11

Computational Approaches to Predict Protein Interaction

Darby Tien-Hao Chang
National Cheng Kung University
Taiwan

1. Introduction

Recently there have been large advances in high-throughput experimental approaches to identifying protein interactions. However, these experimental verified interactions still account for a small proportion of the complete interaction network. For example, based on current understanding (Stumpf, Thorne et al. 2008), less than 10% of interactions of the human protein interaction network (PIN) are identified and collected in the Human Protein Reference Database (HPRD) (Peri, Navarro et al. 2003; Stumpf, Thorne et al. 2008). The low coverage can be complemented by the computational approaches methods to predict protein interaction. This chapter describes approaches based on different biological observations and/or different computational techniques. Another focus of this chapter is to highlight the importance of creating a benchmark - especially negative samples since there are very limited techniques developed to confirm that two proteins do not interact (Doerr 2010; Smialowski, Pagel et al. 2010) - in evaluating computational approaches.

Computational methods can be roughly divided into two categories. Methods in the first category utilize the observation that functionally related proteins have patterns of co-occurrence, such as co-evolution or co-expression; while methods in the second category compile proteins into features potentially related to protein interaction, such as protein surface area, and resort to machine learning (ML) techniques for prediction. Different co-occurrence-based methods are distinct in where, namely which biological properties, the co-occurrence is observed and in the implementation details to record the co-occurrence. For example, Salgado et al. suggested that some related genes are close in genome to make the transcription more efficient (Salgado, Moreno-Hagelsieb et al. 2000). Methods based on this observation utilize co-localization in genome and could, for example, use the distance between two genes to record the co-occurrence. Section 2 will introduce seven categories of co-occurrence as follows.

1. **Genomic location**—some genes producing proteins that will interact are close in genome to facilitate transcription;
2. **Cellular compartment**—interacting proteins should appear in the same area in a cell to interact, another co-localization pattern which consider the cellular location;

3. **Phylogenetic tree**—if one protein was mutated in evolution, its cooperating protein should have a corresponding mutation to keep their interaction/function and thus the species survival, i.e. cooperating proteins should have similar phylogenetic trees;
4. **Existence in close species**—if two proteins co-work for a function to a species, then the species will have both of them, otherwise the species will have none of them, i.e. some related proteins are present in/absent from species together;
5. **Interacting domains**—interacting proteins usually have complementary parts of a interacting domain pair;
6. **Literature**—related proteins, since there must be some papers describing their relations, are prone to be mentioned together in literature, as opposed to other proteins existing only in articles describing their individual characteristics;
7. **Gene fusion**—some interacting proteins whose homologues form a fused protein chain, a special biological phenomenon named Rosetta Stone protein.

Different ML-based (or feature-based) methods, however, may share partial features to previous studies but develop new features at the same time. This led to more complicated relations than that among co-occurrence-based methods. For example, Shen et al. (Shen, Zhang et al. 2007) proposed to use a composition of short sequences as protein features and a following work by Chang et al. (Chang, Syu et al. 2010) combined these features with protein surface information. In addition to the overlap of features among different ML-based methods, they may use identical or different ML techniques. Using the two ML-based methods as an example, Shen et al. chose the widely used support vector machine (SVM) (Vapnik and Vapnik 1998), while Chang et al. used a relaxed variable kernel density estimator (RVKDE) (Oyang, Hwang et al. 2005) developed by their group. Thus to keep the description structure compact, we will focus on the features in section 3. We will provide only a minimum introduction to several well-known ML techniques in section 4 since they are beyond the scope of this chapter. Knowing the concepts of these ML techniques may help to understand the design of different ML-based PPI predictors and to select appropriate features. This chapter roughly divides features into four categories.

1. **Sequence information**—many studies extracted features only from protein sequences. Methods using only such features are very challenging but provide much applicability. Some derived features such as protein polarity (by summing the polarity index of its amino acids) are also included in this category.
2. **Evolution information**—features involving alignment with other sequences fall into this category. Methods using such features usually require a collection of protein sequences of many species.
3. **Structure information**—methods of this category can perform geometry and even energy analyses. Many useful features such as protein surface, secondary structure and binding affinity can be derived. These methods are usually time-consuming, where researchers will expect to obtain extremely accurate predictions.
4. **Auxiliary information**—some studies used auxiliary information such as function annotation. These studies usually used such features to analyze rather than to predict protein interactions, since some features were manually curated. It was hard to perform a fair comparison with other methods not using such features.

After the features used in recent ML-based methods there is an introduction to three well-known ML techniques.

1. **Decision tree**—a time-honored tool, which is less accurate than modern ML tools but preferred by many biologists because its learning model is more interpretable to human;
2. **SVM**—a state-of-the-art tool that overwhelmingly prevails in the field of computational biology because of its accuracy;
3. **RVKDE**—another modern ML tool that solves the most critical problem of SVM, unacceptable execution time on large data, by slightly sacrificing accuracy.

This chapter ends up with the important issue of computational approaches - evaluation. Computational approaches of identifying protein interactions have a fateful difference to experimental approaches. That is, their results are considered as "prediction" rather than the answer. So it is an inevitable step for the studies of computational methods that they must test their algorithms and report the prediction accuracy compared to a benchmark with the answers already known.

As a summary, this chapter will first introduce the concept of co-occurrence pattern and the implementation details of some co-occurrence-based methods. For the ML-based methods, this chapter focuses on the features and a little on the ML techniques. Finally, three contradictions are used to describe to readers the importance of evaluating these computational methods and explain how to interpret the accuracy they see in literature.

2. Co-occurrence-based approaches

This section introduces seven concepts of co-occurrence patterns that have been adopted to predict protein interactions. An identical concept, based on the available materials, may have different implementation details. In this section, the concept of each co-occurrence pattern is first introduced followed by the implementation details of several methods as examples of that co-occurrence pattern.

2.1 Genomic location

The advance of sequencing leads to the opportunity not only of identifying the genomic locations of genes, but also of analyzing genomic context to predict interactions between genes (Huynen and Snel 2000). The genomic location, also known as genomic context, co-occurrence pattern relies on the fact that operons and some adjacent genes are likely to encode functionally related proteins (Rogozin, Makarova et al. 2002). Huynen and Snel proposed a method to assess the probability that two genes occur as neighbours in a genome only by chance (Huynen and Snel 2000). They randomized the genes in each genome over the loci in that genome. The expected number of the co-occurrences of two genes, namely those that occur as neighbours, in the randomized genomes was less than one. A functional interaction between genes was inferred if the observed number of co-occurrences is significantly higher than the expectation. Rogozin et al. proposed a procedure to compare the orders of orthologous genes (Rogozin, Makarova et al. 2002). They clustered genes into orthologous groups which were then projected onto genomes to identify the neighborhood genes. The results show that the gene neighbours have good functional coherence.

2.2 Cellular compartment

Proteins that occur in different cellular compartments are, in principle, considered not to interact since they do not have the chance to meet. However, some *in vitro* experiments such as tandem affinity purification-mass spectroscopy (TAP-MS) method (Krogan, Cagney et al. 2006), might report such interactions of two proteins in different cellular compartments. It is difficult to determine if these *in vitro* interactions are correct. Thus, this co-occurrence pattern is usually used to increase the prediction reliability of another method or to generate a reliable benchmark rather than an individual interaction predictor. For example, the Eukaryotic Linear Motif (ELM) server used cellular compartment information as a filter to double-check gene function (Davey, Van Roey et al. 2011). The gene function, represented by its Gene Ontology (GO) terms (Ashburner, Ball et al. 2000), was required to be consistent with its cellular compartment. Guo et al. used cellular compartment information to build the negative data of protein interaction (Guo, Yu et al. 2008). They assumed that proteins that occur in different cellular compartments do not interact. They grouped proteins into eight subsets based on the eight main types of cellular compartment—cytoplasm, nucleus, mitochondrion, endoplasmic reticulum, golgi apparatus, peroxisome, vacuole and cytoplasm and nucleus. The negative samples of non-interacting pairs were generated by pairing proteins from different subsets.

2.3 Phylogenetic tree

The phylogenetic tree was proposed to reflect the evolution information. Thus, the similarity between phylogenetic tress provides a good measure of gene co-evolution. Interacting proteins usually co-evolve since mutations in one protein led to the loss of function or a compensation mutation of the other protein to preserve the interaction (Walhout, Sordella et al. 2000). Jothi et al. proposed the MORPH, an algorithm to search the best superimposition between evolutionary trees based on the tree *automorphism* group in 2005 (Jothi, Kann et al. 2005). The search was done by Monte Carlo algorithm that probes the search space of all possible superimpositions, which is computationally intensive. In graph theory, two trees are isomorphic if there is a one-to-one mapping between their vertices (genes) and edges (interactions). Jothi et al. extended this definition to automorphic whereby a tree is isomorphic to itself. The search space was largely reduced to the automorphism group of a phylogenetic tree. The same group proposed another method to assess the degree of co-evolution of domain pairs in interacting proteins in 2006 (Jothi, Cherukuri et al. 2006). Multiple sequence alignments of two proteins/domains to a reference set of genomes were used to construct phylogenetic trees and similarity matrices. The degree of co-evolution of two domains was then estimated by the correlation coefficient of the two corresponding similarity matrices.

2.4 Existence in close species

The co-occurrence pattern of the existence in close species, known as phylogenetic profile, is based on the fact that functionally related proteins usually co-evolve and have homologues in the close genomes (Snitkin, Gustafson et al. 2006). A phylogenetic profile of a gene is a vector, representing the presence or absence of homologues to that gene across a collection

of reference organisms. There are two major components in a phylogenetic profile-based method: i) how to construct a phylogenetic profile of a given gene and ii) how to determine the similarity of two phylogenetic profiles. First, the presence or absence of homologues can be determined by sequence alignment scores, such as a BLAST (Altschul, Madden et al. 1997) E-value, with a threshold of presence (Sun, Xu et al. 2005). Such binary vectors were improved as real valued vectors of normalized alignment scores without arbitrarily determining a score threshold (Enault, Suhre et al. 2003). Second, any similarity or distance function between two vectors can be used to define the similarity of two phylogenetic profile vectors. Enault et al. have examined two Euclidean-like distance funtions and another two correlation coefficient variants (Enault, Suhre et al. 2003). They concluded that inner product, shown as follows, is a good indicator in predicting *Escherichia coli* protein interactions.

$$Sim(i,j) = \frac{\sum_{k=1}^{n} R_{ik} \times R_{jk}}{\left[\left(\sum_{k=1}^{n} R_{ij}^2\right) \times \left(\sum_{k=1}^{n} R_{jk}^2\right)\right]^{1/2}}$$

2.5 Interacting domains

Proteins usually depend on a short sequence of residues to perform interactions with other molecules. The functional short sequences between two interacting proteins form the contact interfaces, also known as interaction sites (Sheu, Lancia et al. 2005). These interaction sites are usually represented by domains/motifs. Li et al. proposed a method to detect interaction sites, which required only protein sequences (Li, Li et al. 2006). They developed an efficient itemset mining algorithm that can identify the most conserved motifs within two interacting protein groups. Here interacting protein groups indicate two groups, *A* and *B*, of proteins where all proteins in group *A* interact with all proteins in group *B*, denoted an *all-versus-all* interaction network. The conserved motifs within group *A* were considered, in principle, related to the conserved motifs within group *B*. The identified interacting motif pairs can be then used to predict novel interacting proteins. Tan et al. proposed a method, D-STAR to find correlated motifs that were overrepresented in interacting protein pairs (Tan, Hugo et al. 2006). The basic idea of D-STAR is to check all possible (l, d)-motif pairs, where (l, d) indicate an alignment of length l with at most d mismatches. Tan et al. speeded up the brute force procedure by transforming the problem into a clique-finding problem (Pevzner and Sze 2000).

2.6 Literature

Owing to the advance of Internet technologies, the scale of public accessible biomedical literature has increased astonishingly in the last decade. Text mining tools are critical to maximize the usage of such a large-scale knowledge base. Extracting protein interactions from literature is generally categorized as *relationship* mining, which aims to detect co-occurrences of a pair of entities of specific types (such as gene, protein, drug or disease) to a pre-specified relationship (such as interact, regulate, activate or inhibit) in the same article (Cohen and Hersh 2005). Albert et al. proposed a method to retrieve abstracts reporting

nuclear receptors (NRs) (Albert, Gaudan et al. 2003). The retrieved data were reviewed manually. Albert et al. generated a dictionary focusing on NRs, cofactors and other NR-binding proteins of human, mouse and rat. The extraction process as follows was performed on MEDLINE abstracts: i) identify abstracts with at least one NR in the generated dictionary, ii) tag entities (proteins) and relationships (interactions) according to the generated dictionary and iii) extract sentences contains two tagged proteins and a tagged interaction. In the current genomic era, the text-minded information is widely applied in database annotation. Many popular protein interaction databases such the Database of Interacting Proteins (DIP) database (Salwinski, Miller et al. 2004) and the Search Tool for the Retrieval of Interacting Genes (STRING) database (Szklarczyk, Franceschini et al. 2011) included automatically extracted literature information as an additional line of evidence.

2.7 Gene fusion

Gene fusion is a special genomic organization whereby some interacting proteins have orthologues in the close genomes fused as a single protein (Enright, Iliopoulos et al. 1999). The fused protein is usually called a Rosetta Stone protein, thus this method is sometime called the Rosetta Stone method. This genomic organization of gene fusion is formed for efficiently transcribing related genes together, thus it is preserved evolutionarily. Marcotte et al. applied the gene fusion method on *Escherichia coli* (Marcotte, Pellegrini et al. 1999). They identified 6,809 protein pairs of which both protein sequences were significantly similar to the same protein sequence of at least a genome. More than half of these 6,809 protein pairs have been shown to be related. This method, unlike previous co-occurrence patterns, is a very specific genomic organization rather than a concept of co-occurrence. Thus, there is very limited space for the algorithm development and implementation details. For any new genome, researchers can always search for Rosetta Stone proteins first. But other methods are required since many interacting proteins are not Rosetta Stone proteins. For example, in the DIP database that deposits experimentally confirmed protein interactions, only 6.4% interacting protein pairs formed Rosetta Stone proteins (Shoemaker and Panchenko 2007).

3. Machine learning-based approaches

This chapter roughly divides features into four categories: sequential, evolutionary, structural and other. Note that the power of ML tools allows researchers to submit any features, with or without obvious biological glues to protein interaction, into a magical black box and wait for the prediction without knowing how the prediction was made. For example, amino acid composition (20 features) and number of search results in PubMed can be used as features. Namely, each co-occurrence pattern can be used as a feature—the only thing to do is designing a rule to record the pattern with one or more real numbers. So in this chapter we only demonstrate several features that have been shown to help the prediction accuracy in published articles, but cannot list all features in a category.

3.1 Sequence information

One of the most widely used data to encode proteins is their primary sequence. Methods that only rely on protein sequences have a great advantage of the wide applicability. Because such methods do not rely on other information, they are sometime called *de novo* (*ab*

initio) predictors of protein interaction. Yu et al. proposed a method that encoded protein sequences as feature vectors by considering the amino acid triads observed in it (Yu, Chou et al. 2010). An amino acid triad regards three continuous residues as a unit. However, considering all 20^3 amino acid triads requires an 8000-dimensional feature vector to represent a protein, which is too large for contemporary machine learning tools. Thus, the 20 amino acid types were clustered into seven groups based on their dipole strength and side chain volumes to reduce the dimensions of the feature vector (Shen, Zhang et al. 2007). The frequencies of the $7^3 = 343$ triads can be used to encode a protein sequence. However, such a frequency is highly correlated to the distribution of amino acids. To overcome this problem, Yu et al. proposed a significance calculation by answering the question: how rare is the number of observed occurrences considering the amino acid composition of the protein? The significances of all triads were used to encode protein sequences.

Methods based on sequence motifs/domains also fall into this category since sequence motifs are mined from protein sequences. One may notice that the co-occurrence-based methods mentioned in subsection 2.5 used similar features. In this regard, the co-occurrence-based methods use domain as features with a straightforward rule: if two proteins have interacting domains, then they are predicted as interacting. On the other hand, ML-based methods use domain as features but resort to ML tools for the final decision/prediction. Depending on the ML tools used, the decision rules could be very complicated models of, for example, non-linear equations or a combination of multiple individual components (it could be a 'sum' of multiple functions). Dijk et al. proposed a ML-based method to select relevant motifs from a set of pre-mined motifs (Van Dijk, Ter Braak et al. 2008). They first invoked the D-STAR (Tan, Hugo et al. 2006) algorithm to identify correlated motifs that overrepresented in interacting protein pairs. The vector of the presence or absence of the identified motif pairs were used to encode proteins.

3.2 Evolution information

Methods that require not only the sequences of the query protein pairs but also a collection of supporting sequences fall into this category. The supporting collection is usually from other species for calculating the conservation score. Position-specific scoring matrix (PSSM) is a widely used scheme to encode a protein sequence while considering its orthologues. For a protein sequence, PSSM describes the likelihood of a particular residue substitution at a specific position based on evolutionary information (Altschul, Madden et al. 1997). It is outputted by BLAST when aligning the query protein seuqence to a seqeucne database, e.g. the non-redundant (NR) database from National Center for Biotechnology Information (NCBI). The likelihood values are scaled to [0,1] using the following logistic function:

$$x' = \frac{1}{1 - \exp(-x)},$$

where x is the raw value in PSSM profile and x' is the value corresponding to x after scaling. Each position of a protein sequence is represented by a 21-dimensional vector where 20 elements take the likelihood values of 20 amino acid types from the scaled PSSM profile and the last element is a terminal flag. Finally, the feature vector of a residue comprises a window of positions. Chang et al. proposed a method based on the assumption that protein interactions are more related to amino acids at the surface than those at the core (Chang, Syu

et al.). They first used PSSM to encode protein sequences for surface prediction and then used the surface sequence for interaction prediction.

Espadaler et al. proposed a method that made use of conservation of protein pairs (Espadaler, Romero-Isart et al. 2005). They first collected 855 protein complexes with known three-dimensional structure with <80% sequence identity. The 855 complexes were further classified into 16 groups. In a protein complex, the distance between a residue pair from two proteins was defined as the distance of the nearest heavy atoms of the two residues. Via setting a cut-off of the contact distance, one can identify the interface of two proteins in these complexes. These identified interfaces were actually unordered sequence fragments, among which Espadaler et al. defined more than five contiguous residues a *patch*. The conservation of the patches obtained by multiple sequence alignment was considered to select the final patches. These conserved structural patch pairs can be used to predict novel protein interactions. Notice that this method proposed by Espadaler et al. also used the structure information which will be introduced in the next subsection. This also reveals that with the ML tools, combining multiple resources becomes relatively easy since it is no longer dependent on a single co-occurrence pattern.

3.3 Structure information

The most critical problem of sequence-based methods is the reliability. Conversely, researchers usually resort to structure-based methods for verification since the results delivered by structure-based methods can be visualized. Aloy and Russell proposed a method to detect interactions based on protein tertiary structures. They used empirical potentials to compute the fitness between two protein structures. Thus, success of such a method is highly dependent on the performance of the underlying potential function. The adopted potential function did not rely on model proteins, which enlarges its applicability. Aloy and Russell defined interacting residues as those having at least one i) hydrogen bonds (N—O distances ≤ 3.5 Å), salt bridges (N—O distances ≤ 5.5 Å), or van de Waals interactions (C—C distances ≤ 5 Å). Buried side-chains were excluded by filtering out residues with relative accessibility ≥10%. The identified interacting residues were used to train the empirical potentials based on a molar-fraction random state model as follows:

$$S_{ab} = \log_{10}\left(\frac{O_{ab}}{E_{ab}}\right) E_{ab} = N \frac{n_a}{\sum_{a=1}^{20} n_a} \frac{n_b}{\sum_{b=1}^{20} n_b},$$

where a and b are amino acid types, O_{ab} and E_{ab} are the number of observed/expected contacts, N is the number of analyzed residue pairs and n_a and n_b are number of residues of the corresponding types. The method of Aloy and Russell provided ranks of analyzed protein pairs so that researchers can pick the most promising prediction for further biological experiments.

3.4 Auxiliary information

Important data that is not mentioned above is microarray data, which has been broadly utilized in various biomedical problems. The Gene Expression Omnibus (GEO) database (Barrett, Troup et al. 2006) of NCBI holds more than 20 thousands microarray experiments.

A problem of microarray data is that they are usually full of noises. Soong et al. used principal component analysis (PCA) to reduce such noises (Soong, Wrzeszczynski et al. 2008). PCA is a statistical technique used to find hidden factors from observed factors, expression values in this case. Lee and Batzoglou have shown that proteins with extreme principal components are prone to participate in relevant biological processes (Lee and Batzoglou 2003). The transformation of expression values to principal components can be represented as follows:

$$PX = Y,$$

where P is a $l \times m$ transformation matrix obtained by PCA, X is a $m \times n$ matrix of the raw expression values from m microarrays and n samples while Y is a $l \times n$ matrix containing every sample's l principal components. The final feature vector of two proteins a and b was the concatenation of a's principal components, b's principal components and the Pearson correlation of both.

This section ends with a method based on literature data, which has been discussed in subsection 2.6. Demonstrating literature data in a ML-based method is to reinforce the impression that in principle any data can be used as features with appropriate encoding schemes. Thus, one can consider combining any of the features discussed in section 2 with ML-based tools. Donaldson et al. proposed an extraction procedure for identifying protein interactions in literature (Donaldson, Martin et al. 2003). They first used a parser to collect synonyms for proteins and their encoding loci. The collected protein names were then used to search the title and abstract of articles in the PubMed literature database. An article was encoded by terms it contained. The weight of each term was the *tf-idf* score (term frequency-inverse document frequency), where term frequency is the number of occurrences of the term in the document and inverse document frequency is the inverse of the number of documents having the term. Here a term was a word or two adjacent words (usually called 2-gram) that appear in at least three documents.

4. Machine learning techniques

After encoding proteins into feature vectors, the next step is to choose a ML tool to generate a model describing these feature vectors. The generated model can be used to predict novel protein interactions. Most ML tools provide a user-friendly interface, where all that researchers need to do is encode their data. The remaining task is very trivial: i) a command to train the model and ii) a command to predict with the trained model. In this regard, researchers who want to adopt ML-based methods can focus on features without caring about the ML algorithms. This section briefly lists three ML algorithms that have been used in recent studies of protein interaction, which can be considered as a basic introduction for researchers who have no idea how to choose an appropriate ML tool.

4.1 Decision tree

Decision trees are usually constructed recursively (Witten, Frank et al. 2011). The first step is to select a feature to split samples (branch the decision tree) based on the selected feature. This step divides the original dataset into several disjointed subsets, each of them can be considered as another dataset. Thus, the same procedure can be applied recursively to each

subset and the further sub-subsets. Such a recursive fashion stops at several conditions of, for example, all samples in a branch belonging to the same class or all features have been examined. The above descriptions, however, missed an important detail in decision trees: how to select a feature to branch. A trivial strategy is to select the feature that can result in the purest subsets, namely most samples in the same subset belong to the same class. Thus, a measure of set purity is required.

So far, there have been many purity measurements proposed. This subsection introduces the most fundamental one, entropy, as follows. Indeed, larger entropy indicates the less purity. Thus, negative entropy, in definition, is a measurement of purity. Many mature decision tree algorithms use variants of entropy.

$$\text{entropy}(p_1, p_2, \ldots, p_n) = -p_1 \log p_1 - p_2 \log p_2 \ldots - p_n \log p_n ,$$

where p_i is the fraction of class-i samples in the subset and n is the number of total classes. For example, suppose that a dataset has nine positive samples and five negative samples. Before any branching, the entropy of the original dataset is –(9/14)log(9/14)-(5/14)log(5/14) = 0.940, where p_1 is 9/14, p_2 is 5/14 and n is 2 (positive and negative). If after a branch, the 14 samples are split into three nodes that contain 2-3, 4-0, 3-2 positive-negative samples, respectively. The entropies of the three nodes are –(2/5)log(2/5)-(3/5)log(3/5) = 0.971, –(4/4)log(4/4)-0 = 0 and –(3/5)log(3/5)-(2/5)log(2/5) = 0.971, respectively. The total entropy of the branched tree became (5/14)×0.971+(4/14)×0+(5/14)×0.971 = 0.693, a weighted sum of the three entropies corresponding to the subset size. It is observed that the entropy decreased from 0.940 to 0.693, revealing that this operation of branch did increase the purity of the dataset. For a purity measurement, the following three conditions must be satisfied:

1. when a subset is pure (all samples belong to the same class), the measurement is zero;
2. when all possible classes appear equally, the measurement is maximized;
3. the measurements must be the same without depending on the order of branches.

The third condition requires that if a dataset is first split into two nodes of *a*-*b* and *c*-*d*, positive-negative samples and then the second node is further split into two more sub-nodes of *e*-*f* and *g*-*h*, the entropy should be the same as split into three nodes of *a*-*b*, *e*-*f* and *g*-*h* in a single branch while using another feature. Entropy is the only one function that fits all these conditions (Witten, Frank et al. 2011). This explains the high popularity of entropy and its variants in decision trees.

4.2 Support Vector Machine (SVM)

Currently, SVM is the state-of-the-art ML tool. It prevails in biomedical data because of its high accuracy. SVM first transforms the original data to a higher dimensional space with a non-linear transformation and then finds the maximum margin hyperplane to separate samples of different classes in the transformed space (Witten, Frank et al. 2011). This strategy has two advantages: i) it can generate non-linear model and ii) it prevent overfitting as the decision boundary is still linear in the transformed space. Overfitting is a critical issue in ML. It indicates that the constructed model overfit the training dataset, so that which cannot be used to predict novel data. This problem becomes more serious when using more complicated model. However, some complex data do need complicated

models to describe. Thus most advanced ML algorithms still favor complicated models and then try to solve the overfitting issue. In this regard, SVM finds an excellent balance, which can generate very complicated models depending on the adopted transformation while choosing a very simple decision, a hyperplane, which equals to a one stage decision tree of two branches.

Mathematically, SVM uses support vectors to model the transformation and hyperplane. That is the reason for the name. Transforming the original data from the sample space with a non-linear function to a new space means that a linear model (a straight line in a two dimensional space, a plane in a three dimensional space and a hyperplane in a higher dimensional space) in the new space becomes non-linear in the original sample space. For example, for a two dimensional sample $x = (a, b)$, a non-linear transformation to a three dimensional space could be $x' = (a^2, ab, b^2)$. If any ML tool finds a decision boundary in the new space, it does not look like a straight line in the original space. Notice that, in principle, any tool, such as a decision tree, could be used to make the decision in the transformed space. SVM advances in already developing a robust mathematical system with efficient optimization algorithms to find good hyperplanes.

4.3 Relaxed Variable Kernel Density Estimation (RVKDE)

The biggest drawback of SVM is the computational cost. Yu et al. reported that using SVM to perform a complete interaction analysis on human genome may take years (Yu, Chou et al. 2010). In this regard, efficient ML algorithms with acceptable accuracy are reasonable alternatives to SVM. The relaxed variable kernel density estimation (RVKDE) algorithm (Oyang, Hwang et al. 2005) has been practically used in recent interaction studies (Chang, Syu et al. 2010; Yu, Chou et al. 2010). The time complexity of RVKDE is an order faster than SVM. Furthermore, unlike other fast ML algorithms, such as decision trees, the descriptive capability of the constructed model of RVKDE is comparable to SVM.

The kernel of RVKDE is an approximate probability density function. Let $\{\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_n\}$ be a set of samples randomly and independently taken from the distribution governed by f_x in a m-dimensional vector space. RVKDE estimates the value of f_x at point $\mathbf{v}$ as follows:

$$\hat{f}(\mathbf{v}) = \frac{1}{n}\sum_{\mathbf{s}_i}\left(\frac{1}{\sqrt{2\pi}\cdot\sigma_i}\right)^m \exp\left(-\frac{\|\mathbf{v}-\mathbf{s}_i\|^2}{2\sigma_i^2}\right), \text{ where}$$

1. $\sigma_i = \beta\dfrac{R(\mathbf{s}_i)\sqrt{\pi}}{\sqrt[m]{(k+1)\Gamma(\frac{m}{2}+1)}}$;
2. $R(\mathbf{s}_i)$ is the maximum distance between $\mathbf{s}_i$ and its ks-th nearest training sample;
3. $\Gamma(\cdot)$ is the Gamma function (Artin 1964);
4. β and ks are parameters to be set either through cross-validation or by the user.

For prediction, a kernel density estimators is constructed to approximate the distribution of each class. Then, a query sample located at $\mathbf{v}$ is predicted to the class that gives the maximum value among the likelihood functions defined as follows:

$$L_j(\mathbf{v}) = \frac{|S_j| \cdot \hat{f}_j(\mathbf{v})}{\sum_h |S_h| \cdot \hat{f}_h(\mathbf{v})},$$

where $|S_j|$ is the number of class-*j* training samples and $\hat{f}_j(\cdot)$ is the kernel density estimator corresponding to class-*j* training samples.

RVKDE belongs to the radial basis function network (RBFN), a special type of neural networks with several distinctive features (Mitchell 1997; Kecman 2001). The decision function of two-class RVKDE can be simplified as follows:

$$f_{\text{RVKDE}}(\mathbf{v}) = \sum_{\mathbf{s}_i} y_i \cdot \frac{1}{\sigma_i} \cdot \exp\left(-\frac{\|\mathbf{v}-\mathbf{s}_i\|^2}{2\sigma_i^2}\right),$$

where $\mathbf{v}$ is a testing sample, y_i is the class value as either +1 (positive) or -1 (negative) of a training sample $\mathbf{s}_i$, and σ_i is the local density of the proximity of $\mathbf{s}_i$, estimated by the kernel density estimation algorithm. The testing sample $\mathbf{v}$ is classified as positive if $f_{\text{RVKDE}}(\mathbf{v}) \geq 0$, and as negative otherwise. Interestingly, the decision function of RVKDE is very similar to that of SVM using the radial basis function (RBF) kernel:

$$f_{\text{SVM}}(\mathbf{v}) = \sum_{\mathbf{s}_i} y_i \cdot \alpha_i \cdot \exp\left(-\gamma\|\mathbf{v}-\mathbf{s}_i\|^2\right),$$

where a_i (corresponds to the inverse of σ_i in f_{RVKDE}) and γ (corresponds to $1/2\sigma_i^2$ in f_{RVKDE}) are user-specified parameters. Thus, the mathematical models of RVKDE and SVM are analogous. The main difference between RVKDE and SVM is the criteria used to determine σ_i and a_i.

5. Evaluation

A paradoxical situation is that a benchmark requires negative samples - proteins known not to interact. A benchmark that contains only interacting protein pairs is useless, since a trivial predictor predicting any protein pairs as interacting can achieve a perfect accuracy. However, there are very limited techniques developed to confirm that two proteins do not interact. Recently, several studies have addressed this problem in evaluating computational methods of identifying protein interactions (Yu, Chou et al. 2010; Yu, Guo et al. 2010). This issue is still in a chaos stage and there is no perfect solution that fit everyone's requirements. Instead, this chapter demonstrates this issue via three major contradictions in this area.

1. **Sampled vs. entire data (also efficiency issue)**—most ML-based methods adopted SVM and have to reduce the data size because of its high time complexity. However, sampled data must lose some information and may bias the evaluation. This contradiction is especially important when comparing co-occurrence- and ML-based methods, where the former usually can be applied on entire data. Using more computing power or switching more efficient ML tool is a compromising solution.

2. **Balanced vs. unbalanced**—once sampled data is adopted, (most studies of ML-based methods adopted using sampled data even without carefully considering the previous contradiction), how to sample is another serious problem. Random sampling can preserve the data distribution (ratio of positive and negative samples) but loss too many positive samples. However, balanced sampling, which forces the inclusion of all positive samples and thus change the data distribution, has also been shown bias the evaluation accuracy (Yu, Chou et al. 2010).
3. **Distinct vs. similar**—one philosophy of creating negative data is to choose the samples which can never be positive. For example, proteins appear in different cellular compartments are possible negative samples. An opposite philosophy is that if a method can discriminate between the negative samples that are very similar to the positive ones, then this method can discriminate those dissimilar ones. The first philosophy prevents collecting negative samples that are actually positive but somehow makes the problem easier while the second philosophy has opposite advantage and disadvantage.

6. Conclusions

In this chapter, various computational methods of protein interaction are reviewed. These methods used various data sources, including localization data, structural data, expression data and/or interactions from orthologs. As a result, all of them are limited to the experimental technologies that generate such data and the incompleteness of verified data. Based on current understanding, the size of protein interaction network (PIN) of human comprises ~650,000 interactions (Stumpf, Thorne et al. 2008). However, the Human Protein Reference Database (HPRD) deposits less than 3% of them (Peri, Navarro et al. 2003; Mishra, Suresh et al. 2006). Even under such a challenging circumstance, computational methods have shown to achieve satisfying performance. This encourages more effort in developing computational methods of protein interaction to complement experimental technologies.

7. References

Albert, S., S. Gaudan, et al. (2003). "Computer-assisted generation of a protein-interaction database for nuclear receptors." *Molecular Endocrinology* 17(8): 1555-1567.

Altschul, S. F., T. L. Madden, et al. (1997). "Gapped BLAST and PSI-BLAST: a new generation of protein database search programs." *Nucleic Acids Research 25*(17): 3389.

Artin, E. (1964). *The Gamma Function*. New York, Holt, Rinehart and Winston.

Ashburner, M., C. A. Ball, et al. (2000). "Gene Ontology: tool for the unification of biology." *Nature genetics 25*(1): 25.

Barrett, T., D. B. Troup, et al. (2006). "NCBI GEO: mining tens of millions of expression profiles—database and tools update." *Nucleic Acids Research 35*(suppl 1): D760.

Chang, D., Y. T. Syu, et al. (2010). "Predicting the protein-protein interactions using primary structures with predicted protein surface." *BMC Bioinformatics 11*(Suppl 1): S3.

Chang, D. T. H., Y. T. Syu, et al. "Predicting the protein-protein interactions using primary structures with predicted protein surface." *BMC bioinformatics 11*.

Cohen, A. M. and W. R. Hersh (2005). "A survey of current work in biomedical text mining." *Briefings in bioinformatics 6*(1): 57.

Davey, N. E., K. Van Roey, et al. (2011). "Attributes of short linear motifs." *Mol. BioSyst.*

Doerr, A. (2010). "The importance of being negative." *Nature Methods 7*(1): 10-11.

Donaldson, I., J. Martin, et al. (2003). "PreBIND and Textomy–mining the biomedical literature for protein-protein interactions using a support vector machine." *BMC bioinformatics 4*(1): 11.

Enault, F., K. Suhre, et al. (2003). "Annotation of bacterial genomes using improved phylogenomic profiles." *Bioinformatics 19*(Suppl 1): i105.

Enault, F., K. Suhre, et al. (2003). "Annotation of bacterial genomes using improved phylogenomic profiles." *Bioinformatics 19*(suppl 1): i105.

Enright, A. J., I. Iliopoulos, et al. (1999). "Protein interaction maps for complete genomes based on gene fusion events." *Nature 402*(6757): 86-90.

Espadaler, J., O. Romero-Isart, et al. (2005). "Prediction of protein–protein interactions using distant conservation of sequence patterns and structure relationships." *Bioinformatics 21*(16): 3360.

Guo, Y., L. Yu, et al. (2008). "Using support vector machine combined with auto covariance to predict protein–protein interactions from protein sequences." *Nucleic Acids Research 36*(9): 3025.

Huynen, M. A. and B. Snel (2000). "Gene and context: integrative approaches to genome analysis." *Advances in Protein Chemistry 54*: 345-379.

Jothi, R., P. F. Cherukuri, et al. (2006). "Co-evolutionary analysis of domains in interacting proteins reveals insights into domain-domain interactions mediating protein-protein interactions." *Journal of molecular biology 362*(4): 861-875.

Jothi, R., M. G. Kann, et al. (2005). "Predicting protein–protein interaction by searching evolutionary tree automorphism space." *Bioinformatics 21*(suppl 1): i241.

Kecman, V. (2001). *Learning and soft computing : support vector machines, neural networks, and fuzzy logic models*. Cambridge, Mass., MIT Press.

Krogan, N. J., G. Cagney, et al. (2006). "Global landscape of protein complexes in the yeast Saccharomyces cerevisiae." *Nature 440*(7084): 637-643.

Lee, S. I. and S. Batzoglou (2003). "Application of independent component analysis to microarrays." *Genome Biology 4*(11): R76.

Li, H., J. Li, et al. (2006). "Discovering motif pairs at interaction sites from protein sequences on a proteome-wide scale." *Bioinformatics 22*(8): 989.

Marcotte, E. M., M. Pellegrini, et al. (1999). "Detecting protein function and protein-protein interactions from genome sequences." *Science 285*(5428): 751.

Mishra, G. R., M. Suresh, et al. (2006). "Human protein reference database - 2006 update." *Nucleic Acids Research 34*: D411-D414.

Mitchell, T. M. (1997). *Machine learning*. New York, McGraw-Hill.

Oyang, Y. J., S. C. Hwang, et al. (2005). "Data classification with radial basis function networks based on a novel kernel density estimation algorithm." *IEEE Transactions on Neural Networks 16*(1): 225-236.

Peri, S., J. D. Navarro, et al. (2003). "Development of human protein reference database as an initial platform for approaching systems biology in humans." *Genome research 13*(10): 2363.

Peri, S., J. D. Navarro, et al. (2003). "Development of human protein reference database as an initial platform for approaching systems biology in humans." *Genome Research 13*(10): 2363-2371.

Pevzner, P. A. and S. H. Sze (2000). *Combinatorial approaches to finding subtle signals in DNA sequences*, Citeseer.

Rogozin, I. B., K. S. Makarova, et al. (2002). "Connected gene neighborhoods in prokaryotic genomes." *Nucleic Acids Research 30*(10): 2212.

Salgado, H., G. Moreno-Hagelsieb, et al. (2000). "Operons in Escherichia coli: genomic analyses and predictions." *Proceedings of the National Academy of Sciences 97*(12): 6652.

Salwinski, L., C. S. Miller, et al. (2004). "The database of interacting proteins: 2004 update." *Nucleic Acids Research 32*(suppl 1): D449.

Shen, J., J. Zhang, et al. (2007). "Predicting protein–protein interactions based only on sequences information." *Proceedings of the National Academy of Sciences 104*(11): 4337.

Sheu, S. H., D. R. Lancia, et al. (2005). "PRECISE: a database of predicted and consensus interaction sites in enzymes." *Nucleic Acids Research 33*(suppl 1): D206.

Shoemaker, B. A. and A. R. Panchenko (2007). "Deciphering protein–protein interactions. Part II. Computational methods to predict protein and domain interaction partners." *PLoS computational biology 3*(4): e43.

Smialowski, P., P. Pagel, et al. (2010). "The Negatome database: a reference set of non-interacting protein pairs." *Nucleic acids research 38*(suppl 1): D540.

Snitkin, E., A. Gustafson, et al. (2006). "Comparative assessment of performance and genome dependence among phylogenetic profiling methods." *BMC bioinformatics 7*(1): 420.

Soong, T., K. O. Wrzeszczynski, et al. (2008). "Physical protein–protein interactions predicted from microarrays." *Bioinformatics* 24(22): 2608-2614.

Stumpf, M. P. H., T. Thorne, et al. (2008). "Estimating the size of the human interactome." *Proceedings of the National Academy of Sciences* 105(19): 6959.Sun, J., J. Xu, et al. (2005). "Refined phylogenetic profiles method for predicting protein–protein interactions." *Bioinformatics 21*(16): 3409.

Szklarczyk, D., A. Franceschini, et al. (2011). "The STRING database in 2011: functional interaction networks of proteins, globally integrated and scored." *Nucleic Acids Research 39*(suppl 1): D561.

Tan, S. H., W. Hugo, et al. (2006). "A correlated motif approach for finding short linear motifs from protein interaction networks." *BMC bioinformatics 7*(1): 502.

Van Dijk, A., C. Ter Braak, et al. (2008). "Predicting and understanding transcription factor interactions based on sequence level determinants of combinatorial control." *Bioinformatics 24*(1): 26.

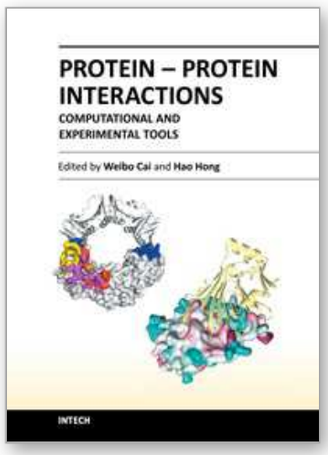
Vapnik, V. and V. Vapnik (1998). *Statistical learning theory*, Wiley New York.

Walhout, A. J. M., R. Sordella, et al. (2000). "Protein interaction mapping in C. elegans using proteins involved in vulval development." *Science 287*(5450): 116.

Witten, I. H., E. Frank, et al. (2011). *Data mining : practical machine learning tools and techniques*. Burlington, MA, Morgan Kaufmann.

Yu, C. Y., L. C. Chou, et al. (2010). "Predicting protein-protein interactions in unbalanced data using the primary structure of proteins." *BMC Bioinformatics 11*(1): 167.

Yu, J., M. Guo, et al. (2010). "Simple sequence-based kernels do not predict protein-protein interactions." *Bioinformatics 26*(20): 2610.



Protein-Protein Interactions - Computational and Experimental Tools

Edited by Dr. Weibo Cai

ISBN 978-953-51-0397-4
Hard cover, 472 pages
Publisher InTech
Published online 30, March, 2012
Published in print edition March, 2012

Proteins are indispensable players in virtually all biological events. The functions of proteins are coordinated through intricate regulatory networks of transient protein-protein interactions (PPIs). To predict and/or study PPIs, a wide variety of techniques have been developed over the last several decades. Many in vitro and in vivo assays have been implemented to explore the mechanism of these ubiquitous interactions. However, despite significant advances in these experimental approaches, many limitations exist such as false-positives/false-negatives, difficulty in obtaining crystal structures of proteins, challenges in the detection of transient PPI, among others. To overcome these limitations, many computational approaches have been developed which are becoming increasingly widely used to facilitate the investigation of PPIs. This book has gathered an ensemble of experts in the field, in 22 chapters, which have been broadly categorized into Computational Approaches, Experimental Approaches, and Others.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Tien-Hao Chang (2012). Computational Approaches to Predict Protein Interaction, Protein-Protein Interactions - Computational and Experimental Tools, Dr. Weibo Cai (Ed.), ISBN: 978-953-51-0397-4, InTech, Available from: http://www.intechopen.com/books/protein-protein-interactions-computational-and-experimental-tools/computational-approaches-to-predict-protein-interaction

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

IntechOpen