1, 2 and 3 respectively in the sequence P
$percent_polarizability_tran(P, T1, T2, T3)$	T1, T2 and T3 are transition percentages of group 1$\Longleftrightarrow$2, 1$\Longleftrightarrow$3, and 2$\Longleftrightarrow$3, respectively in the sequence P
$location_dis_polarizability_1(P, D1, \ldots, D5)$	D1,. . . ,D5 are the locations that 0%,25%, 50%, 75%, 100% of the group 1 appear in the sequence P divided by the sequence length
$location_dis_polarizability_2(P, D1, \ldots, D5)$	D1,. . . ,D5 are the locations that 0%,25%, 50%, 75%, 100% of the group 2 appear in the sequence P divided by the sequence length
$location_dis_polarizability_3(P, D1, \ldots, D5)$	D1,. . . ,D5 are the locations that 0%,25%, 50%, 75%, 100% of the group 3 appear in the sequence P divided by the sequence length
$percent_amino_acid_pi_com(P, C1, C2, C3)$	C1, C2 and C3 are compositions of group 1, 2 and 3 respectively in the sequence P
$percent_amino_acid_pi_tran(P, T1, T2, T3)$	T1, T2 and T3 are transition percentages of group 1$\Longleftrightarrow$2, 1$\Longleftrightarrow$3, and 2$\Longleftrightarrow$3, respectively in the sequence P

Table 3.9: **Background knowledge: Global Attributes 3**

Predicates	**Interpretation**
$location_dis_amino_acid_pi_1(P, D1, \ldots, D5)$	D1,. . . ,D5 are the locations that 0%,25%, 50%, 75%, 100% of the group 1 appear in the sequence P divided by the sequence length
$location_dis_amino_acid_pi_2(P, D1, \ldots, D5)$	D1,. . . ,D5 are the locations that 0%,25%, 50%, 75%, 100% of the group 2 appear in the sequence P divided by the sequence length
$location_dis_amino_acid_pi_3(P, D1, \ldots, D5)$	D1,. . . ,D5 are the locations that 0%,25%, 50%, 75%, 100% of the group 3 appear in the sequence P divided by the sequence length
$percent_charge_com(P, C1, C2, C3)$	C1, C2 and C3 are compositions of group 1, 2 and 3 respectively in the sequence P
$percent_charge_tran(P, T1, T2, T3)$	T1, T2 and T3 are transition percentages of group 1$\Longleftrightarrow$2, 1$\Longleftrightarrow$3, and 2$\Longleftrightarrow$3, respectively in the sequence P
$location_dis_charge_1(P, D1, \ldots, D5)$	D1,. . . ,D5 are the locations that 0%,25%, 50%, 75%, 100% of the group 1 appear in the sequence P divided by the sequence length
$location_dis_charge_2(P, D1, \ldots, D5)$	D1,. . . ,D5 are the locations that 0%,25%, 50%, 75%, 100% of the group 2 appear in the sequence P divided by the sequence length
$location_dis_charge_3(P, D1, \ldots, D5)$	D1,. . . ,D5 are the locations that 0%,25%, 50%, 75%, 100% of the group 3 appear in the sequence P divided by the sequence length
$percent_aromatic_com(P, C1, C2, C3)$	C1, C2 and C3 are compositions of group 1, 2 and 3 respectively in the sequence P
$percent_aromatic_tran(P, T1, T2, T3)$	T1, T2 and T3 are transition percentages of group 1$\Longleftrightarrow$2, 1$\Longleftrightarrow$3, and 2$\Longleftrightarrow$3, respectively in the sequence P

Table 3.10: **Background knowledge: Global Attributes 4**

Predicates	**Interpretation**
$location_dis_aromatic_1(P, D1, \ldots, D5)$	D1,. . . ,D5 are the locations that 0%,25%, 50%, 75%, 100% of the group 1 appear in the sequence P divided by the sequence length
$location_dis_aromatic_2(P, D1, \ldots, D5)$	D1,. . . ,D5 are the locations that 0%,25%, 50%, 75%, 100% of the group 2 appear in the sequence P divided by the sequence length
$location_dis_aromatic_3(P, D1, \ldots, D5)$	D1,. . . ,D5 are the locations that 0%,25%, 50%, 75%, 100% of the group 3 appear in the sequence P divided by the sequence length
$percent_sulfur_com(P, C1, C2, C3)$	C1, C2 and C3 are compositions of group 1, 2 and 3 respectively in the sequence P
$percent_sulfur_tran(P, T1, T2, T3)$	T1, T2 and T3 are transition percentages of group 1$\Longleftrightarrow$2, 1$\Longleftrightarrow$3, and 2$\Longleftrightarrow$3, respectively in the sequence P
$location_dis_sulfur_1(P, D1, \ldots, D5)$	D1,. . . ,D5 are the locations that 0%,25%, 50%, 75%, 100% of the group 1 appear in the sequence P divided by the sequence length
$location_dis_sulfur_2(P, D1, \ldots, D5)$	D1,. . . ,D5 are the locations that 0%,25%, 50%, 75%, 100% of the group 2 appear in the sequence P divided by the sequence length
$location_dis_sulfur_3(P, D1, \ldots, D5)$	D1,. . . ,D5 are the locations that 0%,25%, 50%, 75%, 100% of the group 3 appear in the sequence P divided by the sequence length
$range(Lo =< Int =< Hi)$	the range of an integer value restricted from both sides
$range_l(Int =< Hi)$	the range of an integer value restricted from left side
$range_r(Lo =< Int)$	the range of an integer value restricted from right side
$interval(Lo =< Float =< Hi)$	the range of a float value restricted from both sides
$interval_l(Float =< Hi)$	the range of a float value restricted from left side
$interval_r(Lo =< Float)$	the range of a float value restricted from right side

Table 3.11: **Background knowledge: Relational Attributes**

Predicates	**Interpretation**
$total_length(Lo =< P =< Hi)$	the total number of amino acids.
$distance(P, A, B, Dist, TypA, TypB)$	this predicate indicates that the secondary structures A and B have a distance of Dist. Furthermore, their respective types are TypA and TypB
$has_helix(P, A)$	Sequence P contains an α-helix A
$has_strand(P, A)$	Sequence P contains an β-strand A
$helix(A, Pos, Len)$	the secondary structure α-helix A starts at position Pos and has length Len
$strand(A, Pos, Len)$	the secondary structure β-strand A starts at position Pos and has length Len
$percent_helix(P, Per)$	α-helix composition
$percent_strand(P, Per)$	β-strand composition
$percent_coil(P, Per)$	coil composition
$percent_helix_strand(P, Per)$	transition composition: from helix $\Longleftrightarrow$ strand
$percent_helix_coil(P, Per)$	transition composition: from helix $\Longleftrightarrow$ coil
$percent_strand_coil(P, Per)$	transition composition: from strand $\Longleftrightarrow$ coil
$location_dis_helix(P, Per1, \ldots, Per5)$	α-helix distribution composition. Identify the relative location of the α-helix (position/sequence length) that with a value of 0%, 25%, 50%, 75% and 100%
$location_dis_strand(P, Per1, \ldots, Per5)$	β-strand distribution composition. Identify the relative location of the β-strand (position/sequence length) that with a value of 0%, 25%, 50%, 75% and 100%
$location_dis_coil(P, Per1, \ldots, Per5)$	coil distribution composition. Identify the relative location of the coil (position/sequence length) that with a value of 0%, 25%, 50%, 75% and 100%

Table 3.12: **Positive Examples Covered by Rules: Inverse Frequency**

Positive Examples	Frequency	Name
cg5102_pa	1	daughterless
cg11094_pa	3	doublesex
cg11094_pb	3	doublesex
cg11094_pc	3	doublesex
cg3496_pa	1	virilizer
cg8704_pa	2	deadpan
cg9019_pa	1	dissatisfaction
cg1007_pa	1	extra macrochaetae
cg6315_pa	3	female lethal d
cg6315_pb	2	female lethal d
cg14307_pa	1	fruitless
cg14307_pb	1	fruitless
cg14307_pc	2	fruitless
cg14307_pd	2	fruitless
cg14307_pe	2	fruitless
cg14307_pf	2	fruitless
cg14307_pg	1	fruitless
cg14307_ph	1	fruitless
cg8384_pa	1	groucho
cg8384_pb	1	groucho
cg8384_pc	1	groucho
cg8384_pd	1	groucho
cg4694_pa	0	hermaphrodite
cg13201_pa	0	intersex
cg11680_pa	1	maleless
cg11680_pb	1	maleless
cg3025_pa	1	males absent on the first
cg10385_pa	0	male_specific lethal 1
cg3241_pa	1	male_specific lethal 2
cg3241_pb	1	male_specific lethal 2
cg8631_pa	2	male_specific lethal 3
cg8631_pb	2	male_specific lethal 3
cg12743_pa	1	ovarian tumor
cg12743_pb	1	ovarian tumor
cg6824_pa	1	ovo
cg6824_pb	2	ovo
cg6824_pc	1	ovo
cg1849_pa	1	runt
cg1641_pa	0	sisterless A
cg3827_pa	0	scute
cg5993_pa	1	outstretched
cg4528_pa	1	sans fille
cg10582_pa	1	Sex_lethal interactor
cg16724_pa	2	transformer
cg10128_pa	1	transformer 2
cg10128_pb	1	transformer 2
cg10128_pc	1	transformer 2
cg10128_pd	1	transformer 2
cg10128_pe	1	transformer 2

Chapter 4

Materials and Methods

A number of computational approaches have been proposed to predict protein-protein interactions, including those based on genomic information [51, 192], integration of multiple genomic datasets [94, 113] and literature mining [120]. Protein-protein interactions can also be predicted on the basis of evolutionary relationship. It has been shown that interacting proteins often exhibit coordinated evolution, so that proteins with similar phylogenetic trees are more likely to interact with each other [151, 66, 161]. In addition, the concept of 'interologs' has been proposed based on the idea that a pair of interacting proteins are co-evolving so that their respective orthologs in other organisms tend to interact as well [202].

Several methods have been proposed to predict protein interactions in *Saccharomyces cerevisiae* on the basis of another important principle, namely, domain-domain interactions. The protein domain as a unit of structure, function and evolution also serves as a unit for protein-protein interactions. Therefore, it is important to

take into account domain-domain interactions when we infer plausible interacting protein pairs. In these methods, proteins are characterized by one or more domains and each domain is responsible for a specific interaction with another domain. Sprinzak [182] identified the domain pairs that are highly correlated with interacting protein pairs using protein-protein interaction data from S.cerevisiae as training data. The information was further used to predict interacting protein pairs that contain an interacting domain pair. Similarly, Gomez [67, 68] and Deng [41] estimated the probabilities of domain-domain interactions using protein-protein interaction data from S.cerevisiae as training data; the estimated domain-domain interaction probabilities can be used to infer protein-protein interaction probabilities. These methods depend highly on the accuracy of the training data and have been mostly applied to protein-protein interaction data from a single organism only, which may be inferior to methods that can incorporate more information in estimating domain-domain interaction probabilities.

Because domains are likely evolutionarily conserved, information from multiple organisms may be integrated together to improve the estimation of domain-domain interaction probabilities. In our study, we incorporate information from three organisms, *S. cerevisiae*, *Caenorhabditis elegans* and *Drosophila melanogaster*, to effectively utilize the domain information as the evolutionary connection among these model organisms. The protein-domain relationship can be extracted from relevant databases

such as PFAM and SMART [14, 112]. By integrating large-scale protein-protein interaction data from these three organisms, we have extended a likelihood approach proposed by Deng [41] to estimate the probabilities of domain-domain interactions based on information from all three organisms. Considering each protein as a collection of domains, we can then estimate the probabilities of protein-protein interactions in *S. cerevisiae* based on the inferred domain-domain interaction probabilities. The protein pairs with interaction probabilities above a certain threshold can then be predicted to interact with each other. In order to assess the performance of our method, we first apply it to the interaction data from *S. cerevisiae* only and compare its performance with that of three other methods that predict protein interactions based on the domain composition of proteins in the cross-validation measurement, and we demonstrate that our method provides comparable performance to the others. Then, we compare our prediction results based on all five organisms with those based on *S. cerevisiae* alone. We find that the integrated analysis provides more reliable inference of protein-protein interactions than the analysis from a single organism based on the analysis of sensitivity and specificity, Gene Ontology term enrichment and gene expression profiles.

Our strategy is to use multiple methods and data sources to independently derive protein-protein interactions, then compare the results obtained.

4.1 Data

In our experiments we use the interaction data from published literature, DIP, InterDom and MIPS. The *D. melanogaster* protein interaction map was published in [65]. A two hybrid-based protein-interaction map of the fly proteome was published and a total of 10,623 predicted transcripts were isolated and screened against standard and normalized complementary DNA libraries to produce a draft map of 7,048 proteins and 20,405 interactions. A computational method of rating two-hybrid interaction confidence was developed to refine this draft map to a higher confidence map of 4,679 proteins and 4,780 interactions. Statistical modelling of the network showed two levels of structure: small clusters, presumably corresponding to multi-protein complexes, and bigger clusters, presumably corresponding to inter-complex connections. The network recapitulated known pathways, extended pathways, and uncovered previously unknown pathway components. This map serves as a starting point for a systems biology modelling of multicellular organisms including mosquitoes.

4.2 Feature Extraction

In order to represent a pair of proteins, we consider two different types of features. The first is computed using a collection of hidden Markov models (HMMs) of protein

domains from the Pfam database [14]. These models represent evolutionarily conserved structures and are assumed to be related to protein function. Using HMMER 2.0 [50], we compute the E-value of the best match of each Pfam domain to each yeast protein. Each such E-value serves as one feature in this representation. The Pfam E-values, as well as the interaction data, are used to reduce the size of the data set. Following [182], we eliminate from consideration all proteins that do not match at least one Pfam model with an E-value less than 0.01. From within this set, we select all proteins that interact with at least one other protein. A total of 1,714 proteins, containing 1,015 different domain types, satisfy these criteria. The resulting set of 1.46 million protein pairs contains 7,735 positive interactions and 19,315 unknown interactions. The unknown interactions are not considered further. This collection of 1,714 proteins is also characterized using a second type of feature. Rather than identifying protein domains, these 4-tuple features attempt to identify short amino acid subsequences that occur in interacting proteins. To compute these features, the sequence alphabet is first reduced from 20 amino acids to six categories of biochemical similarity IVLM, FYW, HKR, DE, QNTP, and ACGS [95]. After this reduction, there are 1,296 possible substrings of length 4. For a given protein sequence, the 4-tuple feature representation is simply a binary vector of length 1,296, in which each bit indicates whether the corresponding length-4 string occurs in the protein.

4.3 Building protein-protein interaction map with orthologous proteins transfer method

The transfer of annotations between two different species using protein-protein interaction method has been reported in several reports, however the combination of orthologous clusters and the protein-protein interaction network of one species to infer that of another species has seldom been seen in literature.

Recently comparative genomic analysis of two distant diptera has been performed by several groups [214, 19, 176]. Similarly, in a comparative genomics aspect, our work has been focusing on predict the protein interaction networks of *A. gambiae* from the protein interaction maps of *D. melanogaster*, with various approaches.

Concepts of orthology and paralogy are become increasingly important as whole-proteome comparison allows their identification. Functional specificity of proteins is assumed to be conserved among orthologs and in-paralogs and is different among out-paralogs. We used this assumption to identify corresponding proteins of different species. Finding such proteins is crucial for understanding protein annotation transfer between different species. The orthologous and in-paralogous proteins information was collected from the Inparanoid database.

4.3.1 The Inparanoid database

Inparanoid database includes the information for 17 organisms and 388,912 sequences as of version 3.0, release on 15 August 2004. Particularly for the ortholog groups between *D. melanogaster* and *A. gambiae*, there are 7,259 clusters, among which 7,724 *D. melanogaster* proteins and 7,993 *A. gambiae* proteins. Inparanoid is available at

4.3.2 The algorithm

For each of the ortholog cluster, there are m *D. melanogaster* proteins and n *A. gambiae* proteins. When m is 1, $n \geq 1$; When $m \geq 1$, n must be 1. The program reads the *D. melanogaster* whole protein interaction map and for each protein interaction pair, it replaces the pair with all the corresponding orthologous *A. gambiae* proteins. If a protein in *D. melanogaster* has no orthologous protein(s) in *A. gambiae*, the program will skip this protein-protein interaction pair in our current design. Further improvements to include such protein-protein interaction pairs will be provided in the next step of the project. The algorithm is illustrated in Fig. 4.1.

We also give an example to explain the algorithm, which is demonstrated in Fig. 4.2.

We have downloaded the *D. melanogaster* protein interaction pairs from latest research in Science (Fig 4.3).

We check one protein-protein interaction pair in *D. melanogaster*, namely CG11094 and CG11154 and their normalised confidence score is 18.269999 (Fig 4.4).

We observe from the inparanoid database that the proteins in blue color (*D. melanogaster*) and red color (*Anopheles*) are put into different clusters according to certain algorithms. By definition of the database, it is quite possible that proteins in the same cluster share similar functions, i.e., proteins in cluster 6335 should have similar properties, so should cluster 1031. Now we can transfer the interaction relation from CG11094 - CG11154 to ENSANGP00000004060 - ENSANGP00000016868 (Fig 4.5).

Considering the confidence score in both Fig 4.4 and Fig 4.5 , we have the following result in Fig 4.6.

Finally, we plot the protein-protein interaction map we predicted Fig 4.7 and a sub-graph Fig 4.9 which is related to some sex-determination related proteins. We also plot the sex-determination related proteins of *Drosophila melanogaster* in Fig 4.8. We can see that the only intersection of Fig 4.9 and Fig 4.8 is the Doublesex protein (CG11094 in *Drosophila melanogaster* and ENSANGP00000029176 in *Anopheles*

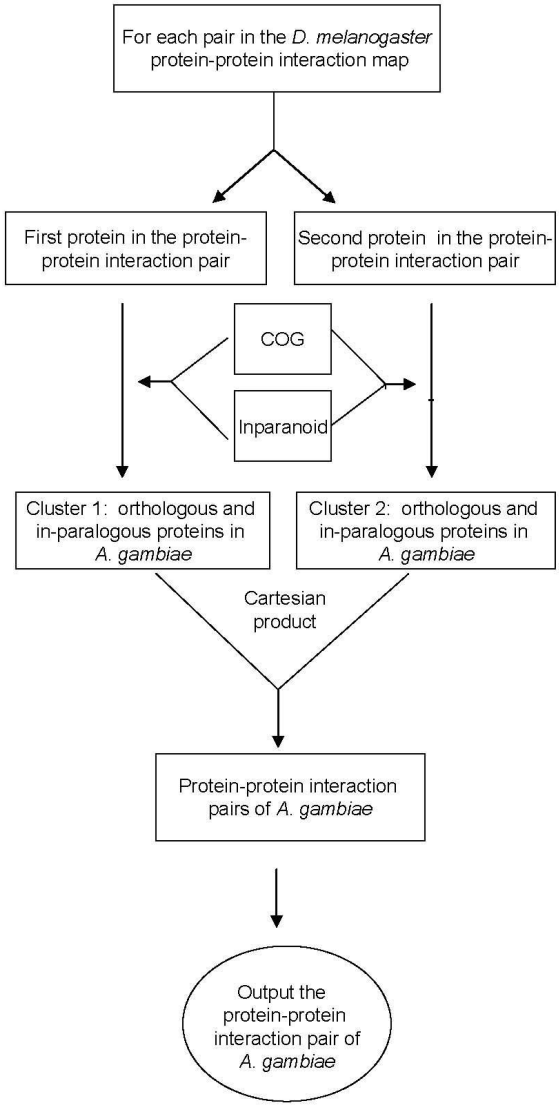


Figure 4.1: Overview of the orthologs transfer algorithm.

Inparalog and Orthologs clusters for Drosophila melanogaster and Anopheles gambiae (7259 clusters in total)

Page 1 of 1 Show page

Cluster #1

Protein ID	Species	Score	Bootstrap	Name
CG18076-PH	D. melanog.	1.0000	100%	CG18076-PC
ENSANGP00000011363	A. gambiae	1.0000	100%	Novel gene ENSANGG00000008874

Cluster #5572

Protein ID	Species	Score	Bootstrap	Name
CG5224-PA	D. melanog.	1.0000	72%	CG5224 protein (LD18692p)
ENSANGP00000025069	A. gambiae	1.0000	96%	Glutathione S-transferase 3-1
ENSANGP00000025106	A. gambiae	0.3980		Glutathione S-transferase 3-2
ENSANGP00000024962	A. gambiae	0.1040		GLUTATHIONE S TRANSFERASE EC_2.5.1.18 GST CLASS THETA

Figure 4.2: Clusters for two species from Inparanoid database

Supplemental Table S7. Drosophila interactions

cg1	cg2	p1	p2	norient	baitprey	preybait	cdna	collection	baitrgn	bait5utr	baitcds	maxneigh	geomneigh	nbridge	jac	hexpert	yexpert	cexpert	cconf
CG10002	CG10032	fkh	CG10032	1	1	0	0	1	CDS_F1:1	0	1	62	30.4959	0	0	NA	NA	NA	0.0624
CG10002	CG11899	fkh	ESTS:39C10S	1	1	0	1	0	CDS_F1:1	0	1	15	10.9545	0	0	NA	NA	NA	0.1697
CG10002	CG15224	fkh	Cklibeta	1	1	0	0	1	CDS_F1:1	0	1	43	25.3969	0	0	NA	NA	NA	0.1116
CG10002	CG15529	fkh	CG15529	1	4	0	0	4	CDS_F1:4	0	4	23	18.5742	0	0	NA	NA	NA	0.3413
CG10002	CG17419	fkh	CG17419	1	1	0	1	0	CDS_F1:1	0	1	19	16.8819	1	0.03225806	NA	NA	NA	0.4194
CG10002	CG17666	fkh	CG17666	1	1	0	0	1	CDS_F1:1	0	1	108	40.2492	0	0	NA	NA	NA	0.0348
CG10002	CG17945	fkh	Mst84Dc	1	1	0	0	1	CDS_F1:1	0	1	29	20.8567	0	0	NA	NA	NA	0.1071
CG10002	CG17946	fkh	Mst84Da	1	1	0	0	1	CDS_F1:1	0	1	20	17.3205	0	0	NA	NA	NA	0.0959
CG10002	CG1925	fkh	mus205	1	1	0	1	0	CDS_F1:1	0	1	15	10.2470	0	0	NA	NA	NA	0.1368
CG10002	CG31054 | CG4849	fkh	CG31054 BG:DS04096.	1	2	0	2	0	CDS_F1:2	0	2	15	8.8603	0	0	NA	NA	NA	0.2574
CG10002	CG4959	fkh	3	1	1	0	0	1	CDS_F1:1	0	1	55	28.7228	0	0	NA	NA	NA	0.0469
CG10002	CG6238	fkh	ssh	1	1	0	1	0	CDS_F1:1	0	1	15	7.7460	0	0	NA	NA	NA	0.0519
CG10002	CG6459	fkh	CG6459	1	36	0	24	12	CDS_F1:36	0	36	64	30.9839	1	0.01315789	NA	NA	NA	0.9974
CG10002	CG7081	fkh	CG7081	1	1	0	1	0	CDS_F1:1	0	1	15	5.4772	0	0	NA	NA	NA	0.2905
CG10002	CG9572	fkh	CG9572	1	1	0	0	1	CDS_F1:1	0	1	64	28.4605	0	0	NA	NA	NA	0.0590
CG10007	CG11205	CG10007	phr	1	0	2	0	2	CDS_F1:2	0	2	5	3.8730	0	0	NA	NA	NA	0.7376
CG10007	CG17686	CG10007	DIP1	1	1	0	1	0	CDS_F1:1	0	1	3	2.4495	0	0	NA	NA	NA	0.3363
CG10007	CG32491	CG10007	mod(mdg4)	1	0	6	6	0	CDS_F1:6	0	6	31	9.6437	0	0	NA	NA	NA	0.1991
CG10009	CG1133	Noa36	opa	1	0	1	0	1	CDS_F3:1	0	0	30	18.1659	0	0	NA	NA	NA	0.1013
CG10009	CG11835	Noa36	CG11835	1	0	9	0	9	CDS_F1:9	0	9	22	15.5563	0	0	NA	NA	NA	0.6199
CG10009	CG12679	Noa36	CG12679	1	0	4	0	4	CDS_F1:4	0	4	15	12.8452	0	0	NA	NA	NA	0.3805
CG10009	CG14546	Noa36	CG14546	1	0	1	1	0	CDS_F1:1	0	1	27	17.2337	0	0	NA	NA	NA	0.0722
CG10009	CG1792	Noa36	CG1792	1	0	1	0	1	CDS_F1:1	0	1	13	11.9583	0	0	NA CDS_F1:3	NA	NA	0.1984

Figure 4.3: Known Drosophila interactions

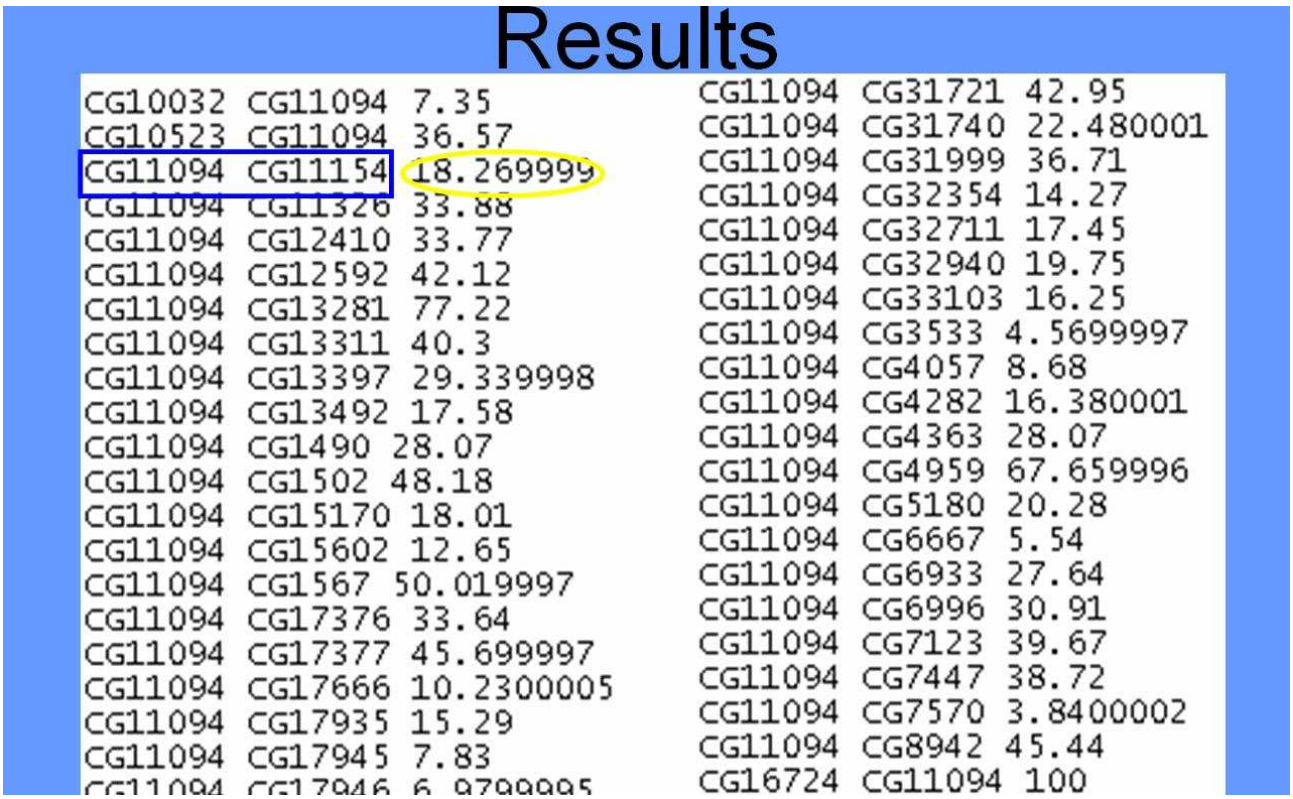


Figure 4.4: A pair of interacting proteins from Drosophila

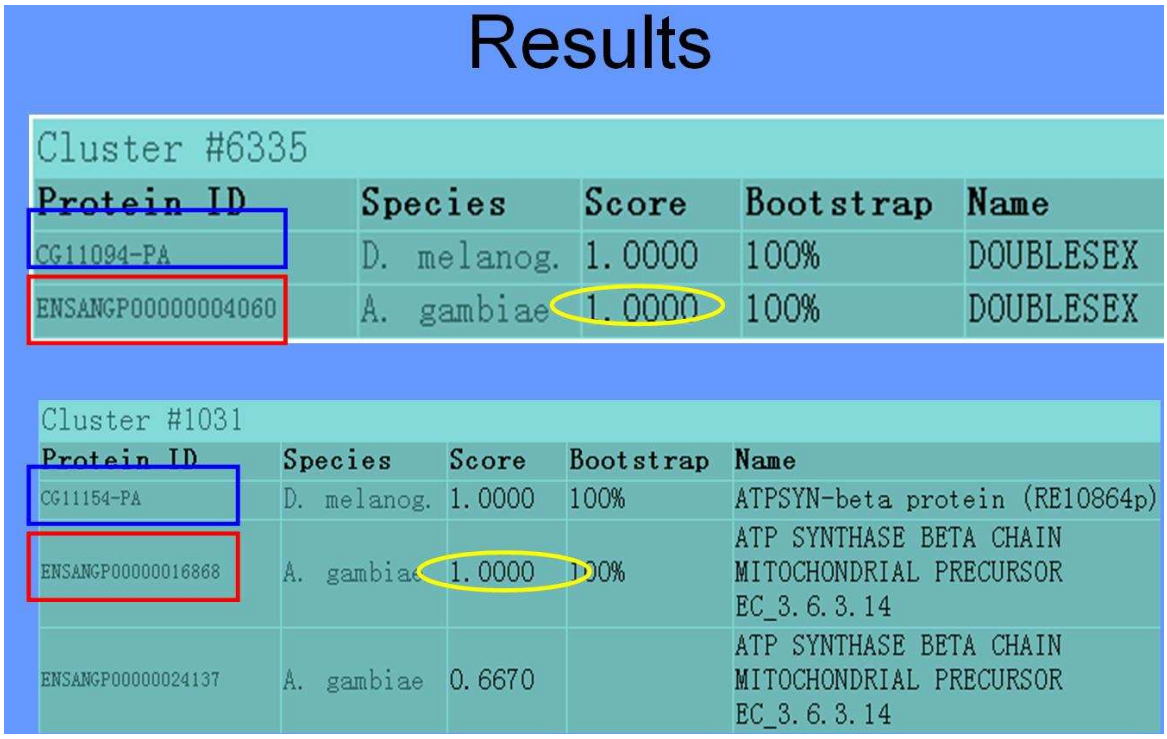


Figure 4.5: Highlighted clusters for two species from Inparanoid database



Figure 4.6: A pair of interacting proteins inferred from the above algorithm

gambiae. Thus the significance and efficiency of this method is further proved to be doubtable.

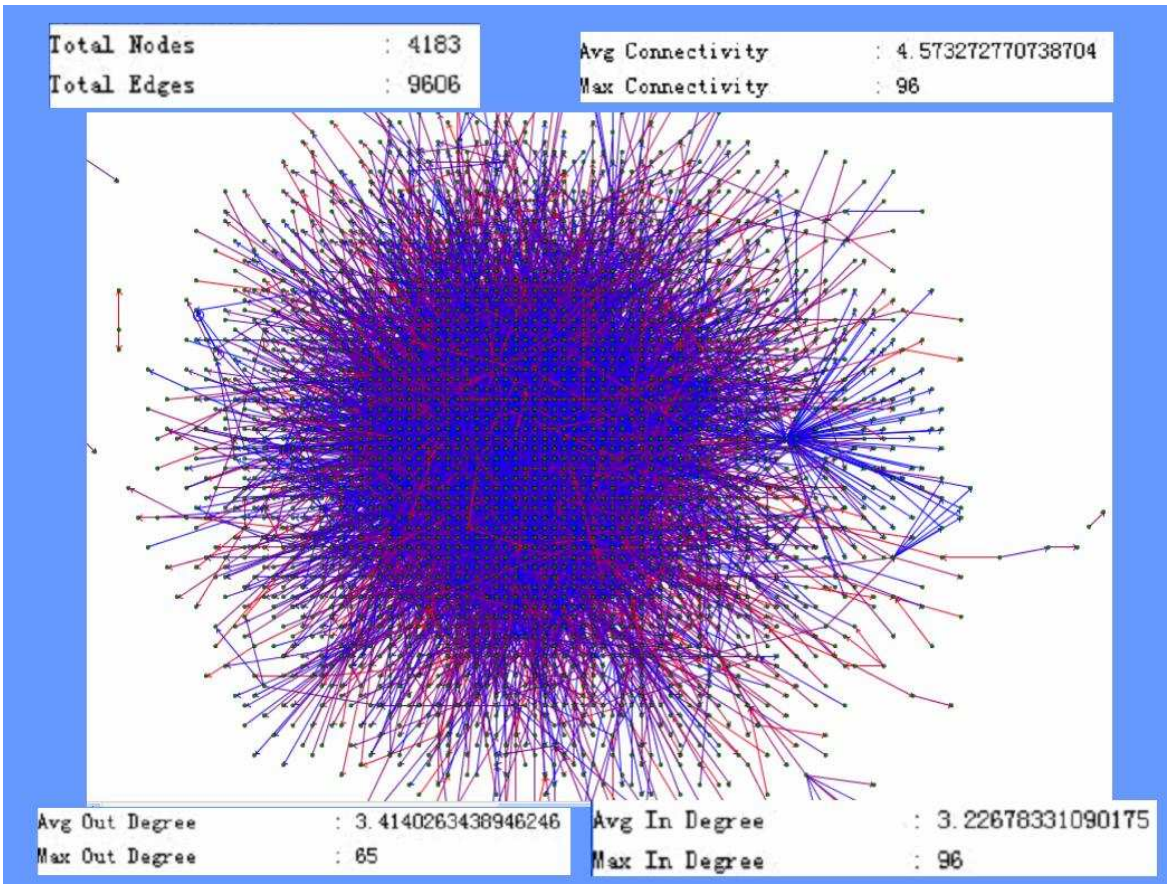


Figure 4.7: Protein interaction map for Anopheles

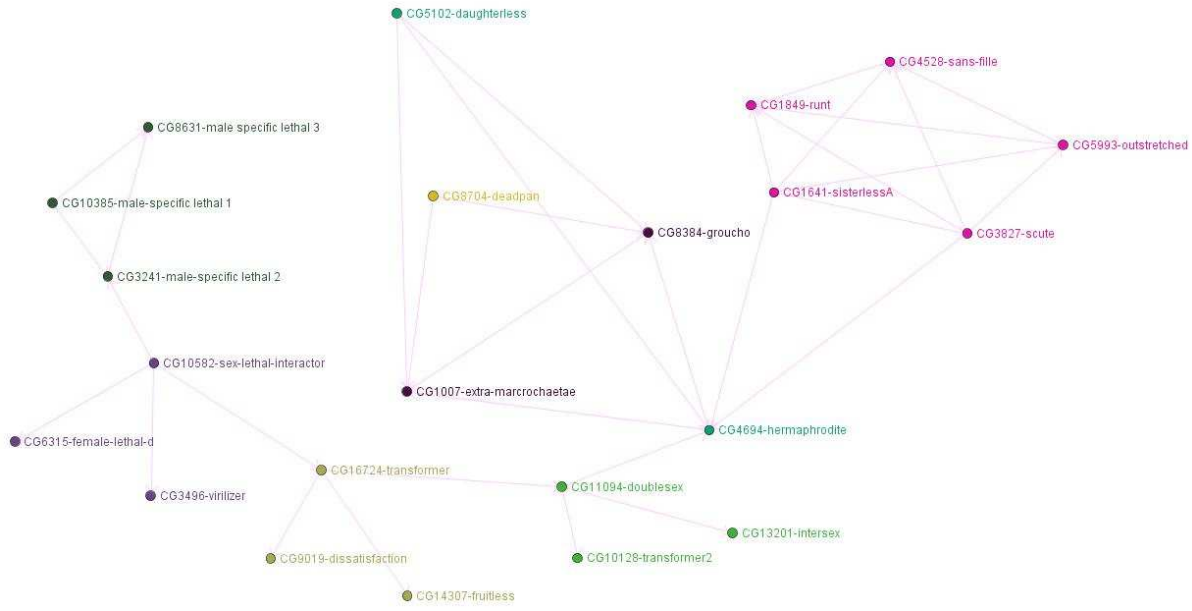


Figure 4.8: Sex-determination related proteins of *Drosophila melanogaster*

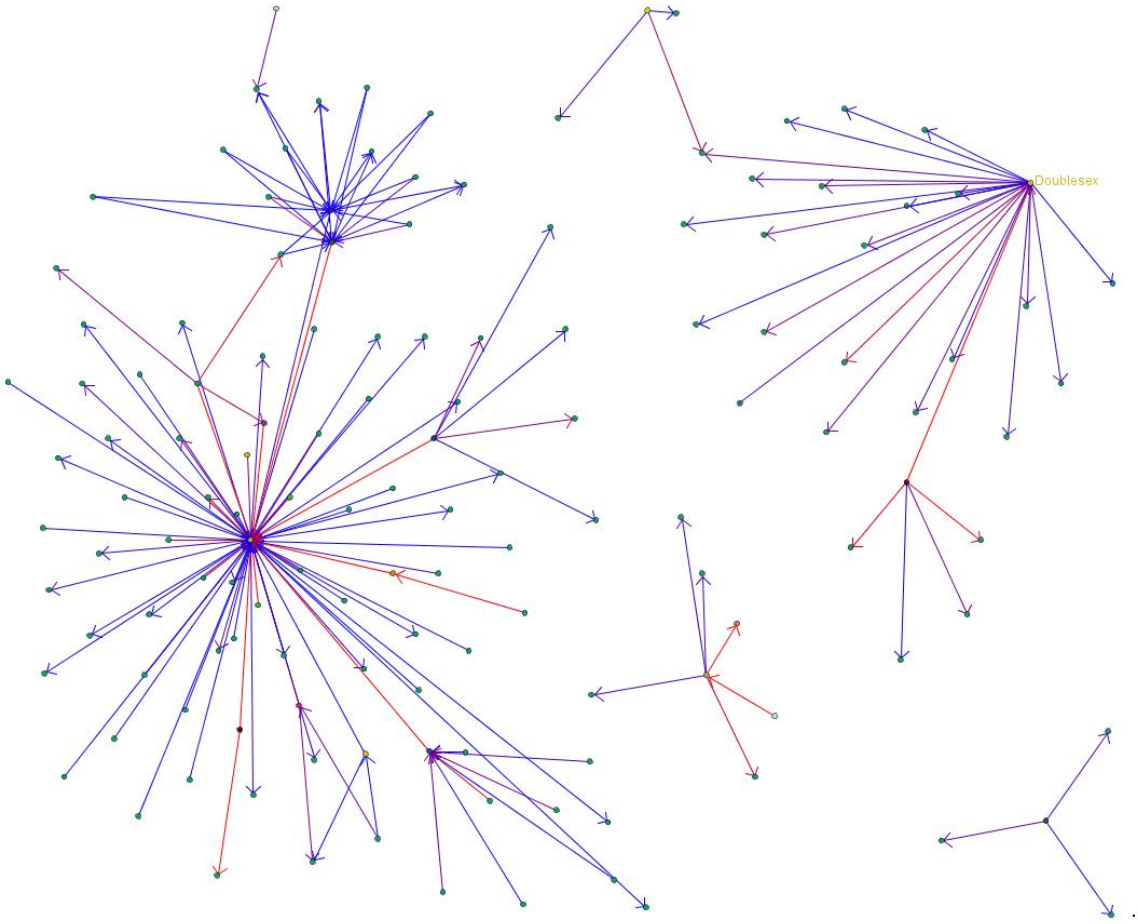


Figure 4.9: Sex-determination related proteins of *Anopheles gambiae*

4.4 Hybrid Bayes method

4.4.1 Introduction

In [94], Jansen et al have developed an approach using Bayesian networks [79] to predict protein-protein interactions genome-wide in yeast. However, this method is based on genomic features such as messenger RNA co-expression, coessentiality, and co-localization). These genomic feature data are neither available for *A. gambiae*, nor does their modelling process share any similarity with our hybrid bayes method other than the basic formula being Bayes.

In a review at [206], Wilkinson introduces recent developments in Bayesian bioinformatics relevant to computational systems biology. Unfortunately, the review is very short and does not have detailed introduction and comparisons to popular Bayesian modelling methods on biological data. Actually, most of the current methods for protein-protein interaction prediction or function assignment rely heavily on genomic data information [94] or protein structure data (2-d and 3-d) [11, 147], which are not available in our case.

Our proposed model described here assesses the probability of a protein-protein interaction by drawing upon what is known of previously observed interactions. The model is first trained from a set of trusted data consisting of a list of known (or at least putatively known) protein-protein interactions and a list of features for each of

the proteins in an interaction. Features can be anything, from stretches of identical charge to structural domains. In practice Pfam domains have been found to be generally informative. The main assumption about features is that they are evolutionarily conserved, and that it is the presence of a subset of these features that are responsible for establishing various interactions. In the process of training, each interaction between proteins in the training data is analyzed by counting the number of times each pair of protein features is observed. If a pair of domains is responsible for an interaction, we would expect to see an increase in the number of times such a pair is seen within the data. From such counts, probabilities are generated for every pair of domain-domain interactions. When given a novel set of proteins with feature information, and using the information collected in the training stage, we can now assign a probability to all possible interactions between these proteins. Again, domain pairs enriched in the training data will provide greater support for an interaction in the novel set of proteins. Rankings of predicted interactions based on probability can then be used in guiding further investigations.

More specifically, this approach is based on the treatment of proteins as collections of conserved domains, where each domain is responsible for a specific interaction with another domain. By characterizing the frequency with which specific domain-domain interactions occur within known interactions, our model can assign a probability to an arbitrary interaction between any two proteins with defined domains. Domain

interaction data is complemented with information on the topology of a network and is incorporated into the model by assigning greater probabilities to networks displaying more biologically realistic topologies. We then use the Markov chain Monte Carlo techniques for the prediction of posterior probabilities of interaction between a set of proteins.

A primary goal of this work is to provide a method for generating predictions that would be useful to the experimental biology community. In particular we feel that the prediction of molecular interactions, along with the ability to assign a probability to a given interaction, could be of significant benefit in the generation of new hypotheses and the prioritizing of appropriate (and perhaps more focused) experiments.

We use a likelihood approach to estimating domain-domain interaction probabilities by integrating large-scale protein interaction data from three organisms, *Saccharomyces cerevisiae*, *Caenorhabditis elegans* and *Drosophila melanogaster*. The estimated domain-domain interaction probabilities are then used to predict protein-protein interactions in *S. cerevisiae*. Furthermore the inferred results combined with the data from DIP and MIPS are compared with those obtained from the orthologous transfer method to evaluate the accuracy of the method. Our predicted protein-protein interactions have a significant overlap with the protein-protein interactions published at MIPS(http://mips.gfs.de) and DIP (http://dip.doe-mbi.ucla.edu/).

4.4.2 Hybrid Bayes method - from protein-protein interaction to domain-domain interaction

We have a training set which contains a known protein-protein interaction network from *D. melanogaster*, and we also know what domains are included in a single protein.

We represent our protein-protein interaction network with undirected graph. An undirected graph G is an ordered pair $G: = (V,E)$ that is subject to the following conditions:

- V is a set of vertices or nodes,
- E is a set of pairs (unordered) of distinct vertices, called edges or lines.
- The vertices belonging to an edge are called the ends, endpoints, or end vertices of the edge.

In our work, each node V_i in the graph G represents a protein, which is $V: = \{P_1,P_2,...,P_i,...,\}$. Each edge A_{st}, denotes an interaction between two proteins P_i and P_j, and $E: = \{A_{11},...,A_{st},...\}$. Thus an edge connects the proteins P_s and P_t with the probability of A_{st}, or not connecting with the value of $(1 - A_{st})$.

We treat each protein as a collection of domains, and each of these domains has a tendency to attract other domains between distinct proteins. We define a probability

of attraction $Pr(D_i, D_j)$ that exists for each upstream and downstream domain D_i and D_j, respectively.

Furthermore we have the following definitions. k_{ij} is the number of edges in the training set that contain at least one domain D_i at the vertex of edge origin and at least one domain D_j at the vertex of edge destination. k_i is the number of distinct vertices that contain at least one domain D_i, and k_j is the number of distinct vertices that contain at least one domain D_j.

The probability of an edge forming between a pair of proteins is dependent on the relative attraction and repulsion of each protein's complement of domains, taken over all upstream-downstream pairwise combinations. This expression is a reasonable assumption as long as the number of edges incoming to or outgoing from a vertex is independent of the number of domains per protein; This assumption was verified previously [67].

Let A_{st} denote the event that proteins P_s and P_t interact, and B_{ij} represent the event that domains D_i and D_j interact. We follow the conventions in current literature to assume an interaction of domains between a pair of domains is independent of other interactions of domains, and an interaction of proteins between a pair of proteins is independent of other interactions of proteins. We have the following formula:

$$\hat{Pr}(B_{ij}) = \frac{k_{ij} + \phi_1}{k_i k_j + \phi_2} \tag{2.2.1}$$

Where ϕ_1 and ϕ_2 are parameters used to compensate for small samples, and are initially assigned to the value of 1 in our work. ϕ_1 and ϕ_2 are adjusted at the calibration stage. $\hat{Pr}(B_{ij})$ is observed from frequency information to estimate the real probability $Pr(B_{ij})$. $\hat{Pr}(B_{ij})$ will converge to real probability $Pr(B_{ij})$ for large k_i, k_j and k_{ij}.

This expression generates domain attraction probabilities greater than or equal to 0.5. As discussed later, probabilities of less than 0.5 are reserved for future modelling of repulsive interactions between domains, as observed, for example, in domain combination studies [10]. We assume that data supporting the existence of a particular interaction is usually backed by several experiments, while experiments showing the absence of an interaction are generally under-represented by having either failed (and these failures not reported) or have not been performed. Thus this expression does not "penalize" for lack of an interaction, but assumes it to be the lack of supporting data. In the absence of any supporting data, all interactions between domains (and hence proteins) are equally likely.

In summary, we observe the frequency with which domain D_i lies immediately upstream or downstream of domain D_j within experimental protein-protein interaction data. For an arbitrary pair of proteins, each with their own set of domains, we are then able to assign a probability to the likelihood of an edge forming between them. A complete network with a defined set of edges can similarly be assigned a

probability; networks with many favorable edges will have a higher probability than a network with many unlikely edges.

We give an example to illustrate the use of this formula before we proceed presenting and deducting the formulas for our hybrid method.

Table 4.1: **Proteins and domains contained**

Proteins	***Domains contained***
CG18745	*Arrestin_N*, *Arrestin_C*
CG14869	*Pep_M12B_propep*, Reprolysin, *TSP_1*, *ADAM_spacer1*
Y00XX0	*Arrestin_N*, Z00001
Y00YY0	*TSP_1*, *ADAM_spacer1*, Z00002
S000X0	*Arrestin_N*, *Arrestin_C*
S000Y0	Z00003

Most of the proteins have specific function groups, i.e., domains. The interaction of two proteins usually means some domains in one protein interact with some domains in another protein. However from computational view, we don't know for sure which domains actually interact with which domains. Table 4.1 shows a number of proteins and the domains they contain. The known interacting protein pairs are CG18745 and CG14869, Y00XX0 and Y00YY0, S000X0 and S000Y0. Protein 'CG18745' has two domains: *Arrestin_N* and *Arrestin_C*; protein 'CG14869' has four domains: *Pep_M12B_propep*, *Reprolysin*, *TSP_1*, and *ADAM_spacer1*. We are interested in knowing the actual interacting domain pair. The maximum possible domain interactions in this example is 8 but these interactions definitely include many false-positives. A biologically meaningful estimation would be an one-one or at

most a two-two interaction. An effective strategy is needed to reduce the number of

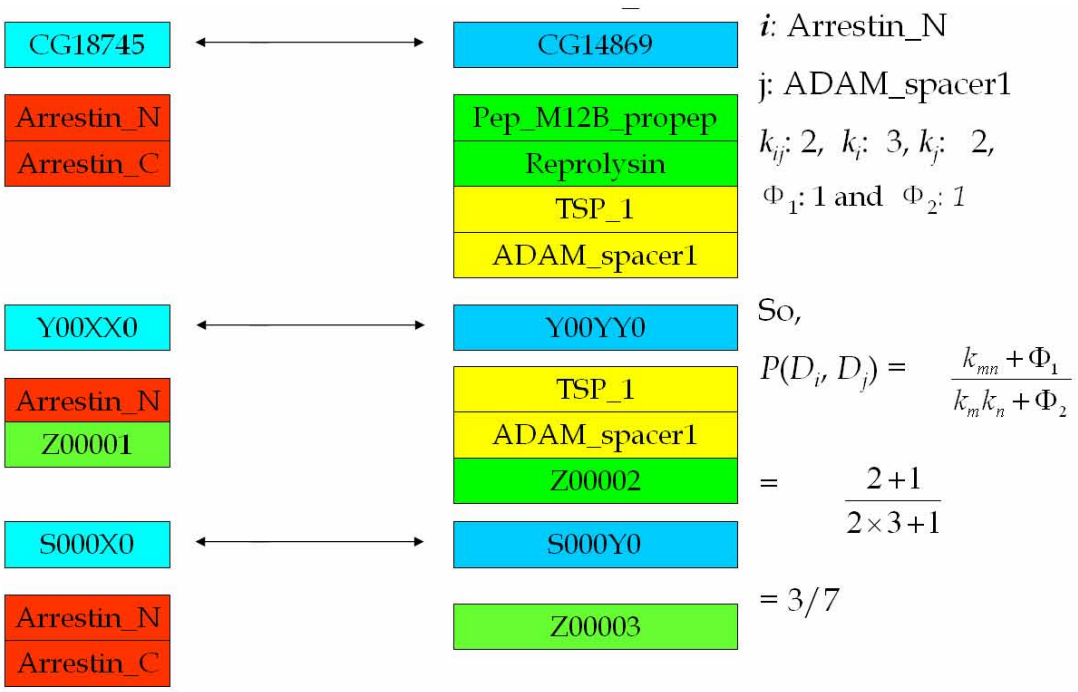


Figure 4.10: Protein interaction pair and domains contained

Bayes' theorem relates the conditional and marginal probabilities of stochastic events A and B:

$$Pr(A|B) = \frac{Pr(B|A)Pr(A)}{Pr(B)} \propto L(A|B)Pr(A) \tag{2.2.2}$$

where $L(A|B)$ is the likelihood of A given fixed B. Each term in Bayes' theorem

has a conventional name:

- $Pr(A)$ is the prior probability or marginal probability of A. It is "prior" in the sense that it does not take into account any information about B.
- $Pr(A|B)$ is the conditional probability of A, given B. It is also called the posterior probability because it is derived from or depends upon the specified value of B.
- $Pr(B|A)$ is the conditional probability of B given A.
- $Pr(B)$ is the prior or marginal probability of B, and acts as a normalizing constant.

With this terminology, the theorem may be paraphrased as

$$posterior = \frac{likelihood \times prior}{normalizing \quad constant} \tag{4.4.1}$$

In words: the posterior probability is proportional to the prior probability times the likelihood. In addition, the ratio $Pr(B|A)/Pr(B)$ is sometimes called the standardized likelihood, so the theorem may also be paraphrased as

$posterior = standardized \quad likelihood \times prior$

Bayes'theorem is often embellished by noting that

$$Pr(B) = Pr(A \bigcap B) + Pr(\bar{A} \bigcap B) = Pr(B|A)Pr(A) + Pr(B|\bar{A})Pr(\bar{A}) \tag{2.2.3}$$

where $\bar{A}$ is the complementary event of A (often called "not A"). So the theorem can be restated as

$$Pr(A|B) = \frac{Pr(B|A)Pr(A)}{Pr(B|A)Pr(A) + Pr(B|\bar{A})Pr(\bar{A})} \tag{4.4.2}$$

More generally, where A_i forms a partition of the event space,

$$Pr(A_i|B) = \frac{Pr(B|A_i)Pr(A_i)}{\sum_j Pr(B|A_j)Pr(A_j)} \tag{4.4.3}$$

for any A_i in the partition.

Now from the alternative forms of Bayes' theorem (2.2.3), we have

$$Pr(B_{ij}) = Pr(B_{ij}|A_{st})Pr(A_{st}) + Pr(B_{ij}|\overline{A}_{st})Pr(\overline{A}_{st}) \tag{2.2.4}$$

Rearrange (2.2.4), and combine with (2.2.1), we have

$$\begin{aligned} Pr(B_{ij}|A_{st})Pr(A_{st}) &= Pr(B_{ij}) - Pr(B_{ij}|\overline{A}_{st})Pr(\overline{A}_{st}) \\ &= Pr(B_{ij}) - \frac{k_{ij} + \phi_1 - 1}{k_i k_j + \phi_2} \times Pr(\overline{A}_{st}) \end{aligned} \tag{2.2.5}$$

Our goal is to find the value for $Pr(A_{st}|B_{ij})$, that is, given a known value of the probability of a pair of domain-domain interacting, we want to know at what probability the two proteins containing certain domains interact. Thus combined with (2.2.1), (2.2.2) and (2.2.5), we have

$$
\begin{aligned}
& Pr(A_{st}|B_{ij}) \\
&= \frac{Pr(B_{ij}|A_{st})Pr(A_{st})}{Pr(B_{ij})} \\
&= \frac{Pr(B_{ij}) - \frac{k_{ij}+\phi_1-1}{k_i k_j+\phi_2} \times Pr(\overline{A}_{st})}{Pr(B_{ij})} \\
&= 1 - \frac{\frac{k_{ij}+\phi_1-1}{k_i k_j+\phi_2} \times Pr(\overline{A}_{st})}{Pr(B_{ij})} \\
&= 1 - \frac{\frac{k_{ij}+\phi_1-1}{k_i k_j+\phi_2} \times Pr(\overline{A}_{st})}{\frac{k_{ij}+\phi_1}{k_i k_j+\phi_2}} \\
&= 1 - \frac{k_{ij}+\phi_1-1}{\phi_1+k_{ij}} \times Pr(\overline{A}_{st}) \qquad (2.2.6)
\end{aligned}
$$

As $\frac{k_{ij}+\phi_1-1}{\phi_1+k_{ij}}$ is less than 1 and $Pr(\overline{A}_{st})$ is a constant value for a fixed organism, the above value is a valid probability value. In our experiment, ϕ_1 is set in the range of [0.8, 1]. It is used as a general weight adjustment factor so small changes of the value will not affect the whole classification curve of the predicted data set, although shifts towards both directions could be observed.

If we can estimate the approximate number of proteins and the number of protein interactions in an organism, we could assume that the value $Pr(\overline{A}_{st})$ is a constant. Suppose the number of proteins in an organism is x and the number of protein interactions is around y, then

$$Pr(\overline{A}_{st}) = max((1 - \frac{1}{x^2}), (1 - \frac{1}{y}))$$

Thus (2.2.6) is a known value. We have successfully met our goal.

For instance, it is reported that the approximate number of proteins in yeast is around 4,751 and the number of protein-protein interactions in yeast is around 40,000, then

$$Pr(\overline{A}_{st}) = max((1 - \frac{1}{4,751^2}), (1 - \frac{1}{40,000})) = \frac{3,9999}{40,000}$$

We use the following formula to represent $Pr(B_{ij}) = \frac{k_{ij}+\phi_1}{k_i k_j+\phi_2}$ in (2.2.1). Alternatively if we model it with the following formula which has been extensively used in the literature:

$$Pr(B_{ij}) = \frac{1}{2} \times (1 + \frac{k_{ij}}{k_i k_j + \phi}) \tag{2.2.7}$$

where ϕ is usually assigned with the value of 1, we will come up with a different formula for

$$
\begin{aligned}
& Pr(A_{st}|B_{ij}) \\
&= \frac{Pr(B_{ij}|A_{st})Pr(A_{st})}{Pr(B_{ij})} \\
&= \frac{Pr(B_{ij}) - \frac{1}{2} \times (1 + \frac{k_{ij}-1}{k_i k_j + \phi}) \times Pr(\overline{A}_{st})}{Pr(B_{ij})} \\
&= 1 - \frac{\frac{1}{2} \times (1 + \frac{k_{ij}-1}{\phi + k_i k_j}) \times Pr(\overline{A}_{st})}{Pr(B_{ij})} \\
&= 1 - \frac{\frac{1}{2} \times (1 + \frac{k_{ij}-1}{\phi + k_i k_j}) \times Pr(\overline{A}_{st})}{\frac{1}{2} \times (1 + \frac{k_{ij}}{k_i k_j + \phi})} \\
&= 1 - \frac{k_i k_j + \phi + k_{ij} - 1}{k_i k_j + \phi + k_{ij}} \times Pr(\overline{A}_{st})
\end{aligned} \tag{2.2.8}
$$

which means the value of $Pr(A_{st}|B_{ij})$ is dependent on the values of k_i, k_j and k_{ij}.

4.4.3 Markov chain Monte Carlo

Gomez et al adopted a Markov Chain Monte Carlo [8] method in their research [63]. We used their method to produce the posterior probabilities of all edges within the predicted maps. This approach is very useful in generating posteriors from complicated distributions, enables us to adequately sample from the very large number of possible network configurations (for N vertices there are 2^{N^2} possible networks). A uniform prior distribution over all networks is used in our experiments. The reason is that we had no prior information that would cause us to prefer one network over another. Starting with an random generated network, and using a reversible jump

methodology [72], edges were both added and removed at each iteration of the algorithm. Addition and removal of edges moves the network from the current state S to a proposed state T. Thus the new state is accepted with probability:

$$\alpha(s,t) = min\{1, \frac{L(T)}{L(S)}\}$$

where L(.) is the likelihood of the network. If the proposed state is accepted, it will become the current state. Thus this method could sample networks from the space of all possible networks while keeping each edge either occupied or unoccupied over time, in proportion to its posterior probability. We follow [63] and generated the posterior distribution from approximately 10^7 samples.

4.4.4 Hybrid Bayes method - from domain-domain interaction to protein-protein interaction

As the second step of building a protein-protein interaction classifier, we have to find efficient and accurate algorithm to calculate the protein-protein interaction networks from the inferred virtual domain-domain interaction maps in the previous step. There are currently two popular methods to do this.

Suppose the probability of two domains d_m and d_n interact is $P(d_m, d_n)$, For a pair of multidomain proteins P_i and P_j, where N_i and N_j are the number of unique

protein domains for each, the probability of an edge forming between the two proteins can be calculated via the following two methods (2.2.9.a) and (2.2.9.b).

$$P_{ij} = \sum_{d_m \in N_i} \sum_{d_n \in N_j} \frac{prob(d_m, d_n)}{|N_i||N_j|} \tag{2.2.9.a}$$

$$P_{ij} = 1.0 - \prod_{1}^{m \times n} (1 - prob(d_m, d_n)) \tag{2.2.9.b}$$

In the results section, we will compare the results generated from each individual method and the combined voting machine method.

4.4.5 Details of the algorithm and domain detection

Now back to our previous example, our goal is to find an effective strategy to reduce the number of false domain interactions.

In order to achieve this goal, an algorithm combining the calculation of the maximum number of domain-domain interaction occurrences and the selection of a meaningful interaction pair from a public domain-domain interaction database $InterDom$ was proposed. The algorithm is shown in (Fig. 4.11)

The program first scans each line of the training protein-protien interacton networks and get the protein-protein interaction pair, then find the corresponding domain information from the dataset generated during domain detection stage. The domain

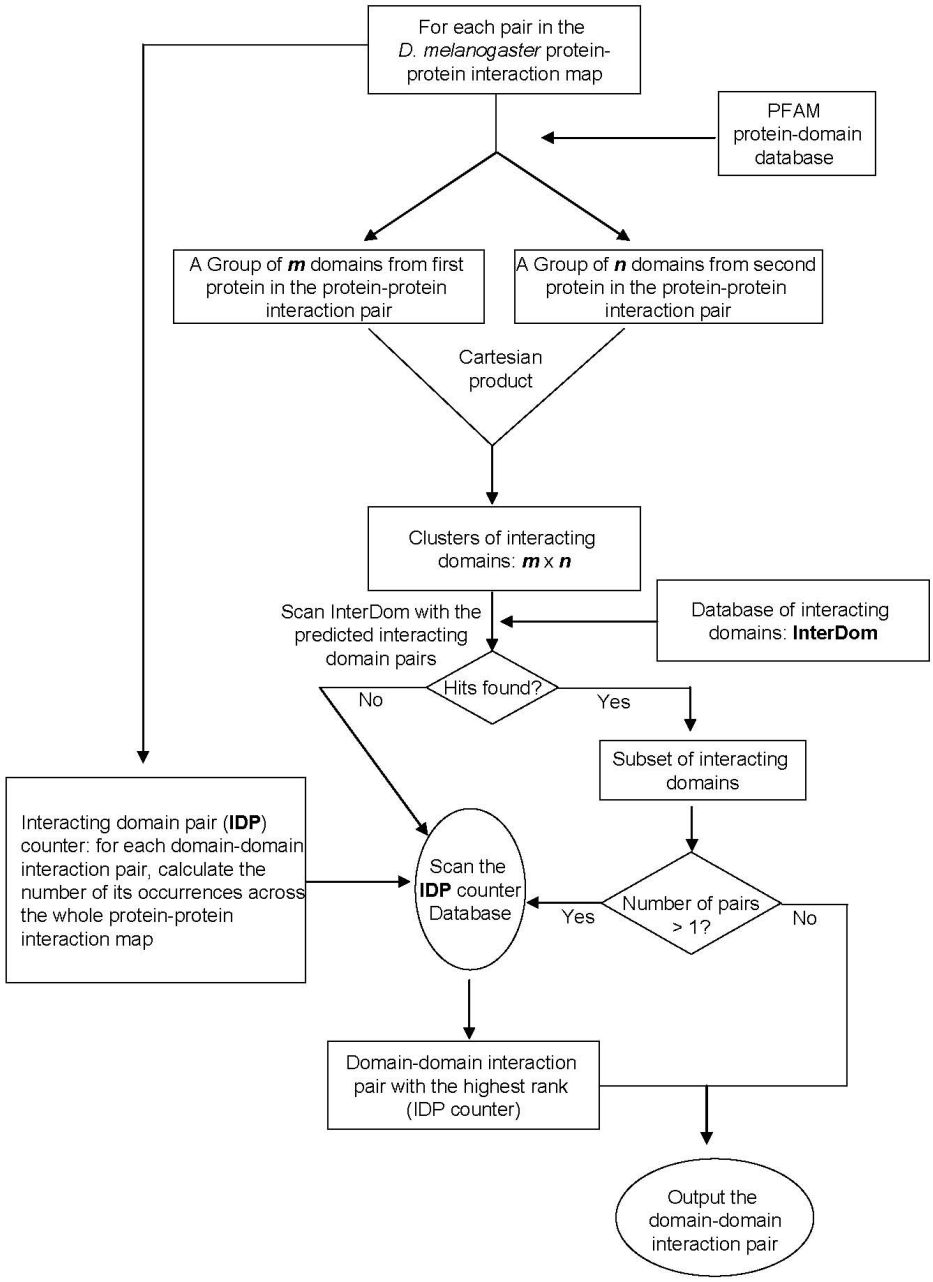


Figure 4.11: Overview of the virtual domain algorithm.

information datasets for both organisms were extracted. We used PFAM, Interdom and SMART [14, 112] to perform the domain detection for a certain protein sequence. If there is no domain reported for a protein in the interaction pair, this pair will be omitted for our current design. A possible improvement is to use PSI-Blast search to identify the closest protein in *A. gambiae* based on sequence similarity.

Secondly, an Interacting domain pair (IDP) counter is built by scanning the whole protein-protein interaction map and adding up all the occurrences of each domain-domain interaction map. A global domain-domain interaction counter for each domain-domain interaction pair is generated after this step.

Next, suppose the number of domains is is m in the first protein and n in the second, a cartesian product of these two groups of domains produces a group of $m \times n$ domain-domain pairs P_{d-d}. Each pair in this group is scanned against the dataset extracted from InterDom. If there is a match, the domain-domain interaction pair will be kept and added into a temporary dataset T_{d-d} as it is deemed as an interaction with certain confidence. If the final number of domain-domain interaction pair in T_{d-d} is 1, then the program write down this pair and goes on to fetch and analyze next protein-protein interaction pair in M_d. If the number of pairs in T_{d-d} is greater than 1, all the pairs in it will be searched against the (IDP) counter and the scores of pairs will be found. The domain-domain pair in T_{d-d} with the highest number of occurrences will be taken as the final output. In the previous comparison, If there is

no match between P_{d-d} and InterDom, all the pairs in P_{d-d} will be thrown directly to the (IDP) counter to find the pair with the highest number of occurrences.

Domain detection was performed using Pfam 21.0 [14] (November 2006 release, 8,957 families). Pfam is a large collection of multiple sequence alignments and hidden Markov models covering many common protein domains and families. A particular useful function is to analyze a protein query sequence to find Pfam family matches. It was reported that 74% of protein sequences have at least one match to Pfam. This number is called the sequence coverage.

As running Pfam locally is very computationally expensive, we decided to extract the results from Pfam website directly. We wrote a program to post the protein sequence one by one and extracted the results and parsed the domain information if available. Pfam is available at

12,208 proteins were reported having domains in *D. melanogaster*, which is around 2/3 of the total number of proteins. In the published *D. melanogaster* protein-protein interaction map M_d, only BDGP 3.1 CG numbers were used, i.e., 'CG11094-PA', 'CG11094-PB', and 'CG11094-PC' were listed as 'CG11094'. In order to build connections with M_d, we combined the domain information of these three different transcripts and generated one unified item. The number of proteins was reduced to 8625 after this step and there are 1,959 different domains in this dataset.

13,276 proteins were reported having domains in *A. gambiae* and there are 2,613

different domains in this dataset.

The two organisms share 1,540 common domains. The number of domains specific to *D. melanogaster* is 1,073 and to *A. gambiae* is 419.

We have also combined the domain interaction information from Interdom [138]. A program was written to download all the webpages containing domain information at InterDom website . A parser was written to extract the domain-domain interaction pairs and the confidence scores. There are 30,037 inferred domain-domain interactions in the dataset. Our calculation shows that the total number of self-interacting domains is 1,198.

Chapter 5

Results

In this chapter, we compare the results produced from our methods and some popular methods.

We suggest that further research in this direction is likely going to reveal additional properties of voting machine methods and thus contribute to our understanding of how proteins really interact and function, and how these predicted properties could assume an important role in reduce the number of candidates in the biological experiments. These will eventually serve the purpose of deciphering the sex-determination pathway of A. gambiae and using STL technique to kill A. gambiae and eliminate Malaria.

5.1 Orthologous protein transfer method

The chi-square experiment results are reported in Table 5.2 and (Fig. 5.2). For significance at the 0.05 level, chi-square should be greater than or equal to 3.84. Our results are well above this threshold and it should not be a random prediction.

We observed from figure Table 5.1 that for *D. melanogaster*, the number of protein interactions published in literature is 4,997 while predicted by orthologous transfer method from inparanoid database is 2,774. They share a common set of 1,218. For *S. cerevisiae*, the number of protein interactions published in literature is 2,366 while predicted by orthologous transfer method from is 1,055. They share a common set of 1,055. The number of protein interactions for *S. cerevisiae* in DIP database is 3,944 and it shares a common set of 1,738 with those predicted via inparanoid method. Finally, our predicted method has a common set of 783 with that from the DIP database.

Table 5.1: **Prediction results vs data from public databases**

	D_map vs D_inparanoid	S_inparanoid vs S_predicted	S_inparanoid vs S_DIP	S_predicted vs S_DIP
Dataset A	4,997	2,366	2,366	1,055
Dataset B	2,774	1,055	3,944 (DIP reported 4,751)	3,944 (DIP reported 4,751)
Intersection C = (A $\bigcap$ B)	1,218	1,055	1,738	783
Inter vs Data1 (C vs A)	24.37%	44.59%	73.46%	74.22%
Inter vs Data2 (C vs B)	43.91%	100%	44.07%	19.85%

Pearson's chi-square (χ^2) test is the best-known of several chi-square tests [154]. It is a statistical procedures whose results are evaluated by reference to the chi-square distribution. Its properties were first investigated by Karl Pearson.

It tests a null hypothesis that the frequency distribution of certain events observed in a sample is consistent with a particular theoretical distribution. The events considered must be mutually exclusive and have total probability 1. A common case for this is where the events each cover an outcome of a categorical variable. A simple example is the hypothesis that an ordinary six-sided dice is "fair", i.e., all six outcomes are equally likely to occur. Pearson's chi-square is the original and most widely-used chi-square test.

The first step in the chi-square test is to calculate the chi-square statistic. The chi-square statistic is calculated by finding the difference between each observed and theoretical frequency for each possible outcome, squaring them, dividing each by the theoretical frequency, and taking the sum of the results.

$$\chi^2 = \sum_{i=1}^{n} \frac{(O_i - E_i)^2}{E_i}$$

where O_i is an observed frequency; E_i is an expected frequency, asserted by the null hypothesis; and n is the number of possible outcomes of each event.

The chi-square statistic can then be used to calculate a p-value by comparing the value of the statistic to a chi-square distribution. The number of degrees of freedom is equal to the number of possible outcomes, minus 1.

Pearson's chi-square is used to assess two types of comparison: tests of goodness of fit and tests of independence. A test of goodness of fit establishes whether or not

an observed frequency distribution differs from a theoretical distribution. A test of independence assesses whether paired observations on two variables, expressed in a contingency table, are independent of each other. For example, whether people from different regions differ in the frequency with which they report that they support a political candidate.

A chi-square probability of 0.05 or less is commonly interpreted by applied workers as justification for rejecting the null hypothesis that the row variable is unrelated (that is, only randomly related) to the column variable. The alternate hypothesis is not rejected when the variables have an associated relationship.

In statistical hypothesis testing, the p-value is the probability of obtaining a value of the test statistic at least as extreme as the one that was actually observed, given that the null hypothesis is true. The fact that p-values are based on this assumption is crucial to their correct interpretation.

More technically, a p-value of an experiment is a random variable defined over the sample space of the experiment such that its distribution under the null hypothesis is uniform on the interval [0,1]. Many p-values can be defined for the same experiment.

Generally, one rejects the null hypothesis if the p-value is smaller than or equal to the significance level. If the level is 0.05, then the results are only 5% likely to be as extraordinary as just seen, given that the null hypothesis is true. The conclusion obtained from comparing the p-value to a significance level yields two results: either

the null hypothesis is rejected, or the null hypothesis cannot be rejected at that significance level.

Our Chi-square tests use a p-value of 0.05. Fig 5.1 shows how the contingency table is constructed from our actual and predicted datasets. Table 5.2 gives the actual test results. From the results, we can plot the Chi-square value vs the protein hypothesis space in Fig 5.2. Our results show that the hypothesis space 123,900 is the break-even point; if the number of the protein-interaction pairs in *S. cerevisiae* is greater than 123,900, our hypothesis that the prediction method produces a significant result could be null; if the number is less than that, our prediction method is significant. Note in current literature, it is reported that the number of pairs could be around 40,000, at which point our Chi-square score is 679.02, which is much higher than the threshold value of 3.84. In this regard, our prediction method can produce significant results and it is highly possible the method is efficient and useful.

5.2 Hybrid domain-domain interaction method

For the first step, namely extract domain-domain interaction information, we propose estimating the probabilities of interactions between domain pairs by pooling information from 5 organisms Using the estimated domain-domain interaction probabilities with a weighted probabilties, we then estimate the probabilities of interactions between each protein pair in a given organism. Because of the experimental errors of large-scale two-hybrid assays, the domain interactions inferred from one organism may

Chi-square test

- *A:* actual number
- ~*A*: *N* – *A* (*not A*)
- *P:* predicted number
- ~*P*: *N* – *P* (*not P*)
- For significance at the .05 level, *chi-square* should be greater than or equal to 3.84.
- Contingency table:

	A	~*A*
P	*P* & *A*	*P* & (*not A*)
~*P*	(*not P*) & *A*	(*not P*) & (*not A*)

Figure 5.1: Chi-square test method

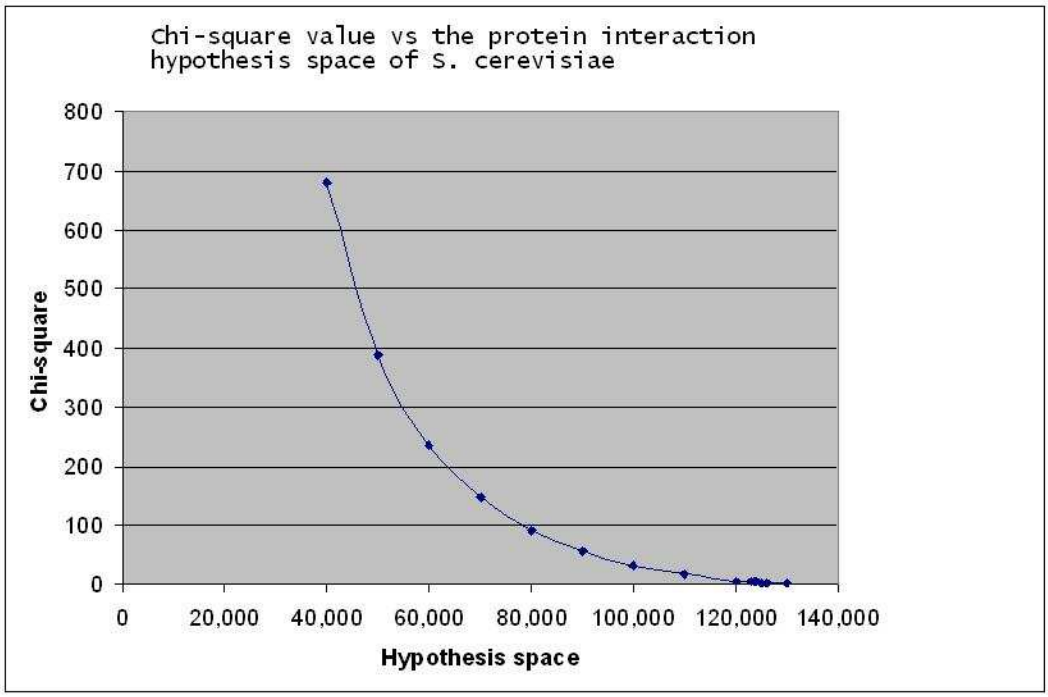


Figure 5.2: Chi square values vs the hypothetical protein interaction space

Table 5.2: **Chi-square tests for *S. cerevisiae* protein interaction data**

Hypothesis space	$\boldsymbol{A \bigcap P}$	$\boldsymbol{A \bigcap \overline{P}}$	$\boldsymbol{\overline{A} \bigcap P}$	$\boldsymbol{\overline{A} \bigcap \overline{P}}$	**Chi-square**	**Significant**
17252	218	$15481 - 218 = 15263$	$1989 - 218 = 1771$	0	15144.82	Yes
20,000	218	15263	1771	2,748	5575.27	Yes
30,000	218	15263	1771	12,748	1408.98	Yes
40,000	218	15263	1771	22,748	679.02	Yes
50,000	218	15263	1771	32,748	387.68	Yes
60,000	218	15263	1771	42,748	236.69	Yes
70,000	218	15263	1771	52,748	147.90	Yes
80,000	218	15263	1771	62,748	92.02	Yes
90,000	218	15263	1771	72,748	55.62	Yes
100,000	218	15263	1771	82,748	31.69	Yes
110,000	218	15263	1771	92,748	16.23	Yes
120,000	218	15263	1771	102,748	6.13	Yes
123,000	218	15263	1771	105,748	4.31	Yes
123,500	218	15263	1771	106,248	4.04	Yes
123,600	218	15263	1771	106,348	3.98	Yes
123,700	218	15263	1771	106,448	3.93	Yes
123,800	218	15263	1771	106,548	3.88	Yes
123,900	218	15263	1771	106,648	$3.83 < 3.84$	No
124,000	218	15263	1771	106,748	$3.78 < 3.84$	No
125,000	218	15263	1771	107,748	$3.29 < 3.84$	No
126,000	218	15263	1771	108,748	$2.84 < 3.84$	No
130,000	218	15263	1771	112,748	$1.40 < 3.84$	No

not be reliable, and the incorporation of data from other organisms can improve the estimated domain-domain and protein-protein interactions. For instance, currently only about two-thirds of the *S. cerevisiae* proteins have a defined domain composition, and we have considered possible interactions only between those proteins with annotated domain information. As a result, the predictions based on domain-domain

interactions will be able to capture only a portion of all actual interactions. The number of protein interactions for *D. melanogaster*, *S. cerevisiae*, *Caenorhabditis elegans*, *Escherichia coli* and *Mus musculus* from DIP are 20,349, 3,944, 4,714, 761 and 288 respectively.

The number of domain pairs we predicted is 27,371, containing 6,589 domains. While Interdom and PDB predicted 30,037 and 25,741 domain pairs respectively. Our method shares a common set of 18,651 with Interdom and 16,023 with PDB. These are quite high coverage, given the noise introduced by various methods and different filtering criteria.

To measure the accuracy of our system's predictions, we ranked all predicted interactions such that the most likely positive interactions were situated at the beginning of the list. With this ranking, we were able to set a likelihood or rank threshold, treating the predictions above the threshold as positive interactions and the predictions below the threshold as negligible or negative interactions.

This threshold allowed us to measure the specificity and sensitivity of our method by comparing its predictions with the known but not directly observable simulated real-world network interactions. Sensitivity (or recall) is defined as the percentage of true positives among true positives plus false negatives; specificity is the percentage of true negatives among true negatives plus false positives. Further, we varied the rank threshold, from the very beginning of the list, where specificity is 1 and sensitivity

is 0, to the very end of the list, at which point sensitivity grows to 1 and specificity drops to 0.

By varying the threshold, we computed the receiver-operator characteristic (ROC) curve [217], which plots sensitivity against specificity at different threshold values. Sensitivity is calculated by dividing the number of true positives (TP) through the number of all positives, which equals the sum of the true positives and the false negatives (FN); specificity is calculated by dividing the number of true negatives (TN) through the number of all negatives, which equals the sum of the true negatives and the false positives (FP).

Sensitivity = TP/(TP + FN), Specificity = TN/(TN + FP).

The ROC curve plot shows 1 - specificity on the X axis and sensitivity on the Y axis. A good classifier has its ROC curve climbing rapidly towards upper left hand corner of the graph. This can also be quantified by measuring the area under the curve. The closer the area is to 1.0, the better the classifier is; and the closer the area is to 0.5, the worse the classifier is.

5.3 The first voting machine

As each prediction has its advantage and disadvantage, we propose to construct a voting machine to collectively make decisions.

Given a protein pair (P_i, P_j), let p_{ij}^k be the probability of interaction returned from method k and (w_k be the weight of the method. The final voting score is

$p_{ij} = \sum_{i=1}^{n} w_k p_{ij}^k$, where $\sum_{i=1}^{n} w_k = 1$.

For each weight, we change other parameters of the methods including the threshold to choose a data point as a positive result, we obtain the ROC curves below (Fig. 5.16). From our experiments, both least square method and the graph show that $w_1 = 0.65$ (domain based method) gives the best performance.

Next, we fix the weights for the voting machine, and do further experiments for our hybrid bayes method and other popular methods.

We have compared our hybrid bayes method (the first voting machine) with the Maximum likelihood method [42], the Logistic Regression method [2] and the Random Forest method [22]. While all methods successfully predict a common interaction subgroups (Table 5.3), our method is shown to have slightly better performance on the whole data set based on the ROC curve (Fig. 5.3).

Table 5.3: **Examples of common protein interactions of *S. cerevisiae* predicted from several methods**

Protein A	**Protein B**	**Function**
YDR299w	YLR208w	Vesicular transport (Golgi network)
YOL018c	YMR117c	Cellular import
YDL154w	YBR133c	Meiosis and budding
YGL192w	YBR057c	Development of asco-basido-zygo spore
YDR299w	YPL085w	Both in vesicular transport
YEL013w	YFL039	Protein targeting and budding

For cross-validated comparison of our hybrid method, we measured the performance of each prediction using a 10-fold cross-validation. Considering the 3,543 yeast

physical interaction pairs in MIPS as positive examples, treat the other possible protein pairs, totally 6,895,215 pairs, as negative examples. At each iteration, we left one-tenth of both the positives and the negatives for testing and used the remaining data for training. The training-test procedure is repeated ten times. The prediction accuracy is also measured using the ROC curve.

We chose four protein-interaction hubs from the protein interaction map of *Drosophila melanogaster* to do more evaluations of different prediction methods used in our experiments. Results are shown in Table. 5.4, 5.5, 5.6 and 5.7. While each of the four prediction methods has its advantages and disadvantages, we can see that our hybrid bayes method has an overall better performance than other three methods.

Table 5.4: **Example 1 of common protein interactions of *Drosophila melanogaster* predicted from several methods**

Protein A	Function	Protein B	Function	Maximum Likelihood	Regression	Random	Bayes
CG10855	Skip-dimerization	CG15782	Unknown	Y	N	Y	Y
CG4643	Skip-dimerization	CG13085	RNI domain	Y	N	Y	N
CG4643	Skip-dimerization	CG9316	RNI domain	N	N	Y	N
CG4643	Skip-dimerization	CG2247	RNI domain	Y	Y	Y	Y
CG9772	Skip-dimerization	CG14937	RNI domain	N	Y	Y	Y
CG9772	Skip-dimerization	CG9790	Unknown	Y	Y	Y	Y
CG1222	Skip-dimerization	CG9316	RNI domain	Y	Y	Y	Y

Table 5.5: **Example 2 of common protein interactions of *Drosophila melanogaster* predicted from several methods**

Protein A	Function	Protein B	Function	Maximum Likelihood	Regression	Random	Bayes
CG2955	Calmodulin binding	CG31958	Calmodulin related	Y	Y	Y	Y
CG13838	Unknown	CG15022	Adaptor SH3-domain binding	Y	Y	N	Y
CG13838	Unknown	CG2079	Adaptor SH2/PTB	N	N	Y	Y
CG13503	Actin reorganization	CG2079	Adaptor SH2/PTB	N	N	Y	Y

Table 5.6: **Example 3 of common protein interactions of *Drosophila melanogaster* predicted from several methods**

Protein A	Function	Protein B	Function	Maximum Likelihood	Regression	Random	Bayes
CG32708	RNA binding	CG3918	RNA binding	Y	N	N	Y
CG32708	RNA binding	CG6843	Novels	Y	N	Y	N
CG9346	RNA binding	CG31211	Novels	Y	N	N	Y
CG3918	RNA binding	CG6843	Novels	Y	Y	Y	Y
CG3918	RNA binding	CG31211	Novels	N	N	Y	Y
CG11274	RNA binding	CG10324	G-patch	N	N	N	Y
CG10324	G-patch	CG10689	RNA helicase	Y	Y	Y	Y
CG8273	G-patch	CG11266	Splicing factors	Y	N	N	N
CG31550	G-patch	CG3075	Transcription/translation	N	N	Y	Y
CG31550	G-patch	CG3162	Splicing factors	Y	N	Y	Y
CG6418	Splicing factors	CG3162	Splicing factors	Y	N	N	Y
CG7757	Splicing factors	CG8079	Splicing factors	N	Y	N	Y
CG7757	Splicing factors	CG4709	Splicing factors	N	Y	N	Y
CG7757	Splicing factors	CG5064	Splicing factors	Y	N	N	N
CG3294	Splicing factors	CG5064	Splicing factors	Y	Y	Y	Y
CG3294	Splicing factors	CG6843	Novels	N	N	Y	Y
CG3294	Splicing factors	CG1420	Splicing factors	Y	N	Y	Y
CG11266	Splicing factors	CG1420	Splicing factors	Y	Y	Y	Y

Table 5.7: **Example 4 of common protein interactions of *Drosophila melanogaster* predicted from several methods**

Protein A	Function	Protein B	Function	Maximum Likelihood	Regression	Random	Bayes
CG13651	Homeodomain	CG4617	HMG	Y	Y	Y	Y
CG13651	Homeodomain	CG10348	C2H2 Zn finger	Y	N	N	N
CG13651	Homeodomain	CG7512	C2H2 Zn finger	Y	N	Y	Y
CG13651	Homeodomain	CG17244	TFIID/TAF	Y	N	Y	N
CG11301	TFIID/TAF	CG17244	TFIID/TAF	Y	N	Y	Y
CG11301	TFIID/TAF	CG7512	C2H2 Zn finger	N	N	Y	Y

5.4 The second voting machine

While both the orthologous cluster and hybrid bayes methods produce encouraging results the second one predicts more protein-protein interaction than the first. Yet these two data sets share a very small fraction of common interactions. We adopt a second voting machine and train it with the data from Inparanoid.

Similarly to the first voting machine, we use a second voting machine to make

decisions and the final voting score is

$p_{ij} = \sum_{i=1}^{n} w^k p_{ij}^k$, where $w^{hybrid} + w^{orthlogous} = 1$.

For each weight (0 to 1, with step 0.01), we conducted experiments, compared the results and recommended to assign the hybrid bayes method a weight of 0.8 and the orthologous method a weight of 0.2.

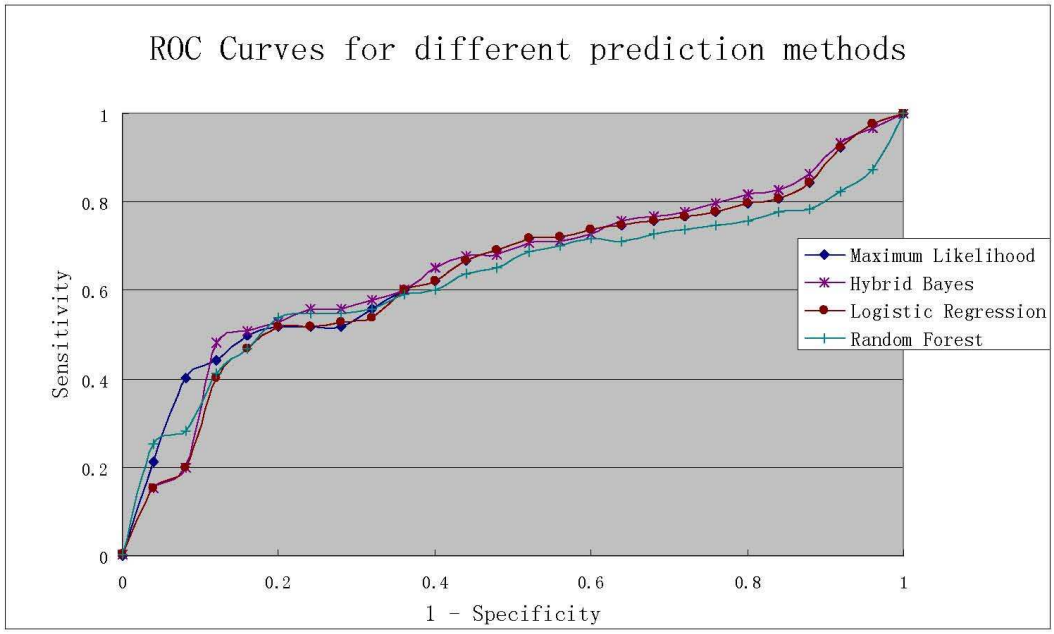


Figure 5.3: ROC curves with different prediction methods

5.5 Comparing with randomly generated network

Random networks were first proposed in [5]. The authors studied networks created by randomly distributing a given number of links between a given number of nodes. Random networks can be created by defining a network with a certain number of nodes, N, and a certain probability, p, of connecting two nodes. In essence, $\frac{1}{2}N(N-1)$, pair of nodes a link is made with the given probability p, that for example can be defined as the desired average number of links, $\bar{s}$, divided by the maximal number of links:

$$p = \frac{\bar{s}}{\frac{1}{2}N(N-1)} \tag{5.5.1}$$

For this type of network the connectivities are binomial distributed around the average, which means that all nodes have similar connectivities. This can be seen by considering the $(N-1)$ possible links a given node can have. For each possibility the probability of a link is p. Now the probability that links are found in c of the $(N-1)$ places is:

$$P(c) = C_{N-1}^{c} p^{c} (1-p)^{N-1-c} \tag{5.5.2}$$

$$\bar{c} = p(N-1) \tag{5.5.3}$$

$$\sigma(c) = \sqrt{(N-1)p(1-p)} \tag{5.5.4}$$

where C_{N-1}^{c} is the binomial coefficient, $\bar{c}$ is the average connectivity and $\sigma(c)$ is the standard deviation of connectivities. We can observe that the total number of links, s, is binomial distributed with the average $\bar{s}$ given by 5.5.1. In random networks we say the network having a scale, for instance the average connectivity that depends on the number of nodes N and links s. For large number of nodes N, the binomial connectivity distribution of these random networks will be close to a normal distribution.

In real life, the distribution is in most cases very wide, which means there might be several nodes with connectivity much higher than expected from a binomial distribution; likewise, there are many nodes with low connectivity, all giving a large standard deviation of connectivity for the network. These networks often have a connectivity distribution following the power law, they are namely scale free network. See equation 5.5.5 and figures 5.4, 5.5, 5.6 and 5.7 for distributions with different γ values.

$$P(c) = \frac{1}{\zeta(\gamma)} \frac{1}{c^{\gamma}} \tag{5.5.5}$$

$$\zeta(\gamma) = \sum_{c=1}^{\infty} c^{-\gamma} \tag{5.5.6}$$

here $p(c)$ is the fraction of nodes with connectivity c, and the normalization constant $\zeta(\gamma)$ is *Riemann*'s Zeta function. Please be noted that in 5.5.5, $c <> 0$ and

$\gamma > 1$, and the mean value and standard deviation of the connectivity c is not defined for all values of γ. The mean value of c is found to be:

$$\bar{c} = \frac{1}{\zeta(\gamma)} \sum_{c=1}^{\infty} c^{-\gamma}.c = \frac{\zeta(\gamma - 1)}{\zeta(\gamma)} \tag{5.5.7}$$

That is to say, $\bar{c}$ is only defined for $\gamma > 2$ and the standard deviation of c, $\sigma(c)$ is defined as:

$$\sigma(c) = \sqrt{\bar{c^2} - \bar{c}^2} \tag{5.5.8}$$

$$\bar{c^2} = \frac{1}{\zeta(\gamma)} \sum_{c=1}^{\infty} c^{-\gamma}.c^2 = \frac{\zeta(\gamma - 2)}{\zeta(\gamma)} \tag{5.5.9}$$

$$\sigma(c) = \sqrt{\frac{\zeta(\gamma - 2)}{\zeta(\gamma)} - \frac{\zeta(\gamma - 1)^2}{\zeta(\gamma)^2}} \tag{5.5.10}$$

therefore $\sigma(c)$ is only defined for $\gamma > 3$. Many real networks have a γ value between 2 and 3, which means that the standard deviation is not defined for an infinite network giving the wide distribution of connectivities. For a finite network, the mean value and the standard deviation can be calculated however they depend on the size of the network.

We also observed that when γ value increases, the mean value of the connectivity decreases and approaches 1 for very large γ. For instance, when $\gamma = 3$, the mean

value of c is smaller than 2. For an infinite network, this means that the network can not be fully connected and it breaks up into disjoint smaller groups.

For a given random number, $y \in [0, 1]$, the connectivity c is:

$$c = (1 - y)^{\frac{1}{1-\gamma}} \tag{5.5.11}$$

Connectivities can be assigned to every network node by picking a random number for each node and setting y in equation 5.5.11 equal to this random number. This will give a connectivity c_i for each node. Because this distribution is continuous, the value of c_i is rounded off to the nearest smaller integer giving a connectivity distribution of integers. To make sure the connectivities are not too large compared to the system size, we make a restriction so that only those y's making $c_i < 90\%$ of the system size are accepted. This will not be a restriction for γ around 2.5, but for smaller γ the probability of choosing a c_i larger than the system size increases. After each node i is given a certain connectivity c_i, the nodes are sorted by descending connectivities: $c_1 \geq c_2 \geq \ldots \geq c_i \geq \ldots$

A threshold of detection is applied to create a network where interactions are either seen or not seen.

Below we plot the total connectivity of a protein in the model network vs. the real connectivity of that protein from a simulated scale-free network below with different γ values. It is difficult to observe good connectivity for large network as they have

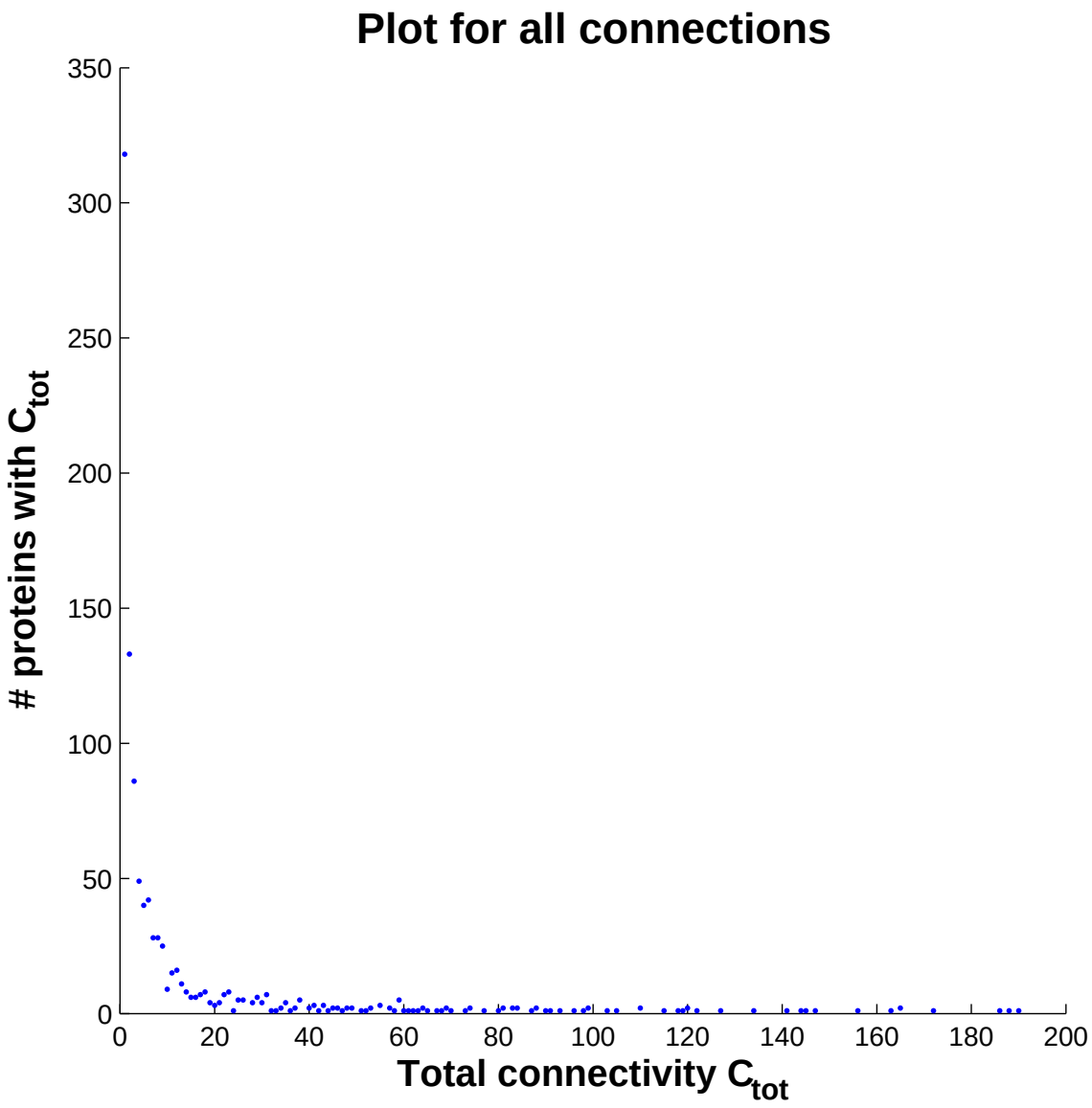


Figure 5.4: γ=1.5, the connectivity distribution of a scale free network that follows power-law

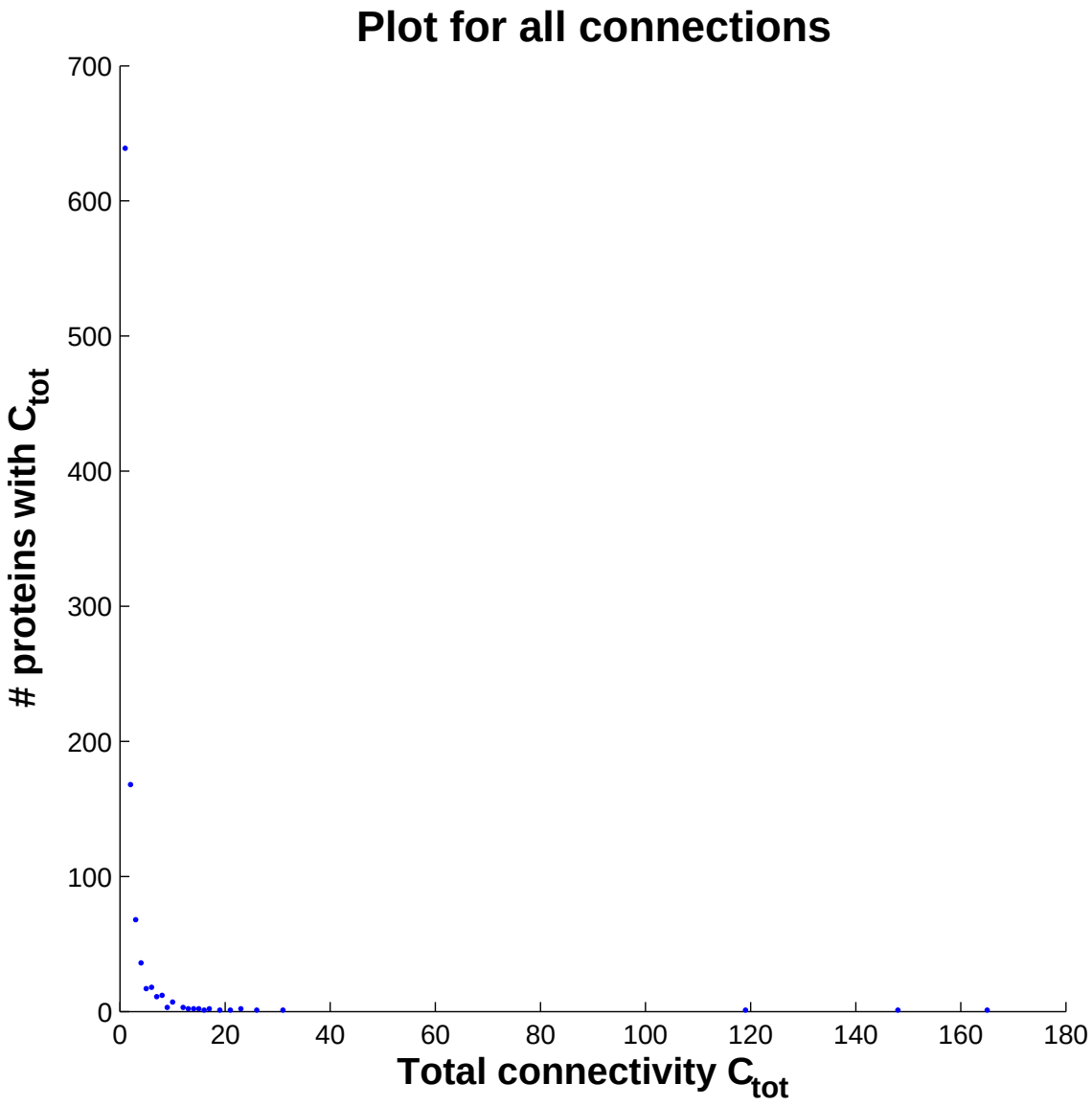


Figure 5.5: γ=2.5, the connectivity distribution of a scale free network that follows power-law

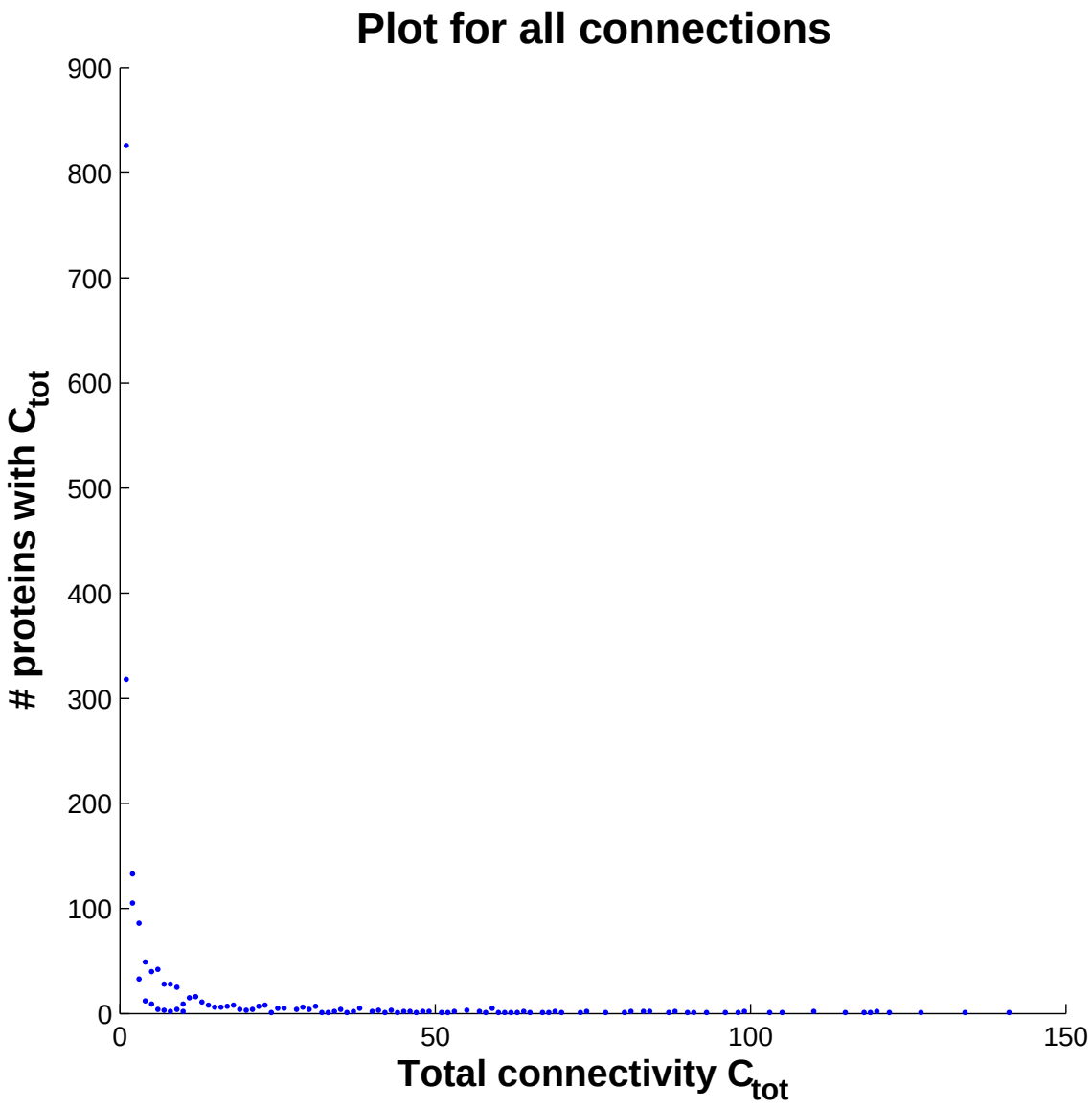


Figure 5.6: γ=3.5, the connectivity distribution of a scale free network that follows power-law

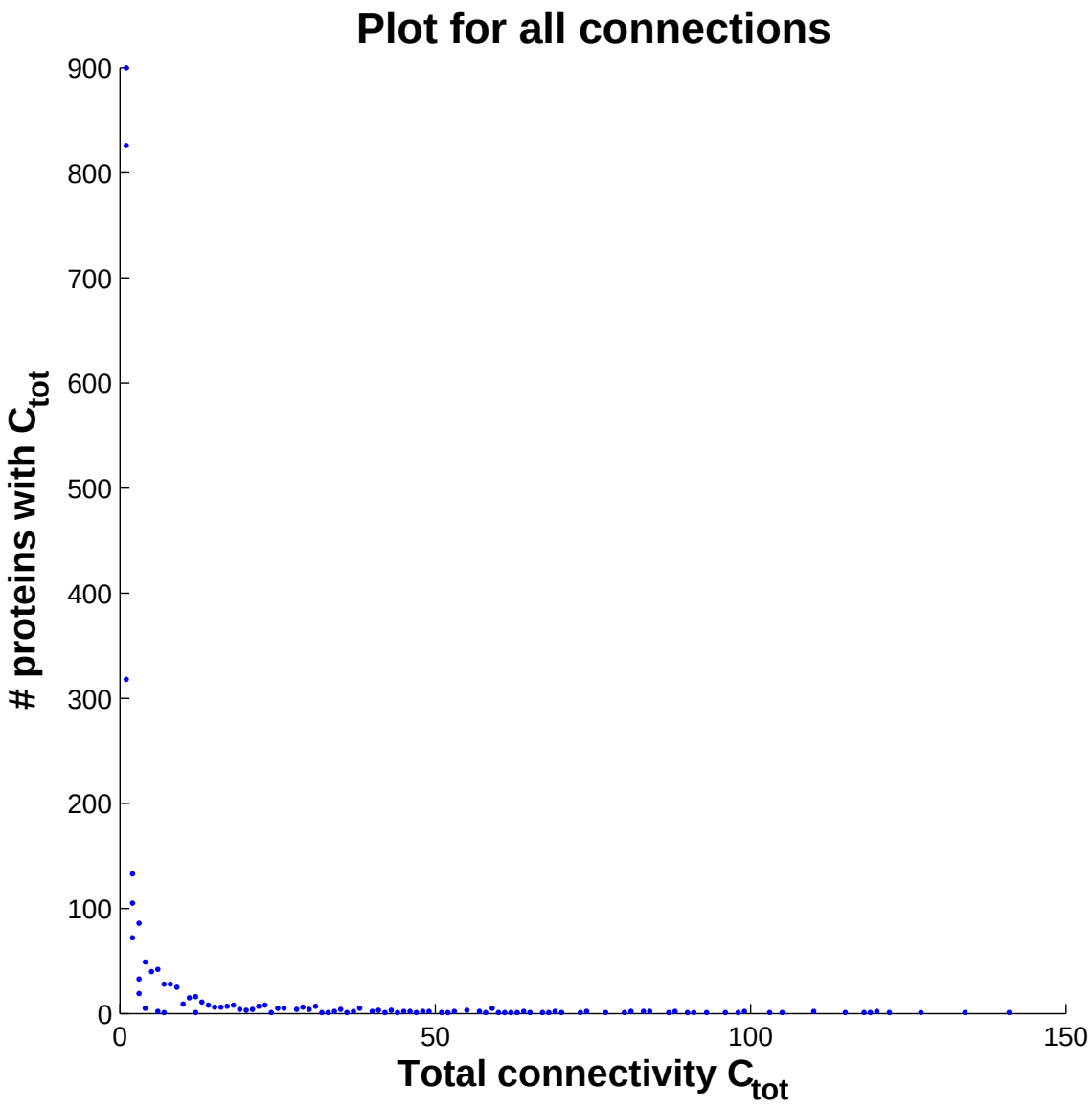


Figure 5.7: γ=4.5, the connectivity distribution of a scale free network that follows power-law

many nodes and are broken into disjoint small subgroups. Thus the connectivity shown in the Fig. 5.8 decreases as the number of proteins increases.

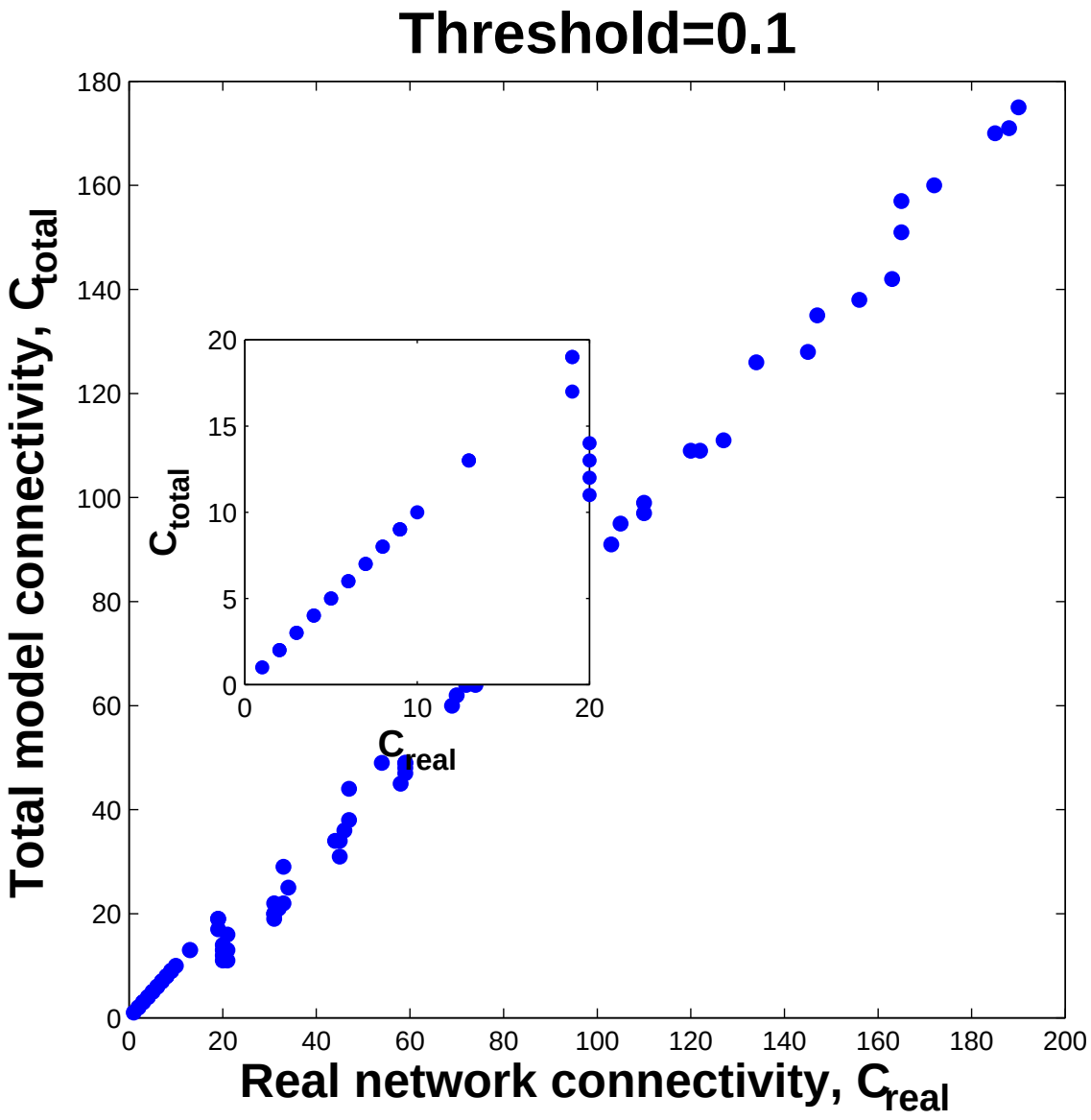


Figure 5.8: γ=1.5, the connections as real vs. the connections as model are plotted. This is obtained by assuming all complexes having higher probability than the threshold are seen in the experiment.

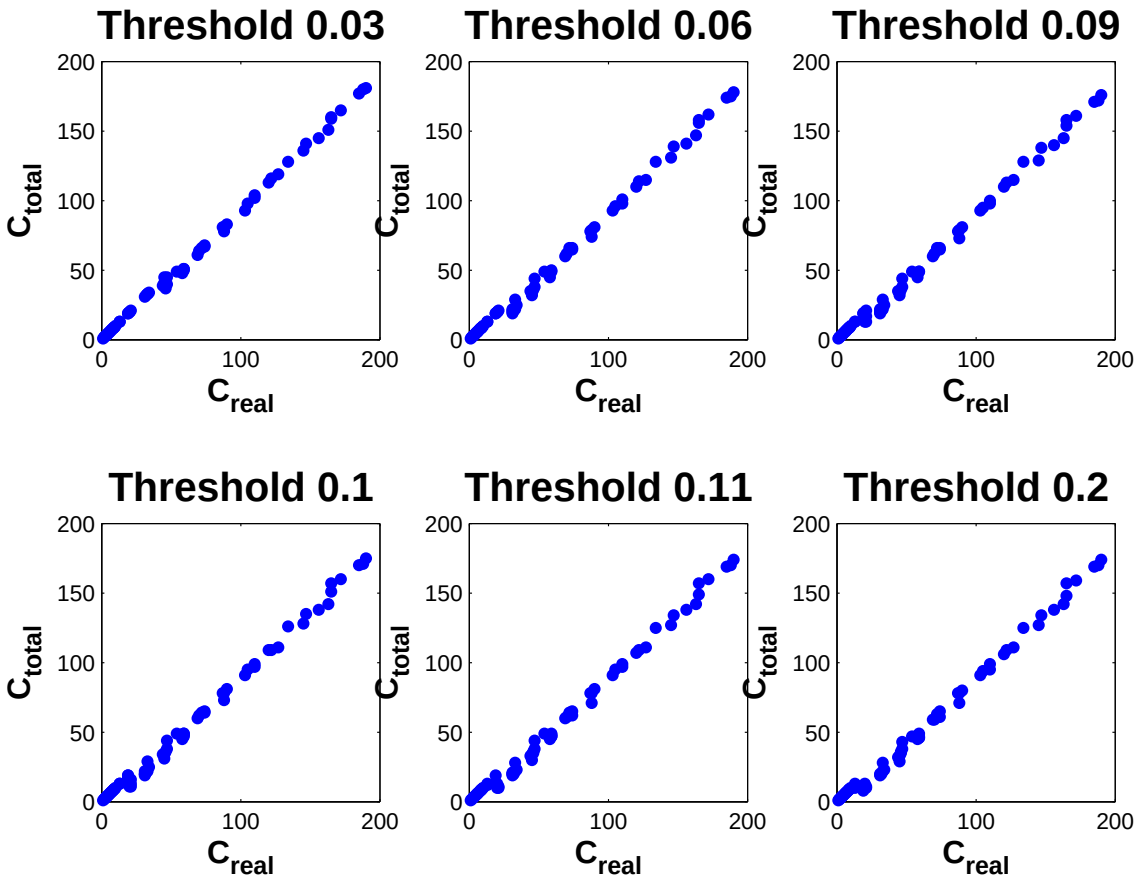


Figure 5.9: γ=1.5, the total connectivity of a protein in the model network is plotted as a function of the real connectivity of that protein from the simulated network

How to choose a suitable γ to generate the scale-free network and then compare with our model predicted network is very important. We follow the literature and choose the γ value to be 2.5 for our experiments.

5.6 *A. gambiae* protein-protein interaction prediction

5.6.1 Orthologous transfer method

There are 2,0485 protein interactions in *D. melanogaster* maps, and 7,259 orthologous clusters for *D. melanogaster* and *A. gambiae* in the inParanoid database. We obtained 9,771 interactions for *A. gambiae*.

5.6.2 Hybrid method and the first voting machine

Using the results from the above experiments, we predict numerous sets of protein interaction networks of *A. gambiae*. As we have two methods to infer protein-protein interaction networks from domain-domain interaction networks, we tried both of them to produce our predictions. At the threshold of 0.5 (probability greater than this denotes a possible interaction between the two nodes of a pair; protein pairs with probability less than this are discarded), we obtained 16,302 and 14,875 interactions respectively. They share a common set of size 8,423.

We also tried the voting machine method. As we have no previous knowledge about how accurate the predicted interactions for *A. gambiae* are, we have to rely on previous calibration results from *S. cerevisiae* and other organisms. Those evaluation

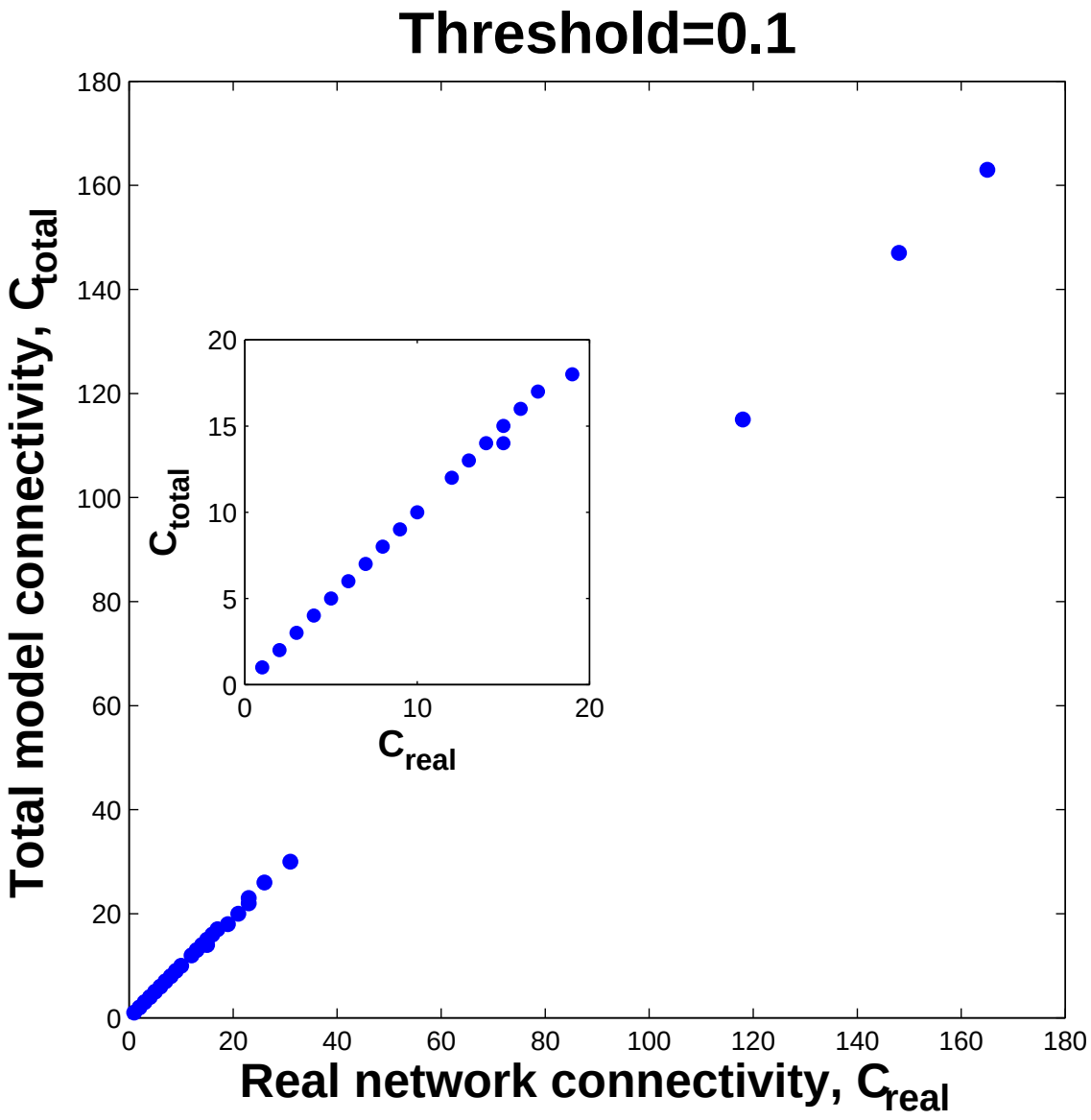


Figure 5.10: γ=2.5, the connections as real vs. the connections as model are plotted. This is obtained by assuming all complexes having higher probability than the threshold are seen in the experiment.

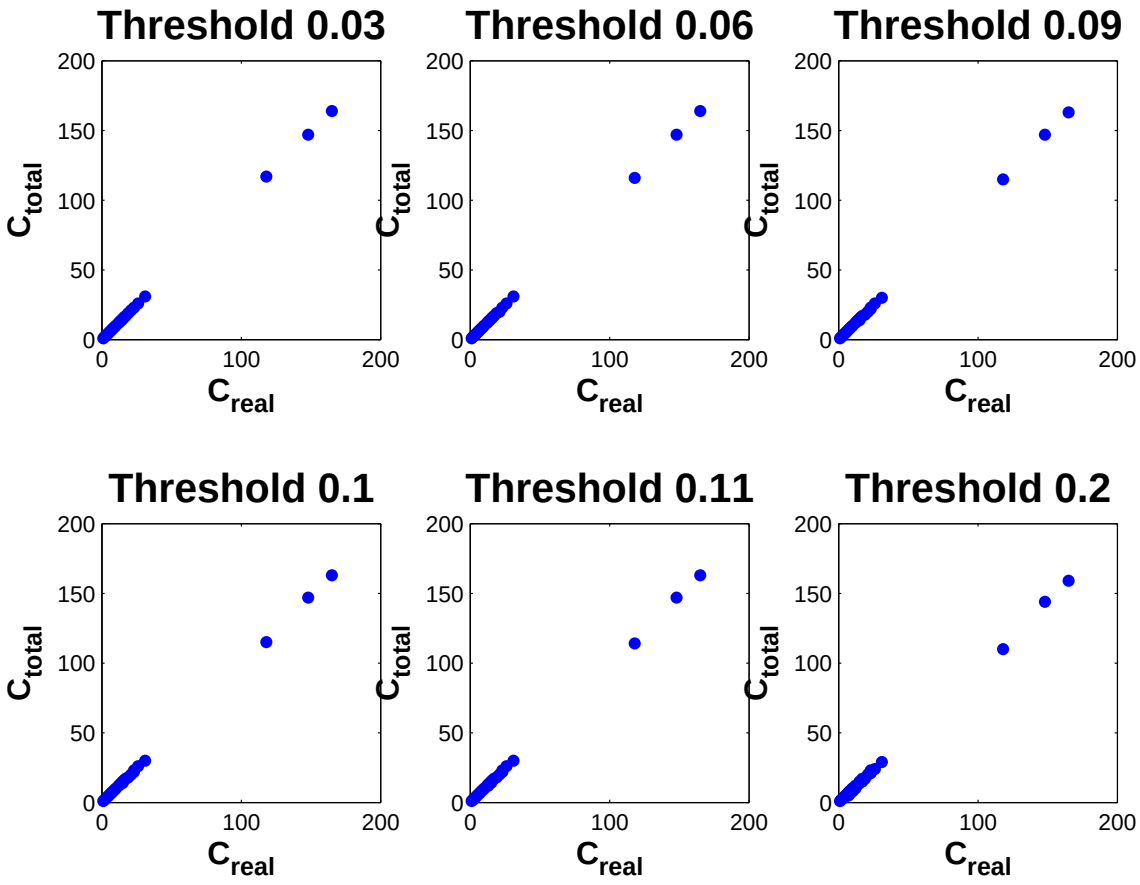


Figure 5.11: γ=2.5, the total connectivity of a protein in the model network is plotted as a function of the real connectivity of that protein from the simulated network

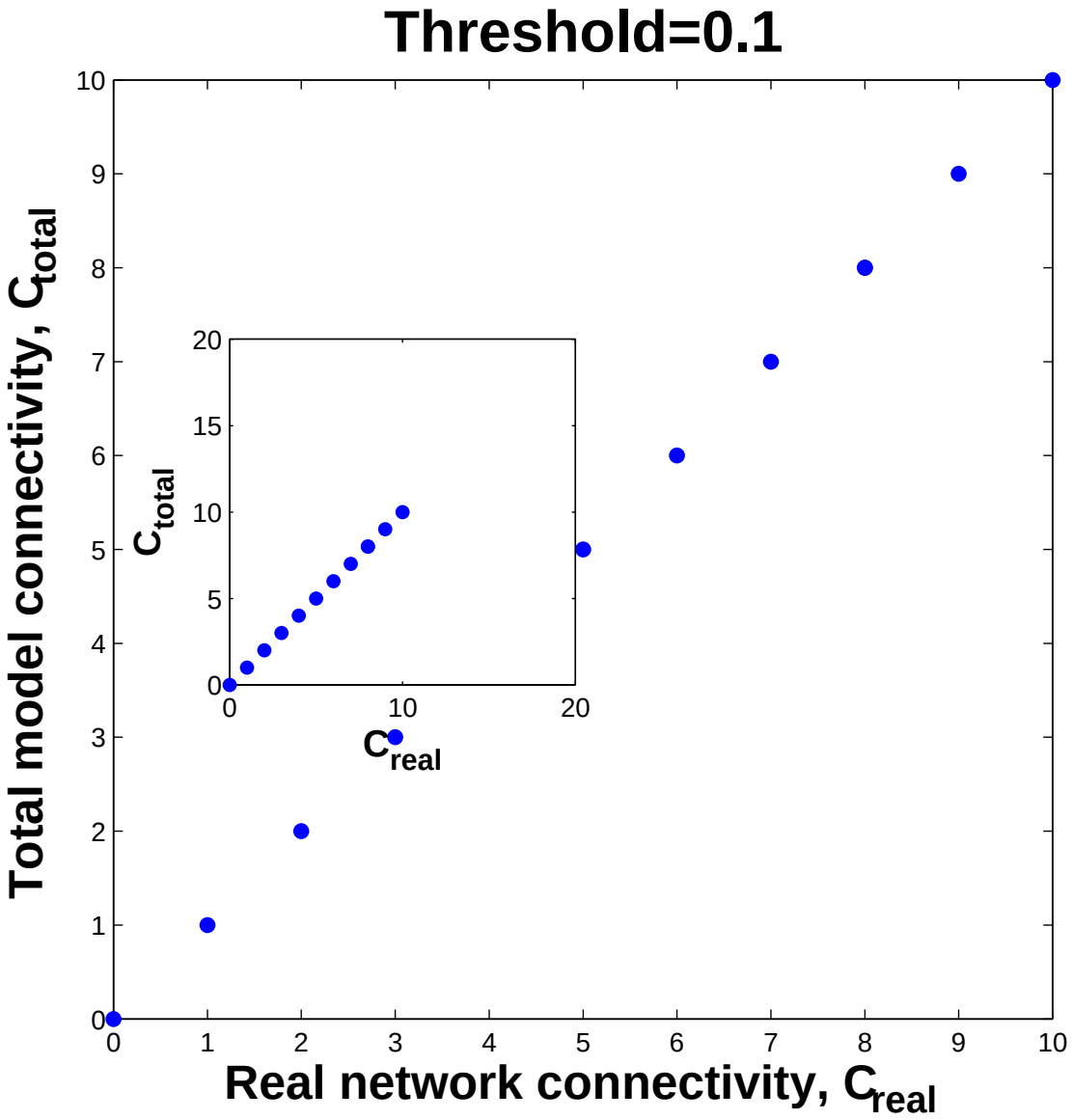


Figure 5.12: γ=3.5, the connections as real vs. the connections as model are plotted. This is obtained by assuming all complexes having higher probability than the threshold are seen in the experiment.

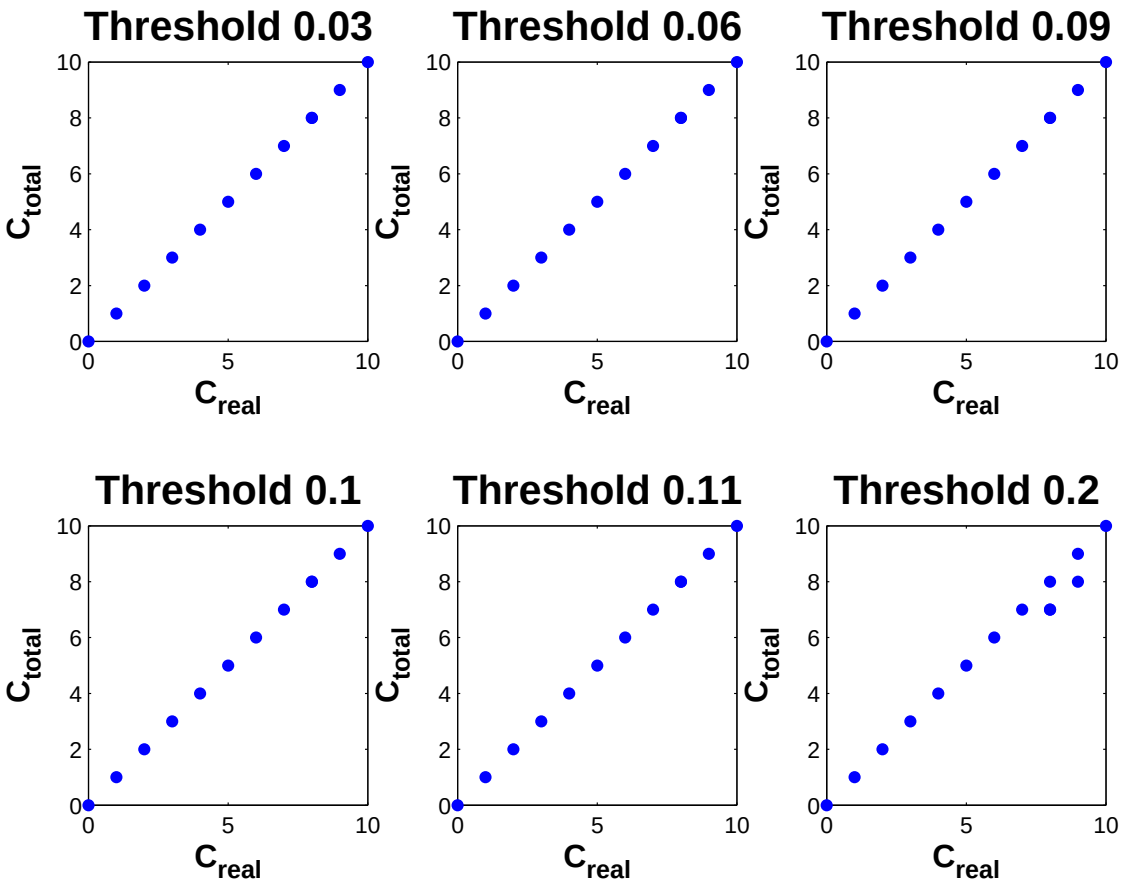


Figure 5.13: γ=3.5, the total connectivity of a protein in the model network is plotted as a function of the real connectivity of that protein from the simulated network

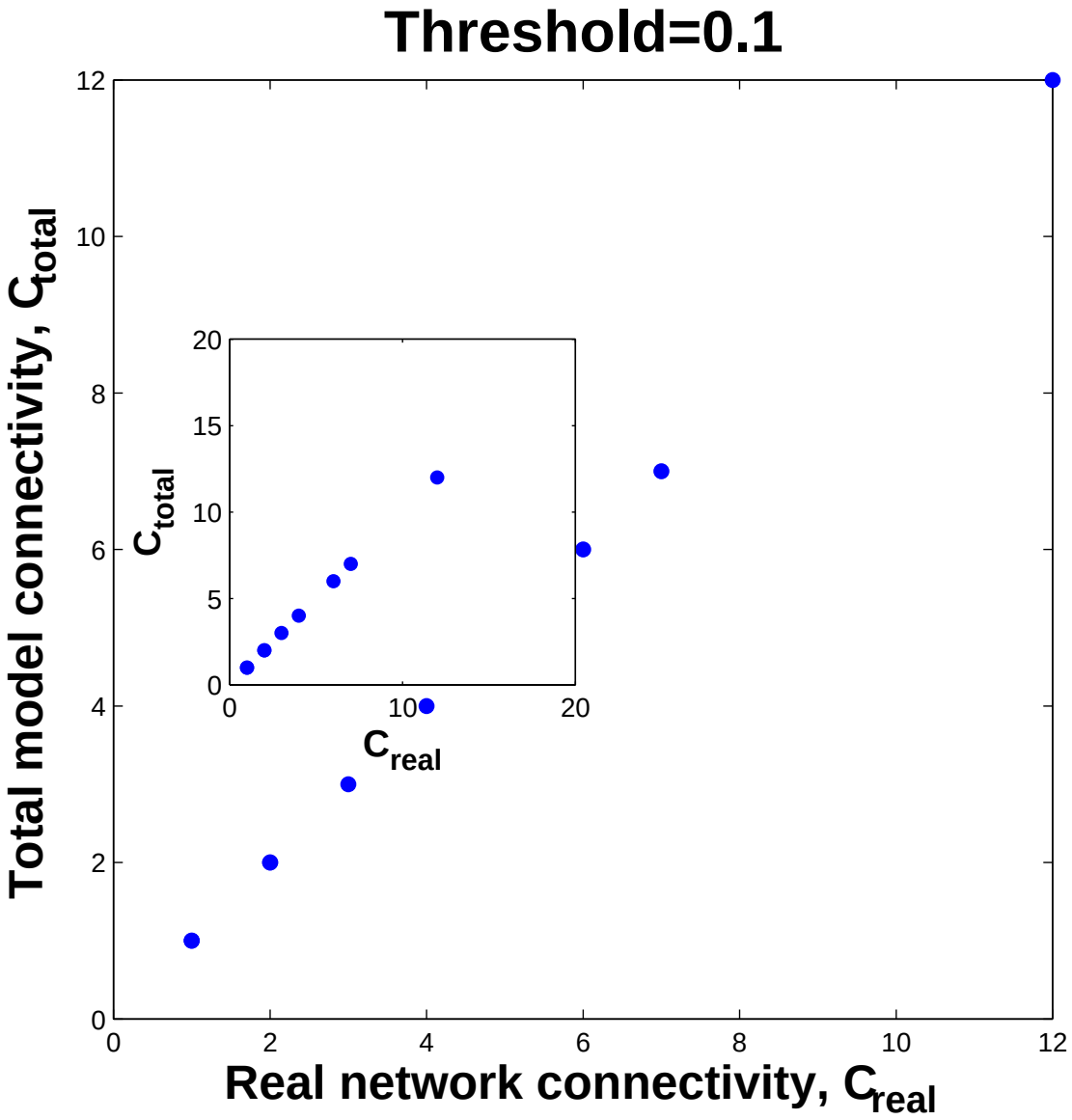


Figure 5.14: γ=4.5, the connections as real vs. the connections as model are plotted. This is obtained by assuming all complexes having higher probability than the threshold are seen in the experiment.

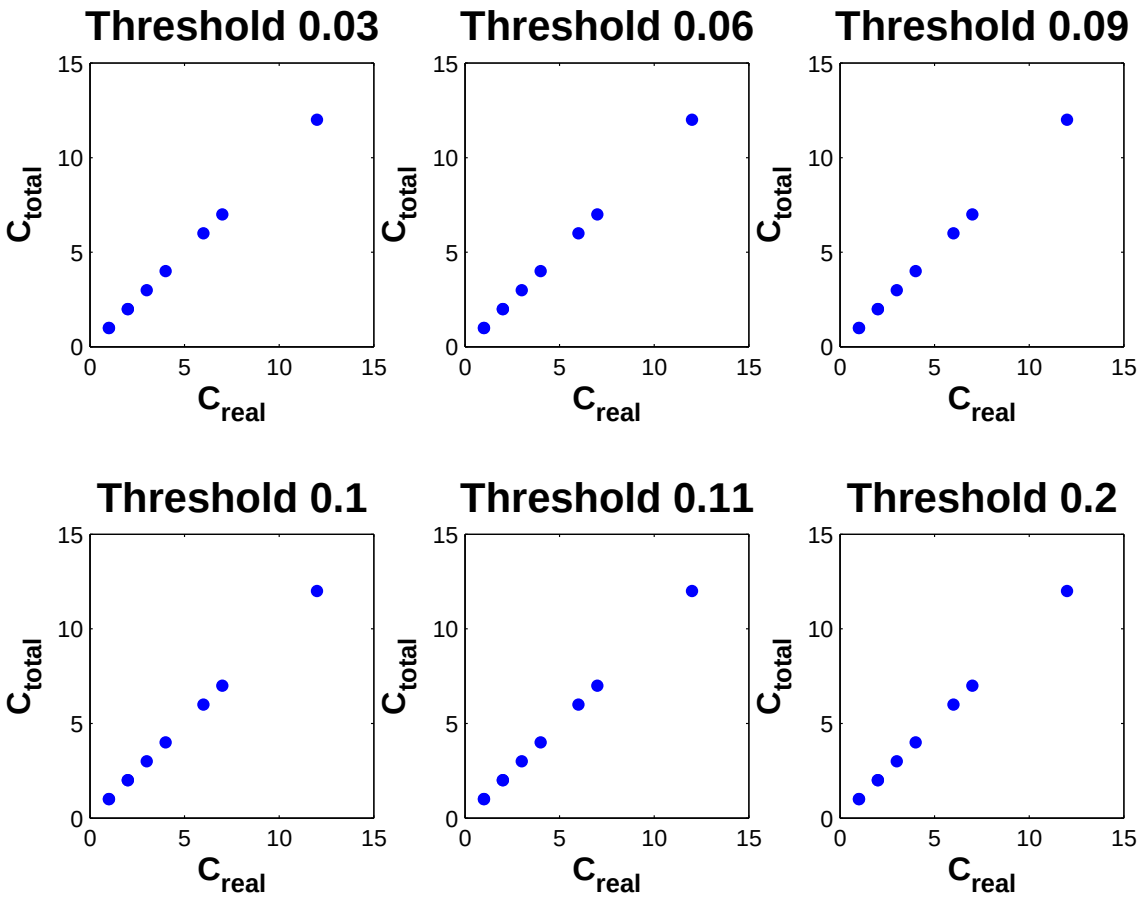


Figure 5.15: γ=4.5, the total connectivity of a protein in the model network is plotted as a function of the real connectivity of that protein from the simulated network

experiments have suggested the best parameter $w_1^{hybrid} = 0.65$. Experiments with this parameter produced 9,021 interactions, which is quite close to the size of the common set produced by each individual method.

The second voting machine

The second voting machine is used to produce the final *A. gambiae* protein-protein interaction networks. The parameters are $w_1^{hybrid} = 0.65$ and $w_1^k = 0.8$, namely the hybrid method has a weight of 0.65 and within the hybrid method, the first domain-domain to protein-protein inference method has a weight of 0.8. With this classifier, we produced 8,209 protein interaction pairs with 1,500 proteins.

Some well-known function groups have been observed in this map and they overlap with some known groups published in the literature. We show a list of ATP-binding cassette protein and the according genes we obtained from Pubmed in Table 5.8. In our inferred interaction maps, these proteins are clustered together in a dense subgroup. There are several other proteins in this cluster that are currently unannoated. We have yet to know their functions and will be working closely with the biologists to see whether they belong to the ATP-binding or not.

We have also noticed that the largest subgroup of the predicted networks contains 29 proteins and 152 edges. From Pubmed, we know that the largest Anopheles subgroup is the ABCC genes which includes one member that can potentially encode ten different isoforms of the protein by differential splicing. The largest subgroup

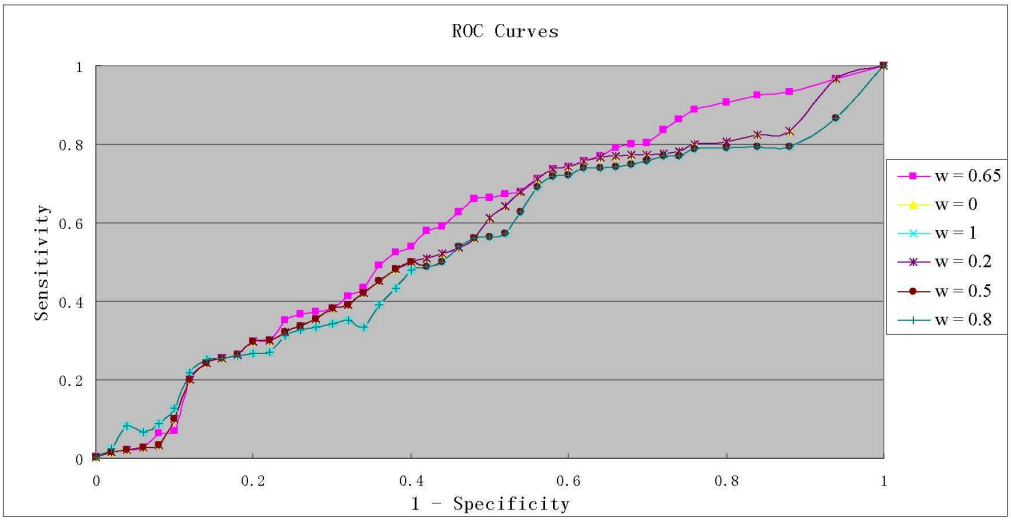


Figure 5.16: ROC curves with different weights

from our prediction contains 14 known ABCC proteins. We do not know whether the rest 15 proteins belong to the ABCC family or they are just noise.

From Pubmed, the second largest Anopheles group is the ABCG subgroup with 12 genes. We have also found the according subgroup in our predicted network, a subgroup with the size of 37. The subgroup has 92 edges and it contains 9 proteins from the ABCG family.

The predicted interactions are huge and we are currently working with the biologists to see whether those groups that have yet to be annotated are of biological interests and importance.

Table 5.8: **The *A. gambiae* ATP-binding proteins identified with our experiments**

ABC gene name	AGP protein	AGP transcript	Amino acids
ABCA1	agCP6247	agCT49435	1579
ABCA2	new		1723
ABCA3	ebiP9427	ebi9427	1777
ABCA4	agCP5051	agCG45520	1679
ABCA5	ebiP9427	ebi9427	1911
ABCA6	ebiP1718	ebi1718	1699
ABCB1	agCP14814	agCG49906	764
ABCB2	agCP6117	agCG53696	1243
ABCB3	agCP6394	agCG46625	726
ABCB4	agCP6334	agCG50982	1302
ABCB5	agCP8914	agCG43517	593
ABCC1	ebiP8456	ebi8456	1610
ABCC2	ebiP8459	ebi8459	1322
ABCC3	agCP2994	agCG55550	1450
ABCC4-	ebiP1450	ebi1450	1452
ABCC5	ebiP1450	ebi1450	1419
ABCC6	agCP4410	agCG56881	1440
ABCC7	ebiP4277	ebi4277	1285
ABCC8	new-like	ABCC9	1507
ABCC9	ebiP6599	ebi6599	1505
ABCC10	ebiP6599	ebi6599	1499
ABCC11	agCP10987	agCG56074	1414
ABCC12	ebiP7239	ebi7239	1517
ABCC13	agCP8668	agCG55063	1625
ABCC14	agCP8352	agCG55087	1510
ABCD1	agCP12768	agCG49658	744
ABCE1	agCP14752	agCG47181	642
ABCF1	agCP1742	agCG50079	623
ABCF2	agCP9633	agCG55801	910
ABCF3	ebiP8671	ebi8671	702
ABCG1	EbiP7325	ebi7325	697
ABCG2	agCP13474	agCG52831	676
ABCG3	agCP11887	agCG53539	618
ABCG4	agCP12225	agCG52594	714
ABCG5	agCP3256	agCG57489	625
ABCG6	ebiP6474	ebi6474	534
ABCG9	new		656
ABCG10	agCP14617	agCG44880	744
ABCG11	new		653
ABCG12	agCP5161	agCG44098	614
ABCH1	agCP1751	agCG49201	765
ABCH2	agCP7841	agCG45840	756

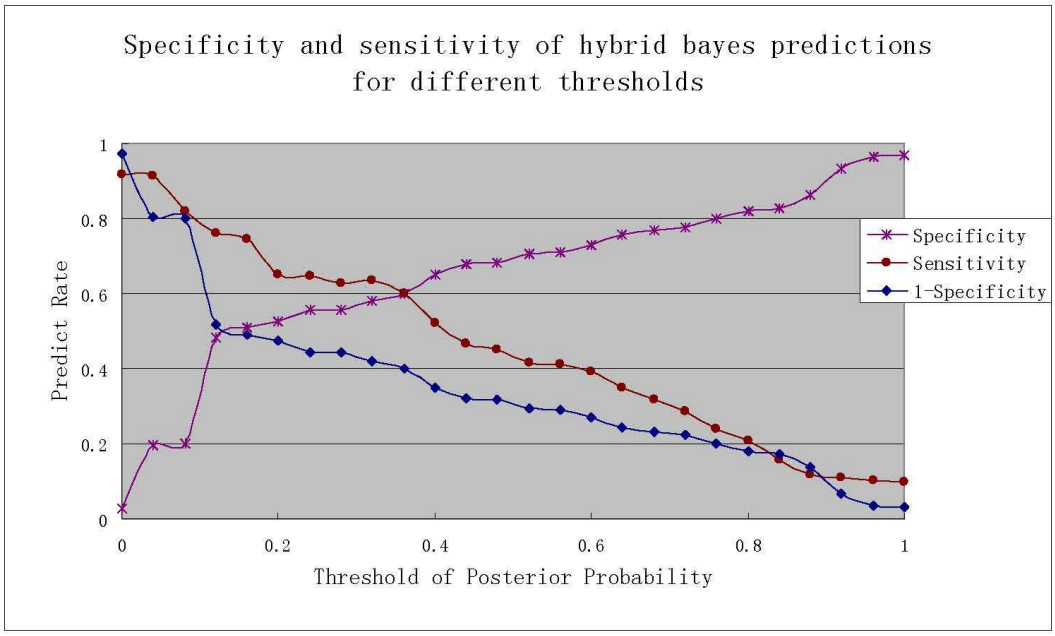


Figure 5.17: Specificity and sensitivity of hybrid bayes for different thresholds

Chapter 6

Discussion and conclusions

6.1 Discussions

We proposed estimating the probabilities of interactions between domain pairs by pooling information from five organisms based on large-scale protein interaction data and using weighted probabilites. Using the estimated domain-domain interaction probabilities with a weighted probabilties, we then estimate the probabilities of interactions between each protein pair in a given organism. We focus our attention on predicting the protein interactions in *S. cerevisiae*, and we have found that, the approach is among the best-performing methods considered in our comparisons. Because of the experimental errors of large-scale two-hybrid assays, the domain interactions inferred from one organism may not be reliable, and the incorporation of data from other organisms can indeed improve the estimated domain-domain and protein-protein interactions.

These methods are being applied to *A. gambiae* as an aid to genome annotation. For proteins with unknown function, they can associate probable function by a "guilt through association" approach - if an unknown protein is believed to bind with one of known function, we can tentatively say that it is involved in the same process or function. For proteins with known function, we can learn of new potential roles for these proteins by looking at previously unknown interactions. Extension of this will further help the study of pathways and the potential interactions between them. With continued development, this and similar methods could not only give us a better understanding of *A. gambiae* and its role in the transmission of malaria, but also provide a useful perspective from which to view and study biological processes as a whole.

The results from our approach can be further improved when the domain information is further and more reliably annotated in the future and additional protein-protein interaction data are introduced. Currently, only about two-thirds of the *S. cerevisiae* proteins have a defined domain composition, and we have considered possible interactions only between those proteins with annotated domain information. As a result, the predictions based on domain-domain interactions will be able to capture only a portion of all interactions, the number of which is estimated to be 20,000-30,000 in *S. cerevisiae*. Our predicted interacting pairs depend on the threshold value used for the estimated interaction probabilities, and the number of predicted pairs increases

as we reduce the threshold. Owing to the unknown number of truly interacting protein pairs as well as the incompleteness of the annotated domain information, it is difficult to set a threshold value to match the expected number of interacting pairs. When we set the threshold at 0.1, 20,088 protein pairs are predicted to interact with each other. At this level, using MIPS physical interaction data as the gold standard, we estimate the sensitivity and specificity to be 38.6 and 99.7%, respectively. As the interacting protein pairs included in MIPS are far from complete, these values calculated based on the MIPS data could be different from the actual values.

The basic principle of our approach is the fact that domain-domain interactions are likely conserved across different organisms, therefore allowing us to borrow information from diverse organisms to improve the predictions of protein-protein interactions in a given organism. Although our current approach has indeed led to improved predictions, it can be further refined to generate more accurate predictions. For example, we may first improve the predictions of protein-protein interactions within the same organism through integrating diverse data sources from that organism [94, 113] and then perform joint analysis across different organisms based on the results from these integrated analyses. This would be better than pooling information together and then try to improve the prediction quality afterwards.

The performance of any protein-interaction prediction method depends to a large extent not only on the method's own merits but also on the real-world properties of

the networks that it is meant to simulate. For example, the frequencies of occurrence of these types in a proteome are extremely far from uniform, and so there are a few domain types with thousands of domain copies per proteome and a large number of domain types with a single copy per proteome.

The data with which we used to start the protein-protein interaction prediction have significant effect on the model we produced. Recalled that in our project we examined a yeast two-hybrid dataset generated for a subset of *Drosophila* proteins [65]. This dataset comprises more than 20,000 experimental interactions among about 7,000 proteins. Yet according to the current estimate, there are approximately 18,000 proteins in the *Drosophila* proteome [1]. Clearly the number of known interactions seems small compared with the total number of all possible interactions [18,000 * (18,000 + 1)/2] $\approx$ 62 millions.

6.1.1 Applying the super-domain concept

Approximately 20% proteins are single-domain proteins. For the rest proteins, we could identify the super-domains of size 2, 3, 4 etc. Once these super-domains are identified, we assign a high probability (a user specified value K, times the raw probability value or set to 1 directly) to any interactions that associate with these super-domains. Where a super-domain exists, we replace the original domain-domain probability value p_{raw} with $P_{ajusted}$ Otherwise, we keep the original probability values.

By doing this, we assign higher probability values to those interactions we think are more important. The weights of the predicted VDMs (Virtual Domain Maps) would be changed, so the results of the final step, from domain-domain interaction to protein-protein interaction prediction, would also be changed.

6.1.2 The domain interaction profile pairs method

Recently a method called IDPP (Interacting domain profile pair) method has been reported to work well in predicting protein interaction maps of one species from another. Wojcik used experimentally derived *Helicobacter pylori* interaction map to predict a virtual map for *Escherichia coli* [209, 208] and the extensive literature concerning *E. coli* was used to assess all predicted interactions and to validate the IDPP method.

The IDPP method clusters protein domains by sequence and connectivity similarities and has a much better heuristic value than methods solely based on protein homology. While a straightforward algorithm predicts interactions by global alignments between full-length protein sequences, the IDPP method uses homology between interacting domains instead. IDPP uses a modified high-throughput yeast two-hybrid method which makes it possible to infer protein-protein interactions and to define restricted interacting domains. Such domains might be functionally equivalent to other well characterized motif-based domains.

The IDPP method and our proposed virtual domain-domain interaction pair method share the same idea of first constructing an abstract domain-domain interaction map from a known protein-protein interaction map. They differ significantly in how to build such a domain-domain interaction map and how to transfer the domain-domain interaction map to a protein-protein interaction map. We will try to combine the IDPP method in the future.

6.1.3 Boosting the individual dataset

While our method is useful, it does not address two important problems well. First, it estimates a set of parameters that are used for all input pairs. However, the biological datasets used contain many missing values and highly correlated features. Thus, different samples may benefit from using different feature sets. Secondly, biologists who want to use the method to select experiments can not easily determine which of the features contributed to the resulting prediction. Since different reserachers may have different opinions regarding the the reliability of the various features, it is useful if the method can indicate, for every pair, which feature contributed the most to the final result. Our next step is to divide the biological datasets into several groups. Each of the groups represents a specific data type and is used by our voting machine to predict interactions. That is, we assign weights to different data sets so that the importance of each data set is indicated. The importance could be some

input parameters for different biologists to determine their confidence in the data sets and models.

6.1.4 Validation of potential protein interactions

While our models can significantly reduce the number of required experiments by identifying a few most likely hypotheses, potential protein interactions must be validated by examining the effect of a loss of protein interaction in vivo. We will collaborate with our lab colleagues to design biological experiments to test the hypotheses we formed. This could be either selective Microarray or gene knock-out experiments. The feedback will be scrutinized and further incorporated into our next step prediction work.

6.2 Conclusions

The the orthologous transfer method and the hybrid Bayes method we proposed are able to produce good results through are experiments. For the hybrid Bayes method, once the virtual domain-domain interaction maps are constructed, we propose two ways to predict the protein-protein interaction maps. These two methods are compared and then combined to form a voting machine to collectively decide a protein-pair's candidacy. The users could adjust the weights for different methods to

flexibly control the output. Parameters are chosen through running different experiments on the training data set. We concluded that our methods are flexible, robust and accurate compared with popular methods.

While both the orthologous cluster and hybrid Bayes methods produce encouraging results the second one predicts more protein-protein interaction than the first. In view of this, we adopt a second voting machine and calibrate the parameters with the putative protein interaction data. Those parameters for the voting machine are used to predict the protein-protein interaction maps of the *A. gambiae* and produces reasonably good results.

Our contributions are as follows. Firstly we propose to use the orthologous transfer method to try to infer protein-protein interaction network. Secondly we also dig into the hybrid Bayes field. We propose the Bayes formula to represent the complex mapping relationships between protein-protein, domain-domain and back to protein-protein, we also propose two different methods to map domain-domain interaction to unknown protein-protein interaction. On top of that, we propose the notion of voting machines, and creatively used the idea in two different stages of the network modelling. It is worth pointing out that further research in any of the three directions is very likely going to reveal additional properties and produce more exciting results.

Bibliography

[1] M.D. Adams, S.E. Celniker, R.A. Holt, C.A. Evans, J.D. Gocayne, P.G. Amanatides, S.E. Scherer, P.W. Li, R.A. Hoskins, and R.F. Galle, *The genome sequence of Drosophila Melanogaster*, Science **287** (2000), 2185–2195.

[2] A Agresti, *Categorical Data Analysis*, New York: Wiley, 2002.

[3] S. Ahmad, M. Gromiha, H. Fawareh, and A. Sarai, *ASAView: Database and tool for solvent accessibility representation in proteins*, BMC Bioinformatics **5** (2004), no. 1, 51–62.

[4] Shandar Ahmad and M. Michael Gromiha, *NETASA: neural network based prediction of solvent accessibility*, Bioinformatics **18** (2002), no. 6, 819–824.

[5] R. Albert and A. Barabasi, *Statistical Mechanics of Complex Networks*, REVIEWS OF MODERN PHYSICS **74** (2002), no. 1, 47–97.

[6] H. Amrein, M. Gorman, and R. Nöthiger, *The sex-determining gene tra-2 of Drosophila encodes a putative RNA binding protein*, Cell **55** (1988), 1025–1035.

[7] W. An, S. Cho, H. Ishii, and P.C. Wensink, *Sex-specific and non-sex-specific oligomerization domains in both of the doublesex transcription factors from Drosophila melanogaster*, Mol. Cell. Biol. **16** (1996), no. 6, 3106–3111.

[8] Christophe Andrieu, Nano Freitas, Arnaud Doucet, and Michael I. Jordan, *An Introduction to MCMC for Machine Learning*, Machine Learning **50** (2003), 5–43.

[9] N. Angelopoulos and S.H. Muggleton, *Machine learning metabolic pathway descriptions using a probabilistic relational representation*, Electronic Transactions in Artificial Intelligence **7** (2002), 1–15.

[10] Gordana Apic, Julian Gough, and Sarah A. Teichmann, *Domain combinations in archaeal, eubacterial and eukaryotic proteomes*, Journal of Molecular Biology **310** (2001), no. 2, 311–325.

[11] A. Selim Aytuna, Attila Gursoy, and Ozlem Keskin, *Prediction of protein-protein interactions by combining structure and sequence conservation in protein interfaces*, Bioinformatics **21** (2005), no. 12, 2850–2855.

[12] D. A. Barbash and T. W. Cline, *Genetic and molecular analysis of the autosomal component of the primary sex determination signal of drosophila melanogaster*, Genetics **141** (1995), no. 4, 1451–1471.

[13] G. J. Bashaw and B. S. Baker, *The msl-2 dosage compensation gene of Drosophila encodes a putative DNA-binding protein whose expression is sex specifically regulated by Sex-lethal*, Development **121** (1995), no. 10, 3245–3258.

[14] Alex Bateman, E. Birney, Lachlan Coin, and Durbin et al, *The Pfam protein families database*, Nucl. Acids Res. **32** (2004), D138–141.

[15] J. M. Belote and B. S. Baker, *Sex determination in Drosophila melanogaster: Analysis of transformer-2, a sex-transforming locus*, Proc. Natl. Acad. Sci. **79** (1982), 1568–1572.

[16] M. Benedict and A. Robinson, *the first releases of transgenic mosquitoes: an argument for the sterile insect technique*, Trends in Parasitology **19** (2003), 349–355.

[17] S. Bhatta, *A model-based approach to analogical reasoning and learning in design*, PhD thesis proposal GIT-CC-92/60, College of Computing, Georgia Institute of Technology, 1992.

[18] B. Bjellqvist, G.J. Hughes, Ch. Pasquali, N. Paquet, F. Ravier, J.-Ch. Sanchez, S. Frutiger, and D.F. Hochstrasser, *The focusing positions of polypeptides in immobilized pH gradients can be predicted from their amino acid sequences*, Electrophoresis **14** (1993), no. 10, 1023–1031.

[19] Viacheslav N. Bolshakov, Pantelis Topalis, Claudia Blass, Elena Kokoza, Alessandra della Torre, Fotis C. Kafatos, and Christos Louis, *A Comparative Genomic Analysis of Two Distant Diptera, the Fruit Fly, Drosophila melanogaster, and the Malaria Mosquito, Anopheles gambiae*, Genome Res. **12** (2002), no. 1, 57–66.

[20] I. Bratko, S. H. Muggleton, and A. Varvsek, *Learning qualitative models of dynamic systems*, Inductive Logic Programming (S. H. Muggleton, ed.), Academic Press, 1992, pp. 437–452.

[21] I. Bratko and S.H. Muggleton, *Applications of inductive logic programming*, Communications of the ACM **38** (1995), no. 11, 65–70.

[22] L Breiman, *Random Forests*, Machine Learning **24** (2001), 123–140.

[23] K. C. Burtis and B. S. Baker, *Drosophila doublesex gene controls somatic sexual differentiation by producing alternatively spliced mRNAs encoding related sex-specific polypeptides*, Cell **56** (1989), 997–1010.

[24] D. Chandler, M.E. McGuffin, J. Piskur, J. Yao, B.S. Baker, and W. Mattox, *Evolutionary conservation of regulatory strategies for the sex determination factor transformer-2*, Mol. Cell. Biol. **17** (1997), no. 5, 2908–2919.

[25] M. Charton and B. I. Charton, *The Structural Dependence of Amino Acid Hydrophobicity Parameters*, J. theor. Biol. **99** (1982), 629–644.

[26] B. A. Chase and B. S. Baker, *A Genetic Analysis of intersex, a Gene Regulating Sexual Differentiation in Drosophila melanogaster Females*, Genetics **139** (1995), no. 4, 1649–1661.

[27] Yu Chen and Dong Xu, *Global protein function annotation through mining genome-scale data in yeast Saccharomyces cerevisiae*, Nucl. Acids Res. **32** (2004), no. 21, 6414–6424.

[28] Sayeon Cho and Pieter Wensink, *Linkage between oligomerization and dna binding in drosophila doublesex proteins*, Biochemistry **37** (1998), no. 6, 11301–11308.

[29] Sayeon Cho and Pieter C. Wensink, *DNA Binding by the Male and Female Doublesex Proteins of Drosophila melanogaster*, J. Biol. Chem. **272** (1997), no. 6, 3185–3189.

[30] C. Chothia and A. V. Finkelstein, *The classification and origins of protein folding patterns*, Annu Rev Biochem **59** (1990), 1007–1039.

[31] John Clement, *Observed methods for generating analogies in scientific problem solving*, Cognitive Science **12** (1988), 563–586.

[32] T. Cline, *The drosophila sex determination signal how do flies count to two*, Genetics **9** (1993), no. 11, 385–390.

[33] T. W. Cline, *Evidence That sisterless-a and sisterless-b Are Two of Several Discrete "Numerator Elements" of the X/A Sex Determination Signal in Drosophila That Switch Sxl Between Two Alternative Stable Expression States*, Genetics **119** (1988), no. 4, 829–862.

[34] Jacques Colinge, Jéróme Magnin, Thierry Dessingy, Marc Giron, and Alexandre Masselot, *Improved peptide charge state assignment*, Proteomics **3** (2002), no. 8, 1434–1440.

[35] A. Cootes, S.H. Muggleton, and M.J.E. Sternberg, *The automatic discovery of structural principles describing protein fold space*, Journal of Molecular Biology **330** (2003), 839–850.

[36] J. L. Cornette, K. B. Cease, H. Margalit, J. L. Spouge, J. A. Berzofsky, and C. DeLisi, *Hydrophobicity scales and computational techniques for detecting amphipathic structures in proteins*, J Mol Biol. **195** (1987), no. 3, 659–685.

[37] K. T. Coschigano and P. C. Wensink, *Sex-specific transcriptional regulation by the male and female doublesex proteins of Drosophila*, Genes Dev. **7** (1993), no. 1, 42–54.

[38] C. Cronmiller and T. W. Schedl, P.and Cline, *Molecular characterization of daughterless, a Drosophila sex determination gene with multiple roles in development*, Genes Dev. **2** (1988), no. 12, 1666–1676.

[39] J. Cussens, D. Page, S. Muggleton, and A. Srinivasan, *Using Inductive Logic Programming for Natural Language Processing*, ECML97 - Workshop Notes on Empirical Learning of Natural Language Tasks (Prague) (W. Daelemans, T. Weijters, and A. van der Bosch, eds.), University of Economics, 1997, Invited keynote paper, pp. 25–34.

[40] Todd R. Davies and Stuart Russell, *A Logical Approach to Reasoning by Analogy*, Proceedings of the Tenth International Joint Conference on Artificial Intelligence, 1987.

[41] Minghua Deng, Shipra Mehta, Fengzhu Sun, and Ting Chen, *Inferring Domain-Domain Interactions From Protein-Protein Interactions*, Genome Res. **12** (2002), no. 10, 1540–1548.

[42] Minghua Deng, Fengzhu Sun, and Ting Chen, *Assessment of the reliability of protein-protein interactions and protein function prediction*, Pacific Symposium on Biocomputing Online Proceedings **8** (2003), 140–151.

[43] G. Deshpande, J. Stukey, and P. Schedl, *scute (sis-b) function in Drosophila sex determination*, Mol. Cell. Biol. **15** (1995), no. 8, 4430–4440.

[44] B. Dolsak and S.H. Muggleton, *The application of Inductive Logic Programming to finite element mesh design*, Inductive Logic Programming (S. H. Muggleton, ed.), Academic Press, London, 1992, pp. 453–472.

[45] Douglas R. Dorer, Jamie A. Rudnick, Etsuko N. Moriyama, and Alan C. Christensen, *A Family of Genes Clustered at the Triplo-lethal Locus of Drosophila melanogaster Has an Unusual Evolutionary History and Significant Synteny With Anopheles gambiae*, Genetics **165** (2003), no. 2, 613–621.

[46] Inna Dubchak, Ilya Muchnik, Christopher Mayor, Igor Dralyuk, and Sung-Hou Kim, *Recognition of a Protein Fold in the Context of the SCOP Classification*, PROTEINS: Structure, Function, and Genetics **35** (1999), no. 4, 401–407.

[47] J.B. Duffy and J.P. Gergen, *The Drosophila segmentation gene runt acts as a position-specific numerator element necessary for the uniform expression of the sex- determining gene Sex-lethal*, Genes Dev. **5** (1991), no. 12, 2176–2187.

[48] S. Dzeroski, Nico Jacobs, M. Molina, and C. Moure, *Ilp experiments in detecting traffic problems*, 10th European Conference on Machine Learning, Lecture Notes in Artificial Intelligence, Springer-Verlag, August 1998, pp. 61–66.

[49] S. Dzeroski and N. Lavrac (eds.), *Relational data mining*, Springer-Verlag, Berlin, September 2001.

[50] S. R. Eddy, *Profile Hidden Markov Models*, Bioinformatics **14** (1998), 755–763.

[51] Anton J. Enright, Ioannis Iliopoulos, Nikos C. Kyrpides, and Christos A. Ouzounis, *Protein interaction maps for complete genomes based on gene fusion events*, Nature **402** (1999), 86–90.

[52] J.W. Erickson and T.W. Cline, *A bZIP protein, sisterless-a, collaborates with bHLH transcription factors early in Drosophila development to determine sex*, Genes Dev. **7** (1993), no. 9, 1688–1702.

[53] P.A. Estes, L.N. Keyes, and P. Schedl, *Multiple response elements in the Sex-lethal early promoter ensure its female-specific expression pattern*, Mol. Cell. Biol. **15** (1995), no. 2, 904–917.

[54] G. D. Fasman, *The handbook of biochemistry and molecular biology*, vol. 1, CRC Press, 1975.

[55] J. L. Fauchere, M. Charton, L. B. Kier, A. Verloop, and V. Pliska, *Amino acid side chain parameters for correlation studies in biology and pharmacology*, Int J Pept Protein Res. **32** (1988), no. 4, 269–278.

[56] M. Fellenberg, K. Albermann, A. Zollner, H. M. Mewes, and J. Hani, *Integrative analysis of protein interaction data*, Intell. Syst. Mol. Biol. **8** (2000), 152–161.

[57] Kim D. Finley, Barbara J. Taylor, Marc Milstein, and Michael McKeown, *dissatisfaction, a gene involved in sex-specific behavior and neural development of Drosophila melanogaster*, PNAS **94** (1997), no. 3, 913–918.

[58] P. Finn, S.H. Muggleton, D. Page, and A. Srinivasan, *Pharmacophore discovery using the inductive logic programming system progol*, Machine Learning **30** (1998), 241–271.

[59] T. W. Flickinger and H. K. Salz, *The Drosophila sex determination gene snf encodes a nuclear protein with sequence and functional similarity to the mammalian U1A snRNP protein*, Genes Dev. **8** (1994), no. 8, 914–925.

[60] Matthias E. Futschik, Gautam Chaurasia, and Hanspeter Herzel, *Comparison of human protein-protein interaction maps*, Bioinformatics **23** (2007), no. 5, 605–611.

[61] C Gaboriaud, V Bissery, T Benchetrit, and Mornon J. P., *Hydrophobic cluster analysis: an efficient new way to compare and analyse amino acid sequences*, FEBS Lett. **224** (1987), no. 1, 149–155.

[62] Carrie M. Garrett-Engele, Mark L. Siegal, Devanand S. Manoli, Byron C. Williams, Hao Li, and Bruce S. Baker, *intersex, a gene required for female sexual development in drosophila, is expressed in both sexes and functions together with doublesex to regulate terminal differentiation*, Development **129** (2002), 4661–4675.

[63] W. R. Gilks, S. Richardson, and David Spiegelhalter, *Markov Chain Monte Carlo in Practice*, Chapman and Hall, 1996.

[64] S.C. Gill and P.H. von Hippel, *Calculation of protein extinction coefficients from amino acid sequence data*, Anal Biochem. **182** (1989), no. 2, 319–326.

[65] L. Giot, J. S. Bader, C. Brouwer, A. Chaudhuri, B. Kuang, Y. Li, Y. L. Hao, C. E. Ooi, B. Godwin, E. Vitols, G. Vijayadamodar, P. Pochart, H. Machineni, M. Welsh, Y. Kong, B. Zerhusen, R. Malcolum, Z. Varrone, A. Collis, M. Minto, S. Burgess, L. Mcdaniel, E. Stimpson, F. Spriggs, J. Williams, K. Neurath,

N. Loime, M. Agee, E. Voss, K. Furtak, R. Renzulli, N. Aanensen, S. Carrolla, E. Bickelhaupt, Y. Lazovatsky, A. Dasilva, J. Zhong, C. A. Stanyon, R. L. Finley Jr., K. P. White, M. Braverman, T. Jarvie, S. Gold, M. Leach, J. Knight, R. A. Shimkets, M. P. Mckenna, J. Chant, and J. M. Rothberg, *A Protein Interaction Map of Drosophila Melanogaster*, Science **302** (2003), 1727–1736.

[66] Chern-Sing Goh and Fred E. Cohen, *Co-evolutionary Analysis Reveals Insights into Protein-Protein Interactions*, Journal of Molecular Biology **324** (2002), no. 1, 177–192.

[67] Shawn M. Gomez, Shaw-Hwa Lo, and Andrey Rzhetsky, *Probabilistic Prediction of Unknown Metabolic and Signal-Transduction Networks*, Genetics **159** (2001), no. 3, 1291–1298.

[68] Shawn M. Gomez, William Stafford Noble, and Andrey Rzhetsky, *Learning to predict protein-protein interactions from protein sequences*, Bioinformatics **19** (2003), no. 15, 1875–1881.

[69] P. Graham, J. Penn, and P. Schedl, *Masters change, slaves remain*, Bioessays **25** (2002), 1–4.

[70] B. Granadino, A. S. Juan, P. Santamaria, and L. Sanchez, *Evidence of a Dual Function in fl(2)d, a Gene Needed for Sex-lethal Expression in Drosophila melanogaster*, Genetics **130** (1992), no. 3, 597–612.

[71] R. Grantham, *Amino acid difference formula to help explain protein evolution*, Science **185** (1974), no. 4154, 862–864.

[72] P. J. Green, *Linear models for field trials,smoothing and cross-validation*, Biometrika **82** (1985), 523.

[73] Russell Greiner, *Learning by understanding analogies*, Artificial Intelligence **35** (1988), no. 1, 81–125.

[74] Andrei Grigoriev, *On the number of protein-protein interactions in the yeast proteome*, Nucleic Acids Research **31** (2003), no. 14, 4157–4161.

[75] D. Gunetti and G. Ruffo, *Intrusion Detection through Behavioral Data*, Proc. of The Third Symposium on Intelligent Data Analysis (Amsterdam, The Netherlands) (S. Wrobel, ed.), Lecture Notes in Artificial Intelligence, Springer-Verlag, 1999.

[76] K. Guruprasad, B.V.B. Reddy, and M.W. Pandit, *Correlation between stability of a protein and its dipeptide composition: a novel approach for predicting in vivo stability of a protein from its primary sequence*, Protein Engineering **4** (1990), no. 2, 155–161.

[77] Juergen Haas and G. Christian Aaronson, Jeffery S.and Overton, *Analogical reasoning for knowledge discovery in a molecular biology database*, CIKM, 1993, pp. 554–564.

[78] Rogers P. Hall, *Computational approaches to analogical reasoning: a comparative analysis*, Artificial Intelligence **39** (1989), no. 1, 39–120.

[79] David Heckerman, *A Tutorial on Learning Bayesian Networks*, Tech. Report MSR-TR-95-06, Microsoft Research, Redmond, WA, March 1995.

[80] A.K. Hickman and J. Larkin, *Internal analogy: A model of transfer within problems*, Proceedings of the 12th Annual Conference of the Cognitive Science Society, 1990, pp. 53–60.

[81] A. Hilfiker, H. Amrein, A. Dubendorfer, R. Schneiter, and R. Nothiger, *The gene virilizer is required for female-specific splicing controlled by Sxl, the master gene for sexual development in Drosophila*, Development **121** (1995), no. 12, 4017–4026.

[82] D. Hilfiker-Kleiner, A. Dubendorfer, A. Hilfiker, and R. Nothiger, *Genetic control of sex determination in the germ line and soma of the housefly, Musca domestica*, Development **120** (1994), no. 9, 2531–2538.

[83] S. Hinson and R. N. Nagoshi, *Regulatory and functional interactions between the somatic sex regulatory gene transformer and the germline genes ovo and ovarian tumor*, Development **126** (1999), no. 5, 861–871.

[84] E. Hirowatari and S. Arikawa, *Partially isomorphic generalization and analogical reasoning*, Machine Learning: ECML-94 - Proc. of the European Conference on Machine Learning (F. Bergadano and L. De Raedt, eds.), Springer, Berlin, Heidelberg, 1994, pp. 363–366.

[85] J. Hodgkin, *Sex determination compared in Drosophila and Caenorhabditis*, Nature **344** (1990), 721–728.

[86] Robert A. Holt, G. Mani Subramanian, Aaron Halpern, Granger G. Sutton, Rosane Charlab, Deborah R. Nusskern, Patrick Wincker, Andrew G. Clark, Jose M. C. Ribeiro, Ron Wides, Steven L. Salzberg, Brendan Loftus, Mark Yandell, William H. Majoros, Douglas B. Rusch, Zhongwu Lai, Cheryl L. Kraft, Josep F. Abril, Veronique Anthouard, Peter Arensburger, Peter W. Atkinson, Holly Baden, Veronique de Berardinis, Danita Baldwin, Vladimir Benes, Jim Biedler, Claudia Blass, Randall Bolanos, Didier Boscus, Mary Barnstead, Shuang Cai, Angela Center, Kabir Chatuverdi, George K. Christophides, Mathew A. Chrystal, Michele Clamp, Anibal Cravchik, Val Curwen, Ali Dana, Art Delcher, Ian Dew, Cheryl A. Evans, Michael Flanigan, Anne Grundschober-Freimoser, Lisa Friedli, Zhiping Gu, Ping Guan, Roderic Guigo, Maureen E. Hillenmeyer, Susanne L. Hladun, James R. Hogan, Young S. Hong, Jeffrey Hoover, Olivier Jaillon, Zhaoxi Ke, Chinnappa Kodira, Elena Kokoza, Anastasios Koutsos, Ivica Letunic, Alex Levitsky, Yong Liang, Jhy-Jhu Lin, Neil F. Lobo, John R.

Lopez, Joel A. Malek, Tina C. McIntosh, Stephan Meister, Jason Miller, Clark Mobarry, Emmanuel Mongin, Sean D. Murphy, David A. O'Brochta, Cynthia Pfannkoch, Rong Qi, Megan A. Regier, Karin Remington, Hongguang Shao, Maria V. Sharakhova, Cynthia D. Sitter, Jyoti Shetty, Thomas J. Smith, Renee Strong, Jingtao Sun, Dana Thomasova, Lucas Q. Ton, Pantelis Topalis, Zhijian Tu, Maria F. Unger, Brian Walenz, Aihui Wang, Jian Wang, Mei Wang, Xuelan Wang, Kerry J. Woodford, Jennifer R. Wortman, Martin Wu, Alison Yao, Evgeny M. Zdobnov, Hongyu Zhang, Qi Zhao, Shaying Zhao, Shiaoping C. Zhu, Igor Zhimulev, Mario Coluzzi, Alessandra della Torre, Charles W. Roth, Christos Louis, Francis Kalush, Richard J. Mural, Eugene W. Myers, Mark D. Adams, Hamilton O. Smith, Samuel Broder, Malcolm J. Gardner, Claire M. Fraser, Ewan Birney, Peer Bork, Paul T. Brey, J. Craig Venter, Jean Weissenbach, Fotis C. Kafatos, Frank H. Collins, and Stephen L. Hoffman, *The Genome Sequence of the Malaria Mosquito Anopheles gambiae*, Science **298** (2002), no. 5591, 129–149.

[87] Paul Holyoak, Keith J.and Thagard1, *Analogical mapping by constraint satisfaction*, Cognitive Science **13** (1989), 295–355.

[88] T. P. Hopp and K. R. Woods, *Prediction of protein antigenic determinants from amino acid sequences*, Proc Natl Acad Sci **78** (1981), no. 6, 3824C3828.

[89] H. Robert Horton, Laurence A. Moran, Raymond S. Ochs, J. David Rawn, and K. Gray Scrimgeour, *Principles of biochemistry*, Prentice Hall, 2002.

[90] K. Hoshijima, K. Inoue, I. Higuchi, H. Sakamoto, and Y. Shimura, *Control of doublesex alternative splicing by transformer and transformer-2 in Drosophila*, Science **252** (1991), 833–836.

[91] A. Ikai, *Thermostability and aliphatic index of globular proteins*, J Biochem **88** (1980), no. 6, 1895–1898.

[92] Takashi Ito, Tomoko Chiba, Ritsuko Ozawa, Mikio Yoshida, Masahira Hattori, and Yoshiyuki Sakaki, *A comprehensive two-hybrid analysis to explore the yeast protein interactome*, PNAS **98** (2001), no. 8, 4569–4574.

[93] Olivier Jaillon, Carole Dossat, Ralph Eckenberg, Karin Eiglmeier, Beatrice Segurens, Jean-Marc Aury, Charles W. Roth, Claude Scarpelli, Paul T. Brey, Jean Weissenbach, and Patrick Wincker, *Assessing the Drosophila melanogaster and Anopheles gambiae Genome Annotations Using Genome-Wide Sequence Comparisons*, Genome Res. **13** (2003), no. 7, 1595–1599.

[94] Ronald Jansen, Haiyuan Yu, Dov Greenbaum, Yuval Kluger, Nevan J. Krogan, Sambath Chung, Andrew Emili, Michael Snyder, Jack F. Greenblatt, and Mark Gerstein, *A Bayesian Networks Approach for Predicting Protein-Protein Interactions from Genomic Data*, Science **302** (2003), no. 5644, 449–453.

[95] D T Jones, C Orengo, W R Taylor, and J M Thornton, *Progress towards recognizing protein folds from amino-acid-sequence*, Protein Engineering **6** (1993), 124.

[96] Susan Jones and Janet M. Thornton, *Principles of protein-protein interactions*, Proc. Natl. Acad. Sci. **93** (1996), no. 1, 13–20.

[97] Thomas C. Kaufman, David W. Severson, and Gene E. Robinson, *The Anopheles Genome and Comparative Insect Genomics*, Science **298** (2002), no. 5591, 97–98.

[98] Ryan Kelley and Trey Ideker, *Systematic interpretation of genetic interactions using protein networks*, Nat Biotech **23** (2005), 561–566.

[99] Manfred Kerber and Erica Melis, *Using exemplary knowledge for justified analogical reasoning*, WOCFAI, 1995, pp. 157–168.

[100] Manfred Kerber, Erica Melis, and Jörg Siekmann, *Analogical reasoning with typical examples*, Seki Report SR-92-13, Fachbereich Informatik, Universität des Saarlandes, Saarbrücken, Germany, 1992.

[101] Manfred Kerber, Erica Melis, and Jörg H. Siekmann, *Analogical reasoning with a hybrid knowledge base*.

[102] Manfred Kerber, Erica Melis, and Jorg H. Siekmann, *Reasoning with assertions and examples*, Working Notes of the AAAI Spring Symposium on AI and Creativity (Stanford, California, USA) (Terry Dartnell, Steven Kim, and Fay Sudweeks, eds.), 1993, pp. 61–66.

[103] K. Kersting and L. De Raedt, *Towards combining inductive logic programming with bayesian networks*, Proceedings of the 11th International Conference on Inductive Logic Programming (Céline Rouveirol and Michèle Sebag, eds.), Lecture Notes in Artificial Intelligence, vol. 2157, Springer-Verlag, September 2001, pp. 118–131.

[104] L. N. Keyes, T. W. Cline, and P. Schedl, *The primary sex determination signal of Drosophila acts at the level of transcription*, Cell **68** (1992), 933–943.

[105] R.D. King, K.E. Whelan, F.M. Jones, P.G. Reiser, C.H. Bryant, S.H. Muggleton, D.B. Kell, and S.G. Oliver, *Functional genomic hypothesis generation and experimentation by a robot scientist*, Nature **427** (2004), no. 6971, 247–252.

[106] S. G. Kramer, T. M. Jinks, P. Schedl, and J. P. Gergen, *Direct activation of Sex-lethal transcription by the Drosophila runt protein*, Development **126** (1999), no. 1, 191–200.

[107] N. Lavravc and S. Dvzeroski, *Inductive logic programming: Techniques and applications*, Ellis Horwood, 1994.

[108] N. Lavravc, S. Dvzeroski, and M. Grobelnik, *Learning nonrecursive definitions of relations with linus*, Proceedings of the 5th European Working Session on Learning (Y. Kodratoff, ed.), Lecture Notes in Artificial Intelligence, vol. 482, Springer-Verlag, 1991, pp. 265–281.

[109] Nada Lavrač and Peter A. Flach, *An extended transformation approach to inductive logic programming*, ACM Trans. Comput. Logic **2** (2001), no. 4, 458–494.

[110] Insuk Lee, Shailesh V. Date, Alex T. Adai, and Edward M. Marcotte, *A Probabilistic Functional Network of Yeast Genes*, Science **306** (2004), no. 5701, 1555–1558.

[111] Pierre Legrain, Jerome Wojcik, and Jean-Michel Gauthier, *Protein-protein interaction maps: a lead towards cellular functions*, Trends in Genetics **17** (2001), no. 6, 346–352.

[112] Ivica Letunic, Richard R. Copley, Steffen Schmidt, Francesca D. Ciccarelli, Tobias Doerks, Jorg Schultz, Chris P. Ponting, and Peer Bork, *SMART 4.0: towards genomic data integration*, Nucl. Acids Res. **32** (2004), no. 90001, D142–144.

[113] Nan Lin, Baolin Wu, Ronald Jansen, Mark Gerstein, and Hongyu Zhao, *Information assessment on predicting protein-protein interactions*, BMC Bioinformatics **5** (2004), no. 1, 154.

[114] T. Y. LIN and S. N. TIMASHEFF, *On the role of surface tension in the stabilization of globular proteins*, Protein Science **5** (1996), no. 2, 372–381.

[115] A. Javier Lopez, *Alternative splicing of pre-mrna: Developmental consequences and mechanisms of regulation*, Annual Review of Genetics **32** (1998), no. 1, 279–305.

[116] Thanasis G. Loukeris, Ioannis Livadaras, Bruno Arca, Sophia Zabalou, and Charalambos Savakis, *Gene transfer into the medfly, ceratitis capitata, with a drosophila hydei transposable element*, Science **270** (1995), no. 5244, 2002–2005.

[117] Van Doren M., H. M. Ellis, and J. W. Posakony, *The Drosophila extramacrochaetae protein antagonizes sequence-specific DNA binding by daughterless/achaete-scute protein complexes*, Development **113** (1991), no. 1, 245–255.

[118] H. Craig Mak, Mike Daly, Bianca Gruebel, and Trey Ideker, *CellCircuits: a database of protein network models*, Nucleic Acids Research **35** (2007), D538D545.

[119] Hiroshi Mamitsuka, *Essential latent knowledge for protein-protein interactions: Analysis by an unsupervised learning approach*, IEEE/ACM Trans. Comput. Biol. Bioinformatics **2** (2005), no. 2, 119–130.

[120] Edward M. Marcotte, Ioannis Xenarios, and David Eisenberg, *Mining literature for protein-protein interactions*, Bioinformatics **17** (2001), no. 4, 359–363.

[121] I. Marín and Bruce S. Baker, *The Evolutionary Dynamics of Sex Determination*, Science **281** (1998), no. 5385, 1990–1994.

[122] Liam J. McGuffin, Kevin Bryson, and David T. Jones, *The PSIPRED protein structure prediction server*, Bioinformatics **16** (2000), no. 4, 404–405.

[123] M. Meise, D. Hilfiker-Kleiner, A. Dubendorfer, C. Brunner, R. Nothiger, and D. Bopp, *Sex-lethal, the master sex-determining gene in Drosophila, is not sex-specifically regulated in Musca domestica*, Development **125** (1998), no. 8, 1487–1494.

[124] Erica Melis and Manuela M. Veloso, *Analogy in problem solving*, Handbook of Practical Reasoning Computational and Theoretical Aspects (L.Farinas del

Cerro, D. Gabbay, and H.J. Ohlbach, eds.), Oxford University Press, 1998, invited contribution, in press.

[125] Thomas M. Mitchell, *Machine learning*, McGraw-Hill Higher Education, 1997.

[126] Emmanuel Mongin, Christos Louis, Robert A. Holt, Ewan Birney, and Frank H. Collins, *The Anopheles gambiae genome: an update*, Trends in Parasitology **20** (2004), no. 2, 49–52.

[127] R.J. Mooney, *Inductive logic programming for natural language processing*, Proceedings of the 6th International Workshop on Inductive Logic Programming (S. H. Muggleton, ed.), Lecture Notes in Artificial Intelligence, vol. 1314, Springer-Verlag, 1996, pp. 3–24.

[128] Lyria Mori and A. L. P. Perondini, *Errors in the elimination of X chromosome in Sciara ocellaris*, Genetics **94** (1980), no. 3, 663–673.

[129] S. Muggleton, *Inverting implication*, Proceedings of the 2nd International Workshop on Inductive Logic Programming (S. Muggleton, ed.), 1992, pp. 19–39.

[130] S. Muggleton and J. Firth, *Cprogol4.4: a tutorial introduction*, Relational Data Mining (S. Dzeroski and N. Lavrac, eds.), Springer-Verlag, 2001, pp. 160–188.

[131] S. H. Muggleton and W. Buntine, *Machine invention of first order predicates by inverting resolution*, ML88, MK, 1988, pp. 339–351.

[132] S. H. Muggleton and C. Feng, *Efficient induction of logic programs*, Proceedings of the 1st Conference on Algorithmic Learning Theory, 1990, pp. 368–381.

[133] S.H. Muggleton, *Inductive logic programming*, Inductive Logic Programming, Academic Press, 1992, pp. 3–28.

[134] S.H. Muggleton and M. Bain, *Analogical prediction*, Proc. of the 9th International Workshop on Inductive Logic Programming (ILP-99) (Berlin), Springer-Verlag, 1999, pp. 234–244.

[135] S.H. Muggleton, A. Tamaddoni-Nezhad, and H. Watanabe, *Induction of enzyme classes from biological databases*, ILP03 (T. Horvath and A. Yamamoto, eds.), LNAI, vol. 2835, SV, 2003, pp. 269–280.

[136] Stephen Muggleton, *Inverting entailment and progol*, Machine intelligence 14: applied machine intelligence (1996), 133–187.

[137] S.-K. Ng, Z. Zhuo, S.-H. Tan, and K. Lin, *InterDom: a database of putative interacting protein domains for validating predicted protein interactions and complexes*, Nucleic Acids Research **31** (2003), no. 1, 251–254.

[138] See-Kiong Ng, Zhuo Zhang, and Soon-Heng Tan, *Integrative approach for computationally inferring protein domain interactions*, Bioinformatics **19** (2003), no. 8, 923–929.

[139] Shan-Hwei. Nienhuys-Cheng and Ronald Wolf, *Foundations of inductive logic programming*, Springer-Verlag Berlin Heidelberg, 1997.

[140] R. Nöthiger and M. Steinmann-Zwicky, *Sex determination in Drosophila* , Genetics **1** (1985), 209–215.

[141] Kevin P. O'Brien, Maido Remm, and Erik L. L. Sonnhammer, *Inparanoid: a comprehensive database of eukaryotic orthologs*, Nucl. Acids Res. **33** (2005), D476–D480.

[142] B. Oliver, Y. J. Kim, and B. S. Baker, *Sex-lethal, master and slave: a hierarchy of germ-line sex determination in Drosophila*, Development **119** (1993), no. 3, 897–908.

[143] M. T. O'Neil and J. M. Belote, *Interspecific Comparison of the transformer Gene of Drosophila Reveals an Unusually High Degree of Evolutionary Divergence*, Genetics **131** (1992), no. 1, 113–128.

[144] Rainer Opgen-Rhein and Korbinian Strimmer, *Learning causal networks from systems biology time course data: An effective model selection procedure for the vector autoregressive process*, Proceedings of PMSB 2006, 2006.

[145] M. Ouali and R. D. King, *Cascaded multiple classifiers for secondary structure prediction*, Protein Science **9** (2000), no. 6, 1162–1176.

[146] C. N. PACE, F. VAJDOS, L. FEE, G. GRIMSLEY, and T. GRAY, *How to measure and predict the molar absorption coefficient of a protein*, Protein Science **4** (1995), no. 11, 2411–2423.

[147] Debnath Pal and David Eisenberg, *Inference of Protein Function from Protein Structure*, Structure **13** (2005), 121–130.

[148] S. M. Parkhurst, D. Bopp, and D. Ish-Horowicz, *X:A ratio, the primary sex-determining signal in Drosophila, is transduced by helix-loop-helix proteins*, Cell **63** (1990), no. 6, 1179–1191.

[149] Z. Paroush, R. L. Finley, T. Kidd, S. M. Wainwright, P. W. Ingham, R. Brent, and D. Ish-Horowicz, *Groucho is required for Drosophila neurogenesis, segmentation, and sex determination and interacts directly with hairy-related bHLH proteins*, Cell **79** (1994), no. 5, 805–815.

[150] D. Pauli, B. Oliver, and A. P. Mahowald, *The role of the ovarian tumor locus in Drosophila melanogaster germ line sex determination*, Development **119** (1993), no. 1, 123–134.

[151] Florencio Pazos and Alfonso Valencia, *Similarity of phylogenetic trees as indicator of protein-protein interaction*, Protein Eng. **14** (2001), no. 9, 609–614.

[152] Jose B. Pereira-Leal, Anton J. Enright, and Christos A. Ouzounis, *Detection of Functional Modules From Protein Interaction Networks*, PROTEINS: Structure, Function, and Bioinformatics **54** (2004), 49–57.

[153] Sylvain Pitre, Frank Dehne, Albert Chan, and Jim et al Cheetham, *PIPE: a protein-protein interaction prediction engine based on the re-occurring short polypeptide sequences between known interacting protein pairs*, BMC Bioinformatics **7** (2006), 365.

[154] R. L. Plackett, *Karl pearson and the chi-squared test*, International Statistical Review / Revue Internationale de Statistique **51** (1983), no. 1, 59–72.

[155] G.D. Plotkin, *A note on inductive generalization*, Machine Intelligence, vol. 5, Edinburgh University Press, 1970, pp. 153–163.

[156] L. Popelinsky, *Knowledge discovery in spatial data by means of ilp*, Principles of Data Mining and Knowledge Discovery. PKDD'98 Nantes France. (J. M. Zytkow and M. Quafafou, eds.), Lecture Notes in Artificial Intelligence, vol. 1510, Springer Verlag, September 1998, pp. 271–279.

[157] M. A. Pultz and B. S. Baker, *The dual role of hermaphrodite in the Drosophila sex determination regulatory hierarchy*, Development **121** (1995), no. 1, 99–111.

[158] M. A. Pultz, G. S. Carson, and B. S. Baker, *A Genetic Analysis of hermaphrodite, a Pleiotropic Sex Determination Gene in Drosophila melanogaster*, Genetics **136** (1994), no. 1, 195–207.

[159] J.R. Quinlan, *Induction of decision trees*, Machine Learning **1** (1986), no. 1, 81–106.

[160] JR. Quinlan, *Learning logical definitions from relations*, Machine Learning **5** (1990), 239–266.

[161] Arun K. Ramani and Edward M. Marcotte, *Exploiting the Co-evolution of Interacting Proteins to Discover Interaction Specificity*, Journal of Molecular Biology **327** (2003), no. 1, 273–284.

[162] C. S. Raymond, Shamu C. E., Shen M. M., Seifert K. J., Hirsch B., Hodgkin J., and Zarkower D., *Evidence for evolutionary conservation of sex-determining genes*, Nature **391** (1998), no. 6668, 691–695.

[163] M Remm, C. E. V. Storm, and E. L. L. Sonnhammer, *Automatic clustering of orthologs and in-paralogs from pairwise species comparisons*, Journal of Molecular Biology **314** (2005), no. 5, 1041–1052.

[164] Robert Riley, Christopher Lee, Chiara Sabatti, and David Eisenberg, *Inferring protein domain interactions from databases of interacting proteins*, Genome Biol **6** (2005), no. 10, R89.

[165] S.I. Robertson and H. Kahney, *Is analogical problem solving always analogical? the case for imitation*, 1997.

[166] G. D. Rose, A. R. Geselowitz, G. J. Lesser, R. H. Lee, and M. H. Zehfus, *Hydrophobicity of amino acid residues in globular proteins*, Science **229** (1985), no. 4716, 834–838.

[167] Stuart Russell and Peter Norvig, *Artificial intelligence: A modern approach*, Prentice Hall, 2002.

[168] G. Saccone, I. Peluso, D. Artiaco, E. Giordano, D. Bopp, and L. C. Polito, *The Ceratitis capitata homologue of the Drosophila sex-determining gene sex-lethal is structurally conserved, but not sex-specifically regulated*, Development **125** (1998), no. 8, 1495–1500.

[169] H. K. Salz, *The Genetic Analysis of snf: A Drosophila Sex Determination Gene Required for Activation of Sex-lethal in Both the Germline and the Soma*, Genetics **130** (1992), no. 3, 547–554.

[170] M. Schapira, M. Totrov, and R. Abagyan, *Prediction of the binding energy for small molecules, peptides and proteins*, J Mol Recognit. **12** (1999), no. 3, 177–190.

[171] R. Schmidt, M. Hediger, R. Nothiger, and A. Dubendorfer, *The Mutation masculinizer (man) Defines a Sex-Determining Gene With Maternal and Zygotic Functions in Musca domestica L*, Genetics **145** (1997), no. 1, 173–183.

[172] C. SchüLtt, A. Hilfiker, and R. Nothiger, *virilizer regulates Sex-lethal in the germline of Drosophila melanogaster*, Development **125** (1998), no. 8, 1501–1507.

[173] Trudi Schüpbach, *normal female germ cell differentiation requires the female X chromosome to autosome ratio and expression of sex-lethal in Drosophila Melanogaster*, Genetics **109** (1985), no. 3, 529–548.

[174] C. Schütt and R. Nöthiger, *Structure, function and evolution of sex-determining systems in Dipteran insects*, Development **127** (2000), no. 4, 667–677.

[175] B. Schwikowski, P. Uetz, and S. Fields, *A network of protein-protein interactions in yeast*, Nature Biotechnology **18** (2000), 1257–1261.

[176] D. W. Severson, B. deBruyn, D. D. Lovin, S. E. Brown, D. L. Knudson, and I. Morlais, *Comparative Genome Analysis of the Yellow Fever Mosquito Aedes aegypti with Drosophila melanogaster and the Malaria Vector Mosquito Anopheles gambiae*, J Hered **95** (2004), no. 2, 103–113.

[177] Igor V. Sharakhov, Andrew C. Serazin, Olga G. Grushko, Ali Dana, Neil Lobo, Maureen E. Hillenmeyer, Richard Westerman, Jeanne Romero-Severson, Carlo

Costantini, N'Fale Sagnon, Frank H. Collins, and Nora J. Besansky, *Inversions and Gene Order Shuffling in Anopheles gambiae and A. funestus*, Science **298** (2002), no. 5591, 182–185.

[178] K. A. Sharp, A. Nicholls, R. F. Fine, and B. Honig, *Reconciling the magnitude of the microscopic and macroscopic hydrophobic effects*, Science **252** (1991), no. 5002, 106–109.

[179] D. C. A. Shearman and M. Frommer, *The Bactrocera tryoni homologue of the Drosophila melanogaster sex-determination gene doublesex*, Insect Molecular Biology **7** (1998), no. 4, 355–366.

[180] B. A. Sosnowski, J. M. Belote, and M. McKeown, *Sex-specific alternative splicing of RNA from the transformer gene results from sequence-dependent splice site blockage*, Cell **58** (1989), no. 3, 449–459.

[181] Einat Sprinzak, Yael Altuvia, and Hanah Margalit, *Characterization and prediction of protein-protein interactions within and between complexes*, PNAS **103** (2006), no. 40, 14718–14723.

[182] Einat Sprinzak and Hanah Margalit, *Correlated sequence-signatures as markers of protein-protein interaction*, Journal of Molecular Biology **311** (2001), no. 4, 681–692.

[183] Einat Sprinzak, Shmuel Sattath, , and Hanah Margalit, *How Reliable are Experimental Protein-Protein Interaction Data?*, Journal of Molecular Biology **327** (2003), 919–923.

[184] A. Srinivasan and R.D. King, *Feature construction with Inductive Logic Programming: a study of quantitative predictions of biological activity aided by structural attributes*, Data Mining and Knowledge Discovery **3** (1999), no. 1, 37–57.

[185] A. Srinivasan, S.H. Muggleton, M.J.E. Sternberg, and R.D. King, *Theories for mutagenicity: A study in first-order and feature-based induction*, Artificial Intelligence **85** (1996), 277–299.

[186] S. Staab, A. Heller, and M. Steinmann-Zwicky, *Somatic sex-determining signals act on XX germ cells in Drosophila embryos*, Development **122** (1996), no. 12, 4065–4071.

[187] M. Steinmann-Zwicky, *Sex determination in Drosophila: sis-b, a major numerator element of the X:A ratio in the soma, does not contribute to the X:A ratio in the germ line*, Development **117** (1993), no. 2, 763–767.

[188] M.J.E. Sternberg and S.H. Muggleton, *Structure activity relationships (SAR) and pharmacophore discovery using inductive logic programming (ILP)*, QSAR and Combinatorial Science **22** (2003), 527–532.

[189] David L. Tabb, *An algorithm for isoelectric point estimation*, 2001.

[190] Roman L Tatusov, Natalie D Fedorova, John D Jackson, Aviva R Jacobs, Boris Kiryutin, Eugene V Koonin, Dmitri M Krylov, Raja Mazumderand, Sergei L Mekhedov, Anastasia N Nikolskaya, B Sridhar Raoand, Sergei Smirnov, Alexander V Sverdlov, Sona Vasudevan, Yuri I Wolfand, Jodie J Yin, and Darren A Natale, *The COG database: an updated version includes eukaryotes*, BMC Bioinformatics **4** (2003), 41.

[191] W. Traut, *Sex Determination in the Fly Megaselia scalaris, a Model System for Primary Steps of Sex Chromosome Evolution*, Genetics **136** (1994), no. 3, 1097–1104.

[192] Sophia Tsoka and Christos A. Ouzounis, *Prediction of protein interactions: metabolic enzymes are frequently involved in gene fusion*, Nature Genetics **26** (2000), 141–142.

[193] Chandra L. Tucker, Joseph F. Gera, and Peter Uetz, *Towards an understanding of complex protein networks*, Trends in Cell Biology **11** (2001), no. 3, 102–106.

[194] M. Turcotte, S.H. Muggleton, and M.J.E. Sternberg, *Application of inductive logic programming to discover rules governing the three-dimensional topology of protein structure*, ILP98 (D. Page, ed.), LNAI, vol. 1446, SV, 1998, pp. 53–64.

[195] M. Turcotte, SH Muggleton, and MJE Sternberg, *Automated discovery of structural signatures of protein fold and function*, Journal of Molecular Biology **306** (2001), 591–605.

[196] Marcel Turcotte, Stephen Muggleton, and Michael Sternberg, *The effect of relational background knowledge on learning of protein three-dimensional fold signatures*, Machine Learning **43** (2001), no. 1/2, 81–95.

[197] Peter Uetz, Loic Giot, Gerard Cagney, Traci A. Mansfield, Richard S. Judson, James R. Knight, Daniel Lockshon, Vaibhav Narayan, Maithreyan Srinivasan, Pascale Pochart, Alia Qureshi-Emili, Ying Li, Brian Godwin, Diana Conover, Theodore Kalbeisch, Govindan Vijayadamodar, Meijia Yang, Mark Johnston, Stanley Fields, and Jonathan M. Rothberg, *A comprehensive analysis of protein-protein interactions in saccharomyces cerevisiae*, Nature **403** (2000), 632–632.

[198] E. Van Baelen and L. De Raedt, *Analysis and prediction of piano performances using inductive logic programming*, ILP96 (S. H. Muggleton, ed.), Lecture Notes in Artificial Intelligence, vol. 1314, Springer-Verlag, 1996, pp. 55–71.

[199] Manuela M. Veloso, *Prodigy/analogy: Analogical reasoning in general problem solving*, EWCBR, 1993, pp. 33–52.

[200] Manuela M. Veloso and Jaime G. Carbonell, *Derivational analogy in prodigy: Automating case acquisition, storage, and utilization*, Machine Learning **10** (1993), no. 3, 249–278.

[201] C. von Mering, R. Krause, B. Snel, M. Cornell, S.G. Oliver, S Fields, and P. Bork, *Comparative assessment of large-scale data sets of protein-rotein interactions*, Nature **417** (2002), 399–403.

[202] Albertha J. M. Walhout, Raffaella Sordella, Xiaowei Lu, James L. Hartley, Gary F. Temple, Michael A. Brasch, Nicolas Thierry-Mieg, and Marc Vidal, *Protein Interaction Mapping in C. elegans Using Proteins Involved in Vulval Development*, Science **287** (2000), no. 5450, 116–122.

[203] R. Alan Whitehurst, *Systemic software reuse through analogical reasoning*, Ph.D. thesis, University of Illinois at Urbana-Champaign, 1995.

[204] A. S. Wilkins, *Moving up the hierarchy: a hypothesis on the evolution of a genetic sex determination pathway*, Bioessays **17** (1995), no. 1, 71–77.

[205] M.R. Wilkins, E. Gasteiger, A. Bairoch, J.-C. Sanchez, K.L. Williams, R.D. Appel, and D.F. Hochstrasser, *Protein Identification and Analysis Tools in the ExPASy Server*, Methods Mol Biol. **112** (1999), 531–52.

[206] Darren J. Wilkinson, *Bayesian methods in bioinformatics and computational systems biology*, Brief Bioinform (2007), bbm007.

[207] Patrick H. Winston, *Learning and reasoning by analogy*, Communications of the ACM **23** (1980), no. 12, 689–703.

[208] Jérôme Wojcik, Ivo G. Boneca, and Pierre Legrain, *Prediction, assessment and validation of protein interaction maps in bacteria*, Journal of Molecular Biology **323** (2002), 763–770.

[209] Jérôme Wojcik and Vincent Schächter, *Protein-protein interaction map inference using interacting domain profile pairs*, Bioinformatics **17** (2001), S296–S305.

[210] Xiaomei Wu, Lei Zhu, Jie Guo, Da-Yong Zhang, and Kui Lin, *Prediction of yeast protein-protein interaction network: insights from the Gene Ontology and annotations*, Nucl. Acids Res. **34** (2006), no. 7, 2137–2150.

[211] Ioannis Xenarios, Danny W. Rice, Lukasz Salwinski, Marisa K. Baron, Edward M. Marcotte, and David Eisenberg, *DIP: the Database of Interacting Proteins*, Nucl. Acids Res. **28** (2000), no. 1, 289–291.

[212] W. Yi and D. Zarkower, *Similarity of DNA binding and transcriptional regulation by Caenorhabditis elegans MAB-3 and Drosophila melanogaster DSX suggests conservation of sex determining mechanisms*, Development **126** (1999), no. 5, 873–881.

[213] S. Younger-Shepherd, H. Vaessin, E. Bier, L.Y. Jan, and Y. N. Jan, *deadpan, an essential pan-neural gene encoding an HLH protein, acts as a denominator in Drosophila sex determination*, Cell **70** (1992), no. 6, 911–922.

[214] Evgeny M. Zdobnov, Christian von Mering, Ivica Letunic, David Torrents, Mikita Suyama, Richard R. Copley, George K. Christophides, Dana Thomasova, Robert A. Holt, G. Mani Subramanian, Hans-Michael Mueller, George Dimopoulos, John H. Law, Michael A. Wells, Ewan Birney, Rosane Charlab, Aaron L. Halpern, Elena Kokoza, Cheryl L. Kraft, Zhongwu Lai, Suzanna Lewis, Christos Louis, Carolina Barillas-Mury, Deborah Nusskern, Gerald M. Rubin, Steven L. Salzberg, Granger G. Sutton, Pantelis Topalis, Ron Wides, Patrick Wincker, Mark Yandell, Frank H. Collins, Jose Ribeiro, William M.

Gelbart, Fotis C. Kafatos, and Peer Bork, *Comparative Genome and Proteome Analysis of Anopheles gambiae and Drosophila melanogaster*, Science **298** (2002), no. 5591, 149–159.

[215] Weiwei Zhong and Paul W. Sternberg, *Genome-Wide Prediction of C. elegans Genetic Interactions*, Science **311** (2006), no. 5766, 1481–1484.

[216] S. Zhou, Y. Yang, M. J. Scott, A. Pannuti, K. C. Fehr, A. Eisen, E. V. Koonin, D. L. Fouts, R. Wrightsman, and J. E. Manning, *Male-specific lethal 2, a dosage compensation gene of Drosophila, undergoes sex-specific regulation and encodes a protein with a RING finger and a metallothionein-like cysteine cluster*, EMBO J. **14** (1995), no. 12, 2884–2895.

[217] XH Zhou, N Obuchowski, and Obuchowski D, *Statistical Methods in Diagnostic Medicine*, New York: Wiley, 2002.

APPENDIX

Commercial software packages used in this project include:

- Biolayout (www.biolayout.org) to visualise the protein interaction network from protein-interaction pairs;
- NCBI-BLAST (www.ncbi.nlm.nih.gov/blast/) to do sequence comparison;
- DIP (http://dip.doe-mbi.ucla.edu/), InterDom (http://interdom.i2r.a-star.edu.sg/), Inparanoid (http://inparanoid.sbc.su.se/cgi-bin/index.cgi) to extract orthologous protein pairs and domain information;
- PROGOL (http://www.doc.ic.ac.uk/ shm/progol.html) used to infer rules with inductive logic programming;
- PROF (www.aber.ac.uk/ phiwww/prof/) to extract secondary structure information;
- PSIPRED (http://bioinf.cs.ucl.ac.uk/psipred/) to extract secondary structure information.

The other programs, were written by the author himself, including but not limited to all the sequence extraction algorithms, Hidden Markov Models, sequence process programs, all the comparison methods, novel algorithms, performance evaluation methods etc. Programming languages used in this project include: C++, Java, Matlab and Prolog.

The author can be contacted at liqiuxiang@hotmail.com. Computer programs are available on request.

The author's publications during his PhD study.

[1] Identification of sex-specific transcripts of the Anopheles gambiae doublesex gene, Journal of Experimental Biology, 2005 October, 208 : 3701-3709.

[2] Profiling the Antibody Immune Response against Blood Stage Malaria Vaccine Candidates, Clinical Chemistry 53: 1244-1253, 2007.