View metadata, citation and similar papers at core.ac.uk
brought to you by CORE
provided by ScholarBank@NUS

PROTEIN FUNCTION PREDICTION VIA PROTEIN-PROTEIN INTERACTION - a Support Vector Machine approach

LO SIAW LING
(B.C.M., UWA (Western Australia))

A THESIS SUBMITTED FOR THE DEGREE OF MASTER OF SCIENCE
DEPARTMENT OF BIOCHEMISTRY
NATIONAL UNIVERSITY OF SINGAPORE
2004

ACKNOWLEDGEMENTS

I would like to thank *Associate Professor Maxey Chung*, for his constant encouragement and advice during the entire period of my postgraduate studies. In particular, he has guided me to make my research applicable to the real world problem. This work would not have been possible without his understanding and kindness in supporting my part-time research work done in the midst of my official duties.

I am also deeply indebted to *Associate Professor Chen Yu Zong*, for his guidance and patience in helping me to shape up the manuscript for publication, especially the many ideas and directions he gave me during the rough times.

I am grateful to Dr. Cai Congzhong for his teaching and explanation of the various mathematical methods and solutions.

I thank Professor Miranda Yap for initiating and giving me the opportunity to embark on this part-time postgraduate research course.

I would also like to thank Dr Eivind Coward of the Department of Informatics, University of Bergen for sharing the shufflet sequence-randomizing code.

I am also thankful for the support from:

- my colleagues/ex-colleagues – Sandra, Gek San, Cynthia, Jason and Justin;
- my bible-study buddies – Niki, Seok Shin, Alvin, Christine and Beatrice;
- my friends – Tzee Ping, Joy, Huang Sing and Selena

I am forever grateful to my parents for giving me the gift of love and trust which surpasses all else. Mum, for always having my best interest at heart and taking care of my needs; Dad, for giving me the best education he could and his constant support, encouragement and quiet strength.

There is only one person who can fully comprehend what I have been through – my husband and best friend, Ling, who is always there for me and encourages me when I was down.

Lastly, this thesis will not be possible without God. He alone knows the many times I had a mental-block and it is He who renewed and refreshed me time and time again. All glory and honor belongs to Him.

The work is supported by the Lee Hiok Kwee (LHK) Fund from the Department of Biological Sciences, National University of Singapore and a Core Competencies grant from the Agency for Science, Technology and Research (A*STAR), Singapore.

CONTENTS

SUMMARY

Motivation:

Knowledge of protein-protein interaction is useful in elucidating protein function via the concept of 'guilt-by-association'. It is not yet feasible to construct complete protein interaction maps by exhaustive experimental studies. Thus efforts have been directed at development of computational methods for facilitating the prediction of protein-protein interactions. One recently explored method is Support Vector Machines (SVM). A SVM statistical learning system, trained from datasets of real sequences of interacting proteins and artificial shuffled sequences of hypothetical non-interacting proteins, has shown promising capability for prediction of protein-protein interactions (Bock, J.R. and Gough, D.A. Predicting protein-protein interactions from primary structure. 2001. *Bioinformatics*, 17, 455-460). It remains unclear how the prediction accuracy is affected if real protein sequences are used to represent non-interacting proteins.

Method:

In this work, protein function prediction using protein-protein interaction data is assessed by comparison of the results derived from the use of real protein sequences with that derived from the use of shuffled sequences in the non-interactive dataset. Three SVM systems are constructed using three types of negative datasets (i.e., the hypothetical non-interacting proteins) which consists of real protein; 1-let shuffled sequence protein and 2-let shuffled sequence protein respectively together with validated dataset from Database of Interacting Proteins (DIP) as the positive dataset. The real protein sequences of hypothetical non-interacting proteins are generated from an exclusion analysis in combination with subcellular localization information of

interacting proteins found in the DIP. The amino acid sequence of each interacting proteins complex is converted to the feature vectors for SVM training and testing. These vectors are assembled from encoded representation of amino acid residue properties including amino acids composition, hydrophobicity, Van der Waals volume, polarity, polarizability, charge and surface tension.

Results:

Prediction accuracy using shuffled sequences as hypothetical non-interacting proteins yields 94.1%, which is comparable to that obtained by Bock and Gough (2001). In contrast, prediction accuracy using real protein sequences is only 76.9%. The factors that might contribute to the reduced accuracy include limited diversity of dataset and the expected higher level of classification difficulty using two sets of real protein sequences as compared to that of one set of real protein sequences and one set of artificial sequences. The potential of the SVM as a prediction tool for putative protein interacting partners is further evaluated by applying all the three SVM classification systems to the prediction of protein partners of a set of thioredoxin related proteins and *D. melanogaster* high-throughput interaction dataset. The classification system using real protein sequences gives better prediction results that are consistent with observations, indicating that it is more practically useful in facilitating protein-protein interaction prediction than those using artificial/shuffled sequences.

LIST OF TABLES

LIST OF FIGURES

ABBREVIATIONS

ASA	Accessible Surface Area
DIP	Database of Interacting Proteins
KEGG	Kyoto Encyclopedia of Genes and Genomes
MIPS	Munich Information Center for Protein Sequence, Germany
OSH	Optimal Separating Hyperplane
PDB	Protein Data Bank
RMS	Root Mean Square
ROC	Receiver Operator Characteristic
SCOP	Structural Classification of Proteins
SRM	Structural Risk Minimization
SVM	Support Vector Machine
TIGR	The Institute for Genome Research
TRIPLES	TRansposon-Insertion Phenotypes, Localization, and Expression in Saccharomyces

CHAPTER ONE

LITERATURE REVIEW

CHAPTER 1 LITERATURE REVIEW

Protein function prediction is very important in this post-genomics era as there are many hypothetical and/or novel proteins with unknown functions arising from many sequencing projects. Often time, these are the proteins that play interesting roles, be it in cancer or other diseases. Hence understanding protein function and its exact role is vital for human health and for any living organisms to function properly.

Besides the experimental means of investigating the function of proteins, there is increasing need to engage in computational or *in silico* methods to aid in designing and planning the detailed experimental steps in order to decipher the true biological meaning of the protein of interest. The focus of this literature review is on the current publicly available *in silico* protein function and protein-protein interaction prediction methods. The study of protein-protein interaction is chosen as it is probably one of the best ways to understand the function of a novel or un-annotated protein. In addition to inferring the functions from its partner proteins, protein-protein interaction data also offer pathway information which will aid in understanding the overall role of the protein in the larger biological context. However knowing the interacting partners may not be sufficient as we still need to grasp the exact interacting mechanism between the proteins so as to do any rational drug design and analyze the detailed metabolic and signal transduction network. As the result, it is necessary to understand the characteristics of the protein interaction sites in order to develop a better protein-protein interaction method.

1.1 Protein function prediction

The function of proteins include building, supporting, recognizing, transporting and transforming cellular activities with incredible speed and accuracy and in many cases

are subject to multiple regulatory mechanisms. The development of high-throughput methods and their applications have generated a large amount of data that are useful for the study of protein functions. Several attempts have been made to predict protein functions using data from sequence homology, protein-protein interactions, protein structural information and gene expression.

A common method used in protein function prediction is sequence homology. This 'homology method' is used widely to extend knowledge of protein function from one protein to other proteins on the basis of sequence similarity, which are presumably descended from the same common ancestral protein. One of the publicly available methods is the suite of BLAST programs[1] which are used to extend experimental knowledge of protein function to new sequences. By using such homology methods, roughly 40–70% of new genome sequences can be assigned to some functions[2,3]. The functional assignments by homology usually involve identification of some molecular functions of the protein, but they do not place the protein in its context of cellular function. For example, thioredoxin, which contains an active reduction and oxidation (redox) disulfide/dithiol domain, is an endogenous multifunctional protein with numerous cellular functions including defense against oxidative stress, control of growth and apoptosis and chemokine activities[4]. Besides that if a protein has multiple domains, this approach may not be able to provide a good prediction of the cellular roles. Hence it is important to determine the function of a protein in cellular context to fully understand its role.

The specialized functions of proteins can be understood in terms of how proteins bind to and interact with other components of living systems, e.g., small molecules,

proteins and much larger entities such as nucleic acids. As such, approaches based on derived protein-protein interaction databases such as Yeast Protein Database (YPD)[5] and Database of Interacting Protein (DIP)[6] are gaining popularity. Besides that, there are also integrated interaction map methods which attempt to address the completeness of protein-protein interaction relationships in order to deduce a particular function of a protein in the map[7].

In order to fully understand the functional properties of a protein, it is necessary to deduce or predict the three-dimensional (3D) protein structure from amino acids to identify the domain that either serve as module(s) for building up large assemblies or provide specific catalytic or binding sites. However, to date, the protein folding problem still remains unsolved. Since the 3D structures of individual proteins cannot be predicted computationally, they must, instead, be determined experimentally by x-ray crystallography, cryo-electron microscopy or nuclear magnetic resonance (NMR) techniques. These experimentally obtained results can be used in developing protein structure prediction methods. One of the protein structure prediction methods, homology modeling, can be adopted to deduce an unknown protein function if the protein has sequence identity of more than 30% [8] with that of a known protein structure.

Clustering analysis of gene expression data can also be used to predict functions of un-annotated proteins based on the idea that genes with similar functions are likely to be co-expressed [9,10].

1.2 Protein-protein interaction prediction

Currently there are many approaches to determine protein-protein interaction experimentally but most of the methods are complementary to each other and none of the methods stand out as a definitive way to identify protein interaction partner(s)[11]. Hence it will be helpful if experimental data can be collated and analyzed computationally to discover underlying rules or relationships of protein interaction. The recent protein-protein interaction data generated from many high-throughput methods[12,13,14] is useful to derive computational methods to predict the possible function of an unknown protein and also to understand the mechanism of interaction, for example, the interaction site of the interacting partners. This set of data is far from being complete and should only be used as a method for development and possible prediction of function. Nevertheless, there is no doubt that with more information, computational prediction will be able to achieve reasonably accurate results which can complement experimental determination of protein-protein interaction and the protein-protein interaction site. Computational methods of predicting possible interacting partners and interacting siyttes are listed in Table 1.

Table 1 Computational methods in predicting protein-protein interaction and interacting site.

	Methods used in prediction of protein-protein interaction	**Databases or training data used**	**Remarks**	**Ref**
Protein-protein interaction prediction – protein sequence based				
1.	A method based on the assumption that proteins that function together in a pathway or structural complex are likely to evolve in a correlated fashion, i.e., phylogenetic profiles of the presence and absence of genes in related species	Computed phylogenetic profiles for the 4,290 proteins encoded by the genome of *Escherichia coli* by aligning each protein sequence with the proteins from 16 other fully sequenced genomes (listed at the web site of The Institute for Genome Research: www.tigr.org)	This method finds pairs of functionally linked proteins by their phylogenetic profiles which cannot be linked by conventional sequence-alignment techniques	15
2.	Observation that gene order in different species is conserved and the proteins encoded by conserved gene pairs appear to interact physically. This comparison among species is used for interaction prediction	Nine bacterial and archaeal genomes	Work is done on prokaryotic gene products. It may not be applicable to eukaryotes as their genome structures are	16

	Methods used in prediction of protein-protein interaction	Databases or training data used	Remarks	Ref
			more complicated.	
3.	Based on the observation that some pairs of interacting proteins have homologues in another organism fused into a single protein chain. Two methods used in confirming the protein pairs are functionally related: 1. Using annotation given in the SWISS-PROT : 3950 *E. coli* pairs of known function, 68% share at least one keyword in their annotations. For yeast, 32% correctly predicted from 9857 pairs of known function 2. Using phylogenetic profiles which detect functional interactions by analyzing correlated evolution of proteins. 5% predicted correctly from 6809 *E. coli* pairs	1. 6809 protein-protein interactions in *Escherichia coli* from 4290 protein sequence in the genome 2. 45,502 protein pairs in yeast from yeast genome (total number of proteins used was not specified)	The domain fusion analysis cannot distinguish between homologs that bind and those that do not, i.e., the inability to distinguish homologs. Hence cannot handle the 'promiscuous' domains such as SH3	17
4.	Protein interaction maps based on gene fusion events,	215 genes or proteins in the complete	Only applicable to completely	18

	Methods used in prediction of protein-protein interaction	**Databases or training data used**	**Remarks**	**Ref**
	solely using sequence alignment in comparison. Each of the three complete genomes of *Escherichia coli*, *Haemophilus influenzae* and *Methanococcus jannaschii* are used in turn as the query genome Q; Q is compared with the other two genomes, plus the genome of the yeast *Saccharomyces cerevisiae*, as reference genome to detect `orthologous' proteins across species	genomes of *Escherichia coli*, *Haemophilus influenzae* and *Methanococcus jannaschii* – 64 unique fusion events	sequenced genome and proteins that occurs in gene fusion event. However it has potential in identifying interacting proteins regardless of distance within the gene fusion event	
5.	Method based on conserved gene neighbouring idea, i.e., there is a correlation between the spatial proximity of genes on the genome and the directness of the interaction between the encoded protein	*Mycoplasma genitalium* database	This method may not be applicable to eukaryotes because their co-regulation of genes is not imposed at genome structure level	19
6.	The study is based on correlated mutations in multiple	multiple sequence alignments were	The limitation of this method is	20

	Methods used in prediction of protein-protein interaction	**Databases or training data used**	**Remarks**	**Ref**
	sequence alignments. The method, in silico two-hybrid, i2h, directly addresses the detection of physically interacting protein pairs and identifies the most likely sequence regions involved in the interactions	obtained by searching for homologous proteins with BLAST and aligning them with Clustalw in the *Escherichia coli* test sets, or taken from the HSSP database in the cases of the structural domains and the interacting proteins of known structure	in obtaining large multiple sequence alignments of corresponding sequences for each possible pair of proteins due to the inadequate amount of currently known interacting protein pairs	
7.	Support Vector Machine (SVM) learning system using feature vectors of residue properties including charge, hydrophobicity and surface tension and shuffled sequences for non-interacting protein pairs	Database of Interacting Proteins (DIP; http://www.dip.doe-mbi.ucla.edu/)	The use of hypothetical shuffled proteins as negative dataset may not be applicable to the real world situation	6
8.	SVM feature vector for each protein is constructed by concatenation of functional domains with amino acid	*Saccharomyces cerevisiae* interaction data from DIP and MIPS as positive data.	This method may not be able to handle proteins with multiple	21

	Methods used in prediction of protein-protein interaction	**Databases or training data used**	**Remarks**	**Ref**
	composition, sequential amino acid usage, hydrophobicity, surface tension, and the localization. It is found that feature vector containing domain and localization has the highest accuracy – 77.63%	All possible pair of proteins that are not recognized as positive in the above databases as negative data. Only consider positive/negative pairs where both proteins have functional domains	domains and 'promiscuous' domains such as SH3	
9.	Using sequence-signatures that appear together in interacting protein pairs more often than expected at random (from InterPro), to predict putative pairs of interacting partners in the cell	• Munich Information Center for Protein Sequence, Germany (MIPS) – MYGD (yeast database) • Data based on large-scale 2-hybrid analysis in *Saccharomyces cerevisiae* • DIP	Sequence-signature generation is limited by the classification from InterPro. There is a need to search for new common motifs among the interacting proteins or use structural domains	22
10.	A multimeric threading algorithm based on protein	• Yeast proteome from KEGG database	The main limitation is due to	23

	Methods used in prediction of protein-protein interaction	Databases or training data used	Remarks	Ref
	structure but without using the query protein structure - Each possible pairwise interaction among more than 6000 encoded proteins is evaluated against a dimer database of 768 complex structures by using a confidence estimate of the fold assignment and the magnitude of the statistical interfacial potentials	(http://www.genome.ad.jp/kegg/) • Subcellular localizations of yeast proteins from the MIPS (http://mips.gsf.de/proj/yeast/CYGD/db/index.html), the TRIPLES database (http://ygac.med.yale.edu/triples/), and Mark Gerstein's Lab Web site (http://bioinfo.mbb.yale.edu)	the small amount of protein structures solved and the difficulties in improving the accuracy of threading. However this method has an advantage of identifying the interaction site	
11.	Used a reference interaction map to predict interaction map in another organism. Sequence similarity searches with clustering based on interaction patterns and interaction domain	Reference map – *Escherichia coli* Predict – human gastric pathogen (*Helicobacter pylori*)	This method relies heavily on the completeness, accuracy and level of detail (definition of protein domains) of the	24

	Methods used in prediction of protein-protein interaction	Databases or training data used	Remarks	Ref
			reference dataset	
Protein-protein interaction prediction – protein structure based				
12.	Protein-protein docking refinement techniques based on modified molecular mechanics force fields and empirical measures of desolvation, combined with minimisations that switch on the short-range interactions gradually	PDB database	This approach still needs to address issues like interaction sites prediction, promiscuity of interactions, the occurrence of weak interactions and the role of water in interfaces	25
13.	Structure based prediction in studying interactions between protein domains in terms of the interactions between structural families	• PDB database • SCOP • Yeast genome	Homology modeling is used to address the problem of limited known structure. However this approach may not be able to handle situation where	26

	Methods used in prediction of protein-protein interaction	Databases or training data used	Remarks	Ref
			members of two families interact in different ways, using different interfaces	
14.	Created position-specific scoring matrices or virtual interaction profiles (VIPs) for a protein. This was generated via sequence prediction algorithm and a novel ensemble averaging calculation to find peptide sequences that have significant affinity for a protein	PDB database	This method performs well in selecting the partners where the structure is already solved but improvement still needed for predicting biological partners	27
Protein-protein interaction site prediction				
15.	Protein–protein interaction sites are predicted from neural network with sequence profiles of neighboring residues and solvent exposure as input	The network was trained on 615 pairs of nonhomologous complex-forming proteins from PDB.	This approach is insensitive to structural changes accompanying complex formation. However only	28

	Methods used in prediction of protein-protein interaction	Databases or training data used	Remarks	Ref
			interfaces with extensive interfacial contacts (at least 20 residues from each side) and dimer interfaces are considered in this study.	
16.	It is observed that correlated sequence changes can be used to predict protein-protein contact regions. The assumption is that the sequence changes accumulated during the evolution of one of the interacting proteins must be compensated by changes in the other	1. A and B haemoglobin to test inter-protein and inter-domain contacts 2. Heat-shock protein Hsp70 to predict contacting residues	Given two examples show promising results but more test cases are needed	29
17.	Predictive method to identify protein-protein interaction sites based on the observation that proline is the most common residue in the flanking segments of interaction	Fibrin fibre Prediction of a fibrin polymerization site	This method only tested on proteins where the short segments (3-7 residues) is	30

	Methods used in prediction of protein-protein interaction	Databases or training data used	Remarks	Ref
	sites		flanked by proline residues.	
18.	Developed a simple physical model to measure free energy changes brought about by alanine mutation at protein-protein interfaces and predicted the energy hot spot as interacting site by alanine scanning	Datasets for single mutations from ProTherm database (http://www.rtc.riken.go.jp/jouhou/protherm/protherm.html) Mutational data for protein complexes from the Alanine Scanning Energetics database (http://mullinslab.ucsf.edu/~kurt/hotspot/index.php)	The preformance of this simple model is affected by the effect of water molecules in the interface and the side-chain conformational changes	31

1.3 Characteristic of protein-protein interaction interfaces

Knowing that two proteins are interacting is not sufficient to decipher the interaction mechanism and to predict the interaction sites on the protein surface which has direct relevance to the design of drugs for blocking or modifying the interactions. Hence, understanding the characteristic of the interacting interface is of much interest. Various studies have been done[32,33] and they have helped in differentiating many types of interaction but thus far, we are still not clear of the exact rules governing the interaction of proteins. Having said this, it is likely that no single rule can be derived but it is of interest to investigate all possible underlying rules or patterns that exist in interaction between protein families or certain domain-domain interaction in order to better understand protein-protein interaction.

Currently there are three approaches in characterizing the protein-protein interaction sites. The first is based on surface or interface study on protein complexes; the second is through the study of protein sequences with known structures and the third is based on residue frequencies and pairing preferences at protein-protein interfaces. The three approaches are examplied by three studies; details of each are listed below:

- The first study involves surface patch analysis of protein using six parameters – solvation potential, residue interface propensity, hydrophobicity, planarity, protrusion and accessible surface area on three types of complexes - homo-dimer; hetero-complexes and antibody-antigen complexes[32]. Even though in general the parameters reveal the following observations on the interacting site, none of the parameters is definitive in distinguishing each type of complex.

  1. RMS of least-squares plane – more planar;

2. Residue propensities – higher propensity;
3. Protrusion – higher protrusion;
4. Hydrophobicity – more hydrophobic;
5. Accessible surface area (ASA) – higher ASA

This study also uncovers the different trends for different type of complexes:

1. Hydrophobicity : homodimer interface is the most hydrophobic while antigen has most polar interface. No trend for hetero-complexes.
2. Solvation potential : lowest for homodimer and highest for antigens. No trend for hetero-complexes
3. Protrusion : most interfaces are protruding except for big hetero-complexes which are the least protruding patches; enzyme interfaces tend to be those surface patches that are folded into clefts, so the residues involved will be among the least protruding

- The second study is a review on interfaces difference between complexes composed of two components, namely, homodimeric proteins, heterodimeric proteins, enzyme-inhibitor complexes and antibody-protein complexes[33]. The summary of the findings are listed below and this study highlights that there is a need to take into account the type of protein-protein complexes when characterizing the interfaces within them.

  1. Size and shape
     a. Heterocomplexes have interfaces that are more planar than the homodimer

   b. Heterodimer that occurs only as heterodimer are less planar compared to the nonpermanent counterparts (can occur as both heterocomplexes and monomers)
   c. Homodimer – general trend is to favor protrusion
2. Complementarity between surfaces
   a. Homodimers, the enzyme-inhibitor complexes and the permanent heterocomplexes are the most complementary
   b. Antibody-antigen complexes and the nonobligatory heterocomplexes are the least complementary
3. Residue interface propensities
   a. Hydrophobic residues show a greater preference for the interfaces of homodimers than heterocomplexes
   b. Lower propensities for hydrophobic residues in the heterocomplex interfaces is balanced by an increased propensity for the polar residues
4. Hydrophobicity including hydrogen bonding
   a. Homodimers' hydrophobic surfaces are permanently buried within a protein-protein complex
   b. Heterocomplexes that occur as both monomers and complexes have relatively more intermolecular hydrogen bonds per ASA.
5. Segmentation and Secondary structure
   a. Interfaces are highly segmented except for enzyme-inhibitor complexes
   b. Most interfaces have mixed secondary structure

6. Conformational Changes on complex formation
    a. Enzyme complexes – domain movement
    b. Antibody-protein recognition – wide range of variation

- The third study is on residue frequencies and pairing preferences at protein-protein interfaces[28]. This study unveils that hydrophobic residues (Trp and Leu) are abundant in large interfaces while polar residues (Gly and Ala) are more abundant in small interfaces. The exception was Arg (charged aa), which is more common in large than small contact surfaces.

The surface study on protein-protein interaction sites seem to agree that complexes that can exist as independent entities, have interfaces that are less hydrophobic while complexes that are bound permanently, or residues that are at the binding sites, are more closely packed and have fewer inter-subunit hydrogen bonds.

In addition, the second approach, a detailed analysis on amino acids sequence of known protein structure, yields the following findings:

- It is observed that proline is the most common residue found in the flanking segments of interaction sites. As such the interaction sites of proteins might be predicted directly from the amino acid sequence based on the presence of proline brackets[30].
- The analysis of hydrophobicity distribution on linear stretches of amino acid sequences can help to identify "receptor-binding domains" with arginine being the most frequently occurring residue[34].

Hence it is important to take into consideration the amino acid composition; hydrophobicity, charge, polarity distribution of the individual protein in protein-protein interaction study.

Even though none of the above research is able to have a conclusive finding to define a method to identify protein interaction sites of every type of complexes, especially when most complexes studied are two components while the important biological functions involve huge multi-component complexes (e.g. ribosome), it has certainly shed some lights in better understanding the underlying principle of protein-protein interaction.

CHAPTER TWO

INTRODUCTION

CHAPTER 2 INTRODUCTION

2.1 Objectives of the study

Protein-protein interactions play important roles in various biological events[35] and are the bases for assemblies of molecular machines, such as RNA polymerase II. The 'guilt-by-association'[36] concept has been used for elucidating functional roles from pairs of interacting proteins. Identification of its partner of known function may provide a useful clue to the possible role(s) for a protein of unknown function.

Even though knowledge of protein-protein interactions is useful for probing biological pathways and regulation of signaling, metabolic, gene expression and replication processes, it is not yet feasible to construct complete protein interaction maps by exhaustive experimental studies. Current experimental methods include yeast two-hybrid systems[37], protein complex purification techniques using mass spectrometry[12,13], protein chip[38], correlated messenger RNA expression profiles[39] and genetic interaction data[40]. However each method has its own strength and weakness or biases in identifying certain group of proteins. For example, yeast two-hybrid technology detects more proteins that are involved in translation than by other methods. Recently, a large-scale comparative assessment of protein-protein interactions data[11] has shown that the highest accuracy is achieved for interactions supported by more than one methods, including *in silico* prediction methods.

As such, there is a growing interest in the exploration of computational methods for the prediction of protein-protein interactions. This method of complementing experimental with computational approach have certainly achieved useful insight in

understanding and predicting protein-protein interaction[41,42]. So far, two different computational approaches have been explored for the prediction of protein-protein interactions. The first is based on pure sequence-based methods such as, co-occurrence of genes[15], conservation of gene order in different species[16], protein fusion[17,18], conserved gene neighboring[19], *in silico* two-hybrid system[20], machine learning using sequence residues' associated physicochemical properties[6], correlated sequence-signatures that recur in concert in various pairs of interacting proteins[22], threading[23] and reference interaction map[24]. The second is concerned with the study of protein structure, these includes docking[25], interaction protein domain between structural families[26] and virtual interaction profiles[27]. The details of each method have been summerised in section 1.2.

Because of the limited availability of protein 3D structures, methods that derive information directly from protein primary structure are of particular interest. A statistical learning method, support vector machines (SVM), has recently been explored for the prediction of protein-protein interactions[6,21] as well as protein structural class prediction[43], protein secondary structure prediction[44], protein fold recognition[45], analysis of protein solvent accessibility[46] and other biological research, including microarray gene expression data analysis[10] and cancer diagnosis[47]. These studies have consistently shown that SVM is usually superior to traditional supervised learning methods. For example, when predicting protein-protein interaction using domain information, the accuracy using SVM is about 25% higher than that of previous result by Deng *et al*.[48] The difference is mainly caused by the difference in the methods used for prediction. Deng *et al.* used Maximum Likelihood Estimation, while Dohkan *et al*.[21] used SVM, which is known to show better performance in two-

class classification problems. In addition, it is easier for SVM training to find globally optimized solution because of the fewer parameters it uses, and it has the potential to deal with a large number of feature vectors. On the other hand, it is more difficult to use Neural Networks (NN), which is in theory a more sophisticated method than SVM, to find a global solution because of the higher number of parameters it uses. Various examples also highlight that SVM performs with comparable accuracy, if not better than NN [49,50].

SVM is a relatively new and very promising type of supervised learning algorithm for two-class or multi-class classification, which was originally developed by Vapnik and his collaborators at the AT & T laboratory[51,52]. Firmly grounded in the framework of statistical learning theory, the SV algorithms generalize well even to unseen data. One attractive property of SVM is that SVM is capable of extracting essential information from a very large number of training samples to provide a condensed representation of these samples by using a relatively small number of support vectors (SVs). Besides that it also have the ability to handle large feature spaces (by using kernel mapping) and can have excellent generalization performance (by maximizing minimum margin). In addition to the work on proteins and biological research mentioned previously, SVM have also been successfully employed in a wide range of real-world problems such as text categorization[53], hand-written digit recognition[54], tone recognition[55], image classification and object detection[56].

Like other statistical learning methods, the accuracy of SVM classification depends on the relevance of training dataset to a particular biological problem. Thus it is important to use a reliable training dataset to achieve a better classification. Since

experimental conditions and, in some cases, types of proteins are known to affect the accuracy of some of the experimental methods[11], caution needs to be exercised in the interpretation of experimental data. Hence, to ensure their high quality, the dataset of interacting proteins (positive dataset) used in this work is from a subset of the data in the Database of Interacting Proteins (DIP)[57] whose reliability has been assessed[58]. Since non-interacting proteins are not readily available, artificially shuffled sequences resembling realistic proteins have been used to construct the dataset of hypothetical non-interacting proteins (negative dataset) for the prediction of protein-protein interactions. Bock and Gough (2001) has shown that it gives an average accuracy of 80.9%[6]. However, shuffling sequences artificially may result in sequences with no specific sequence patterns like motifs or domains while real protein sequences are known to contain these conserved sequences patterns that play important functional roles. It is unclear whether a classification system derived from artificial sequences is sufficiently effective in prediction of protein-protein interactions since artificial shuffled sequences, having the possibility of not containing any motifs or domains, are unlikely to be functional proteins. It is thus desirable to use only real protein sequences for developing a SVM classification system which might be more relevant to the prediction of protein-protein interactions.

It is of interest to evaluate how the prediction accuracy can be affected by using realistic sequences as negative dataset. For such a purpose, real protein sequences are used to construct a negative dataset. This negative dataset of hypothetical non-interacting proteins is derived from an exclusion analysis in combination with subcellular localization information of interacting proteins in DIP. The prediction accuracy of a SVM system trained from this dataset is compared with those from

shuffled sequences generated from the same principle as described in the literature[6]. The prediction performance of both systems is further evaluated by using them for the identification of putative interacting partners of a set of thioredoxin related proteins and the high-throughput *D. melanogaster* interaction dataset[59].

CHAPTER THREE

MATERIALS AND METHODS

CHAPTER 3 MATERIALS AND METHODS

3.1 Data collection and dataset construction

The positive dataset is from *Saccharomyces cerevisiae* core subset of DIP database[58]. This dataset is validated by two methods described by Deane and colleagues[58]. The first is to use the expression profile reliability (EPR) index to estimate the biologically relevant fraction of protein interactions by comparing the RNA expression profiles of the proteins with expression profiles of known interacting and non-interacting pairs of proteins. The second is to use the paralogous verification method (PVM) to test the reliability of a putative interaction pair by examining whether there is a known paralog that also interacts with its partner protein.

Since a non-interacting proteins dataset is not readily available, a hypothetical non-interacting proteins dataset is generated based on subcellular localization information and it consists of protein pairs that do not co-localize together. The subcellular localization source is retrieved from MIPS[60] and only the four main types of localization are considered in this study – cytoplasm, nucleus, mitochondria and endoplasmic reticulum. The yeast proteins used in the positive dataset are assigned with the four types of localization information and those with multiple localizations are removed to minimize the introduction of possible noise in the training process. Four sets of proteins with respect to the four types of localization are generated and proteins from each set are subsequently paired with proteins from a different localization. Due to the enormous number of possible pairings, 4,810 protein pairs have been randomly selected and used in this work. After removing duplication and performing exclusion analysis of the whole DIP yeast interacting proteins, a total of 4,662 protein pairs are used as the hypothetical non-interacting dataset.

As a comparison, a second type of negative dataset composed of artificial protein sequences of the hypothetical non-interacting dataset are derived by using the Shufflet program[61] with *k*-let (k = [1,2]) counts. The *k*-let (the exact words equal to or shorter than a given length *k*) are kept conserved in generating random shuffled sequences. In addition to preserving the amino acid composition which correlates with protein-protein interfaces[62], such a shuffling[61,63] also maintains the frequencies of di-peptides, tri-peptides etc. The algorithm ensures that every expected occurrence of each possible *k*-let has the same probability, which is expected to generate datasets with conserved properties that are closer to real protein sequences than pure randomly generated sequences. In general, a *k*-let works well for sequences up to 20^k amino acids in length. Hence in order to maintain the random uniform permutation, only 1-let and 2-let shuffled protein sequences are considered in this study.

Each dataset is further divided in a random fashion into a training set and a testing set while maintaining representatives of distinct protein pairs in each set whenever possible. For example, if the positive dataset has four interacting protein pairs of 'protein D', then each of the two pairs will be randomly distributed to positive training and testing set respectively. The training dataset is evaluated to remove homologous sequences using BLASTCLUST[1,64] with identity threshold of 30% and length coverage threshold of 90% to ensure the classifier is not biased to homologous sequences. This gives a positive training set of 2,080 interacting proteins, a negative training set of 2,331 non-interacting proteins, a positive testing set of 2,208 interacting proteins and a negative testing set of 2,331 non-interacting proteins.

As such, three SVM systems are constructed using the same positive training dataset together with three types of negative datasets (i.e., the hypothetical non-interacting proteins) which consists of real protein; 1-let shuffled sequence protein and 2-let shuffled sequence protein respectively. Similarly for the testing dataset, the same positive testing dataset is used together with three types of negative datasets to form three testing datasets. The diagrammatic dataset construction steps is shown in Figure 1.

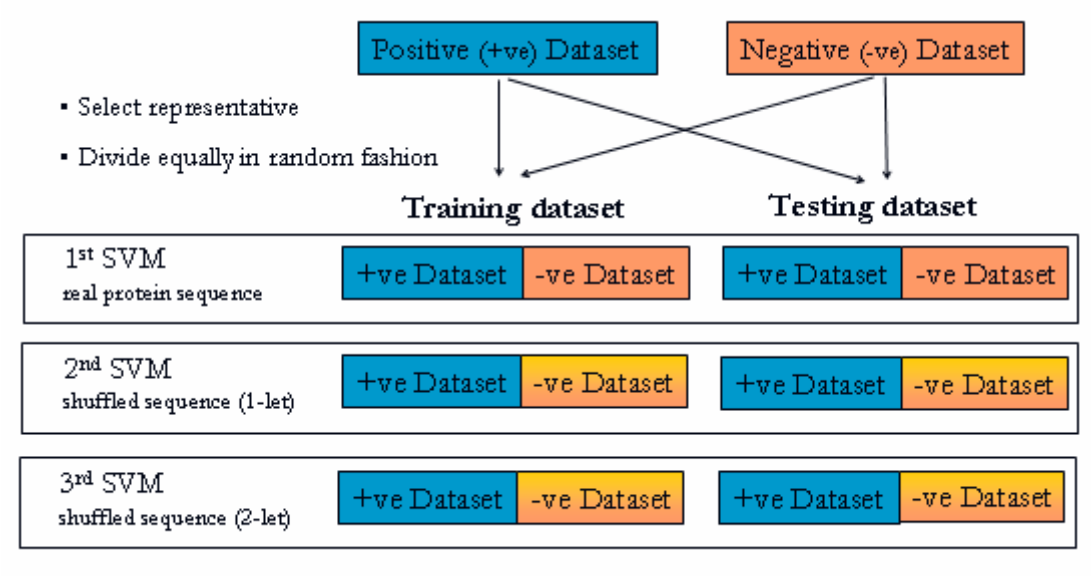


Figure 1 Diagrammatic dataset construction in the three SVM used for protein-protein interaction assessment

3.2 Features extraction and representation

The feature vector extraction is a key technique to a successful classification. For each protein sequence, feature vectors are assembled from encoded representations of tabulated residue properties including amino acids composition, hydrophobicity[65], Van der Waals volume[66], polarity[67], polarizability[68], charge[69] and surface tension[70,71] for each residue in sequence. Each protein sequence is converted to feature vector using amino acids composition percentage and the feature extraction method[45] based on three descriptors. The first is 'Composition' (*C*), percent composition of three

constituents (e.g. polar, neutral and hydrophobic residues in hydrophobicity). The second is 'Transition' (*T*), which describes the transition frequencies (polar to neutral, neutral to hydrophobic, etc.). The third is 'Distribution' (*D*), which represents the distribution pattern of constituents (where the first residue of a given constituent is located, and where 25, 50, 75 and 100% of that constituent are contained).

As an example to illustrate the feature representation of a protein, a hypothetical protein sequence AKAAAKAKKAAAAAKAKKKAAKKAKKKAAK is adopted for the purpose. As shown in Figure 2, the protein has 16 Alanines [A] ($n1$=16) and 14 Lysines [K] ($n2$=14). The composition for these two amino acids are $n1$*100.00/($n1$ + $n2$)=53.33 and $n2$*100.00/($n1$ + $n2$)=46.67 respectively. There are 15 transitions from A to K or from K to A in this sequence and the percent frequency of these transitions is (15/29)*100.00=51.72. The first, 25, 50, 75 and 100% of As are located within the first 1, 5, 12, 20 and 29 residues, respectively. The *D* descriptor for As is thus (1/30)*100.00=3.33, (5/30)*100.00=16.67, (12/30)*100.00=40.0, (20/30)*100.00= 66.67, (29/30)*100.00=96.67. Likewise, the *D* descriptor for Ks is 6.67, 26.67, 60.0, 76.67, 100.0. Overall, the amino acid composition descriptors for this sequence are *C*=(53.33, 46.67), *T*=(51.72) and *D*=(3.33, 16.67, 40.0, 66.67, 96.67, 6.67, 26.67, 60.0, 76.67, 100.0), respectively.

```
Sequence        A K A A A K A K K A A A A A K A K K K A A K K A K K K A A K
Index           1       5         10        15        20        25        30
Index for A     1   2 3 4   5     6 7 8 9 10  11      1213    14      1516
Index for K       1       2   3 4           5   6 7 8     9 10  111213    14
A/K transitions * *     * * *   *         * * *     *   *   * *     *   *
```

Figure 2 Hypothetical sequence for illustration of derivation of the feature vector of a protein

Descriptors for other properties can be computed by a similar procedure and all the descriptors are combined to form the feature vector. In most studies, amino acids are divided into three groups for each feature and thus the three descriptors for each feature consist of twenty-one elements: three for *C*, three for *T* and fifteen for *D*[44,45,46]. The similar approach is adopted in this study and the three groups of each feature, except for amino acid composition (which is the percentage composition of each amino acid), are listed in Table 2. As such, the total dimension of each vector representing a protein is 146 (Table 3). The formulas for calculating the twenty-one elements for each feature are shown in Table 4. Thus the feature vector of an interacting protein pairs is a concatenation of the feature vector of the two interacting proteins which is 292 in total dimension.

Table 2 Representative amino acids in three classes of each feature

Feature	**Group1 (G1)**	**Group2 (G2)**	**Group3 (G3)**
Hydrophobicity	Polar R, K, E, D, Q, N	Neutral G, A, S, T, P, H, Y	Hydrophobic C, V, L, I, M, F, W
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
Charge	Positive K	Neutral R, A, N, C, Q, G, H, I, L, M, F, P, S, T, W, Y, V	Negative D, E
Surface Tension	G, Q, D, N, A, H, R	K, T, S, E, C	I, L, M, F, P, W, Y, V

Table 3 Dimension of feature vector representing a protein

Feature No	**Feature**	**Dimension**
1	Amino acids composition	20
2	Hydrophobicity	21
3	Van der Waals volume	21

Feature No	Feature	Dimension
4	Polarity	21
5	Polarizability	21
6	Charge	21
7	Surface Tension	21
Total		*146*

Table 4 Formulas for calculating descriptors for a feature

The following table contains the formula for calculating the feature vector of a protein for features 2 to 7 in Table 3 while feature 1 – amino acids composition, is the percentage composition of each amino acid in the protein. G1, G2 and G3 can be referenced from Table 2. Dim stands for Dimension from Table 3.

Dim	Descriptor	Calculation	Remarks
1	1st composition	G1/(G1+G2+G3) * 100%	
2	2nd composition	G2/(G1+G2+G3) * 100%	
3	3rd composition	G3/(G1+G2+G3) * 100%	
4	1st transition	F1/(G1+G2+G3-1) * 100%	F1=Frequency[(G1 to G2) or (G2 to G1)]
5	2nd transition	F2/(G1+G2+G3-1) * 100%	F2=Frequency[(G1 to G3) or (G3 to G1)]
6	3rd transition	F3/(G1+G2+G3-1) * 100%	F3=Frequency[(G2 to G3) or (G3 to G2)]
7	1st distribution – 1%	P1,1/(G1+G2+G3) *100%	Dy = total distribution number at y percentage, e.g. if Gx = 10, y = 50% distribution then Dy = 5 Pxy = position where the total distribution number of amino acid representative of the Gx (Dy) is found in the whole protein If the first residue of Gx coincides with the beginning of the chain, the first number of the D descriptor equals 0.0 Else the above rule follows
8	1st distribution – 25%	P1,25/(G1+G2+G3) *100%	
9	1st distribution – 50%	P1,50/(G1+G2+G3) *100%	
10	1st distribution – 75%	P1,75/(G1+G2+G3) *100%	
11	1st distribution – 100%	P1,100/(G1+G2+G3) *100%	
12	2nd distribution – 1%	P2,1/(G1+G2+G3) *100%	
13	2nd distribution – 25%	P2,25/(G1+G2+G3) *100%	
14	2nd distribution – 50%	P2,50/(G1+G2+G3) *100%	
15	2nd distribution – 75%	P2,75/(G1+G2+G3) *100%	
16	2nd distribution – 100%	P2,100/(G1+G2+G3) *100%	

Dim	Descriptor	Calculation	Remarks
17	3rd distribution – 1%	P3,1/(G1+G2+G3) *100%	
18	3rd distribution – 25%	P3,25/(G1+G2+G3) *100%	
19	3rd distribution – 50%	P3,50/(G1+G2+G3) *100%	
20	3rd distribution – 75%	P3,75/(G1+G2+G3) *100%	
21	3rd distribution – 100%	P3,100/(G1+G2+G3) *100%	

3.3 Support Vector Machine

In this work, we employed SVM^{Light} (http://svmlight.joachims.org)[72] for the classification. SVM is based on the structural risk minimization (SRM) principle from statistical learning theory[52]. In linearly separable cases, SVMs separate two different groups of feature vectors (a given known set of {+1, -1} labeled training data) via a hyperplane that is maximally distant from the positive and negative samples (known as Optimal Separating Hyperplane, OSH), then 'plot' the test data at the high dimensional space, distinguishing whether it belongs to positive or negative according to the OSH (Figure 3).

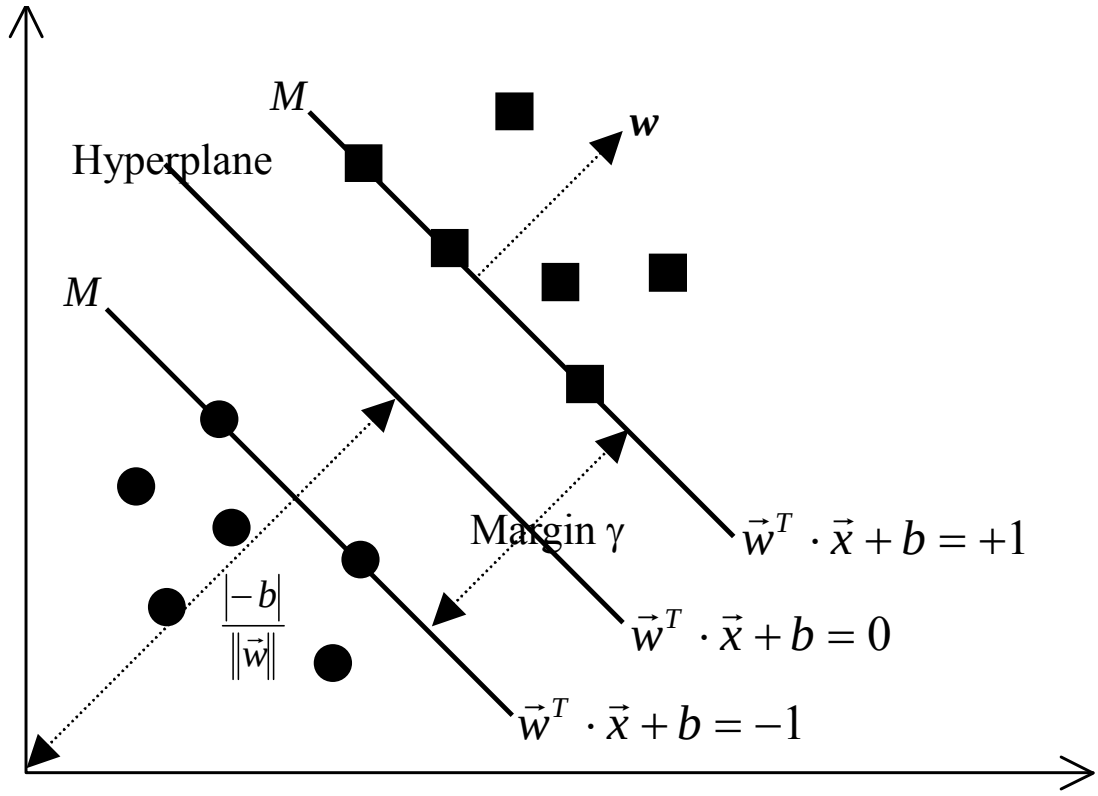


Figure 3 The definition of Hyperplane and Margin

The circular dots and square dots represent samples of class -1 and class +1, respectively.

A feature vector is represented by $\mathbf{x}_i$ with physicochemical descriptors of a protein as its components. The hyperplane is constructed by finding another vector **w** and a parameter *b* that minimizes $||\mathbf{w}||^2$ and satisfies the following conditions:

$\mathbf{w} \cdot \mathbf{x}_i + b \geq +1$, for $y_i = +1$ Group 1 (positive) …… **1**

$\mathbf{w} \cdot \mathbf{x}_i + b \leq -1$, for $y_i = -1$ Group 2 (negative) …… **2**

where y_i is the group index, **w** is a vector normal to the hyperplane, $|b|/||\mathbf{w}||$ is the perpendicular distance from the hyperplane to the origin and $||\mathbf{w}||^2$ is the Euclidean norm of **w**. After the determination of **w** and *b*, a given vector **x** can be classified by:

$\text{sign}[(\mathbf{w} \cdot \mathbf{x}) + b]$ …….. **3**

As most of real-world problems is not linearly separable, SVM can work in combination with the technique of 'kernel' or kernel function, $K(\mathbf{x}_i, \mathbf{x}_j)$, that automatically realizes a nonlinear mapping onto a feature space (Figure 4). The OSH

found by the SVM in feature space corresponds to a nonlinear decision boundary in the input space. An example of a kernel function is the Gaussian kernel :

$$K(\mathbf{x}_i, \mathbf{x}_j) = e^{-\gamma \| \mathbf{x}j - \mathbf{x}i \|2} \quad \text{..... 4}$$

This kernel is commonly used by many other reseachers[73,74,75] and often performed with higher accuracy than other kernel functions in biological research. As such, we chose Gaussian function for the prediction of protein-protein interaction from DIP.

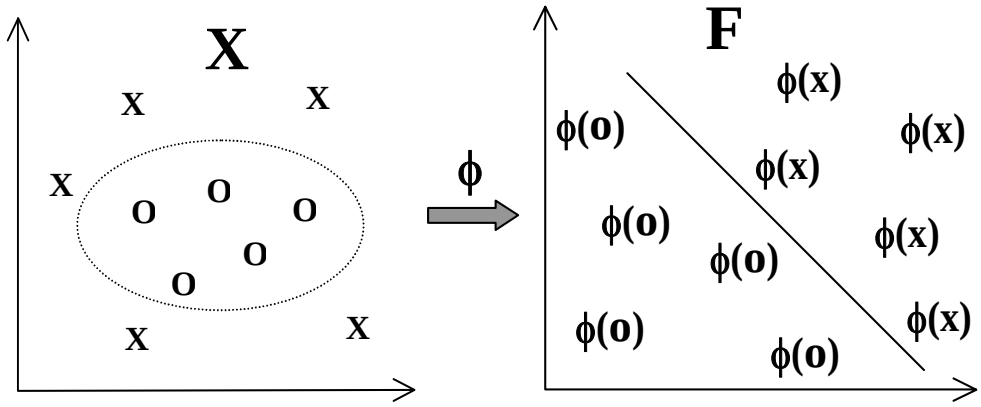


Figure 4 The idea of SV machines

Project the training data nonlinearly into a higher-dimensional feature space via ϕ, and construct a separating hyperplane with maximum margin there.

Linear SVM is applied to this feature space and then the decision function is given by:

$$f(\mathrm{x}) = \mathrm{sign}(\sum \alpha^0_i \, y_i \, K(\mathbf{x}_i, \mathbf{x}_j) + b) \quad \text{..... 5}$$

where the coefficients α^0_i and b are determined by maximizing the following Langrangian expression:

$$\sum \alpha_i - \tfrac{1}{2} \sum\sum \alpha_i \, \alpha_j \, y_i \, y_j \, K(\mathbf{x}_i, \mathbf{x}_j) \quad \text{..... 6}$$

under conditions:

$$\alpha_i \geq 0 \text{ and } \sum \alpha_i \, y_i = 0 \quad \text{..... 7}$$

A positive or negative value from Eq. 3 or Eq. 5 indicates that the vector **x** belongs to the positive or negative group respectively.

As in other statistical learning studies, SVM prediction accuracy can be described by means of the overall classification accuracy Q, precision and recall.

$$Q = \frac{TP + TN}{TP + TN + FP + FN}$$

$$precision = TP/(TP + FP), \qquad recall = TP/(TP + FN)$$

where TP, TN, FP and FN represents true positive, true negative, false positive, and false negative respectively.

Scoring of SVM classification of proteins is estimated by a reliability index (RI) and the *RI* is defined as:

$$RI = \begin{cases} 0 & \text{if } d < 0.2 \\ (d/0.2) & \text{if } 0.2 <= d < 1.8 \\ 9 & \text{if } d >= 1.8 \end{cases}$$

where *d* is the distance between the position of the vector of the classified protein and the optimal separating hyperplane in the hyperspace. The relationship between *RI* value and accuracy percentage or statistical P-value is shown in Table 4 and Figure 5 while the Receiver Operator Characteristic (ROC) plot of each *RI* value can be found in Figure 6. In general, the absolute value of *d* is in the interval [0,2] and *RI* is a value range from 0 to 9 with *RI*=9 corresponding to a rather reliable prediction.

Table 5 Details of *RI* value calculation.

The total number of interactions predicted for each RI range (where P-Positive; N-Negative; TP-True Positive; FN-False Negative; TN-True Negative; FP-False Positive):

RI	**Total P**	**Total N**	**TP**	**FN**	**TN**	**FP**	**Total**	**Correct**	**%**
0-<1	558	384	320	238	218	166	942	538	57.1
1-<2	484	405	308	176	308	97	889	616	69.3
2-<3	394	448	267	127	395	53	842	662	78.6
3-<4	313	439	240	73	412	27	752	652	86.7
4-<5	223	334	182	41	320	14	557	502	90.1
5-<6	144	193	126	18	185	8	337	311	92.3

RI	Total P	Total N	TP	FN	TN	FP	Total	Correct	%
6-<7	67	85	61	6	82	3	152	143	94.1
7-<8	11	36	11	0	36	0	47	47	100
8-<9	10	6	10	0	6	0	16	16	100
9-<10	4	1	4	0	1	0	5	5	100

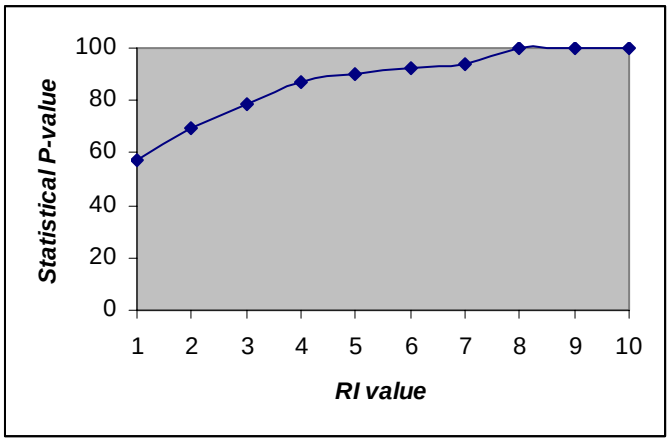


Figure 5 Statistical relationship between *RI* value and P-value

Statistical relationship between the *RI* value and P-value (probability of correct classification) derived from analysis of 2208 positive and 2331 negative protein-protein interaction dataset.

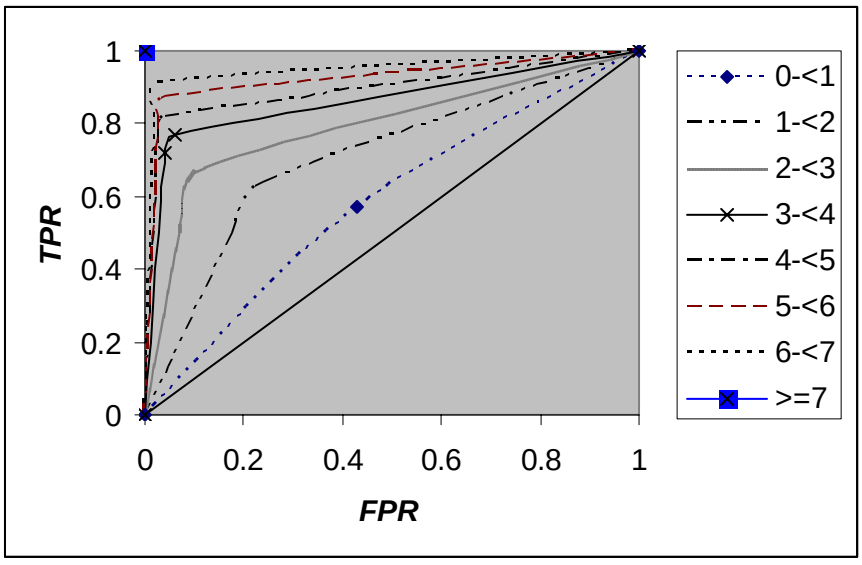


Figure 6 ROC plot of *RI* value

The legend indicate the range of RI value and its corresponding ROC curve. *TPR* is True Positive Rate (Sensitivity) and *FPR* is False Positive Rate (1-Specificity).

3.4 Implementation

MySQL database is used to store the downloaded DIP, MIPS, Swiss-Prot[76] and InterPro[77] databases for further processing. The database structure can be referenced from Appendix A. Swiss-Prot and NCBI databases are also used for extraction of protein sequence and associated features/functions. Java programs are created to parse the various downloaded databases, construct dataset, derive hypothetical negative datasets, generate feature vectors and analyze results. The list of programs and its description can be found in Appendix B. The detailed implementation steps are listed in Appendix C.

CHAPTER FOUR

RESULTS

CHAPTER 4 RESULTS

4.1 Prediction accuracy of three SVMs

Table 5 gives the accuracy of SVM prediction of interacting proteins using both artificial shuffled protein sequences and real protein sequences as the negative datasets. It is found that the prediction accuracy using 1-let shuffled protein sequences as negative dataset is 94.1% while 2-let shuffled protein sequences yields 89.3%, which is comparable to the accuracy of 80.9% from an earlier work[6]. The slight improvement is probably due to the different feature representation and dataset construction methods. In contrast, the prediction accuracy using real protein sequences as negative datasets is 76.9%, which is substantially lower than that derived from the use of shuffled protein sequences as negative datasets.

Table 6 Prediction accuracy of SVM classification of interacting proteins

Prediction accuracy of SVM classification of interacting proteins using shuffled sequences and real protein sequences as negative dataset (dataset for non-interacting proteins). TP, TN, FP and FN represents true positive, true negative, false positive, and false negative respectively. Details of the negative datasets construction are given in the text. A total of 2208 interacting proteins are used as positive testing dataset while 2331 non-interacting proteins are in negative testing dataset. Combined results of five-fold cross validation are shown. The numbers in parentheses under Prediction Accuracy column are corresponding to the standard deviations with five-fold cross validation (Detailed calculation is shown in Appendix D).

Negative dataset	TP	FN	TN	FP	Precision (%)	Recall (%)	Prediction Accuracy (%)
Shuffled sequences (1-let)	2039	169	2233	98	95.4	92.3	94.1 (1.3)
(2-let)	1935	273	2117	212	90.1	87.6	89.3 (0.7)
Real protein sequences	1527	679	1963	368	80.6	69.2	76.9 (1.7)

This result seems to indicate a correlation between the degree of random shuffling of protein sequences in the negative datasets and the computed classification accuracy.

The increasing randomness of the negative dataset tends to give better prediction accuracy, which is expected as increasingly artificial random shuffled sequences are likely to be more easily distinguished from real protein sequences. As shown in Figure 7, even though classifiers trained by shuffled sequences achieve a higher accuracy on shuffled sequences testing datasets, they are not able to perform as well when applied on real sequences testing dataset. This is understandable as the level of difficulty for classifying two datasets of real protein sequences is expected to be higher than that of one set of real protein sequences and one set of shuffled sequences, which partly contributes to the lower classification accuracy derived from the use of real protein sequences. In order to determine the effect of sequence randomness on the performance of the SVM classification, the average distance of support vectors to the respective optimal separating hyperplane for each of the three models is computed. The average distance generated from the negative dataset of real sequence (dr) is 0.54, while that of the shuffled 1-let sequences (ds*1*) and shuffled 2-let sequences (ds*2*) is 0.73 and 0.70 respectively. The classification system of the 1-let shuffled protein sequences gives the largest average distance while that of the real protein sequence gives the smallest average distance. Figure 8 explains the effect of using different negative datasets in a simplified two-dimensional diagram. The larger margin in between the two classes of dataset implies that the model is capable of classifying a given test data better than those with a smaller margin. For example, assuming that the 'data point A' is a positive test data, 'data point A' will be classified correctly when 1-let shuffled sequence model is used but this is not the case when it is classified using the real sequence model.

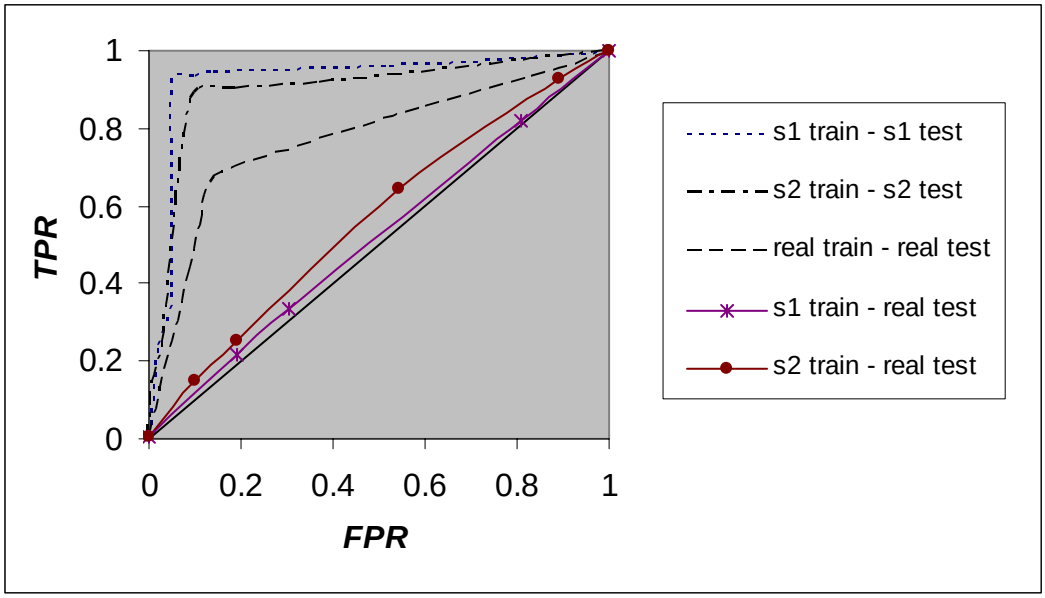


Figure 7 ROC plot of various SVM classifications

s1, s2 and real represents training or testing dataset containing shuffled 1-let, 2-let sequences and real protein sequences as negative dataset respectively while train and test in the legend indicates SVM training and testing dataset. For example, s2 train – real test in the legend means the ROC curve of the classification using SVM model trained with shuffled 2-let protein sequences as negative dataset on real sequences testing dataset.

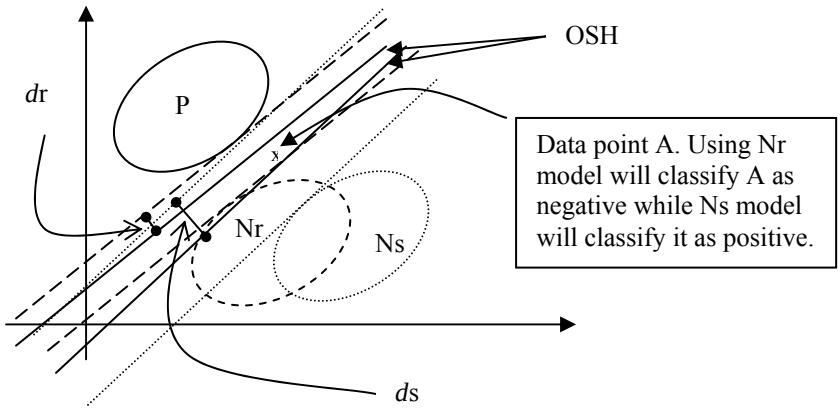


Figure 8 Effect of using different negative datasets

The effect of using different negative datasets in simplified two-dimensional diagram. The larger margin or the distance ($ds > dr$) between the position of the support vector and the optimal separating hyperplane (OSH) in the hyperspace implies that it is able to distinguish positive real sequence dataset (P) and the shuffled sequence negative dataset (Ns) better than real sequence negative dataset (Nr).

4.2 Putative protein partners prediction

4.2.1 Thioredoxin related proteins

To further evaluate the performance of SVM classification systems trained by using different types of negative datasets, a set of thioredoxin related proteins are used as a preliminary test of the prediction capability of these systems in real case studies. Thioredoxins play a critical role in redox regulation of protein function and signaling via thiol redox control. Moreover, they are also known to facilitate DNA binding and to be involved in a number of functions in defense against oxidative stress, control of growth and apoptosis and if secreted, has chemokine activities[4]. Several human thioredoxin related proteins from the Swiss-Prot database[76] are used in this study. The details of the proteins are listed in Table 6.

Table 7 Details of human thioredoxin related proteins from Swiss-Prot

Entry Name	Accession Number	Protein Name	Annotated Functions
PDI_HUMAN	P07237	Protein disulfide isomerase precursor	procollagen-proline 4-dioxygenase activity; protein disulfide isomerase activity
TXN1_HUMAN	Q16881	Thioredoxin reductase	thioreodxin-disulfide reductase activity
TXN5_HUMAN	Q8NBS9	Thioredoxin domain containing 5	potential redox activity
TXNL_HUMAN	O43396	Thioredoxin-like protein 1	plays a role in apoptosis; protein-disulfide reduction and signal transduction

A total of 7,985 human proteins are extracted from Swiss-Prot database as the candidates of potential interacting partners of each of these thioredoxin related proteins. Each of the 7,985 candidate proteins is paired with each thioredoxin related protein to generate feature vectors which are submitted to the three SVM classification systems by the procedure outlined in the Materials and Methods section.

The results in Table 7 suggests that the SVM classification system using artificial shuffled protein sequences (both 1-let and 2-let shuffling) as the negative training datasets may not be practically useful as their ability in identifying potential interacting protein partners seems limited. For example, the dual specificity mitogen-activated protein kinase kinase 4 (P45985), which is involved in signal transduction, is predicted as a putative partner of TXNL_HUMAN by the SVM system of the 1-let shuffled sequences. However, this prediction result maybe questionable as the same protein is also predicted as a partner of TXN1_HUMAN and TXN5_HUMAN which are not known to be involved in signal transduction. On the other hand, the SVM system of the 2-let shuffled sequences predicts a probable ATP-dependent RNA helicase p54 (P26196) as a potential partner of PDI_HUMAN, which appears to be consistent with the entry12097 of the BIND[78] database. This entry describes a protein-protein complex between *Saccharomyces cerevisiae* PDI1 and DBP2 ATP-dependent RNA helicase. Besides that, Sepiapterin reductase (P35270) is also shown to be a possible partner of TXN1_HUMAN[79]. To further assess these prediction results, the two sets of putative protein partners are ranked by the reliability index. As shown in Table 7, the reliability index for the top five protein partners of these two sets is low and thus they may not be confidently predicted as potential partners.

Table 8 Top five prediction results from SVM classifiers trained by shuffled sequences

Top five prediction results (in descending order) from SVM classification of putative interacting protein partners of thioredoxin proteins when shuffled 1-let and 2-let sequences are used as negative dataset. Underlined proteins have evidences of being the putative protein partners or having similar function.

Prediction results using shuffled 1-let sequences as negative dataset			
Thioredoxin proteins (Swissprot ID)	**Putative protein partner**	**(Swissprot ID)**	**[RI]**
PDI_HUMAN Protein disulfide isomerase precursor (P07237)	Leucine carboxyl methyltransferase	(Q9UIC8)	[4.87]
	Desmin	(P17661)	[3.84]
	Oxysterols receptor LXR-alpha	(Q13133)	[3.78]
	ATP-dependent CLP protease ATP-binding subunit ClpX-like	(O76031)	[3.59]
	Replication protein A 30 kDa subunit	(Q13156)	[3.56]
TXN1_HUMAN Thioredoxin reductase (Q16881)	Leucine carboxyl methyltransferase	(Q9UIC8)	[2.87]
	Dual specificity mitogen-activated protein kinase kinase 4	(P45985)	[2.86]
	Keratin, type I cytoskeletal 17	(Q04695)	[2.70]
	Oxysterols receptor LXR-alpha	(Q13133)	[2.56]
	Replication protein A 30 kDa subunit	(Q13156)	[2.27]
TXN5_HUMAN Thioredoxin domain containing 5 (Q8NBS9)	Leucine carboxyl methyltransferase	(Q9UIC8)	[3.65]
	Oxysterols receptor LXR-alpha	(Q13133)	[3.41]
	Dual specificity mitogen-activated protein kinase kinase 4	(P45985)	[3.16]
	Replication protein A 30 kDa subunit	(Q13156)	[3.07]
	Desmin	(P17661)	[2.99]
TXNL_HUMAN Thioredoxin-like protein 1 (O43396)	Leucine carboxyl methyltransferase	(Q9UIC8)	[5.16]
	Oxysterols receptor LXR-alpha	(Q13133)	[4.13]
	Desmin	(P17661)	[3.98]
	Dual specificity mitogen-activated protein kinase kinase 4	(P45985)	[3.93]
	Replication protein A 30 kDa subunit	(Q13156)	[3.84]
Prediction results using shuffled 2-let sequences as negative dataset			
Thioredoxin proteins (Swissprot ID)	**Putative protein partner**	**(Swissprot ID)**	**[RI]**
PDI_HUMAN Protein disulfide isomerase precursor (P07237)	Probable ATP-dependent RNA helicase p54	(P26196)	[2.60]
	MutS protein homolog 4	(O15457)	[2.20]
	Short transient receptor potential channel 6 (TrpC6)	(Q9Y210)	[2.16]
	High-affinity cGMP-specific 3,5-cyclic phosphodiesterase 9A	(O76083)	[2.15]
	Protein-arginine deiminase type II (Peptidylarginine deiminase II)	(Q9Y2J8)	[1.96]
TXN1_HUMAN Thioredoxin reductase (Q16881)	Torsin A precursor (Dystonia 1 protein)	(O14656)	[1.15]
	Ethanolamine kinase (EKI)	(Q9HBU6)	[1.02]
	Pendrin (Sodium-independent chloride/iodide transporter)	(O43511)	[0.34]
	Sepiapterin reductase (SPR)	(P35270)	[0.32]
	MutS protein homolog 4	(O15457)	[0.84]
TXN5_HUMAN Thioredoxin domain containing 5 (Q8NBS9)	MutS protein homolog 4	(O15457)	[1.91]
	Torsin A precursor (Dystonia 1 protein)	(O14656)	[1.51]
	Ethanolamine kinase (EKI)	(Q9HBU6)	[1.31]
	Cholinesterase precursor	(P06276)	[1.11]
	Polycystin 2	(Q13563)	[1.10]
TXNL_HUMAN Thioredoxin-like protein 1 (O43396)	MutS protein homolog 4	(O15457)	[2.47]
	Probable ATP-dependent RNA helicase p54	(P26196)	[2.41]
	Ethanolamine kinase (EKI)	(Q9HBU6)	[2.21]
	Polycystin 2	(Q13563)	[2.21]
	Torsin A precursor (Dystonia 1 protein)	(O14656)	[1.93]

In contrast, the SVM system trained by real protein sequences as the negative training dataset appears to be more capable in identifying potential partners (Table 8). For instance, the proto-oncogene serine/threonine-protein kinase pim-1 (P11309) is predicted as one of the top-five potential partners for each of the three thioredoxin related proteins TXN1_HUMAN, TXN5_HUMAN, TXNL_HUMAN and the top potential partner for TXN1_HUMAN. While there is no direct evidence showing thioredoxin-related proteins interacts with Pim-1 kinase, recent research findings have revealed both proteins are regulated via the NF-kB pathway[80,81,82]. Another protein, mitogen-activated protein kinase 1 (P28482), is also predicted as a potential interacting candidate for TXNL_HUMAN which is consistent with its functional roles in signal transduction and apoptosis[83]. In addition, the 26S proteasome non-ATPase regulatory protein (Q15008) is identified as a putative partner of PDI_HUMAN. It is noted that the same complex has been found in *Saccharomyces cerevisiae* (entry 12123 in BIND database). Besides that, several proteins with redox functions such as pyruvate dehydrogenase (P08559); 24-dehydrocholesterol reductase (Q15392) and soluble epoxide hydrolase (P34913) are also identified. Pyruvate dehydogenase (P08559) is known to play a role together with thioredoxin in the redox regulation of mitochondria[84] while 24-dehydrocholesterol reductase (Q15392), which is involved in cholesterol biosynthesis, regulates mitochondria initiated apoptotic pathways that is sensitive to the redox environment[85]. Although there may not be a direct interaction between soluble epoxide hydrolase (P34913) and TXN1_HUMAN, a recent publication has shown that the expression of both proteins in the prostate apoptosis pathway may be correlated[86].

Table 9 Top five prediction results from SVM classifiers trained by real sequence

Top five prediction results (in descending order) from SVM classification of putative interacting protein partners of thioredoxin proteins when real sequences are used as negative dataset. Underlined proteins have evidences of being the putative protein partners. * proteins are most probably false positive as they are currently not known to be interacting with thioredoxin related proteins.

Prediction results using real sequences as negative dataset			
Thioredoxin proteins (Swissprot ID)	**Putative protein partner**	**(Swissprot ID)**	**[RI]**
PDI_HUMAN Protein disulfide isomerase precursor (P07237)	Alpha-2,8-polysialyltransferase*	(Q92187)	[7.77]
	24-dehydrocholesterol reductase precursor	(Q15392)	[7.34]
	Pyruvate dehydrogenase E1 component alpha subunit	(P08559)	[7.15]
	Beta-parvin (Affixin) (CGI-56)*	(Q9HBI1)	[7.08]
	26S proteasome non-ATPase regulatory subunit 6	(Q15008)	[7.02]
TXN1_HUMAN Thioredoxin reductase (Q16881)	Proto-oncogene serine/threonine-protein kinase pim-1	(P11309)	[9.72]
	Exostosin-like 3 (Putative tumor suppressor protein EXTL3)*	(O43909)	[9.50]
	Soluble epoxide hydrolase	(P34913)	[9.24]
	Brain mitochondrial carrier protein-1*	(O95258)	[9.20]
	Alpha-2,8-polysialyltransferase*	(Q92187)	[9.04]
TXN5_HUMAN Thioredoxin domain containing 5 (Q8NBS9)	24-dehydrocholesterol reductase precursor	(Q15392)	[7.17]
	Pyruvate dehydrogenase E1 component alpha subunit	(P08559)	[7.16]
	cAMP-dependent 3,5-cyclic phosphodiesterase 4C*	(Q08493)	[6.78]
	Proto-oncogene serine/threonine-protein kinase pim-1	(P11309)	[6.77]
	Angiotensinogen precursor*	(P01019)	[6.75]
TXNL_HUMAN Thioredoxin-like protein 1 (O43396)	Alpha-2,8-polysialyltransferase*	(Q92187)	[7.93]
	Proto-oncogene serine/threonine-protein kinase pim-1	(P11309)	[7.61]
	24-dehydrocholesterol reductase precursor	(Q15392)	[7.42]
	Acidic fibroblast growth factor intracellular binding protein*	(O43427)	[7.17]
	Mitogen-activated protein kinase 1(MAP kinase 2)	(P28482)	[6.91]

These results show that the predicted protein interaction pairs derived from the SVM system of real sequences are more consistent with experimental findings than those from artificial sequences, which suggest that SVM classification systems trained by using real protein sequences may be more practically useful in facilitating the prediction of putative potential interacting partners. Moreover, through the concept of 'guilt-by-association', such systems may also find potential application in facilitating protein function prediction of a novel protein by probing its interaction with other proteins of known function.

It is of interest to note that the four thioredoxin related proteins used in this study have less than 30% sequence identity with each other. The ability of the SVM system trained by the real sequences to predict protein with redox function for all of the four proteins and identify putative protein partners having specific functions for individual protein can be partially attributed to the use of feature vectors which are based on physicochemical property of amino acids sequences rather than sequence similarity. From Table 8, one can see that the false positive rate is not small (indicated by *), which is likely due in part to the limited diversity of the negative datasets used for training the SVM systems.

4.2.2 D. melanogaster interaction dataset

While the thioredoxin examples have shown the potential of SVM classification system trained using real protein sequences as the negative training dataset, it may be more realistic to apply the three classification systems on a larger and more comprehensive dataset. The *D. melanogaster* interaction dataset from DIP which consists of 20988 interactions from 7052 proteins is selected as it is the biggest interaction dataset in DIP at the time of writing. Out of the 20988 interactions, 99.7% are extracted from high-throughput yeast two-hybrid approach[59]. The real sequences classifier predicts 64% as possible interacting protein pairs which is much lower than the shuffled sequences trained classifiers (91.2% and 85.5% for shuffled 1-let and shuffled 2-let sequences respectively). However, the recent quality check on DIP yeast dataset (about 8000 interactions) indicates that only 50% of the dataset is reliable[58] while Sprinzak *et al.*[22] has shown that the reliability of high-throughput yeast two-hybrid assays is about 50% which may imply that the false positive rate in the *D. melanogaster* dataset can be close to 50%. This result suggests classifiers

trained by shuffled sequences are not very capable in differentiating the true positive or real interacting protein pairs from a false positive, a non-interacting protein pairs when applying in real testing dataset. Nevertheless, there is a need to include reliability check, as suggested by Deane *et al.*[58], in addition to the *RI* value generated by classifer trained by real protein sequences in order to minimize the false positive rate.

CHAPTER FIVE

DISCUSSION

CHAPTER 5 DISCUSSION

5.1 Significance of results

Our results suggest that the ability of SVM in prediction of putative protein partners is improved when real protein sequences instead of shuffled sequences are used as negative dataset when applied in real life situation. Even though the accuracy of the three SVM systems tested seem to imply that the shuffled sequences trained SVM is the better classifier, it is clearly shown in the results that the higher accuracy is due to the ability of the shuffled sequences classifiers differentiating between real sequences and shuffled sequences rather than between true positive (interacting real protein pairs) and false positive (non-interacting real protein pairs) (refer to Figure 7). Hence in order to develop a SVM system that can be applicable to the real world, it is essential to train the system using real protein sequences.

However it is important to recognize that the SVM classifier trained using real sequences, while providing some predictive power, it is not performing well enough to be used on its own due to a high level of false positives. The main limiting factor is probably due to the amount and quality of the currently available interaction dataset, in additional to the dataset selection and representation. While effort can be applied in improving the later, the accuracy of the SVM classification system in protein-protein interaction is still clearly limited by the reliability of the training dataset. Hence there is a need to include reliability check, as suggested by Deane *et al.*[58], in combination with the verification of subcellular localization and the interaction sequence signature of interacting proteins (method 9 in Table 1), so as to complement the *RI* value (Table 5) in minimizing the false positive rate.

5.2 Dataset selection

Since the negative dataset is obtained from exclusion study of currently published yeast interaction dataset in combination of subcellular localization information as the addition verification, the possible representatives of the protein pairs are enormous. As the result, the dataset is acquired via random sampling and steps have been taken to ensure that representatives are selected (negative datasets have been verified with both positive dataset and negative dataset to ensure only those less than 30% identity with 90% sequence length are chosen). This is done in order to select a manageable set that at least more or less evenly distributed in the protein-pair space. However it is understandable that density of representatives has been reduced, but they are still representative of the protein-pair space. As SVM classifies proteins by a hyperplane (border line), the reduction in the density of representatives likely introduce errors for protein-pairs near the border line as a fine-detailed border line is more difficult to draw without more details. But overall, a rough border line is still useful for distinguishing a majority of protein pairs that are away from the border line. Thus the reduction of overall accuracies may be limited.

The main aim is to minimize the error or noise of the SVM training process by keeping the ratio between the positive and negative dataset close to 1:1 in order not to introduce any overfitting or bias into the classification.

In addition to the selection of the negative dataset, it is of equal importance to have a quality positive dataset as well. As currently the only publicly available and validated positive interacting protein pairs were extracted from yeast interaction data of DIP, this set of positive training dataset may not be representative of all interacting

proteins. Hence further improvement in the prediction capability is expected if a more comprehensive training data is used.

5.3 Data representation

Besides the dataset selection, the feature vector representation also plays important role in improving the classification. Previous approach[6] has represented protein sequence using three features: charge, hydrophobicity and surface tension to achieve 80.96% accuracy when shuffled sequences are used to train the SVM. A different feature representation method has been adopted in this study to assess if it can increase SVM accuracy. Seven features which include amino acids composition, hydrophobicity, Van der Waals volume, polarity, polarizability, charge and surface tension and a more comprehensive feature extraction method[25] has been used. As shown in the result, the accuracy for 1-let hypothetical shuffled sequence dataset is much better at 94.1%. The different dataset representation may have improved the ability of SVM to classify the dataset as similar approach has been successfully applied to various protein structural prediction including protein secondary structure[44], protein fold[45] and protein structural class[43].

5.4 Possible improvement on SVM model

Recursive feature elimination (RFE) has been successfully used in SVM gene selection and classification[47]. However RFE has relatively high computational cost as it constructs a pair (classifier, ranked gene set) from samples in a training set and evaluated on a test set at each model building step. The contribution of each variable is defined through a function of the corresponding weight coefficient that appears in the formula defining the SVM model. The elimination of a single variable at each step

is inefficient. Recently Furlanello *et al*.[87] introduce entropy-based recursive feature elimination (E-RFE) as a non-parametric procedure for gene ranking, which eliminates chunks of genes at every loop without reducing accuracy. This method may be able to assist in selecting representative protein pairs for building a more suitable training dataset.

Since the support vectors generated are solely based on the training dataset, it will be useful to explore both training and testing dataset to identify possible support vectors to train for the best SVM system. An independent dataset can then be introduced to assess the accuracy of the system.

5.5 Multi-class SVM for InterPro groups prediction

In order to address a possible way to predict protein-protein interaction without the completeness of interaction data, the focus is narrowed to analyze protein interaction data of individual protein domain. Three InterPro groups are selected for their basic function in interacting with DNA/RNA [IPR001163 small nuclear ribonucleoprotein; IPR000504 RNA-binding region RNP-1; IPR001138 Fungal transcriptional regulatory protein]. SVM^{light} is then used to differentiate the groups based on the derived feature sets from the protein and its partner. The encouraging result shown in Table 9 confirmed that SVM can indeed classify the various InterPro groups even though their underlying basic function is similar. This preliminary assessment of Interpro groups classification complement the observation that the usage of real protein sequence in machine learning application of protein-protein interaction play a significant role in improving the prediction.

Table 10 Prediction result for InterPro groups

Dataset type	InterPro	Number of records	Accuracy
Positive	IPR000504	131	86.03%
Negative	IPR001163	126	
Positive	IPR000504	131	94.57%
Negative	IPR001138	140	
Positive	IPR001163	126	93.98%
Negative	IPR001138	140	

The binary classification method in the protein-protein interaction prediction using InterPro groups can be extended to address a multi-class problem by using one-versus-others technique[15]. Given the positive results shown in Table 9, we can include more InterPro groups to classify and differentiate among the different groups. Besides that we maybe able to use this approach to predict the possible interacting partner or ligand of a given protein based on the domain, such as WW and SH3 domain which both bind proline rich ligands[88]. As WW and SH3 domains play significant role in signal transduction, it is of interest to understand the ligand recognition by the two domains in order to design drug or ligand to selectively targeting these interactions.

5.6 Prediction and experimental proof

The SVM based protein-protein prediction method has certainly shown its value in predicting putative protein partners of an unknown protein by merely using its amino acids sequence. However as it is a machine learning system trained from a set of input data, its accuracy fully depends on the quality and amount of training data that is currently available. It is essential to have experimental proof of the putative interaction data before a conclusion can be made.

Recently, cluster analysis of gene expression data has shown that genes with similar functions are likely to be co-expressed[10], hence prediction of protein-protein

interactions by combining computer classification with additional information such as protein cellular localization and co-expression profile will definitely help in building a better prediction tool, not only as a tool to predict putative interacting partners but also provides a valuable clue to the role of a novel or un-annotated protein.

CHAPTER SIX

CONCLUSION

CHAPTER 6 CONCLUSION

6.1 Concluding remarks

Our study shows that the SVM classification system trained using real protein sequences as the negative training dataset performs better in real testing cases than that using artificial shuffled sequences. Even though the computed prediction accuracy of the former appears to be lower than the later, the later may not adequately reflect the true prediction capability because of the intrinsically higher level of difficulty for distinguishing real protein sequences than that for separating real protein sequences from artificial ones. This suggests the importance of using real protein sequences in developing SVM classification systems into a practical tool for protein analysis. Further improvement in the diversity and quality of datasets and classification algorithm may be useful in increasing the prediction accuracy of SVM. These, combined with the analysis of additional information such as co-expression profile, may be of help in developing SVM and other classification methods into a useful tool for the protein-protein interaction and protein function prediction.

BIBLIOGRAPHY

BIBLIOGRAPHY

1. Altschul, S. F., Madden, T. L., Schaffer, A. A., Zhang. J., Zhang, Z., Miller, W., Lipman, D.J. (1997) Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Research* **25**, 3389-3402

2. The *C. elegans* Sequencing Consortium (1998) Genome sequence of the nematode *C. elegans*: a platform for investigating biology. *Science* **282**, 2012-2018

3. Tatusov, R. L., Galperin, M. Y., Natale, D. A. & Koonin, E. V. (2000) The COG database: a tool for genome-scale analysis of protein functions and evolution. *Nucleic Acids Research* **28**, 33-36

4. Amer, E. S. J., and Holmgren, A. (2000) Physiological functions of thioredoxin and thioredoxin reductase. *European Journal of Biochemistry* **267**, 6102-6109

5. Deng, M., Zhang, K., Mehta, S., Chen, T., Sun, F. (2002) Prediction of protein function using protein-protein interaction data. *IEEE Computer Society in Bioinformatics*

6. Bock, J. R. and Gough, D. A. (2001) Predicting protein-protein interactions from primary structure. *Bioinformatics* **17**, 455-460

7. Gomez, S. M., Rzhetsky, A. (2002) Towards the prediction of complete protein--protein interaction networks. *Pacific Symposium on Biocomputing* 413-424

8. Brenner, S. E., Chothia, C. and Hubbard, T. J. P. (1998) *Proceedings of the National Academy of Sciences of the United States of America* **95**, 6073-6078

9. Eisen, M. B., Spellman, P. T., Brown, P. O. and Bostein, D. (1998) Cluster analysis and display of genomewide expression patterns. *Proceedings of the National Academy of Sciences of the United States of America* **95**, 14863-14868

10. Brown, M. P. S., Grundy, W. N., Lin, D., Cristianini, N., Sugnet, C. W., Furey, T. S., Ares, M. Jr, Haussler, D. (2000) Knowledge-based analysis of microarray gene expression data by using support vector machines. *Proceedings of the National Academy of Sciences of the United States of America* **97**, 262-267

11. von Mering, C., Krause, R., Snel, B., Cornell, M., Oliver, S. G., Fields, S., Bork, P. (2002) Comparative assessment of large-scale data sets of protein-protein interactions. *Nature* **417**, 399-403

12. Gavin, A.C., Bosche, M., Krause, R., Grandi, P., Marzioch, M., Bauer, A., Schultz, J., Rick, J. M., Michon, A. M., Cruciat, C. M., Remor, M., Hofert, C., Schelder, M., Brajenovic, M., Ruffner, H., Merino, A., Klein, K., Hudak, M., Dickson, D., Rudi, T., Gnau, V., Bauch, A., Bastuck, S., Huhse, B., Leutwein, C., Heurtier, M. A., Copley, R. R., Edelmann, A., Querfurth, E., Rybin, V., Drewes, G., Raida, M., Bouwmeester, T., Bork, P., Seraphin, B., Kuster, B., Neubauer, G., Superti-Furga, G. (2002) Functional organization of yeast proteome by systematic analysis of protein complexes. *Nature* **415**, 141-147

13. Ho, Y., Gruhler, A., Heilbut, A., Bader, G. D., Moore, L., Adams, S. L., Millar, A., Taylor, P., Bennett, K., Boutilier, K., Yang, L., Wolting, C., Donaldson, I., Schandorff, S., Shewnarane, J., Vo, M., Taggart, J., Goudreault, M., Muskat, B., Alfarano, C., Dewar, D., Lin, Z., Michalickova, K., Willems, A. R., Sassi, H., Nielsen, P. A., Rasmussen, K. J., Andersen, J. R., Johansen, L. E., Hansen, L. H., Jespersen, H., Podtelejnikov, A., Nielsen, E., Crawford, J., Poulsen, V., Sorensen, B. D., Matthiesen, J., Hendrickson, R. C., Gleeson, F., Pawson, T., Moran, M. F., Durocher, D., Mann, M., Hogue, C. W., Figeys, D., Tyers, M. (2002) Systematic identification of protein complexes in Saccharomyces cerevisiae by mass spectrometry. *Nature* **415**, 180-183

14. Ita, T., Chiba, T., Ozawa, R., Yoshida, M., Hattori, M. and Sakaki, Y. (2001) A comprehensive two hybrid analysis to explore the yeast protein interactome. *Proceedings of the National Academy of Sciences of the United States of America* **98**, 4569-4574

15. Pellegrini, M., Marcotte, E. M., Thompson, M. J., Eisenberg, D., Yeates, T. O. (1999) Assigning protein functions by comparative genome analysis: protein phylogenetic profiles. *Proceedings of the National Academy of Sciences of the United States of America* **96**, 4285-4288

16. Dandekar, T., Snel, B., Huynen, M., Bork, P. (1998) Conservation of gene order: a fingerprint of proteins that physically interact. *Trends in Biochemical Sciences* **23**, 324-328

17. Marcotte, E. M., Pellegrini, M., Ng, H. L., Rice, D. W., Yeates, T. O., Eisenberg, D. (1999) Detecting protein function and protein-protein interactions from genome sequences. *Science* **285**, 751-753

18. Enright, A. J., Iliopoulos, I., Kyrpides, N. C., Ouzounis, C. A. (1999) Protein interaction maps for complete genomes based on gene fusion events. *Nature* **402**, 86-90

19. Huynen, M., Snel, B., Lathe III, W. and Bork, P. (2000). Predicting protein function by genomic context: Quantitative evaluation and qualitative inferences. *Genome Research* **10,** 1204–1210

20. Pazos, F. and Valencia, A. (2002). In silico two-hybrid system for the selection of physically interacting protein pairs. *Proteins* **47,** 219–227.

21. Dohkan S., Koike, A., Takagi, T., (2003) Support Vector Machines for Predicting Protein-Protein Interactions. *Genome Informatics* 14: 502-503

22. Sprinzak, E. and Hanah, M. (2001) Correlated sequence-signatures as markers of protein-protein interaction. *Journal of Molecular Biology* **311**, 681-692

23. Lu, L., Arakaki, A.K., Lu, H., Skolnick, J. (2003) Multimeric threading-based prediction of protein–protein interactions on a genomic scale: application to the *Saccharomyces cerevisiae* proteome. *Genome Research* **6A**, 1146-1154

24. Wojcik, J. and Schachter, V. (2001) Protein-protein interaction map inference using interacting domain profile pairs. *Bioinformatics* **17**, S296-S305

25. Smith, G. R. and Sternberg, M. J. E. (2002) Prediction of protein-protein interactions by docking methods. *Current Opinion in Structural Biology* **12**, 28-35

26. Park, J., Lappe, M., Teichmann, S. A. (2001) Mapping protein family interactions: intramolecular and intermolecular protein family interaction repertories in the PDB and yeast. *J. Mol Biol.* **307**, 929-938

27. Wollacott, A. M. and Desjarlais, J. R. (2001). Virtual interaction profiles of proteins. *J. Mol Biol.* **313**, 317-342

28. Zhou, H. and Shan, Y. (2001) Prediction of protein interaction sites from sequence profile and residue neighbor list. *Proteins*, **44**, 336-343

29. Pazos, F., Helmer-Citterich, M., Ausiello, G., Valencia, A. (1997) Correlated mutations contain information about protein-protein interaction. *Journal of Molecular Biology* **271**, 511-523

30. Kini, R. M., Evans, H. J. (1996) Prediction of potential protein-protein interaction sites from amino acid sequence. *FEBS Letters* **385**, 81-86

31. Kortemme, T. and Baker, D. (2002) A simple physical model for binding energy hot spots in protein-protein complexes. *Proceedings of the National Academy of Sciences of the United States of America* **99**, 14116-14121

32. Jones, S. and Thornton, J. M. (1997) Analysis of protein-protein interaction sites using surface patches. *Journal of Molecular Biology* **272**, 121-132

33. Jones, S and Thornton, J. M. (1996) Principles of protein-protein interactions. *Proceedings of the National Academy of Sciences of the United States of America* **93**, 13-20

34. Gallet, X., Charloteaux, B., Thomas, A., Brasseur, R. (2000) A fast method to predict protein interaction sites from sequences. *Journal of Molecular Biology* **302**, 917-926

35. Pawson, T., Gish, G. D. and Nash, P. (2001) SH2 domains, interaction modules and cellular wiring. *Trends in Cellular Biology* **11**, 504-511

36. Oliver, S. (2002) Guilt-by-association goes global. *Nature* **403**, 601-603

37. Fields, S., Song, O. (1989) A novel genetic system to detect protein-protein interactions. *Nature* **340**, 245-246

38. Zhu, H., Bilgin, M., Bangham, R., Hall, D., Casamayor, A., Bertone, P., Lan, N., Jansen, R., Bidlingmaier, S., Houfek, T., Mitchell, T., Miller, P., Dean, R. A.,

Gerstein, M., Snyder, M. (2001) Global analysis of protein activities using proteome chips. *Science* **293**, 2101-2105

39. Hughes, T.R., Marton, M. J., Jones, A. R., Roberts, C. J., Stoughton, R., Armour, C. D., Bennett, H. A., Coffey, E., Dai, H., He, Y. D., Kidd, M. J., King, A. M., Meyer, M. R., Slade, D., Lum, P. Y., Stepaniants, S. B., Shoemaker, D. D., Gachotte, D., Chakraburtty, K., Simon, J., Bard, M., Friend, S. H. (2000) Functional discovery via a compendium of expression profiles. *Cell* **102**, 109-126

40. Tong, A. H., Evangelista, M., Parsons, A. B., Xu, H., Bader, G. D., Page, N., Robinson, M., Raghibizadeh, S., Hogue, C. W., Bussey, H., Andrews, B., Tyers, M., Boone, C. (2001) Systematic genetic analysis with ordered arrays of yeast deletion mutants. *Science* **294**, 2364-2368

41. Bader, G. D. and Hogue, C.W. (2002) Analyzing yeast protein-protein interaction data obtained from different sources. *Nature Biotechnology* **20**, 991-997

42. Rain, J-C., Selig, L., Reuse, H., Battaglia, V., Reverdy, C., Simon, S., Lenzen, G., Petel, F., Wojcik, J., Schachter, V., Chememe, Y., Labigne, A. and Legrain, P. (2001) The protein-protein interaction map of *Helicobacter pylori*. *Nature* **409**, 211-215

43. Cai, Y. D., Liu, X. J., Xu, X. B., Chou, K. C. (2002) Prediction of protein structural classes by support vector machines. *Journal of Computational Chemistry* **26**, 293-296

44. Hua, S. J. and Sun, Z. R. (2001) A novel method of protein secondary structure prediction with high segment overlap measure: support vector machine approach. *Journal of Molecular Biology* **308**, 397-407

45. Ding, C. H. Q. and Dubchak, I. (2001) Multi-class protein fold recognition using support vector machines and neural networks. *Bioinformatics* **17**, 349-358

46. Yuan, Z., Burrage, K., Mattick, J. S. (2002) Prediction of protein solvent accessibility using support vector machines. *Proteins* **48**, 566-570

47. Ramaswamy, S., Tamayo, P., Rifkin, R., Mukherjee, S., Yeang, C-H., Angelo, M., Ladd, C., Reich, M., Latulippe, E., Mesirov, J. P., Poggio, T., Gerald, W., Loda, M., Lander, E. S., and Golub, T. R. (2001) Multiclass cancer diagnosis using tumor gene expression signatures. *Proceedings of the National Academy of Sciences of the United States of America* **98**, 15149-15154

48. Deng, M., Mehta, S., Sun, F., and Chen, T., Inferring domain-domain interactions from protein-protein interactions. (2002) *Proceedings 6th International Conference on Computaional Biology (RECOMB02)*, 117–126

49. Cai, Y. D., Liu, X. J., Xu, X. B., Chou, K. C. (2002). Support Vector Machines for predicting HIV protease cleavage sites in protein. *Journal of Computational Chemistry* **23**, 267-274

50. Natt, K. N., Kaur, H. and Raghava, G. P. S. (2004) Prediction of transmembrane regions of β-barrel proteins Using ANN- and SVM-based methods. *Proteins* **56**, 11-18

51. Vapnik, V. (1995) The nature of statistical learning theory. New York: Springer-Verlag

52. Burges, C. J. C. (1998) A tutorial on Support Vector Machine for pattern recognition. *Data Mining and Knowledge Discovery* **2**, 121-167

53. Joachims, T. (1999) Transductive inference for text classification using Support Vector Machines. *International Conference on Machine Learning (ICML)*

54. Bellili, A., Gilloux, M., Gallinari, P. (2001) An hybrid MLP-SVM handwritten digit recognizer. *Sixth International Conference on Document Analysis and Recognition (ICDAR '01)* September 10 - 13, 2001, Seattle, Washington. P. 0028

55. Thubthong, N. and Kijsirikul, B. (2001) Support Vector Machines for thai phoneme recognition. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* **9**, 803-813

56. Heisele, B., Ho, P., and Poggio, T. (2001) Face recognition with support vector machines: global versus component based approach. *Proceedings of 8th International Conference on Computer Vision* **2**, 688–694

57. Xenarios, I., Salwinski, L., Duan, X. J., Higney, P., Kim, S., Eisenberg, D. (2002) DIP: The Database of Interacting Proteins. A research tool for studying cellular networks of protein interactions. *Nucleic Acids Research* **30**, 303-305

58. Deane, C. M., Salwinski, L., Xenarios, I., Eisenberg, D. (2002) Protein interactions – Two methods for assessment of the reliability of high throughput observations. *Molecular & Cellular Proteomics* **1.5**, 349-356

59. Giot, L., Bader, J. S., Brouwer, C., Chaudhuri, A., Kuang, B., Li, Y., Hao, Y. L., Ooi, C. E., Godwin, B., Vitols, E., Vijayadamodar, G., Pochart, P., Machineni, H., Welsh, M., Kong, Y., Zerhusen, B., Malcolm, R., Varrone, Z., Collis, A., Minto, M., Burgess, S., McDaniel, L., Stimpson, E., Spriggs, F., Williams, J., Neurath, K., Ioime, N., Agee, M., Voss, E., Furtak, K., Renzulli, R., Aanensen, N., Carrolla, S., Bickelhaupt, E., Lazovatsky, Y., DaSilva, A., Zhong, J., Stanyon, C. A., Finley, R. L. Jr, White, K.P., Braverman, M., Jarvie, T., Gold, S., Leach, M., Knight, J., Shimkets, R. A., McKenna, M. P., Chant, J., Rothberg, J. M. (2003) A protein interaction map of Drosophila melanogaster. *Science* **302**(5651), 1727-1736

60. Mewes, H. W., Amid, C., Arnold, R., Frishman, D., Guldener, U., Mannhaupt, G., Munsterkotter, M., Pagel, P., Strack, N., Stumpflen, V., Warfsmann, J., Ruepp, A. (2004) MIPS: analysis and annotation of proteins from whole genomes. *Nucleic Acids Research* **32** Database issue:D41-D44

61. Coward, E. (1999) Shufflet: shuffling sequences while conserving the k-let counts. *Bioinformatics* **15**, 1058-1059

62. Ofran, Y. and Rost, B. (2003) Analysing six types of protein-protein interfaces. *Journal of Molecular Biology* **325**, 377-387

63. Kandel, D., Matias, Y., Unger, R. and Winkler, P. (1996) Shuffling biological sequences. *Discrete Applied Mathematics* **71**, 171-185

64. BLASTCLUST - BLAST score-based single-linkage clustering [ftp://ftp.ncbi.nih.gov/blast/documents/blastclust.txt]

65. Chothia, C., Finkelstein, A. V. (1990) The classification and origins of protein folding patterns. *Annual Review of Biochemistry* **59**, 1007-1039

66. Fauchere, J. L., Charton, M., Kier, L. B., Verloop, A., Pliska, V. (1988) Amino acid side chain parameters for correlation studies in biology and pharmacology. *International Journal of Peptide and Protein Research* **3**, 269-278

67. Gratham, R. (1974) Amino acid difference formula to help explain protein evolution. *Science* 185, 862-864

68. Charton, M., Charton, B. I. (1982) The structural dependence of amino acid hydrophobicity parameters. *Journal of Theoretical Biology* **99**, 629-644

69. Taylor, W. R. (1986) The classification of Amino Acid Conservation. *Journal of Theoretical Biology* **119**, 205-218

70. Bull, H. B., Breese, K. (1974) Surface tension of amino acid solutions: a hydrophobicity scale of the amino acid residues. *Archives of Biochemistry and Biophysics* **161**, 665-670

71. Levitt, M. (1976) A simplified representation of protein conformations for rapid simulation of protein folding. *Journal of Molecular Biology* **104**, 59-107

72. Joachims T. (1999) Making large-scale SVM learning practical. In: Scholkopf B, Burges C, Smola A, editors. *Advanced in kernel methods: support vector learning*. Cambridge, MA: MIT Press. 42-56

73. Burbidge, R., Trotter, M., Buxton, B. and Holden, S. (2001) Drug design by machine learning: support vector machines for pharmaceutical data analysis. *Computational Chemistry* **26**, 5-14

74. Czerminski, R., Yasri, A. and Hartsough, D. (2001) Use of Support Vector Machine in Pattern Classification: Application to QSAR studies. *Quantitative Structure-Activity Relationships* **20**, 227-240

75. Trotter, M. W. B., Buxton, B. F. and Holden, S. B. (2001) Support Vector Machines in combinational chemistry. *Measurement and Control* **34**, 235-239.

76. Boeckmann, B., Bairoch, A., Apweiler, R., Blatter, M.-C., Estreicher, A., Gasteiger, E., Martin, M. J., Michoud, K., O'Donovan, C., Phan, I., Pilbout, S.,

Schneider, M. (2003) The SWISS-PROT protein knowledgebase and its supplement TrEMBL in 2003. *Nucleic Acids Research* **31**, 365-370

77. Apweiler, R., Attwood, T. K., Bairoch, A., Bateman, A., Birney, E., Biswas, M., Bucher, P., Cerutti, L., Corpet, F., Croning, M. D. R., Durbin, R., Falquet, L., Fleischmann, W., Gouzy, J., Hermjakob, H., Hulo, N., Jonassen, I., Kahn, D., Kanapin, A., Karavidopoulou, Y., Lopez, R., Marx, B., Mulder, N. J., Oinn, T. M., Pagni, M., Servant, F., Sigrist, C. J. A., Zdobnov, E. M. (2001) The InterPro database, an integrated documentation resource for protein families, domains and functional sites. *Nucleic Acids Research* **29**, 37-40

78. Bader, G. D., Donaldson, I., Wolting, C., Ouellette, B. F., Pawson, T., Hogue, C. W. (2001). BIND--The Biomolecular Interaction Network Database. *Nucleic Acids Research* **29**, 242-245

79. Schallreuter, K. U., Buttner, G., Pittelkow, M. R., Wood, J. M., Swanson, N. N., Korner, C. (1994) Cytotoxicity of 6-biopterin to human melanocytes. *Biochemical and Biophysical Research Communications* **204**, 43-48

80. Sakurai, A., Yuasa, K., Shoji, Y., Himeno, S., Tsujimoto M, Kunimoto M, Imura N, Hara S. (2004) Overexpression of thioredoxin reductase 1 regulates NF-kappa B activation. *Journal of Cellular Physiology* **198**, 22-30

81. Zhang, J., Velsor, L. W., Patel, J. M., Postlethwait, E. M. and Block, E. R. (1999) Nitric oxide-induced reduction of lung cell and whole lung thioredoxin expression is regulated by NF-kappaB. *American Journal of Physiology* **277**, 787-793

82. Zhu, N., Ramirez, L. M., Lee, R. L., Magnuson, N.S., Bishop GA, Gold MR. (2002) CD40 signaling in B cells regulates the expression of the Pim-1 kinase via the NF-kappa B pathway. *Journal of Immunology* **168**, 744-754

83. Shao, L.-E., Tanaka, T., Gribi, R. and Yu, J. (2002) Thioredoxin-related regulation of NO/NOS activities. *Annals of the New York Academy of Sciences* **962**, 140-150

84. Bunik, VI. (2003) 2-Oxo acid dehydrogenase complexes in redox regulation. *European Journal of Biochemistry* **270**, 1036-1042

85. Fernandez-Checa, J. C. (2003) Redox regulation and signaling lipids in mitochondrial apoptosis. *Biochemical and Biophysical Research Communications* **304**, 471-479

86. Pang, S. T., Dillner, K., Wu, X., Pousette, A., Norstedt, G., Flores-Morales, A. (2002) Gene expression profiling of androgen deficiency predicts a pathway of prostate apoptosis that involves genes related to oxidative stress. *Endocrinology* **143**, 4897-4906

87. Furlanello, C., Serafini, M., Merler, S., and Jurman, G.(2003) Entropy-based gene ranking without selection bias for the predictive classification of microarray data. *BMC Bioinformatics* **4**, 54

88. Aghazadeh, B., and Rosen, M. K. (1999) Ligand recognition by SH3 and WW domains: the role of N-alkylation in PPII helices. *Chemistry & Biology* **6**, R241–R246

89. Bateman, A., Birney, E., Cerruti, L., Durbin, R., Etwiller, L., Eddy, S. R., Griffiths-Jones, S., Howe, K. L., Marshall, M., Sonnhammer, E. L. (2002) The Pfam Protein Families Database. *Nucleic Acids Research* **30**, 276-280

APPENDICES

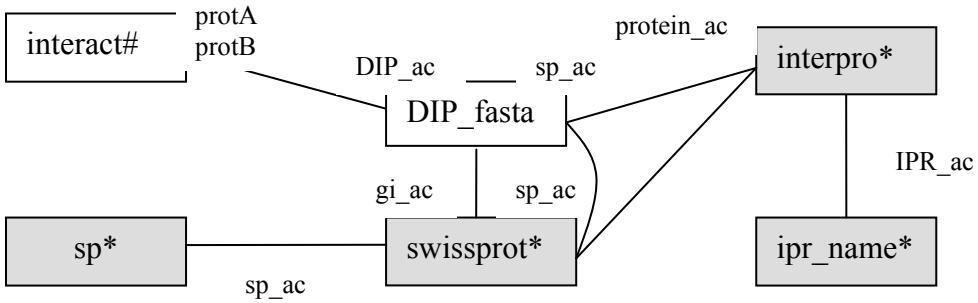
APPENDICES

Appendix A Database structure
Appendix B List of Programs
Appendix C Implementation details
Appendix D Calculation details
Appendix E List of Datasets and models

All softcopy of the database creation script; database content; programs and datasets are available in a CD upon request.

Appendix A: Database structure

Entity-Relation diagram:



* main supporting tables

The *core* table containing the validated positive dataset mentioned in 3.4 Implementation Step 1, has the same structure as *interact* hence its detailed table structure is not listed in the next section. As *DIP_fasta* data was downloaded earlier (05/09/2001), it does not correspond well with the *core* table which has dataset version 04/04/2003, hence a newer DIP dataset (01/06/2003) was downloaded to *dip_all*. The structure of *dip_all* is similar with *DIP_fasta* except that *dip_all* does not have the link to *swissprot* and *interpro* tables.

List of tables and its structure

Protein-protein interaction:
Main tables are *DIP_fasta*, *interact, dip_all*.
Supporting tables includes *interpro*; *IPR_name*; *swissport*; *sp, pfamseed* for DIP dataset analysis.

DIP_fasta
Table that stores data from DIP database (version05/09/2001) with its DIP_ac and sequence information together with flag to identify type of record for program processing and InterPro information.

Field name	Type	Remarks
ID	int(7)	primary key (unique) to identify each record
DIP_ac1	varchar(10)	DIP accession no in text ('N' suffix in fasta.txt is removed)
sp_ac	varchar(10)	Swiss-Prot accession no – manually generated using sp_name
sp_name	varchar(250)	Swiss-Prot name – from DIP-fasta.txt
pir_ac	varchar(10)	PIR accession no – from DIP-fasta.txt
gi_ac	varchar(10)	GI accession no – from DIP-fasta.txt
seq	text	Amino acids sequence – from DIP-fasta.txt
flag	varchar(4)	Type of records – TM1: Testing set with unique InterPro but having > 1 protein from the same InterPro TM2: Testing set with multiple InterPro but having >1 one protein from the same InterPro RM1: Training set with unique InterPro but having >1 protein from the same InterPro RM2: Training set with multiple InterPro but having >1 protein from the same InterPro RU1: Training set with unique InterPro and only one in the grp RU2: Training set with multiple InterPro and only one in the grp
interpro_count	int(2)	Total number of InterPro domains found
interpro	varchar(12)	The last InterPro accession no [note: this approach is not good enough as the last InterPro is not a domain representative. Should keep all InterPro domains for future work]
DIP_ac	int(5) unsigned	DIP accession no in integer (currently used)
random_seq	text	Randomly generated sequence based on seq
random_seq2	text	Partial randomly generated sequence (only 50-100 positions are randomly changed) based on seq

Interact
Table that stores DIP interact pairs information.

Field name	Type	Remarks
ID	int(7)	primary key (unique) to identify each record
protA	int(5)	1st protein in the interaction pair
protB	int(5)	2nd protein in the interaction pair

interpro
Table that stores the internet downloaded InterPro database.

Field name	Type	Remarks
ID	int(7)	primary key (unique) to identify each record
protein_ac	varchar(10)	Swiss-Prot accession no
method_ac	varchar(10)	Method to derive the InterPro information
pos_from	int(7)	Protein sequence start position
pos_to	int(7)	Protein sequence end position
IPR_ac	varchar(12)	InterPro accession no
IPR_name	text	InterPro name

IPR_name (concise version of Table InterPro)
Table that stores InterPro accession no and name.

Field name	Type	Remarks
ipr_ac	varchar(12)	InterPro accession no
ipr_name	text	InterPro name

swissprot
Table that stores the internet downloaded Swiss-Prot database.

Field name	Type	Remarks
ID	int(7)	primary key (unique) to identify each record
gi_ac	varchar(10)	GI (genbank) accession no
sp_ac	varchar(10)	Swiss-Prot accession no
sp_desc	text	Swiss-Prot description

sp
Table that stores the internet downloaded Swiss-Prot database with sequence information.

Field name	Type	Remarks
sp_ac	varchar(10)	Swiss-Prot accession no
name	text	Swiss-Prot description
seq	text	Protein sequence

dip_all
Table that stores data downloaded from DIP database version 01/06/2003.

Field name	Type	Remarks
ID	int(7)	primary key (unique) to identify each record
name	varchar(250)	DIP information of the protein including Swiss-Prot, PIR and GI accession no
seq	text	Amino acids sequence
dip_ac	varchar(10)	DIP accession no
shuffled_seq	text	shuffled 1-let sequence based on seq

Field name	Type	Remarks
shuffled_seq2	text	shuffled 2-let sequence based on seq
shuffled_seq3	text	shuffled 3-let sequence based on seq (no sequence as 3-let is too big to generate any meaningful shuffling)

pfamseed

Table that stores the seed protein from Pfam-A.seed version 30/09/2003 database.

Field name	Type	Remarks
sp_ac	varchar(10)	Swiss-Prot accession no. primary key (unique) to identify each record
name	varchar(250)	Swiss-Prot accession no and its description
seq	text	Amino acids sequence

Appendix B : List of programs

List of programs are sorted by directory and its function. Some of the programs are utility or common module that are used to support the function of the programs or simplify activities mentioned in 3.4 Implemention.

Main directory	Program name	Brief Description
java\common	convertFasta.java	convert fasta file(s) to database
	Database.java	common module for database connection and processing
	evenProt.java	extract interact pairs and evenly distributed them to training and testing file based on input file with info of protA/protB and total count. Uniquely found record is extract to single file
	extractS.java	take in single ac and extract interacting partners from interact table in database.
	format.java	convert the comma-delimited in a file to a 8 characters long separator (including the item). i.e., A,B will become A<7 spaces>B. This is to align with Cai's SVM input file format
	genInfo.java	generate info or seq (fasta) using input file which contains the list of input dip_ac no
	genStat.java	generate statistics of the svmlight output for analysis and publication
	getFasta.java	extract all the sequences in interact database in Fasta format
	getFasta2.java	extract interaction pair sequences from input file and database in Fasta format.
	getRand.java	extract the random fixed no of records (set at 3500) from a list of records
	getRep.java	extract the representative of training dataset using BLASTCLUST output
	getSV.java	extract the SV and nonSV of training set using generated alpha file from svm_learn
	ioFasta.java	get sequence from database and generate output in Fasta format or get the sequences from a Fasta file to insert into database depending on input type
	randomSeq.java	generate a random string based on an input string from database with few type of randomness generation
	randomStr.java	randomize position of amino acids
	randomStr2.java	internal randomise the amino acids position - i.e., the aa composition is the same
	randomStr3.java	randomize amino acids according to the length
	readFasta.java	read from Fasta file

Main directory	Program name	Brief Description
	readLine.java	output number of line of the input file
	Splitfile.java/ Splitfile5.java	will split a file 'filename (without .txt)' content to 2 or 3 files depending on no. entered in the second argument
java\database	DIP_interpro.java	fill up interpro (IPR_ac) and interpro in DIP_fasta table under SVM dataset database
	EvenType.java	take records from a table which is generated from DIP_fasta and make training and test records even in number
	getSeed.java	This program will select the 1st representative proteins from pfamA seed file
	IPRname.java	extract ipr_name info from internet INTERPRO record and write to database
	Match.java	match.java is an implementation on MS Access. Matchsp.java is using MySQL
	Matchsp.java	fill up sp_ac (Swiss-Prot accession no) in DIP_fasta table
	Negfile2.java	similar algorithm as Negfile.java but the output are split to Train and Test based on the type in DIP_fasta 'RM1' and 'TM1'
	Negfile3.java	use all RM1 and TM1 from DIP_fasta to generate all possible negative set instead of checking on every record in interact table
	Negfile.java	generate negative training set output file in format that is recognised by cai's SVM
	Negfile_i.java	written to handle another type – Independent data type 'IM1'
	QCneg2.java	second quality check on the negative record to handle duplicated records in the negative set
	QCneg.java	quality check (QC) the negative dataset with the positive dataset
	SetType.java	assign training set (R) and test set (T) and INTERPRO count to DIP_fasta on field 'flag'
	SetType_i.java	similar as SetType.java but handle I' (independent) type
	Splittype.java	will split a file 'filename (without .txt)' content to 2 files and ensurely training file has all representatives.
	SVMfile.java	generate 3 output files in format that is recognised by cai's SVM input
	SVMfile_i.java	SVMfile_i.java is an enhancement version that handles 'I' (independent) type
java\feature	genfeature.java	The main purpose of this program is to generate the feature vectors of a test protein

Main directory	Program name	Brief Description
		given in a file and pair it with every protein (in feature vectors form) in another input file (protfile) to form the feature vector file for classifying purpose.
	qc.java	use original data file to counter check database
	qcdata.java	compare original data file and generated feature vector file to find missing protein id from original data file
	qcdata2.java	check feature vector file and extract those with full record to filename_ok file and those that is not alright to filename_nok file
	sp_feature.java	This program converts 'seq' field from table specified to feature vector. The generated file will be used as protfile in genfeature.java.
	svm_feature.java	This program converts generated SVM files (protA<>protB<>type) to SVM light format - <class> <feature>:<value> ... <feature>:<value> where <class> is +1;-1 <feature> is no of feature (integer) <value> is a real number which represents the calculated value of each features
java\shuffleSeq	exFasta.java	extract the shufflet generated Fasta sequences (with many trials) to individual file containing each trial
	shuffleSeq.java	generate a shuffled sequence conserving the exact k-let counts for a given k (self written, not comprehensive to be used as it is may not have uniformed permutation)
java\weight	distance.java	calculate the distance of a given point to the OSH, optimal separating hyperplane. The distance can then be used to rank the reliability of the prediction
java\appendix*	genSource.java	This program will generate the source information for train.txt and test.txt file by using used='Y' field in DIP_fasta. the format is ID<>sp_ac|gi_ac|pir_ac where total length of ID<> is 8 characters.
	insertName.java	This program is used to fill up sp_desc (swissprot description or name) in DIP_fasta. It used sp_ac to check in sp first and later gi_ac to check in swissprot.
	insertUsed.java	This program will check the interaction pairs

Main directory	Program name	Brief Description
		in format id1<>id2<>type (where id1<> and id2<> are each 8 characters long) and extract id1 and id2 to indicate in DIP_fasta - used field as 'Y' i.e.,the corresponding DIP_ac is used. The default of used is 'N'. The main reason of doing this is to extract only proteins that are used in training and testing interaction pairs by using used field in DIP_fasta.

* detailed explanation is not included in this document as the program under this directory is mainly for generating information for publication purposes. However the source files are included in the softcopy distribution CD.

The details of directory: java\database

This directory contains programs that are related to manipulate data in database and to populate and massage data into format that can be used for analysis.

The relationship of each programs under this directory:

- To generate positive (SVMfile.java) and negative (Negfile.java) dataset for SVM testing:
  Matchsp.java → DIP_interpro.java → SetType.java → SVMfile.java and Negfile.java

- To quality check the negatibe dataset generated:
  Negfile.java → QCneg.java → QCneg2.java

- For supporting and formating purposes:
  - EvenType.java
  - IPRname.java
  - Splittype.java

DIP_interpro.java

Program	DIP_interpro.java
Usage	This program is used to fill up interpro (IPR_ac) and interpro in DIP_fasta table under SVM dataset database These 2 data are important for assignment of training and testing set for SVM.
Algorithm	o connect to database o For each record in DIP_fasta, get Swiss-Prot acc no (sp_ac_dip) o get distinct IPR_ac from interpro table where the Swiss-Prot acc no is found o get the total count and assign the last IPR_ac to DIP_fasta fields, interpro_count and interpro respectively.
Remark	This approach has simplified method but has also truncated important data as for protein with multiple INTERPRO records, only one is recorded. It is necessary to consider keeping the complete INTERPRO domains information with each protein.

EvenType.java

Program	EvenType.java
Usage	This program takes records from a table which is generated from DIP_fasta and make training and test records even in number. The table contains count of training and test set from each interpro type. The aim is to make sure all test set must have a training set and have examples more than test set.
Algorithm	o connect to database

	o create a temporary table ‘data’ to keep records for TM1, RM1 and RU1 (refer to Remark) o get the train and test set count and find the mean o for each record of the same INTERPRO, assign ‘RM1’ to the first half of records before reaching mean and the rest are assigned at ‘TM1’.
Remark	Program is written to consider unique INTERPRO only. Proteins with multiple INTERPRO identified are not considered. It is necessary to address this groups of proteins too. TM1: Testing set from protein with unique interpro but having more than one protein from the same interpro RM1: Training set from protein with unique interpro but having more than one protein from the same interpro RU1: Training set from protein with unique interpro and only one in the grp

getSeed.java

Program	getSeed.java
Usage	This program will select the 1st representative proteins from pfamA seed file (from pfam ftp site). The index file is created using UNIX script 'grep' function on "#=Gf SQ" to get the no of proteins from each family. Then this no is import into Excel to find the 1st no for each group. This no is used as an index. The list of AC (protein accession no) can be extracted from pfamA seed file using UNIX script 'grep' on " AC ". usage: java getSeed index_file ac_file
Algorithm	o read the input file containing index of the 1st seed protein (row no) o read the file containing the Swiss-Prot no of all the seed protein o match the 2 arrays, if the row no is found then the Swiss-Prot no is output to output file, indicating that it is the 1st seed protein of the family
Remark	• only the 1st representative is used from the PfamA seed file to pair with the test protein of interest to find the putative interacting partner. • The generated output file containing Swiss-Prot accession no and this no is then used to link with *swissprot* table in database to extract the protein sequence • For those that is not found, the accession no is used to extract sequence from internet using Entrez.

IPRname.java

Program	IPRname.java
Usage	Extract ipr_name info from INTERPRO record using http://www.ebi.ac.uk/interpro/ISimpleSearch?query= write the ipr_name to database table IPR_name so that all records can be analysed
Algorithm	o read each line of the input file (containing INTERPRO accession no) o For each INTERPRO accession no, go to INTERPRO internet site to get description (IEntry). If found, update database table IPR_name with the accession no, ipr_ac and the description, ipr_name
Remark	Input file is 'input.txt'. Output file method was commented. If not, output file will contain IPR_ac and IPR_name

matchsp.java (and its variant match.java)

Program	Matchsp.java
Usage	This program is used to fill up sp_ac (Swiss-Prot accession no) in DIP_fasta table. The main reason is without sp_ac, we cannot link InterPro data to DIP data. InterPro data is necessary so that the dataset can be split meaningfully into 2 datasets – training and testing set with relevent domains for testing.
Algorithm	o connect to database o get sp_name from DIP_fasta o get sp_ac from swissprot where sp_desc has part of sp_name o if found, update sp_ac to DIP_fasta
Remark	match.java is an implementation on MS Access. Matchsp.java is using MySQL

Negfile.java (and its variant – Negfile2.java; Negfile3.java; Negfile_i.java)

Program	Negfile.java
Usage	This program will generate negative training set output file in format that is recognised by cai's SVM. negativeddMMyy.txt : interaction eg used as training set. e.g. ID1<>ID2<>P (positive set); N (negative) ID*<> is 8 characters long.
Algorithm	o Connect to database o Select distinct protB from interact table o For each protB, get interacting protA o Using the protA, get all protB but it should not equal to original protB o Output result to output file 'negativeddmmyy.txt' The concept is based on idea that if A-B and B-C but no A-C then A-C is considered as negative dataset
Remark	Negfile.java program is used on *interact* table to retrieve hypothetical

	non-interacting protein pairs based on concept that if A-B and B-C but no A-C is found then A-C is considered as negative dataset. This concept is not used in the latest version as subcellular localization is used for negative dataset selection instead. Negfile2.java : similar algorithm as Negfile.java but the output are split to Train and Test based on the type in DIP_fasta 'RM1' and 'TM1'. Negfile3.java : This version will use all RM1 and TM1 from DIP_fasta to generate all possible negative set instead of checking on every record in interact table. Outfile will be Negfileddmmyy.txt Negfile_i.java : This version is written to handle another type – Independent data type 'IM1'. Output files are created under a subdirectory 'file' with filenames as trainNddmmyy.txt, testNddmmyy.txt, indNddmmyy.txt Need to do a quality check on the list of records to ensure that • no match is found in positive dataset (done in QCneg.java program) • no duplication (done in Qcneg2.java program) • apply to both combinations (protAB; protBA)

QCneg.java

Program	QCneg.java
Usage	This program will quality check (QC) the negative dataset with the positive dataset. The input is the the negative data set where each line is a record. If the record (protAB or protBA) is found in the database (protAB or protBA), the record is output to filename_f.txt. The 'not found' record is output to filename_qc.txt. Positive dataset is stored in 'positive' table in 'interact' database in MySQL. Usage java QCneg filename(without .txt [original has .txt])
Algorithm	o connect to database o read each record from input file (file generated by Negfile.java and its variant) and concatenate protA and protB (both protAB and protBA) to compare with positive dataset o assign a flag as 'N', if the same record is found in positive dataset (protAB and protBA. 4 comparisons protAB-protAB; protAB-protBA; protBA-protAB; protBA-protBA) o output found to filename_f.txt and not found to filename_qc.txt
Remark	A temporary table –positive table is created to contain protAB and protBA generated from positive dataset. (defunct – used in exclusion study)

QCneg2.java

Program	Qcneg2.java
Usage	This program is the second quality check on the negative record. There are duplication records in the negative set. Identify negative record in Access database for comparison. The negative record that is found in the database will be output to filename_f.txt. The 'not found' record is output to filename_qc.txt. Duplicate dataset is stored in 'DupAB2' table in 'duplicate' database in MySQL. Use database to check file. Access database - cannot handle too many transaction Usage java QCneg2 filename(without .txt [original has .txt])
Algorithm	o connect to database o get each record from a temporary table DupAB2 which have all the records of duplicate interacting pairs o read each record from input file (file generated by Negfile.java) and concatenate protA and protB (protAB) to compare with each record with DupAB2 – write the matched record, i.e., duplicated record, to filename_f.txt o read each line of filename_f.txt to check with input file, for each record that is not found, output the record to filename_qc.txt
Remark	A temporary table –DupAB2 table is created in another database 'duplicate' which contains all the duplicates (defunct – used in exclusion study)

SetType.java (and its variant SetType_i.java)

Program	SetType.java
Usage	This program assigned training set (R) and test set (T) to DIP_fasta on field 'flag'. As there are records which has unique INTERPRO record, these set of data will be assigned as R1 and T1. For those who more than one INTERPRO records, the data will be assigned as R2 and T2. Currently for multiple INTERPRO records, only the last is stored. TM1: Testing set from protein with unique interpro but having more than one protein from the same interpro TM2: Testing set from protein with multiple interpro but having more than one protein from the same interpro RM1: Training set from protein with unique interpro but having more than one protein from the same interpro RM2: Training set from protein with multiple interpro but having more than one protein from the same interpro RU1: Training set from protein with unique interpro and only one in the grp RU2: Training set from protein with multiple interpro and only one in

	the grp
Algorithm	o connect to database o create 2 temp tables – testset and testset2 where testset contains record with single INTERPRO and testset2 contains record with multiple INTERPRO o join both testset and testset2 with DIP_fasta using INTERPRO accession no so that we can find out how many records can be found with the INTERPRO no and separate the types. o update DIP_fasta with appropriate types and assign even no record as R (train) and odd no as T (test). However if only one INTERPRO is found then the record is assigned as RU1 or RU2 (training dataset).
Remark	A separate database update program is built for updating the different types after the identification using SQL join and grouping. SetType_i.java is another version • that will assign Independent dataset – I' to another field 'type2'. • The dataset is evenly divided to 3 portions where (mod)%3 =0 is R (train); %3=1 is T (test) and the rest is assigned as I (independent). • All first record is assigned to R (train) to ensure all INTERPRO has a representative in R

Splittype.java

Program	Splittype.java
Usage	This program will split a file 'filename (without .txt)' content to 2 files. 'filename'_r.txt'-training set and 'filename'_t.txt-test set Usage java Splittype filename(without .txt [original has .txt])
Algorithm	o Read in file o Ensure unique record will be written to traning set but duplicate record to testing set o Using protA to check if the record is a new record o Count the total no of records and store the content to an ArrayList o Divide the total in half and get the first half to training and the rest to test
Remark	

SVMfile.java (and its variant SVMfile_i.java)

Program	SVMfile.java
Usage	This program will generate 3 output files in format that is recognised by cai's SVM input. sourceddMMyy.txt : source file with ID and sequence, e.g. ID1<>seq1 ID2<>seq2 trainddMMyy.txt : interaction eg used as training set. e.g. ID1<>ID2<>P (positive set); N (negative) testddMMyy.txt : interaction eg used as test set. same format as training.txt ID*<> is 8 characters long.
Algorithm	o Connect to database o Extract DIP_ac and seq from DIP_fasta and protA, protB from interact where type is R (train) or T (test). However here it is restricted to RM1 and TM1 only o Format (ID*<> as 8 characters long) DIP_ac and seq to output to sourceddMMyy.txt o Format protA and protB and type to train and test output file (type is hardcoded. Only 'P' (positive) is handled, negative file is generated by Negfile.java)
Remark	SVMfile_i.java is an enhancement version that handles 'I' (independent) type so after SetType_i.java has assigned the appropriate type, this program can be used to generate positive dataset for independent dataset testing.

The details of directory: java\common

This directory is used to store programs that is mainly a utility tool or data manipulation tool

convertFasta.java

Program	convertFasta.java
Usage	This program will convert fasta file(s) to database. It will take each files from subdirectory 'fasta' and extract the data from each file (must have the word as the first column of table.txt file in order to corresponding to the setting to table.txt for the database). It works for 2 types – DIP format or just no. o Database configuration is found in config.txt. o It will read table name from table.txt where the format: file identifier|table1 name|columns Usage java convertFasta file [dip/no]
Algorithm	o Connect to database via Database.class o Extract the seq from fasta file using tokenization method. o Format extracted data based on type and insert into database
Remark	

Database.java

Program	Database.java
Usage	generic database program which will handle connection; update; delete; insert data
Algorithm	o Database(driver, url) or Database(driver, url, username, password) module for connection o dataExist(table, condition) module o selectData(sqlstmt) o updateData(table, values, condition) o insertData(table, columns, values) o deleteData(table) o closeConnection()
Remark	DatabaseException class will report error message via DatabaseException(message)

evenProt.java

Program	evenProt.java
Usage	This program will extract interact pairs and evenly distributed them to training and testing file based on input file with info of protA/protB and total count. Odd number line to 'train' output file Even number line to 'test' output file Single records to another file - 'single' output file

	Usage java evenProt filename [protA/protB] type table where filename is the file that contains proteinID and count protA/protB is the column in interact table for proteinID type is the [P/N]
Algorithm	o connect to database o if uniquely found, write to single file o else if ((count%2)==1) odd number, write to train output file o the rest to test output file
Remark	

extractS.java

Program	extractS.java
Usage	This program will take in single ac and extract interacting partners from interact table in database. These data is then converted to the comma-delimited format in a file with a 8 characters long separator (including the item). i.e., A,B will become A<7 spaces>B. This is to align with SVM output file format. Output file will be filename_ex.txt Usage java extractS filename(without .txt [original has .txt]) type
Algorithm	o connect to database o take each of the ac from input file o extract interacting partners from interact table where protA=ac and protB=ac and output to output file
Remark	

format.java

Program	format.java
Usage	This program convert the comma-delimited in a file to a 8 characters long separator (including the item). i.e., A,B will become A<7 spaces>B. This is to align with Cai's SVM input file format. Usage java format filename(without .txt [original has .txt]) Output file will be filename_ok.txt
Algorithm	o read each line of the input file o convert the comma to the space and ensure that the total character count is 8 before the second item. o Output the converted format to output file
Remark	

genInfo.java

Program	genInfo.java
Usage	This program will generate the information using input file which contains the list of input dip_ac no from DIP_fasta database table. The output file has options of either data or fasta input_file - list of dip_ac output_file - list of dip_ac|sp_ac|pir_ac|gi_ac|sp_name or fasta file usage: java genInfo input_file type(f/d) – fasta and data
Algorithm	o connect to database o get dip_ac from input file o extract info (dip_ac|sp_ac|pir_ac|gi_ac|sp_name) and seq and depending on the type entered (f/d) and output relevent info to output file
Remark	

genStat.java

Program	genStat.java
Usage	This program generates statistics of the svmlight output (based on selected or best parameter generated output file). The main purpose of this info is used for publication and also analysis. Input: original protein file, i.e. A<>B<>P/N svmlight out file output: result file (orginal protein file with "_out.txt" appended)containing and details of TP/FN/TN/FP and non SV of original protein file usage: java genStat org_protein_file svmlight_outfile
Algorithm	o getInput – get the total no of record and number of positive and negative records from org_protein_file. Output the details. o countPN - extract details using information from 'getInput' function and svmlight output i.e., true positive(TP); false negative(FN); true negative(TN); false positive(FP) and output the details together with calculate of accuracy [(TP+TN)*100/total]; precision [TP*100/(TP+FP)] and recall [TP*100/(TP+FN)]. o getXSV - extract the line of test records that is not a support vector in order to build up the training set for independent test. The line no is then used to extract the list of not SV from org_protein_file to output result file for further training.
Remark	This program is a merge of countPN.java and getXSV.java programs. Those two programs are now defunct

getFasta.java

Program	getFasta.java
Usage	This program will extract all the sequences in Fasta format according to the seq_column, table name given. Usage java getFasta seq_column table_name outfile
Algorithm	o connect to database o extract dip_ac and seq from database o format to Fasta format to outfile
Remark	Main purpose - for generating shuffled sequences using shufflet and preparation of source files for sharing and reporting

getFasta2.java

Program	getFasta2.java
Usage	This program will extract interaction pair sequences from database in Fasta format. The program will read from file to get protA and protB and use them to retrieve seq from database. usage java getFasta2 inputfile outfile seq_column
Algorithm	o read protA and protB from inputfile o connect to database o extract seq_column from dip_all using dip_ac with protA and protB o append both sequences o format to Fasta format to outfile
Remark	Main purpose - for checking if the interacting pairs are homologous.

getRand.java

Program	getRand.java
Usage	This program is used to extract the random fixed no of records (set at 3500) from a list of records. This is because the total possible number generated is too big so it is necessary to get random sampling. This is for generating negative dataset from cytoplasm and nucleus subcellular types. Input: input file containing total number of records output: output file with randomly selected fixed no of records usage: java getRand total_file fixed_no
Algorithm	o Read in input file and store in array o Use java random function to random select 3500 records o Output to getRand.txt
Remark	

getRep.java

Program	getRep.java
Usage	This program is used to extract the representative of training dataset. input: blastclust output file and either vector or src file as another input. output: 2 files that contain representative seq (appended with _rep) and homologuous seq (appended with _xrep) files usage: java getRep org_protein_file (svmlight vector or original sequence) blastclust_out
Algorithm	o read blastclust_out file and get representing seq no. For the list of homologous seq, only the first seq no is selected as representative. o Read the org_protein_file and extract row no the same as the seq no o Output match row to _rep file and unmatch to _xrep file
Remark	Main purpose is to improve the classification and remove redundancy in sequences

getSV.java

Program	getSV.java
Usage	This program is used to extract the SV and nonSV of training set. The program will take the 'alpha' file after svm_learn and compare with the source file. The line no that is not zero is SV and zero is not. Input: original protein file, i.e. A<>B<>P/N svmlight alpha output:result file containing SV and nonSV of original protein file usage: java getSV org_protein_file svmlight_alpha
Algorithm	o read in svmlight alpha file and store the line no that is 'zero' in array. Keep the other value in another array o read in original profile file – line no found output to nonSV file. Line no not found to be output together with alpha value to SV file
Remark	Main purpose is to improve the classification and independent test

ioFasta.java

Program	ioFasta.java
Usage	This program will either get sequence from database and generate output in Fasta format or get the sequences from a Fasta file to insert into database. The Fasta file format is >id followed by sequence. usage java ioFasta type column <input_file>

	where 1. type (mandatory): i : input - need input_file to get sequences to b inserted into database o : output - generate fasta out file from database 2. column (mandatory when type is 'i') column name to insert the seq 3. input_file (mandatory when type is 'i')
Algorithm	o connect to database o depending on type – 'I' – readFasta module to read fasta file and update column in dip_all o 'O' – extract all the sequence (when not null) from dip_all and output into Fasta file format
Remark	Mainly used in managing shuffled sequences dataset (currently table is hard-coded as dip_all. To be enhanced to any table)

randomSeq.java (and is variant randomSeq2.java)

Program	randomSeq.java
Usage	This program generates a random string based on an input string from database. The randomness is within the position of the string (type 1). The result will be of same length but it is not really a random of amino acids seq. 3 type of random seqs generation: 1). randomStr - randomized position of amino acids 2). randomStr2 - internal randomised the amino acids position - i.e., the aa composition is the same. 3). (not implement) randomized amino acids according to the length
Algorithm	o Connect to database o Get seq, ID from DIP_fasta where seq is not null o Randomize the seq and update the new seq to random_seq (using randomStr) and random_seq2 (using randomStr2) o randomStr – for the whole length, get a random number in between and substring the position from the original. It is possible to have the same position appearing again so the aa composition is not the same. o randomStr2 – store the whole sequence in an array and swap the position of the seq x times. Getting x from rand function (between 50-100). The swapping is also done by finding 2 random numbers using rand function and swapping with these 2 positions. o Rand – this function is using java – rn.nextInt()%n to find the random number between a hi and lo with n=hi-lo+1. After getting the random number i, return lo+i as the final random number between the hi and lo.
Remark	Implement to get database configuration from config.txt randomSeq2.java is the precursor which generates a random string based on an input string. The randomness is swapping the position of the character in the string so composition of the string remains the same. The minimum swapping chance is 50 and max is 100.

readFasta.java

Program	readFasta.java
Usage	This program will read in Fasta file using input stream reader instead of buffer reader. This will improve performance.
Algorithm	o Read in file using InputStreamReader o Convert the buffer to string and access the string via 'toString()' function
Remark	This program is an embedded program for any main program which has Fasta file processing .

readLine.java

Program	readLine.java
Usage	This program output number of line of the input file Usage java Splitfile filename
Algorithm	o Read in file o Count the no of lines and output the result to the screen
Remark	

Splitfile.java (and its variant Splitfile5.java)

Program	Splitfile.java
Usage	This program will split a file 'filename (without .txt)' content to 2 or 3 files depending on no. entered in the second argument. If 2 is entered, 2 files created, 'filename'a.txt and 'filename'b.txt If 3 is entered, 3 files created, 'filename'a.txt, 'filename'b.txt, 'filename'c.txt Usage java Splitfile filename(without .txt [original has .txt]) no (split to 2 or 3) Usage java Splitfile5 filename [no split number is required. Will split the input file equally to 5 portions]
Algorithm	o Read in file o If the args[1] is 2, Even number line (linecount%2=0) content to filename'a.txt and odd number line (linecount%2=1) content to filename'b.txt o If the args[1] is 3, (linecount%3=0) content to 'filename'a.txt; (linecount%3=1) content to 'filename'b.txt; (linecount%3=2) content to 'filename'c.txt
Remark	

The details of directory: java\feature

This directory is used to generate SVM feature.

genfeature.java

Program	genfeature.java
Usage	The main purpose of this program is to generate the feature vectors of a test protein given in a file to pair with all proteins (in feature vectors form) from another input file to form the feature vector file for classifying purpose. This program will read in 2 files: 1. seqfile - file that stores the seq of test protein 2. protfile - file that contains the feature vector of all the proteins program will convert the seq in seqfile to feature vector and attach it to every proteins in the protfile. Since the protfile will contain feature from 1 to 145, the newly calculated seq will be from 146-290. USAGE : java genfeature seqfile protfile feature outfile
Algorithm	o Read in file o Connect to database o Generate seq from seqfile to feature vector (similar algorithm as svm_feature.java) o Create outfile o Get each feature vector from protfile and pair it with the newly generate feature vector of the given protein and output to outfile
Remark	This program is written to generate the SVM input file to test how good the system can identify the potential partners of a given protein.

qc.java

Program	qc.java
Usage	This program will quality check the input file of svmlight (original data file) to ensure that all the protein ID is valid so that svm_feature will generate correct vector file input : input file of svmlight output: valid file invalid file with comments java qc inputfile table column
Algorithm	o Read in file o Connect to database o Extract both protA and protB from input file o Check if both existed in the database, if so, write to valid file; if not, write to invalid file.
Remark	

qcdata.java

Program	qcdata.java
Usage	This program compares original data file and generated feature vector file to find missing protein id from original data file. Each line of the feature vector file is searched and those without "146:" are treated as incomplete line. Complete line has 290 features. Input: original protein file, i.e. A<>B<>P/N feature vector file output: original protein file - valid and invalid files usage: java qcdata original_protein_file feature_vector_file
Algorithm	o Read in feature vector file o Find any line that does not contain “146:” and store in array. o If the line found in original protein file, write to invalid file; if not, write to valid file.
Remark	

qcdata2.java

Program	qcdata2.java
Usage	The program is used to quality check the feature vector file to ensure that only complete record is found. some records are incomplete, only until 145 as the other protein is missing. (due to new table used - dip_all instead of the old table - dip_fasta) usage: java qcdata2 svmlight_feature_file
Algorithm	o Read in feature vector file o Find any line that does not contain “146:”. If the line found, write to _ok file; if not, write to _nok file.
Remark	This program is written to avoid the need to run svm_feature again for the original protein file.

sp_feature.java

Program	sp_feature.java
Usage	This program converts 'seq' field from table specified to feature vector. If there is a need to selectively convert part of the seq in the table, then the list of protein_id (most likely sp_ac) can be provided in infile. The feature generation is similar to svm_feature.java. USAGE : java svm_feature infile table feature outfile
Algorithm	o Read infile and parameter (table) o Connect to database o Extract both seq from specified table based on infile selection o Convert the selected seq to feature vector and prefix each feature vector generated with '+1' to test whether the generated protein pair is interacting before output to outfile
Remark	Outfile is the input file (protfile) of genfeature.java

svm_feature.java

Program	svm_feature.java
Usage	This program converts generated SVM files (in cai's SVM format, protA<>protB<>type) - train and test to fit in to SVM light format - <class> <feature>:<value> ... <feature>:<value> where <class> is +1;-1 <feature> is integer which represents the amino acid (aa) composition based on Ding and Dubchak using frequency; distribution of both sequences in database. <value> is a real number which represents the calculated value of each features USAGE : java svm_feature file table id type feature table2 id2 ntype where file - contains interaction pairs file [without .txt] table - database table that contain the seq information id - the unique database field type - to identify if the id is a int or string - 'i' or 'c' feature - 1: aa composition 2: hydrophobicity 3: Van der Waals 4: Polarity 5: Polariability 6: Charge 7: Surface Tension table2 - additional databse table for negative dataset id2 - additional database field for negative dataset ntype - if the negative dataset should be randomized - used 'r' else using 'n' as normal
Algorithm	o Read in file o Extract seq from database (depending on ntype, if 'n' then get seq else if 'r', get random_seq or random_seq2 from DIP_fasta) o Calculate aa composition of each seq using writeAACom o Calculate feature using writeFeature o Convert P/N to +1/-1 and add count to each feature depending to number of features entered using addCount. This is to format the feature list to fit SVMlight format o writeAACom – get each AA and count its total appearance and calculate its percentage o writeFeature – each feature is separated into 3 groups. Each group percentage distribution in 1%, 25%, 50%, 75% and 100% of the seq is calculated as the feature distribution. Percentage of group transition with each other is calculated as the feature transition.

The details of directory: java\shuffleSeq

This directory contains the programs to manipulate the shuffled sequence generated by shufflet program before inserting the sequences into the database.

exFasta.java

Program	exFasta.java
Usage	This program will extract the shuffled sequences to individual file. The input file is the Fasta formated file generated from shufflet program. There are many times to shuffle a sequence and each shuffled seq is represented by '_SHF?' (where ? is the no of time the seq is shuffled or no of trials) after the key. For example, all _SHF1 will be extracted to inputfile_1.txt. usage java -Xms256m -Xmx256m exFasta inputfile no where inputfile : file that contains the shuffled seq no : total no of times shuffled
Algorithm	o Read in shufflet generated file using readFasta and tokenization o Based on the no of shuffling (x), extract sequences of each shuffling by finding name where "_SHF" x is found in the id_no. o Output the id_no and sequence of the same shuffling to each individual file
Remark	must use the -X option or else will hit out of memory

shuffleSeq.java

Program	shuffleSeq.java
Usage	This program generates a shuffled sequence conserving the exact k-let counts for a given k. It is sampled uniformly from all the valid permutations. java shuffleSeq k
Algorithm	o Connect to database o Get every sequence from dip_fasta o For k-let shuffling of each sequence, find 2 segments of sequence where aa_start=aa_end where aa_end is k+1+aa_start position. o Swap the 2 segments o Move on to next segment after aa_end
Remark	Not advisable to use the program as it is not comprehensive enough. There is no need for aa_start=aa_end but as long as first k-1 = aa_start and last k-1 = aa_end.The main reason of writing the program is because the Shufflet program was not available after a few attempts to get from the author. This is an attempt to generate shuffled seq using the algorithm published in Kandel et. al.[55] . However the program may not have considered all situation like Shufflet[56] and hence it is better to use Shufflet (author responsed after 3 weeks) as it has been used in the original protein-protein interaction paper using SVM[18]

The details of directory: java\weight

This directory and programs created to introduce reliability index (RI) into predicted result so that the system can be applicable to the real world. The RI can be calculated from distance of a given point to the OSH, optimal separating hyperplane.

distance.java

Program	distance.java
Usage	This program is used to calculate the distance of a given point to the OSH, optimal separating hyperplane. The distance can then be used to rank the reliability of the prediction. $d=\sum\alpha_i y_i K(\mathbf{x},\mathbf{x}_i) + b$ where $K(\mathbf{x}_i,\mathbf{x}_j)$ [kernel] = gamma*$e^{(-\|xj-xi\|2)}$ The reliability index (RI) is calculated as (distance/0.2). However this is not ready calculated in the program. The program output list of distance together with its identifier in filename of sv_file with '_w' as suffix. usage: java -Xms64m -Xmx128m distance model_file sv_file no_of_sv no_of_feature predicted_no_sv b gamma where model_file : model file generated by svmlight training dataset (containing the feature vectors of all the support vectors) sv_file : feature vectors dataset of proteins for prediction no_of_sv : number of support vectors from learn output file no_of_feature : total dimension of feature vector predicted_no_sv : total number of correctly predicted dataset b : threshold generated by svm_learn gamma : one of the parameter in Gaussian kernel (the one that generated the best accuracy)
Algorithm	o Read in all files o Extract alpha and feature from model file and store in x_i array o Extract feature value from sv file and store in x array o Calculate the kernel value (gamma*$e^{(-\|xj-xi\|2)}$) using x and x_i and gamma for each point o Calculate distance for each point using the formula $d=\sum\alpha_i y_i K(\mathbf{x},\mathbf{x}_i) + b$
Remark	must use the -X option to solve OutOfMemoryError To run program with an initial heap size of 64Mb and allow this to increase to 128Mb if needed. The mx option actually sets the maximum heap memory (not stack) while the associated ms option sets the initial heap size. The idea is extracted from JMB (2001) 308, 397-407 by Sujun Hua & Zhirong Sun: p. 400 - the absolute value of distance(I) is in the interval [0,2]. RI (reliability index) is 0 when distance(I) < 0.2 INTEGER(distance(I)/0.2) if 0.2 <= distance(I) < 1.8 9 if distance(I) > 1.8

Appendix C: Implementation details

The italicized word represents table name in database and details of table can be referenced in Appendix A while word with suffix '.java' representing java program used (Appendix B):

1. Build the interaction database : Get DIP core database from http://dip.doe-mbi.ucla.edu/dip/Download.cgi?SM=4 version 04/04/2003 and store it in *core*. The interaction dataset in this database has been validated by Expression Profile Reliability (EPR) index and Paralogous Verification Method (PVM).

2. Build the protein database 1 – basic sequence database : Get DIP database from http://dip.doe-mbi.ucla.edu/dip version 01/06/2003. This data file is used to create tables *dip_all* (protein information) and *Interact* (interaction information) so as to correspond to the validated dataset (*core*) in step 1. The data file to database conversion is done by convertFasta.java.

3. Build the protein database 2 – merge InterPro protein domain information to DIP data for protein function reference:
   - Check BIND database[78] but no readily available domain data with interaction information as at 30/08/2001
   - As Pfam[89] dataset format is harder to extract and InterPro is a more complete resource for domain information, InterPro data protein2ipr.dat from ftp://ftp.ebi.ac.uk/pub/databases/InterPro version 05/12/2001 is downloaded to a table - *interpro*.

- As InterPro can be linked using Swiss-Prot accession no but DIP data is mainly represented by PIR and GI accession no, there is a need to update the DIP data with Swiss-Prot accession no before it can link with InterPro.
    a. Download Swiss-Prot data from ftp://ncbi.nlm.nih.gov/blast/db/swissprot.z (swissprot.021201 version) to table *swissprot*.
    b. However since not all gi_ac in *DIP_fasta* (same structure as *dip_all* but downloaded on 05/09/2001) can be found in *swissprot* table. For those that can be found, a program is written to match the record – matchsp.java.
    c. As the result, for all the unmatched record – use Swiss-Prot name available in *DIP_fasta* to get from online database, i.e., http://tw.expasy.org/cgi-bin/sprot-search-de?sp_name where sp_name is the field data in *DIP_fasta*.
    d. After the above processing, out of the 5943 records in *DIP_fasta*, 4135 records has sp_ac (Swiss-Prot accession no) while 1808 records cannot be further processed as there is no link between the InterPro database and the protein.
    e. DIP_interpro.java is written to fill up interpro and interpro_count field in *DIP_fasta* table by using sp_ac as a link between *DIP_fasta* and *interpro* table. SetType.java and EvenType.java are used in assigned proteins with 'R' (training) and 'T' (testing) type in 'flag' field of *DIP_fasta* in order to aid in dataset construction previously. This dataset construction method is aborted later because it is only limited to protein with InterPro assignment.

4. Construct positive dataset : The interacting proteins pairs in *core* dataset are extracted for further validation to remove homologous sequence. The list of protein pairs are checked using qc.java and the combined protein sequences (sequence of protA + sequence of protB) are retrieved using getFasta2.java program. The fasta file generated is then used as an input to BLASTCLUST[1,86] to retrieve list of proteins pairs with identity 30% and sequence coverage of 90% (blastclust -i train_fasta2.txt -o train2_out30.txt -p T -S 30). The output file of BLASTCLUST is then used to extract the representative for training dataset using getRep.java which select the first pair of the homologous matched group as the representative. The protA from the resulting protein pairs list is used to find all proteins with multiple partners and evenProt.java is used to split the protein pairs to training and testing dataset. The unique protA (without multiple partners) are selected to retrieve list of protB with multiple partners. This set of protB is split similarly using evenProt.java to obtain a training and testing dataset. The two set of training and testing datasets generated are used to form the final training and testing positive dataset after eliminating the duplicate.

5. Construct negative dataset : The generated positive dataset is used as the basis for negative interaction dataset construction. The idea is based on subcellular localization exclusion as proteins which localize in different areas of the cell are unlikely to interact together. Proteins with multiple localization are omitted in this study. The yeast subcellular localization data is extracted from MIPS[60] (downloaded on 21-04-2004) (http://mips.gsf.de/genre/proj/yeast/searchCatalogFirstAction.do;jsessionid=D

FA24BA3FA8167B5AAE843F08D6A074B?db=CYGD). Four types of localizations are considered – nucleus; cytoplasm, endoplasmic reticulum (ER), mitochondria. Out of the positive dataset, a total of 718 proteins are found with unique subcellular localization (nucleus – 304; cytoplasm – 318; ER – 30; mitochondria – 66). As the combination for negative dataset is huge, getRand.java is used to randomly pair up the following :

protA	**protB**	**Total**
ER	cytoplasm + nucleus	304 + 318
mitochondria	cytoplasm + nucleus	304 + 318
ER	mitochondria	66
cytoplasm	nucleus	3500

As the result, there are 4810 records. This set of record is quality checked to remove duplication within the group and from all the available DIP yeast interaction dataset (using *dip_all*). The resulting dataset has 4662 records and the file is evenly split using Splitfile.java to a training and testing negative dataset of 2331 records each.

6. Construct training and testing dataset : The generated negative and positive dataset are split according to Step 4 and 5 above before forming the training and testing dataset with a positive and a negative components. The training dataset is further quality check by removing homologous sequences with 30% identity. The training dataset is first extracted using getFasta2.java to append two sequences of the protein pairs of each record into Fasta format. The resulting Fasta file containing combined sequences of the protein pairs are checked using BLASTCLUST to remove protein pairs with identity 30% and sequence coverage of 90%.

7. Generate shuffled sequence : Shufflet program[61] is used to generate shuffled protein sequences for construction of negative dataset. The program is installed in Linux and the same set of sequences in the negative dataset, as mentioned in Step 5, is shuffled in two modes. Command 'shufflet 1 1 < fasta.txt > 1let.txt' is used to generate 1-let shuffled sequences and 'shufflet 1 2 < fasta.txt > 2let.txt' is used to generate 2-let shuffled sequences where fasta.txt is the negative dataset sequence in FASTA format. The 1let.txt and 2let.txt files generated are inserted to *DIP_fasta* and *dip_all* table using ioFasta.java program.

8. Generate feature vectors : svm_feature.java is written to generate the feature vector of each protein sequence based on feature representation method mentioned in section 3.2. The output file format is catered to SVMlight input format. qc.java and qcdata.java are used to quality check the generated feature vector file to ensure that feature vectors of the protein pairs is generated successfully.

9. Train SVMs : The same set of positive dataset constructed in Step 4 are matched with 3 set negative training and testing dataset generated in Step 5 and Step 7 to train three SVMs using SVM^{light} program – svm_learn. The list of datasets and generated models can be found in Appendix E. The output generated by SVM is analysed using genStat.java.

10. Generate putative interaction protein partners : As our research main focus in on human protein, a total of 7,985 Swiss-Prot human proteins (extracted from *swissprot* and *sp* tables) are used to pair with the protein of interest.

genfeature.java and sp_feature.java are used to convert the protein pairs to corresponding feature vectors.

11. *D. melanogaster* interaction dataset: The dataset is downloaded from DIP website on 23-09-2004. The total interaction protein pairs are 20,988. format.java, qc.java, qcdata2.java are used to quality check the dataset with *dip_all* table. Svm_feature.java is then used to generate feature vectors for the list of protein pairs.

12. Analyze prediction result : Even though SVM^{light} provides prediction accuracy; recall and precision measure as the output result, it is of interest to understand which are the support vectors; which are the correctly or wrongly predicted and lastly, how reliable is the prediction. A set of java programs are written for the purposes, they are getSV.java, genInfo.java, distance.java respectively. In order to validate the result, five-fold cross validation is used. The training dataset of the three classification systems is split to 5 portions using Splitfile5.java, each portion are used as the testing dataset to the remaining four combined dataset. Besides that, ROC graph is used to compare the classifiers. The performance is measured by calculating the area under the ROC curve.

Appendix D: Calculation details

Cross validation is calculated based on standard deviation (sd) sampled from five portions of training dataset. Each training dataset are split to five portions – a, b, c, d and e.

$$sd = [\sum(xi-u)^2/N]^{1/2}$$

Where xi is the accuracy from each test; u is the average of the accuracy and N is the total number of test, i.e., 5.

Table D-1 Five–fold cross validation results (Shuffled sequence (1-let)):

Train dataset	Test dataset	Accuracy (%)	Gamma	xi-u	(xi-u)^2
train abcd	train e	96.42	0.00045	0.29	0.08
train abce	train d	96.09	0.00035	0.04	0.002
train abde	train c	97.04	0.00045	0.91	0.83
train acde	train b	95.31	0.00025	0.82	0.67
train bcde	train a	95.81	0.00035	0.32	0.1

Total : 480.67; Average : 96.13

sd = [1.682/5]^1/2 = 1.3

Table D-2 Five–fold cross validation results (Shuffled sequence (2-let)):

Train dataset	Test dataset	Accuracy (%)	Gamma	xi-u	(xi-u)^2
train abcd	train e	91.95	0.00045	0.1	0.01
train abce	train d	91.68	0.00035	-0.17	0.41
train abde	train c	92.69	0.00055	0.84	0.71
train acde	train b	92.13	0.00055	0.28	0.08
train bcde	train a	90.78	0.00045	-1.01	1.14

Total : 459.23; Average : 91.85

sd = [2.35/5]^1/2 = 0.68

Table D-3 Five–fold cross validation results (Real sequence):

Train dataset	Test dataset	Accuracy (%)	Gamma	xi-u	(xi-u)^2
train abcd	train e	79.88	0.00055	1.26	1.59
train abce	train d	82.29	0.00055	1.15	1.32
train abde	train c	81.02	0.00045	0.12	0.01
train acde	train b	81.46	0.00045	0.32	0.1
train bcde	train a	81.05	0.00055	0.09	0.01

Total : 405.7; Average : 81.14

sd = [3.03/5]^1/2 = 1.74

Appendix E: List of datasets and models

Dataset type	Filename	Source directory	Database table
protein-protein interaction – source file	train_subloc.txt test_subloc.txt	Material_Methods\model	table : dip_all field : dip_ac
protein-protein interaction (real sequence) – feature vectors file	train_subloc_1234567.dat test_subloc_1234567.dat	Material_Methods\model	table : dip_all field : dip_ac, seq
protein-protein interaction (shuffled sequence) – feature vectors file	train_subloc_1234567_s1.dat test_subloc_1234567_s1.dat train_subloc_1234567_s2.dat test_subloc_1234567_s2.dat	Material_Methods\model	table : dip_all field : dip_ac, shuffled_seq1 (s1), shuffled_seq2 (s2)
SVMlight generated model	model_t00035_real model_t00035_s1 model_t00055_s2	Material_Methods\model\svm_model	NA
Thioredoxin dataset	thioredoxin.zip	Results	NA
D. melanogaster dataset	species.zip – check for all files with 'fly'	Results	NA

LIST OF PUBLICATIONS

I. Publication in International Peer-reviewed Journal:

Lo, S. L., Cai, C. Z., Chen, Y. Z., Chung, M. C. M. (2005) Effect of training datasets on support vector machine prediction of protein-protein interactions. *Proteomics* **5**(4), 876-884

II. Presentation at International Scientific Conference:

Lo, S. L., Cai, C. Z., Chen, Y. Z., Chung, M. C. M. Effect of training datasets on support vector machine prediction of protein-protein interactions. Poster no: PA19. Third International Proteomics Conference (IPC'03) May 14-17, 2004, Taipei, Taiwan.
Award : Travel Fellowship Award

III. Presentations at Local Scientific Conferences:

Lo, S. L., Cai, C. Z., Chen, Y. Z., Chung, M. C. M. Effect of training datasets on support vector machine prediction of protein-protein interactions. Poster no: 92. Third International Conference on Structural Biology and Functional Genomics, December 2-4, 2004, Singapore

Lo, S. L., Cai, C. Z., Chen, Y. Z., Chung, M. C. M. Rules governing dimer interaction using Support Vector Machine. Poster no: 52. Second International Conference on Structural Biology and Functional Genomics, December 2-4, 2002, Singapore.