Extracting Coevolutionary Features from Protein Sequences for Predicting Protein-Protein Interactions

Lun Hu and Keith C.C. Chan

Abstract—Knowing the ways proteins interact with each other are crucial to our understanding of the functional mechanisms of proteins. It is for this reason that different approaches have been developed in attempts to predict protein-protein interactions (PPIs) computationally. Among them, the sequence-based approaches are preferred to the others as they do not require any information about protein properties to perform their tasks. Instead, most sequence-based approaches make use of feature extraction methods to extract features directly from protein sequences so that for each protein sequence, we can construct a feature vector. The feature vectors of every pair of proteins are then concatenated to form two classes of interacting and non-interacting proteins. The prediction of whether or not two proteins interact with each other is then formulated as a classification problem. How accurate PPI predictions can be made therefore depends on how good the features are that can be extracted from the protein sequences to allow interacting or non-interacting to be best distinguished. To do so, instead of extracting such features from individual protein sequences independently of the other protein in the same pair, we propose to jointly consider features from both sequences in a protein pair during the feature extraction process through using a novel coevolutionary feature extraction approach called CoFex. Coevolutionary features extracted by CoFex refer to the covariations found at coevolving positions. Based on the presence and absence of these coevolutionary features in the sequences of two proteins, feature vectors can be composed for pairs of proteins rather than individual proteins. The experiment results show that CoFex is a promising feature extraction approach and can improve the performance of PPI prediction.

Index Terms—Coevolutionary information, covariations, protein-protein interaction prediction, sequence information

1 Introduction

Rather than individually and independently, proteins perform their functions by interacting with each other as a whole [1]. By studying how proteins interact, we can understand the molecular mechanisms of many biological processes, such as DNA regulation, cell signaling and protein complex assembly. We may also be able to discover unknown functions of a protein based on the known functions of those that it interacts with. Given how important it is for us to know how proteins interact with each other, various attempts have been made to develop techniques to effectively predict protein-protein interactions (PPIs).

Laboratory techniques can be classified into two categories based on the volume of interactions that can be identified every time: the low-throughput and the high-throughput techniques. Low-throughput techniques, such as glutathione S-transferase [43] and TAP-tag [44], analyze proteins in a serial, one-at-a-time manner which is considerably time-consuming [2]. High-throughput techniques, such as the two-hybrid systems [3], [4], mass spectrometry [5], [6], and microarray analysis [7], have been developed for systematic and large-scale identification of PPIs. Though more efficient than low-throughput techniques, high-throughput techniques may only predict PPIs with relatively high false-positive rates [8]. Their coverage of PPIs is also usually rather incomplete [9].

To predict PPIs more efficiently, there have been many attempts to make use of computational techniques. When performing their tasks, these techniques normally require that genomic information, such as gene fusion [10], the conversion of gene-order [11], and the calculation of prior probabilities of genomic features between interacting proteins [12], etc., be known ahead of time. As there is evidence that proteins that coevolve with each other are more likely to interact with each other, some techniques predict the existence of PPI between two proteins based on the knowledge of co-evolutionary information between them. Knowledge of phylogenetics [14], [25], protein domains [15], [16], [26], [27] and topological properties of proteins in PPI networks [17], for example, have thus been considered.

As genomic and co-evolutionary information are not always readily available, there have been some recent attempts to make use of protein-sequence information to predict PPIs [31], [32] as such information can be obtained directly from protein sequences if necessary. In predicting if two proteins interact with each other, therefore, protein sequence information is utilized together with whatever is known about the proteins, such as residue properties and

- *L. Hu is with the School of Computer Science and Technology, Wuhan University of Technology, Wuhan, Hubei 430070, China, and the Department of Computing, The Hong Kong Polytechnic University, Kowloon, Hong Kong, China. E-mail: hulun@whut.edu.cn.*
- *K.C.C. Chan is with the Department of Computing, The Hong Kong Polytechnic University, Kowloon, Hong Kong, China. E-mail: keith.chan@polyu.edu.hk.*

Manuscript received 6 Aug. 2014; revised 20 Sept. 2015; accepted 13 Dec. 2015. Date of publication 22 Jan. 2016; date of current version 2 Feb. 2017.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TCBB.2016.2520923

gene ontology (GO), etc. [18]. Based on the results of the investigative studies in [19], [20], [21], [22], however, it can be concluded that, based on sequence information alone, it is possible for PPIs to be rather accurately predicted. As a result, what have been referred to as the sequence-based approaches to PPI prediction are becoming popular. With these approaches, there is no requirement for any prior knowledge about protein properties to be obtained ahead of time. However, as they do not rely on any other information other than that available within protein sequences, for these approaches to predict PPIs accurately, the extraction of appropriate features from protein sequences is of utmost importance.

For feature extraction from protein sequences, most existing sequence-based approaches attempt to discover k-mers, which are amino acid sequence segments that are of length k. These k-mers are then used to compose a feature vector for each protein sequence.

Different examples of how these k-mers can be used to predict PPIs can be found in different literatures. For example, according to [20], all possible combinations of 3-mers have been used to construct a feature vector for a given protein sequence so as to train a support vector machine (SVM) to distinguish between interacting and non-interacting proteins.

Even though the sequence-based approaches have been shown to be quite promising, they are also considered to have some drawbacks. First, the features that are extracted from protein sequences are of fixed-length, k. For example, in [20] and [21], in composing feature vectors for interacting and non-interacting proteins, only k-mers of length, $k = 3$ are considered. In another example, in the case of the PIPE algorithm [22], k-mers of length, $k = 20$ are considered.

The second drawback of existing sequence-based approaches is that the feature vector associated with each protein is obtained independently of each other rather than in pairs and it is for this reason that it cannot best capture the interaction relationship between proteins. To consider the interaction between protein pairs, there is a need to tackle the problem of combining the feature vectors of proteins in a protein pair for the training of classifiers such as the SVM. In the case of existing approaches, this is typically done by concatenating the feature vectors of two proteins when training SVM classifiers. For example, in [20] and [21], pairwise kernel and the S-Kernel functions are used respectively. To overcome these drawbacks, we propose here a sequence-based PPI prediction approach that makes use of a novel feature extraction method, called CoFex (the Coevolutionary Feature Extraction method), that can take into consideration co-evolutionary sequence information of a protein pair while predicting PPIs.

Based on CoFex, for two residues at different positions in a protein sequence, if the mutation of one residue is always accompanied by the mutation of the other residue, CoFex considers this pair of positions as coevolving and their corresponding amino acids, after mutation, as covariation and are believed to be correlated. This is because coevolving positions have been demonstrated to play an important functional role for proteins [36], [37], [38], therefore, we have reason to believe that we will be able to find more of such covariations of coevolving positions in protein pairs that interact with each other than those that do not. In other words, if we are able to extract features related to the discovering of covariations of coevolving positions in protein pairs, we will be better able to predict PPIs based on sequence information alone.

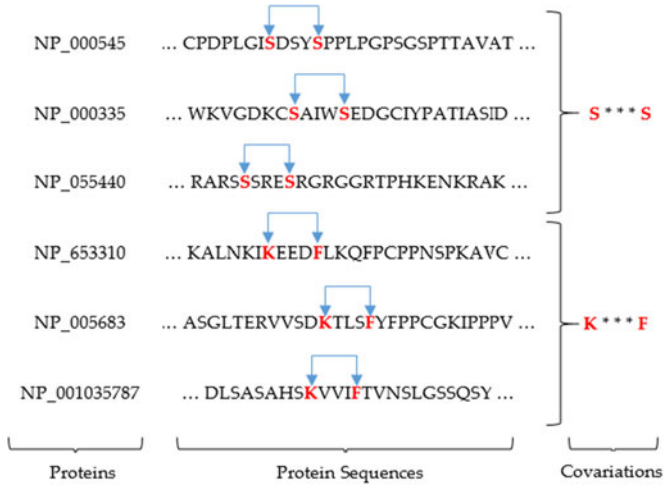

Fig. 1. An example of covariations of coevolving postions in multiple protein sequences. Note that the presence of $*$ in a covariation means that the corresponding position can be replaced with any amino acid.

Unfortunately, however, detecting coevolving positions without additional evolutionary information is difficult. Instead of detecting for such positions directly, CoFex, therefore, looks for covariations in pairs of protein sequences through detection of statistically significant co-occurring patterns in amino acid pairs.

To illustrate the idea, let us consider a list of proteins in Fig. 1. For the first three proteins (i.e., NP_000545, NP_000335 and NP_055440) in Fig. 1, it can be observed that an amino acid S is frequently found four positions to the right of another amino acid S. Representing this co-occurrence as $\mathrm{S}***\mathrm{S}$ (where $*$ denotes any amino acid in-between), one can also observe $\mathrm{K}***\mathrm{F}$ to be the case. Both frequently co-occurring sub-sequence $\mathrm{S}***\mathrm{S}$ and $\mathrm{K}***\mathrm{F}$ in these protein sequences may indicate that their corresponding positions in the sequences of the protein pairs are coevolving and if this is the case, $\mathrm{S}***\mathrm{S}$ and $\mathrm{K}***\mathrm{F}$ can be considered as the covariations of coevolving positions with three don't care amino acid molecules between them.

Given a database of protein sequences consisting of protein pairs that are known to either interact or not-interact, CoFex is able to identify all covariations of coevolving positions based on the discovering of statistically significant co-occurring amino acid sub-sequences. These sub-sequences may provide evidence supporting or refuting the existence of an interaction relationship between protein pairs. By considering these covariations as features of interest, a feature vector can then be obtained for the protein pair based on the presence and absence of covariations in both the sequences of the proteins in the pair. With the feature vectors of all pairs of proteins in the database obtained, CoFex can then construct classifiers to predict PPIs. The rest of this paper is organized as follows. In the next section, we will present a detailed literature review about the related works of PPI prediction. In Section 3, the preliminaries about the mathematical symbols used in this paper are introduced. In Section 4, the details of CoFex are presented and we also demonstrate how to compose feature

vectors for pairs of proteins based on the covariations identified by CoFex. For performance evaluation, we have tested CoFex with the integration of several well-known classifiers and also compared the performance against existing sequence-based prediction approaches, the experiment results are presented and discussed in Section 5. Finally, we summarize our work in Section 6.

2 Related Works

In this section, we present a detailed literature review on the related works of PPI prediction. For a clear demonstration, the computational approaches proposed for predicting PPIs are classified into three categories based on the sources of biological information they make use of for the prediction task. The three categories include genomic approaches, evolutionary approaches and sequence-based approaches. Therefore, the remaining part of this section will be unfolded from these three categories.

2.1 Genomic Approaches

Due to the availability of whole genomic sequencing, it has been pointed out that genes located in genome sequences can hint at the interaction between proteins at a comprehensive level.

Dandekhar et al. [11] find that proteins encoded by conversed gene pairs are more likely to interact with each other and such conserved gene pairs are within a low level conservation of gene-order. Therefore, with this observation, the conservation of gene-order can be exploited to help predict PPIs. Though promising, this approach cannot predict PPIs for proteins where the conservation of gene-order is not found, such as proteins encoded by distantly located genes.

Another discovery about the use of genomic information w.r.t. the formation of PPI is that pairs of interacting proteins are found to have homologs in another genome where they are fused into a single protein [24]. In this regard, Enright et al. [10] have developed a computational method to seek for such fusion events in different genomes so that proteins involved in a fusion event are expected to interact with each other. However, the disadvantage of this approach is also obvious as it cannot work with proteins where no fusion events are uncovered through the analysis of genomic sequencing.

In [12], different genomic features, such as messenger RNA coexpression, coessentiality and colocalization, are used to quantify the associations between them and PPIs. Based on these quantified associations, Bayesian networks are constructed to predict PPIs when the genomic features of query proteins are given.

2.2 Evolutionary Approaches

Evolutionary information discloses the procedure of how proteins evolve across different species. Since proteins that co-evolve are more likely to interact with each other, the similarity in evolutionary information is of potential relevance to the prediction of PPIs as it indicates to what extent the two proteins co-evolve.

Among various evolutionary information, Pazos and Valencia [25] make use of phylogenetic trees of proteins to indicate PPIs. They propose a distance measure to compute the similarity between the phylogenetic trees of proteins, thus determining whether there is a possible interaction between them. Similar to phylogenetic trees, phylogenetic profiles are also adopted by Pellegrini et al. [14] to predict PPIs. Under the assumption that for two interacting proteins one cannot exist if the other one is lost during evolution, Pellegrini et al. characterize this co-evolution property by the use of phylogenetic profiles of proteins and hence proteins with similar profiles are strongly expected to interact.

Another source of evolutionary information that we can use for the prediction of PPIs is the domain knowledge of proteins. It is believed that proteins are to interact as a result of their interacting domains, and it is for this reason that many computations approaches have been proposed to solve the PPI prediction based on domain knowledge. In [26], domains that interact more often than expected are found to compose the signatures of proteins, which in return are used to predict PPIs. Rather than simply resting on frequent interacting domains, Deng et al. [27] apply a Maximum Likelihood Estimation method to identify interacting domains that infer curated PPIs and then with such inferred interacting domains the interactions between proteins can be predicted. Similarly, [15] makes use of random forest of decision trees that are trained by taking into consideration all the proteins domains, thus performing the prediction task. In addition, Kanaan et al. [16] employs a set cover approach to partition pairs of domains so that the desired partitioning can best explain the underlying protein interaction in terms of specificity score. Then with the partitioned pairs of domains, the method of Maximum Specificity Set Cover is introduced to predict potential PPIs.

Instead of considering the knowledge of proteins to infer evolutionary information, You et al. make use of the topological information of PPI networks to perform the task of predicting PPIs. First of all, a PPI network given is transformed into a low-dimensional metric space where each dimension represents a kind of topological property. With such metric space, the likelihood of how two proteins are interacting can be estimated by measuring the similarity of the corresponding points in the matrix space.

Recently, Maetschke et al. [46] introduce the concept of inducers as a method make use of Gene Ontology information more effectively. Given two GO terms, its induced term set is composed of all GO terms along the shortest path between the two terms. To predict PPIs, they propose GO2PPI by integrating all induced GO terms with classification techniques. As the consideration of all induced GO terms is richer in information, the performances of classifiers are better than those without considering induced GO terms.

2.3 Sequence-Based Approaches

Protein sequences, composed of amino acids, are the primary structures of proteins and the motivation behind the use of protein sequences for predicting PPIs derives from the hypothesis that sequence information may contribute to mediate PPIs.

In general, most of sequence-based approaches take advantage of the learning ability of SVM to perform the task. These SVM-based approaches are distinguished by the definition of feature vectors extracted from protein sequences

and also by the proposal of kernel function for the purpose of concatenating the feature vectors of individual proteins.

As the first strike among these SVM-based approaches, Bock and Gough [18] assembly the feature vector for each protein sequence based on a set of residue properties, such as charge, hydrophobicity and surface tension. After transforming these vectors with variable lengths into vectors with a fixed length through a concatenation operation, Bock and Gough then train several SVMs with different standard kernel functions by taking these new vectors as input and make a prediction based on the average results from trained SVMs for query proteins.

Since the sequence information has been proved to be useful for predicting PPIs as indicated by [18], subsequent studies have been concentrating on purely adopting sequence information for PPI prediction. Martin et al. [19] extend the signature descriptors to compose a fixed-length feature vector for each of proteins and then introduce a signature product kernel to concatenate the feature vectors of proteins in protein pairs so that SVM can be applied for the problem of predicting PPIs.

In addition to the signature product kernel proposed by [19], Benhur and Nobel [20] also introduce a pairwise kernel function that measures the similarity between pairs of proteins based on the similarities between individual proteins. To extract the feature vectors from sequence information, Benhur and Noble adopt the spectrum vector composed of *k*-mers with length 3.

Later Shen et al. [21] propose an S-kernel function that is specifically designed for PPIs by considering the symmetry property of PPI. Before assembling feature vectors of proteins, Shen et al. classify the amino acids into seven classes and hence the number of unique elements in the protein sequence is now reduced to 7. With the new protein sequences, Shen et al. introduce a conjoint triad method to create feature vectors for proteins. When considering the difference of feature vector between [20] and [21], the length of the later one, i.e., 7^3, is much shorter than the former one, i.e., 20^3.

In addition to SVM-based approaches, Pitre et al. [22] propose PIPE to tackle the problem of predicting PPIs from a different angle and the idea behind PIPE is to measure how often pairs of subsequences in the two query proteins co-occur in pairs of proteins that are known to interact. Given two sets of *20*-mers respectively found in the sequences of query proteins, PIPE creates a matrix where columns are to denote the *20*-mers of the first query protein and rows are for the *20*-mers of the other query protein. For each cell in this matrix, its value is the number of sequences in the corresponding neighbor list of the column *20*-mer and each of counted sequences is found to contain the row *20*-mer. With this matrix, PIPE scores the confidence about the hypothesis that the query protein are interacting.

In [45], PPIevo has been proposed to extract the feature vectors from Position-Specific Scoring Matrix for each of proteins based on sequence information. To represent a protein pair, PPIevo combines some statistics of the feature vectors of two proteins in this protein pair, the mutual information between the feature vectors and the simple concatenation of the feature vectors to construct the feature vector of the protein pair. Based on the feature vectors of all protein pairs involved, PPIevo then adopts Random Forest to build classifiers for the prediction of PPIs.

In addition to these three kinds of approaches for predicting PPIs, some attempt has been made to explore the possibility of fusing genomic, evolutionary and sequence information for predicting PPIs. In particular, Zahiri et al. [48] use eight different genomic and proteomic features to compose feature vectors for human PPIs and then develop an ensemble learning method, LocFuse, to predict unknown PPIs with four types of different classifiers.

3 Preliminaries

Given an alphabet set $\Gamma = \{\tau_1, \tau_2, \ldots, \tau_{n_\Gamma}\}$ consisting of total n_Γ amino acids, a protein sequence S is represented as $S = (s_t)_{1 \le t \le n_S}$, where $s_t \in \Gamma$ and n_S is the length of S. Therefore, a *k*-mer segment starting from the position t in S is denoted as $S_{t,k} = (s_t, s_{t+1}, \ldots, s_{t+k-1})$, where $1 \le t \le n_S - k + 1$.

A pair of coevolving positions with length k means that there are *k*-2 don't care positions between them. For the sake of convenience, we use$(\tau_i, \tau_j)_k$, where $\tau_i, \tau_j \in \Gamma$, to represent a possible covariation of coevolving positions with length *k*. If S contains $(\tau_i, \tau_j)_k$, it means that $\exists S_{t,k} : s_t = \tau_i \text{ and } s_{t+k-1} = \tau_j$. $\mathbf{CoV}_k = \{(\tau_i, \tau_j)_k\}$ is the set of all verified covariations of coevolving positions with length *k*. Assuming that k_{max} is the maximum value of *k*, the set of all covariations is denoted as $\mathbf{CoV} = \bigcup_{k=2}^{k_{max}} \mathbf{CoV}_k$.

Therefore, for two query proteins, the problem we will address in this paper is to predict whether or not these two proteins are interacting with each other according to coevolution features of covariations found in **CoV**. We use int and $\overline{int}$ to respectively denote the existence and the non-existence of the interaction relationship between two proteins.

4 Methodology

Generally speaking, the procedure of predicting PPIs is comprised of two phases, the first phase is to compose feature vectors for protein pairs based on **CoV** identified by CoFex and the second phase is to train a classifier with these feature vectors, thus predicting PPIs. To implement these two phases, there are several steps as described below.

4.1 Details of CoFex

The purpose of CoFex is to identify the coevolutionary features represented by covariations of coevolving positions so that feature vectors of protein pairs can be constructed. CoFex is composed of two steps, the first step is to identify all covariations in **CoV** from protein sequences and the second step is to weight the verified covariations based on their abilities of providing evidence for int or $\overline{int}$. The details of CoFex are presented as below.

4.1.1 Identifying CoV

To identify **CoV**, let us first consider the problem of determining whether $(\tau_i, \tau_j)_k$ is frequently found with statistical significance in all the protein sequences so that we can verify whether it is a covariation. To do so, we make use of some statistical knowledge.

Assuming that we have total n proteins for consideration in the training dataset, it is necessary for us to verify whether $o_{(\tau_i,\tau_j)_k}$, which is the number of proteins whose sequences contain $(\tau_i,\tau_j)_k$, is large enough to reach the conclusion that the co-occurrences of $(\tau_i,\tau_j)_k$ are significantly frequently observed among the sequences of all proteins. Hence, it is explicit to derive (1) to indicate to what extent $(\tau_i,\tau_j)_k$ is frequently observed

$$\mathrm{freq}((\tau_i,\tau_j)_k) = \mathrm{p}((\tau_i,\tau_j)_k) - \mathrm{p}((\tau_i,*)_k)\mathrm{p}((*,\tau_j)_k). \tag{1}$$

In (1), $\mathrm{p}((\tau_i,\tau_j)_k)$, $\mathrm{p}((\tau_i,*)_k)$ and $\mathrm{p}((*,\tau_j)_k)$ are defined as:

$$\mathrm{p}((\tau_i,\tau_j)_k) = \frac{o_{(\tau_i,\tau_j)_k}}{n}, \tag{2}$$

$$\mathrm{p}((\tau_i,*)_k) = \frac{o_{(\tau_i,*)_k}}{n}, \tag{3}$$

$$\mathrm{p}((*,\tau_j)_k) = \frac{o_{(*,\tau_j)_k}}{n}, \tag{4}$$

where $o_{(\tau_i,*)_k} = \sum_{j=1}^{n_\Gamma} o_{(\tau_i,\tau_j)_k}$ and $o_{(*,\tau_j)_k} = \sum_{i=1}^{n_\Gamma} o_{(\tau_i,\tau_j)_k}$. The product of $\mathrm{p}((\tau_i,*)_k)$ and $\mathrm{p}((*,\tau_j)_k)$ denotes the expected probability of $(\tau_i,\tau_j)_k$ that is found in a protein sequence. In this regard, if $\mathrm{freq}((\tau_i,\tau_j)_k) > 0$, it can be inferred that the exact probability of finding $(\tau_i,\tau_j)_k$ in a protein sequence is larger than expected. However, it is still unknown that to what extent the value of $\mathrm{freq}((\tau_i,\tau_j)_k)$ is large enough to indicate that $(\tau_i,\tau_j)_k$ is frequently observed. Also, the value of $\mathrm{freq}((\tau_i,\tau_j)_k)$ is subject to the magnitudes of $o_{(\tau_i,\tau_j)_k}$, $o_{(\tau_i,*)_k}$ and $o_{(\tau_i,*)_k}$. Therefore, we rewrite the definition of $\mathrm{freq}((\tau_i,\tau_j)_k)$ as:

$$\mathrm{freq}((\tau_i,\tau_j)_k) = \frac{\mathrm{p}((\tau_i,\tau_j)_k) - \mathrm{p}((\tau_i,*)_k)\mathrm{p}((*,\tau_j)_k)}{\sqrt{\frac{\mathrm{p}((\tau_i,*)_k)\mathrm{p}((*,\tau_j)_k)}{n}(1-\mathrm{p}((\tau_i,*)_k))(1-\mathrm{p}((*,\tau_j)_k))}}. \tag{5}$$

It has been pointed out by [28] that the value of $\mathrm{freq}((\tau_i,\tau_j)_k)$ computed with (5) follows a normal distribution. Therefore, we have:

Definition 1. *If* $\mathrm{freq}((\tau_i,\tau_j)_k) \geq 1.96$, *we reckon that* $(\tau_i,\tau_j)_k$ *is significantly frequently found among all protein sequences at a confidence level of 95 percent and hence it is eligible to be considered as a covariation of coevolving positions with length* k.

After evaluating all possible covariations with (5), we will obtain $\mathbf{CoV}_k$ according to *Definition 1*. By applying the same procedure to all values of k, **CoV** can be identified.

4.1.2 Weighting CoV

Once obtaining **CoV**, we then move forward to the problem of weighting each covariation in it. The weights of covariations are used to quantitatively indicate the ability of providing evidence for int or $\overline{int}$. To solve it, we perform the weighting operation from the viewpoint of decrease in uncertainty.

Assuming that $(\tau_i,\tau_j)_k \in \mathbf{CoV}_k$, we will describe how $(\tau_i,\tau_j)_k$ is weighted. But before that, we have to explain the symbols used for weighting $(\tau_i,\tau_j)_k$. Given a training dataset, n_{int} is the number of pairs of interacting proteins, $n_{\overline{\mathrm{int}}}$ is the number of pairs of non-interacting proteins, $n_{int}^{(\tau_i,\tau_j)_k}$ is the number of pairs of interacting proteins whose sequences contain $(\tau_i,\tau_j)_k$, $n_{int}^{\overline{(\tau_i,\tau_j)_k}}$ is the number of pairs of interacting proteins whose sequences do not contain $(\tau_i,\tau_j)_k$, $n_{\overline{int}}^{(\tau_i,\tau_j)_k}$ is the number of pairs of non-interacting proteins whose sequences contain $(\tau_i,\tau_j)_k$, and $n_{\overline{int}}^{\overline{(\tau_i,\tau_j)_k}}$ is the number of pairs of non-interacting proteins whose sequences do not contain $(\tau_i,\tau_j)_k$.

Firstly, the amount of evidence provided by the presence of $(\tau_i,\tau_j)_k$ for int can be estimated by mutual information as below,

$$\mathrm{MI}(int|(\tau_i,\tau_j)_k) = \log\left(\frac{\mathrm{p}(int|(\tau_i,\tau_j)_k)}{\mathrm{p}(int)}\right), \tag{6}$$

where

$$\mathrm{p}(int|(\tau_i,\tau_j)_k) = \frac{n_{int}^{(\tau_i,\tau_j)_k}}{n_{int}^{(\tau_i,\tau_j)_k} + n_{\overline{int}}^{(\tau_i,\tau_j)_k}},$$

$$\mathrm{p}(int) = \frac{n_{int}}{n_{int} + n_{\overline{int}}}.$$

In (6), $\mathrm{p}(int|(\tau_i,\tau_j)_k)$ is the conditional probability of proteins that are interacting with each other given that $(\tau_i,\tau_j)_k$ is found in both of their sequences and $\mathrm{p}(int)$ is the prior probability of two proteins that are interacting. From (6), we note that the value of $\mathrm{MI}(int|(\tau_i,\tau_j)_k)$ is only positive when $\mathrm{p}(int|(\tau_i,\tau_j)_k) > \mathrm{p}(int)$. In other words, the mutual information defined by (6) shows the decrease in uncertainty about the interaction relationship between two proteins when $(\tau_i,\tau_j)_k$ is found in both of their sequences.

Given the presence of $(\tau_i,\tau_j)_k$ in both sequences of two proteins, the difference in mutual information when the two proteins are interacting and when they are not interacting is an estimation of to what extent $(\tau_i,\tau_j)_k$ is likely to indicate int. Therefore, we can use such difference, denoted by $\mathrm{DMI}((\tau_i,\tau_j)_k)$, to measure the evidence provided by the observation of the presence of $(\tau_i,\tau_j)_k$ in both sequences of two proteins in favor of indicating that these two proteins are interacting with each other, and its formula is

$$\mathrm{DMI}((\tau_i,\tau_j)_k) = \mathrm{MI}(int|(\tau_i,\tau_j)_k) - \mathrm{MI}(\overline{int}|(\tau_i,\tau_j)_k). \tag{7}$$

Substituting (6) for both $\mathrm{MI}(int|(\tau_i,\tau_j)_k)$ and $\mathrm{MI}(\overline{int}|(\tau_i,\tau_j)_k)$ in (7), we obtain

$$\mathrm{DMI}((\tau_i,\tau_j)_k) = \log\left(\frac{\mathrm{p}(int|(\tau_i,\tau_j)_k)(1-\mathrm{p}(int))}{\mathrm{p}(int)(1-\mathrm{p}(int|(\tau_i,\tau_j)_k))}\right). \tag{8}$$

From (8), if $\mathrm{DMI}((\tau_i,\tau_j)_k) > 0$, the presence of $(\tau_i,\tau_j)_k$ will provide some evidence to int; if $\mathrm{DMI}((\tau_i,\tau_j)_k) < 0$, the presence of $(\tau_i,\tau_j)_k$ will provide some evidence to $\overline{int}$. The strength of this kind of evidence is computed with (8).

Procedure: Composing Feature Vector

Input: **wCoV** , S^l , S^m

Output: V^{lm}

1: initialize $\mathrm{V}^{lm} = (v_t^{lm})$ as the feature vector
2: $\forall\, wCoV \in \mathbf{wCoV}$: t_{wCoV} is the corresponding position of $wCoV$ in V^{lm}
3: **for** each $wCoV$ in **wCoV do**
4: **if** $wCoV$ is found in both S^l and S^m
5: $v_{t_{wCoV}}^{lm}$ = weight of $wCoV$ computed with (8) or (9)
6: **else**
7: $v_{t_{wCoV}}^{lm} = 0$
8: **end if**
9: **end for**
10: **return** V^{lm}

Fig. 2. A complete description of Composing Feature Vector.

In addition to the presence of $(\tau_i, \tau_j)_k$, it is also possible that the sequences of interacting proteins do not contain $(\tau_i, \tau_j)_k$ after coevolution. Therefore, we also consider the situation of the absence of $(\tau_i, \tau_j)_k$ for the prediction of PPIs.

Assuming that $\overline{(\tau_i, \tau_j)_k}$ represent the absence of $(\tau_i, \tau_j)_k$ in a protein sequence, similar to the processing of the presence of $(\tau_i, \tau_j)_k$, we can obtain $\mathrm{DMI}(\overline{(\tau_i, \tau_j)_k})$ with (9). $\mathrm{DMI}(\overline{(\tau_i, \tau_j)_k})$ shows the ability of $\overline{(\tau_i, \tau_j)_k}$ to indicate int or $\overline{int}$ if $\overline{(\tau_i, \tau_j)_k}$ is not found in both sequences of the two proteins

$$\mathrm{DMI}(\overline{(\tau_i, \tau_j)_k}) = \log\left(\frac{\mathrm{p}(int|\overline{(\tau_i, \tau_j)_k})(1-\mathrm{p}(int))}{\mathrm{p}(int)(1-\mathrm{p}(int|\overline{(\tau_i, \tau_j)_k}))}\right). \quad (9)$$

In (9), $\mathrm{p}(int|\overline{(\tau_i, \tau_j)_k}) = \frac{n_{int}^{\overline{(\tau_i,\tau_j)_k}}}{n_{int}^{(\tau_i,\tau_j)_k} + n_{\overline{int}}^{\overline{(\tau_i,\tau_j)_k}}}$ and it is the conditional probability of proteins that are interacting with each other given that $(\tau_i, \tau_j)_k$ is not found in both of their sequences.

Combing the presence and the absence of all covariations in $\mathbf{CoV}_k$, we can obtain a set of weighted covariations, denoted as $\mathbf{wCoV}_k$, according to (8) and (9). Corresponding to **CoV**, we have $\mathbf{wCoV} = \bigcup_{k=2}^{k_{max}} \mathbf{wCoV}_k$. Obviously, if $|\mathbf{CoV}|$ is the size of **CoV** and $|\mathbf{wCoV}|$ is the size of **wCoV**, we have $|\mathbf{wCoV}| = 2|\mathbf{CoV}|$.

So far, the details of CoFex have been presented. Next we will compose feature vectors for pairs of proteins based on **wCoV**.

4.2 Composing Feature Vectors

Given **wCoV**, a feature vector with length $|\mathbf{wCoV}|$ will be composed for each pair of proteins in the training dataset. A complete description of how to compose such a feature vector is presented in Fig. 2. Line 1 initializes a feature vector V^{lm} for a pair of proteins whose sequences are S^l and S^m respectively. Line 2 defines a map between each weighted covariation and the corresponding position in V^{lm}. Lines 3-9 are related to the details of composing a feature vector given S^l and S^m. In particular, each $wCoV \in \mathbf{wCoV}$ will be iteratively verified to demine the value of the corresponding element in V^{lm}. Note that as both the presence and the absence of each covariation in $\mathbf{CoV}_k$ are considered when deriving **wCoV**, finding $wCoV$ in a protein sequence has different interpretations. Specifically speaking, if $wCoV = (\tau_i, \tau_j)_k$, the scenario that $wCoV$ is found in $S^l = (s_t^l)$ means that $\exists 1 \le t \le n_{S^l} - k + 1 : s_t^l = \tau_i$ and $s_{t+k-1}^l = \tau_j$; if $wCoV = (\tau_i, \tau_j)_k$, the scenario that $wCoV$ is found in $S^l = (s_t^l)$ means that $\forall 1 \le t \le n_{S^l} - k + 1 : s_t^l \neq \tau_i$ or $s_{t+k-1}^l \neq \tau_j$.

4.3 Predicting PPIs

Regarding the choice of classifier, since existing prediction approaches normally compose feature vectors for individual proteins, they have to consider the problem of concatenating feature vectors of proteins in a protein pair. It is for this reason that SVM is widely adopted by existing sequence-based approaches, as SVM is the few classifiers that can allow them to do so. By specifically designing new kernel functions, such as the pairwise kernel function of [20] and S-Kernel function of [21], the concatenating process can be implemented by SVM. But with CoFex, feature vectors are composed for pairs of proteins instead of individual proteins. Therefore, our choice of classifier in this step is more flexible, as we do not have to consider the concatenation problem when performing the task of predicting PPIs.

In the experiments, we selected several well-known classifiers, including Random Forest [39], Naïve Bayes Classifier [40] and SVM, to predict PPIs based on feature vectors identified by CoFex. The experimental results are presented in the next section.

5 Experiments and Results

To evaluate the performance of CoFex, we applied it with the classifiers of Random Forest, Naïve Bayes Classifier and SVM, and conducted experiments with several PPI datasets. For the implementation of SVM, we adopted the sigmoid kernel function which is quite popular for SVM. For simplicity, we used RF+CoFex, NBC+CoFex and SVM+CoFex to denote the integration of CoFex with Random Forest, Naïve Bayes Classifier and SVM respectively.

The results were compared against those obtained by two sequence-based approaches proposed by [20] and [21] respectively and one evolutionary approach, i.e., GO2PPI.

Although the prediction approaches of [20] and [21] were not provided, they were chosen for performance evaluation as there are sufficient details presented in related works for a full implementation. Both these two sequence-based approaches make use of 3-mers to compose feature vectors for individual proteins and then propose different kernel functions to concatenate feature vectors for each of protein pairs so that SVM can be applied for prediction. When compared with SVM+CoFex, both [20] and [21] have to design specific kernel functions so feature vectors can be concatenated for SVM implementation. In particular, [20] uses a pairwise kernel function while [21] introduces an S-Kernel function.

TABLE 1
Statistics of Benchmarking Datasets

	Number of Paris of Interacting Proteins	Number of Pairs of Non-interacting Proteins
Yeast	3,870	385,301
Human	17,434	1,743,105
AT	541	541
EC	1,167	1,167
SP	742	742

For GO2PPI, its software is available to download.[1] Regarding the Gene Ontology information, we used the tool provided by GO2PPI to extract the GO terms for each of proteins in the PPI datasets.

5.1 Benchmarking Datasets

In the experiments, five benchmarking datasets were used and they were obtained from different species including Yeast, Human, Arabidopsis thaliana (AT), Escherichia coli (EC) and Schizosaccharomyces pombe (SP). The PPI datasets of Yeast and Human were made available by [30] while the other three datasets were published by the STRING database [47]. The statistics of these five PPI datasets are given in Table 1.

In [30], the yeast PPI dataset was generated based on the core set of Saccharomyces Cerevisiae in the *database of interacting proteins* (DIP) [31] and the human PPI dataset was obtained from the Human Protein Reference Database [32]. Recognizing that the PPI datasets published by [30] contained a considerably rate of noisy data, the PPIs of AT, EC and SP were selected from the STRING database in such a way that the PPIs in these data sets have all been experimentally validated with a confidence score of not less than 0.9. Proteins that are not reported to interact within the STRING database were randomly paired up to compose the set of non-interacting proteins for each of the AT, EC and SP datasets. These three datasets were therefore more precise and less noisy than the Yeast and Human datasets.

Sequence information of human proteins was obtained from the E-utilities service provided by the *national center for biotechnology information* (NCBI) [33] while the sequence information of yeast proteins was obtained from the DIP database as of February 18, 2012 which was the latest version available for our use when we performed our experiments.

5.2 Experiment Setup

According to Table 1, we noted that the number of non-interacting proteins was much more than that of interacting proteins in the benchmarking datasets of Yeast and Human. Due to the consideration of avoiding heavy computation when training classifiers, not all non-interacting proteins were selected to compose the datasets used for performance evaluation. In the experiments, we randomly selected a certain number of interacting and non-interacting proteins from the benchmarking datasets of Yeast and Human. In particular, all approaches were tested using six kinds of dataset obtained by combining interacting and non-interacting proteins from the Yeast and Human benchmarking datasets with variation in size. Of the six kinds of datasets, three of them,Yeast-500, Yeast-1000 and Yeast-1500, were obtained from yeast benchmarking dataset by randomly selecting pairs of interacting proteins in the size of 500, 1000 and 1500 respectively. Similarly, the other three of them, Human-500, Human-1000 and Human-1500, were obtained from human PPI data by randomly selecting pairs of interacting proteins in the size of 500, 1000 and 1500 respectively. Note that for each kind of datasets, the number of non-interacting proteins was equal to that of interacting proteins.

To reduce the bias resulted from the random selection of interacting and non-interacting proteins, we randomly generated five datasets for each kind of dataset. Taking Yeast-500 as an example, we first generated five datasets each of which was composed of 500 pairs of interacting proteins that were randomly selected from the yeast benchmarking dataset and 500 pairs of non-interacting proteins that should have at least one protein found in the pairs of interacting proteins selected. Note that as the number of non-interacting proteins was relatively small in each of the AT, EC and SP PPI datasets, such processing was not applied to them.

To evaluate the performance, we applied five-fold cross-validation to each dataset we prepared for the experiments and then compared the performance of each approach with the ROC curve and the area under it. A ROC curve presents the performance as a trade-off between sensitivity and specificity. It is a curve of true versus false positive rate when a threshold parameter is set. The area under the ROC curve (i.e., AUC) is widely accepted as an index of the accuracy for performance comparison. AUC values are within the range from 0 to 1. The higher an AUC value is, the more accurate a corresponding algorithm is.

Regarding the parameter settings in the experiments, for the prediction approach of [20], the parameter was set according to the value as recommended, i.e., γ, was set to be 0.25. Regarding [21], since there was no particular parameter value to recommend, the best setting for S-Kernel was determined experimentally by trial-and-error based on ROC performance. For SVM+CoFex, [20] and [21] that used SVM as classifier, the parameter setting w.r.t. SVM was the same so that the impact of SVM to the performance can be ignored for these methods. For CoFex, we set the maximum value k_{max} can take equal to 10 in the experiments. Given a PPI dataset, we iteratively chose the value of k_{max} varying from 2 to 10 with an interval 1 to determine an optimal setting of k_{max} that can maximize the performance of CoFex.

For the implementation, SVM were implemented with the libsvm package [34] while Random Forest and Naïve Bayes Classifier were implemented with Weka [42]. All approaches were implemented by JAVA.

5.3 Results and Discussions

To demonstrate the advantage of CoFex when applied to the task of predicting PPIs, the experiment results are presented in Fig. 3 and Table 2.

For the subfigures (a)-(f) of Fig. 3, each of them was corresponding to a kind of Yeast or Human PPI dataset as indicated by its title, and the best ROC curves obtained by prediction approaches were presented. Subfigures (g)-(i) show the actual ROC curves of all prediction approaches in the datasets of AT, EC and SP respectively.

1. http://bioinformatics.org.au/tools/go2ppi/

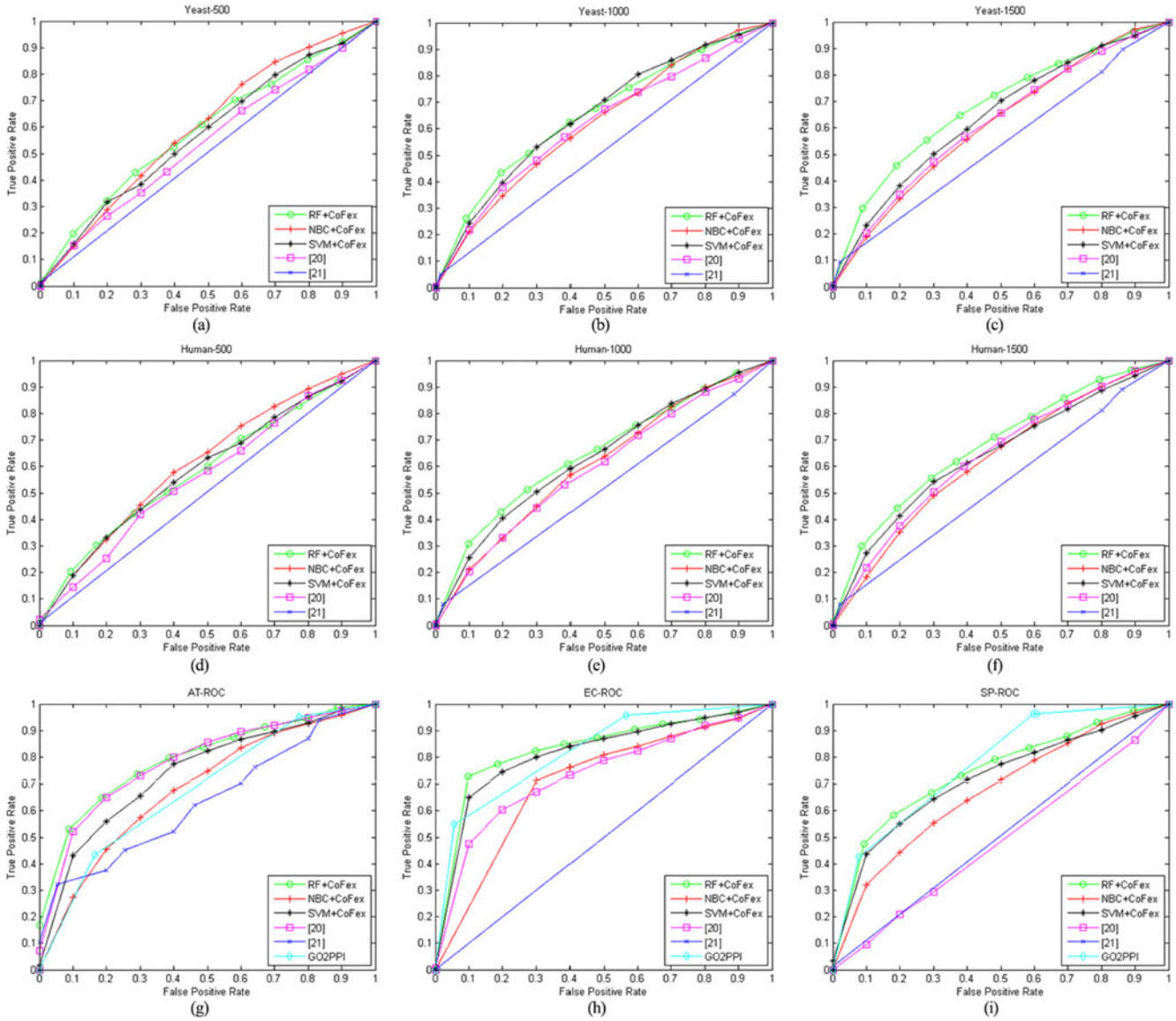

Fig. 3. (a)-(f) are the best ROC curves obtained by the prediction approaches in the datasets of Yeast-500, Yeast-1000, Yeast-1500, Human-500, Human-1000 and Human 1500, respectively; (g)-(i) are the ROC curves obtained by the prediction approaches in the datasets of AT, EC, and SP.

In Table 2, we used the average AUC values to show the performances of prediction approaches in the Yeast and Human PPI datasets and used the actual AUC values for the remaining datasets, i.e., AT, EC and SP. In addition, we tested GO2PPI using all datasets except the Yeast and Human datasets. In the Yeast and Human datasets, gene identifiers were used to refer to different proteins. However, as the current version of GO2PPI only supports Uniprot accession numbers, the GO information for the proteins in Yeast and Human datasets were therefore not obtainable. Hence, we only provided the performance statistics in Table 2 for GO2PPI with the datasets of AT, EC

TABLE 2
The AUC Values of Prediction Approaches for All Datasets

	RF+CoFex	NBC+CoFex	SVM+CoFex	[20]	[21]	GO2PPI
Yeast-500	$0.57^{(2nd)}$	$0.58^{(1st)}$	$0.57^{(2nd)}$	$0.55^{(3rd)}$	0.5	N/A
Yeast-1000	$0.65^{(1st)}$	$0.6^{(3rd)}$	$0.64^{(2nd)}$	$0.6^{(3rd)}$	0.51	N/A
Yeast-1500	$0.68^{(1st)}$	0.6	$0.63^{(2nd)}$	$0.61^{(3rd)}$	0.53	N/A
Yeast-500	$0.57^{(2nd)}$	$0.58^{(1st)}$	$0.58^{(1st)}$	$0.51^{(3rd)}$	0.5	N/A
Yeast-1000	$0.64^{(1st)}$	$0.59^{(3rd)}$	$0.63^{(2nd)}$	$0.59^{(3rd)}$	0.51	N/A
Yeast-1500	$0.7^{(1st)}$	0.6	$0.64^{(2nd)}$	$0.61^{(3rd)}$	0.53	N/A
AT	$0.8^{(1st)}$	0.7	$0.74^{(3rd)}$	$0.79^{(2nd)}$	0.63	0.69
EC	$0.87^{(1st)}$	$0.75^{(3rd)}$	$0.83^{(2nd)}$	0.74	0.5	$0.83^{(2nd)}$
SP	$0.76^{(2nd)}$	0.68	$0.72^{(3rd)}$	0.48	0.51	$0.78^{(1st)}$

and SP, as they used the Uniprot accession numbers to denote proteins.

From Table 2, we noted that RF+CoFex and SVM+CoFex had a very promising performance as they were consistently among the best three approaches for all datasets in terms of the AUC value. Especially for RF+CoFex, its AUC value in the EC dataset was as high as 0.87. Regarding the performance of NBC+CoFex, although it obtained the best performance in the datasets of Yeast-500 and Human-500 and was the third best approach in the datasets of Yeast-1000, Human-1000 and EC, it did not perform as well as either of RF+CoFex and SVM+CoFex in the other datasets but was still comparable to the third best approach in each of these datasets.

When comparing with the approaches, i.e., [20] and [21], that also used sequence information to predict PPIs, RF+CoFex and SVM+CoFex show the superior prediction accuracies as they outperformed [20] and [21] in almost all PPI datasets used in the experiments. For NBC+CoFex, though it was worse than [20] in some PPI datasets, the difference between NBC+CoFex and [20] was very small, which could be mostly ignored. Regarding the performance of [21], we noted that the AUC values of [21] were around 0.5 for all PPI datasets except AT. That is to say, [21] was only slightly better than a random classifier.

In the datasets of AT, EC and SP where PPIs were precisely selected, the performances of RF+CoFex, NBC+CoFex, SVM+CoFex, [20] and [21] had a considerable improvement when compared with their performances in the Yeast and Human datasets. Taking RF+CoFex as an example, its AUC value obtained in the EC dataset performed 26.5 and 22.9 percent better than the best AUC values RF+CoFex obtained in the Yeast and Human datasets respectively.

When comparing classifiers integrated with CoFex with GO2PPI, we noted from Table 2 that all of the classifiers integrated with CoFex performed better than GO2PPI in the AT dataset. But regarding the performance in the EC dataset, although RF+CoFex was still better than GO2PPI, NBC+CoFex performed worse than GO2PPI while the performance of SVM+CoFex was comparable to that of GO2PPI. In the SP dataset, GO2PPI was the best approach in terms of the AUC value, but the difference in the AUC value between RF+CoFex, which was the second best approach in the SP dataset, and GO2PPI was only 0.02. Hence, based on the results in Table 2, we found that most of classifiers integrated with CoFex had at least a comparable performance when compared with GO2PPI. Moreover, if we particularly focused our discussion on improving the performance of Random Forest that was also used by GO2PPI, a conclusion can be made that the integration with CoFex can better improve the prediction accuracies of Random Forest according to Table 2.

Other than the differences in prediction accuracies, another point worth noting is that not all the proteins we considered in the experiments had the GO information. In such cases, GO2PPI could not be applied to predict PPIs involving proteins whose GO information was not known. Comparing sequence information for proteins with GO information, the set of proteins with sequence information is more complete than that with GO information. In this regard, sequence-based approaches are more preferable to approaches that predict PPIs using GO information and this is especially the case when the interactions between new proteins are to be predicted.

Comparing SVM+CoFex with the prediction approaches of [20] and [21], we noted that all of them adopted SVM as the classifier to perform the prediction task. However, generally speaking, the prediction approaches of [20] and [21] did not perform as well as SVM+CoFex in our experiments. For yeast datasets, SVM+CoFex performed 7 and 19.5 percent better than [20] and [21] respectively. For human datasets, SVM+CoFex performed 6 and 20 percent better than [20] and [21] respectively. Such difference became even larger in the PPI datasets of EC and SP. We argued that it was the consideration of concatenating feature vectors of individual proteins that caused the unsatisfactory performance of using SVM for [20] and [21]. In order to concatenate feature vectors of proteins, [20] and [21] have to design specific kernel functions that are simple and also weak in terms of the ability of classifying objects. As has been pointed out by [41], simple sequence-based kernels do not predict PPIs as accurate as they claim. This conclusion is consistent with our experiment results. Since CoFex is capable of composing feature vectors for pairs of proteins by jointly considering the sequence information of proteins in a protein pair, it is possible for us to adopt a more complicated and well-recognized kernel, such as the sigmoid kernel used by SVM+CoFex, other than designing a specific but simple kernel for the purpose of concatenation.

When comparing RF+CoFex, NBC+CoFex and SVM+CoFex, a conclusion can be made that the use of Random Forest obtained the best performance for predicting PPIs especially when the training PPI datasets were more precise. After RF+CoFex, it was SVM+CoFex that performed as the second best method. In this regard, SVM was still a promising classifier for predicting PPIs. NBC+CoFex was the last one among the three classifiers integrated with CoFex. However, we noted that NBC+CoFex had a comparable performance on small datasets (i.e., Yeast-500 and Human-500) when compared with RF+CoFex and SVM+CoFex. Though better, the extent of improvement made by NBC+CoFex was not as significant as both RF+CoFex and SVM+CoFex when the size of dataset increased. Hence, if we want to choose a classifier to integrate with CoFex for the task of predicting PPIs, the Random Forest classifier is preferred.

For CoFex, it is seen that the performance of classifiers integrated with CoFex was generally better than [20] and [21] that also considered sequence information and also better than GO2PPI that made use of other information instead of sequence information. In this regard, we believe that the covariations of coevolving positions were effectively and accurately identified by CoFex and the resulting feature vectors were more useful for classifiers to predict PPIs.

Overall, classifiers integrated with CoFex had a very promising performance when applied to predict PPIs. That is to say, covariations of coevolving positions can facilitate the prediction of PPIs and CoFex is an efficient method of extracting such kind of coevolutionary information from protein sequences so as to compose feature vectors for pairs of proteins. Furthermore, among Random Forest, Naïve Bayes Classifier and SVM, Random Forest obtained the best performance, indicating that Random Forest was more appropriate than both SVM and Naïve Bayes Classifier for predicting PPIs.

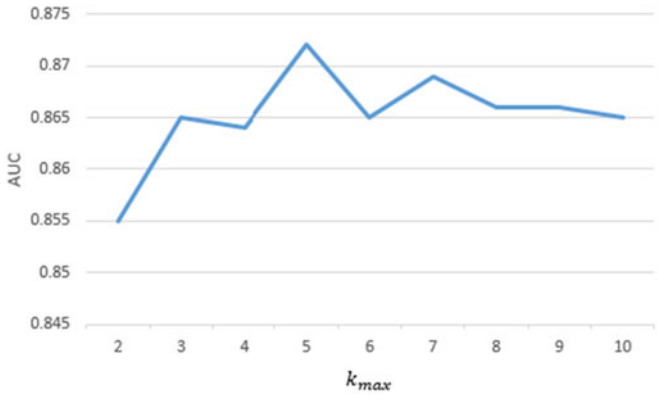

Fig. 4. The AUC performance of RF+CoFex in the EC dataset.

5.4 Computational Complexity of CoFex

Regarding the computational complexity of CoFex, As the most time-consuming part of CoFex is in the identification of all covariations in **CoV**, our discussion focused mainly on this process.

To identify all covariations with a specific length, i.e., k, the computational complexity of CoFex is comparable to the sequence-based approaches [20] and [21], as all of them have to scan all the protein sequences once. However, if the covariations with different lengths are taken into consideration, the increase in computation complexity of CoFex, with respect to magnitude of k, is linear. But yet, the computational time of this identification process can be relatively easily minimized. As a part of our future work, a parallel version of CoFex will be developed to speed up the process.

5.5 Sensitivity Analysis of k_{max} on the Performance of CoFex

According to the discussion in Section 5.4, the value of k_{max} plays an important role on the efficiency of CoFex, as a large value of k_{max} will result in a heavy computation of CoFex. Although we explain the strategy of how to select the best value of k_{max} in the Section 5.2, an in-depth analysis is given in this section to show how the performance of CoFex is affected by the changes in k_{max}. In the following analysis, we take the performance of RF+CoFex in the EC dataset as an example.

It is observed from Fig. 4 that the performance of RF+CoFex increased steadily with the value of k_{max} from 2 to 5 and the performance of RF+CoFex was at its best when k_{max} was set to 5. Furthermore, we also observed that the increasing of k_{max} beyond 5 did not lead to a better performance of RF+CoFex. A possible reason to explain it is that the information of covariations identified by CoFex when k_{max} was set to 5 was sufficient enough to generate a promising Random Forest classifier and hence covariations with length larger than 5 could not improve the performance of the Random Forest classifier any further.

While RF+CoFex performed at its best when k_{max} was set to 5 in the EC dataset, this was not the same with the other classifiers or datasets. However, according to our experiences learned from experiments, we found that the value of k_{max} with which CoFex performed at its best was less than 10 for most of cases, and hence we recommend the maximal value of k_{max} to be set as 10 when integrating CoFex with various classifiers for PPI prediction.

TABLE 3
The Percentage of Covariations that Passed P-Value Tests

	$p \leq 0.1$	$p \leq 0.05$	$p \leq 0.01$
Yeast-500	99.2%	99%	98.5%
Yeast-1000	99.4%	99.2%	99%
Yeast-1500	99.8%	99.6%	99.3%
Human-500	99.2%	99%	98.6%
Human-1000	99.5%	99.3%	99%
Human-1500	99.7%	99.6%	99.3%
AT	99.6%	99.5%	99%
EC	99.4%	99.1%	98.7%
SP	99.9%	99.7%	99.3%

5.6 Biological Significance of CoV

To determine the statistical significance of each covariation in **CoV**, we adopted a p-value test which is a popular statistical significance test commonly used in many applications, such as the identification of protein complexes [35] and social network analysis [23].

Assuming that we have a covariation $(\tau_i, \tau_j)_k \in \mathbf{CoV}$ and $\mathrm{DMI}((\tau_i, \tau_j)_k) > 0$, the p-value of $(\tau_i, \tau_j)_k$ can be computed as:

$$p\text{-value} = 1 - \sum_{i=1}^{n_{int}^{(\tau_i,\tau_j)_k}-1} \frac{\binom{n_{int}}{i}\binom{n_{\overline{int}}}{n_{int}^{(\tau_i,\tau_j)_k}-i}}{\binom{n_{int}+n_{\overline{int}}}{n_{int}^{(\tau_i,\tau_j)_k}}}. \quad (10)$$

For our analysis, covariations with p-values smaller than or equal to the significant thresholds of 0.1, 0.05 and 0.01 in the training set are of biological significance to indicate the existence or non-existence of interaction between proteins. The experimental results of p-value tests are presented in Table 3. It is noted that for almost all datasets, more than 99 percent covariations identified by CoFex were significant at all thresholds of p-values. Hence, for proteins with unknown interacting relationship, these covariations were believed to be able to facilitate prediction.

5.7 Comparison with Genomic and Evolutionary Features

To demonstrate the advantages of CoFex when compared with other genomic and evolutionary approaches for PPI prediction, what we have managed to do in this section is to show how effective CoFex can be by comparing the features it extracts for PPI prediction with other genomic and evolutionary features typically used with standard classifiers. For this purpose, we made use of LocFuse, which is developed based on an ensemble learning method that can allow us to use different genomic and proteomic features to predict unknown PPIs. It does so using an ensemble of four types of different classifiers including random forests, Naive Bayes, multi-layer perceptron and radial basis function network.

We conducted experiments on a independent testing set of human PPIs provided by LocFuse. The genomic and evolutionary features we chose to compare included Post Translational Modification (PTM) types, Tissue Terms

TABLE 4
Performance Comparison of Different Features in Predicting Human PPIs

Feature	Accuracy
CoFex	52.1%
PTM	27.6%
TSU	13.7%
CDN	41.6%

(TSU) and Codon Usage (CDN). Taking PTM features as an example, LocFuse extracts the feature vectors from the PTM database for all the proteins we used in our testing set. The PTM feature vectors for each pair of proteins in the testing set are then used as input to LocFuse so that we can obtain a score to indicate whether a pair of proteins interact with each other or not. Given a testing set of protein pairs, we can then find out, using LocFuse, how useful PTM feature vectors are in predicting PPIs. Note that since not all human proteins in the testing set have their sequence information found in the database of NCBI, we removed PPIs composed of proteins with unknown sequence information and finally 4,072 PPIs were retained in the testing set.

Based on the aforementioned procedure, we have obtained additional experimental results, comparing the feature sets extracted by CoFex, with those based on PTM, TSU and CDN. The results are shown in Table 4. From the table, despite the use of only one single classifier rather than an ensemble of four classifiers adopted by LocFuse, CoFex is better able to accurately predict PPIs with the sequence features that it extracted. In particular, CoFex performed 25.2 percent better than CDN which was ranked as the second best feature. When comparing the use of features extracted by CoFex with that of PTM and TSU in the prediction tasks, the difference in prediction accuracy was even more significant. The accuracy score of CoFex was more than twice as accurate. Based on these results, we conclude that CoFex is a promising feature extraction approach for predicting PPIs.

6 Conclusion

In this paper, we have studied the problem of extracting coevolutionary features from sequence information for the prediction of PPIs. A major concern w.r.t. existing sequence-based prediction methods is that they normally compose feature vectors for individual proteins so that a lot of efforts have been made to tackle the problem of concatenating feature vectors when adopting those well-known classifiers to predict PPIs. Hence, we consider the problem of feature extraction from an alternative view so that feature vectors can be composed for pairs of proteins instead of individual proteins. To do so, we focus our research on the coevolutionary features found in the sequence information of proteins in a protein pair. In particular, the coevolutionary features of interest are covariations of coevolving positions in protein sequences, which have been demonstrated to have a significant role for proteins to function well. To identify such covariations, we propose CoFex by considering covariations of coevolving positions as pairs of amino acids at different positions that are significantly frequently observed in protein sequences. The verified covariations are then weighted by CoFex so that their abilities of providing evidence to the interaction or non-interaction relationship of proteins can be indicated. Given a pair of proteins, the corresponding feature vector can be composed by considering the presence and absence of covariations in the sequences of the two proteins.

To evaluate the performance of CoFex, we integrated it with several well-known classifiers. The experiment results show that the coevolutionary information carried by the covariations at coevolving positions is useful for predicting PPIs and CoFex is an efficient approach to extract such kind of information from protein sequences, especially for a large amount of PPI data. Furthermore, among several well-known classifiers integrated with CoFex, since Random Forest outperformed the other two classifiers, Random Forest could be an alternative and promising classifier other than SVM that is widely adopted by existing sequence-based prediction approaches. Regarding the future work, since we are aware of the fact that the computational complexity of CoFex is linear to the size of different values of k when identifying all covariations, we will continue to develop a parallel version of CoFex so that covariations with different lengths can be identified in separate processes.

Acknowledgments

This work was supported by the Fundamental Research Funds for the Central Universities under Grant No. 2016IVA051.

References

[1] T. Pawson and P. Nash, "Assembly of cell regulatory systems through protein interaction domains," *Sci. Signaling*, vol. 300, no. 5618, p. 445, 2003.

[2] M. R. Wilkins, "Hares and tortoises: The high-versus low-throughput proteomic race," *Electrophoresis*, vol. 30, no. S1, pp. S150–S155, 2009.

[3] C. Chien, P. L. Bartel, R. Sternglanz, and S. Fields, "The two-hybrid system: A method to identify and clone genes for proteins that interact with a protein of interest," *Proc. Nat. Acad. Sci. USA*, vol. 88, no. 21, pp. 9578–9582, 1991.

[4] T. Ito, T. Chiba, R. Ozawa, M. Yoshida, M. Hattori, and Y. Sakaki, "A comprehensive two-hybrid analysis to explore the yeast protein interactome," *Proc. Nat. Acad. Sci. USA*, vol. 98, no. 8, pp. 4569–4574, 2001.

[5] Y. Ho, et al., "Systematic identification of protein complexes in saccharomyces cerevisiae by mass spectrometry," *Nature*, vol. 415, no. 6868, p. 180, 2002.

[6] L. Trinkle-Mulcahy, S. Boulon, Y. W. Lam, R. Urcia, F. Boisvert, F. Vandermoere, N. A. Morrice, S. Swift, U. Rothbauer, and H. Leonhardt, "Identifying specific protein interaction partners using quantitative mass spectrometry and bead proteomes," *J. Cell Biol.*, vol. 183, no. 2, pp. 223–239, 2008.

[7] M. F. Templin, D. Stoll, J. M. Schwenk, O. Pötz, S. Kramer, and T. O. Joos, "Protein microarrays: Promising tools for proteomic research," *Proteomics*, vol. 3, no. 11, pp. 2155–2166, 2003.

[8] A. W. Rives and T. Galitski, "Modular organization of cellular networks," *Proc. Nat. Acad. Sci. USA*, 2003, vol. 100, no. 3, pp. 1128–1133.

[9] J. R. Parrish, K. D. Gulyas, and R. L. Finley Jr., "Yeast two-hybrid contributions to interactome mapping," *Curr. Opin. Biotechnol*, vol. 17, no. 4, pp. 387–393, 2006.

[10] A. J. Enright, I. Iliopoulos, N. C. Kyrpides, and C. A. Ouzounis, "Protein interaction maps for complete genomes based on gene fusion events," *Nature*, vol. 402, no. 6757, pp. 86–90, 1999.

[11] T. Dandekar, B. Snel, M. Huynen, and P. Bork, "Conservation of gene order: A fingerprint of proteins that physically interact," *Trends Biochem. Sci.*, vol. 23, no. 9, pp. 324–328, 1998.

[12] R. Jansen, H. Yu, D. Greenbaum, Y. Kluger, N. J. Krogan, S. Chung, A. Emili, M. Snyder, J. F. Greenblatt, and M. Gerstein, "A bayesian networks approach for predicting protein-protein interactions from genomic data," *Science*, vol. 302, no. 5644, pp. 449–453, 2003.
[13] C. Goh, A. A. Bogan, M. Joachimiak, D. Walther, and F. E. Cohen, "Co-evolution of proteins with their interaction partners," *J. Mol. Biol.*, vol. 299, no. 2, pp. 283–293, 2000.
[14] M. Pellegrini, E. M. Marcotte, M. J. Thompson, D. Eisenberg, and T. O. Yeates, "Assigning protein functions by comparative genome analysis: Protein phylogenetic profiles," *Proc. Nat. Acad. Sci. USA*, vol. 96, no. 8, pp. 4285–4288, 1999.
[15] X. W. Chen and M. Liu, "Prediction of protein-protein interactions using random decision forest framework," *Bioinf.*, vol. 21, no. 24, pp. 4394–4400, 2005.
[16] S. P. Kanaan, C. Huang, S. Wuchty, D. Z. Chen, and J. A. Izaguirre, "Inferring protein–protein interactions from multiple protein domain combinations," *Comput. Syst. Biol.*, vol. 541, pp. 43–59, 2009.
[17] Z. H. You, Y. K. Lei, J. Gui, D. S. Huang, and X. Zhou, "Using manifold embedding for assessing and predicting protein interactions from high-throughput experimental data," *Bioinf.*, vol. 26, no. 21, pp. 2744–2751, 2010.
[18] J. R. Bock and D. A. Gough, "Predicting protein–protein interactions from primary structure," *Bioinf.*, vol. 17, no. 5, pp. 455–460, 2001.
[19] S. Martin, D. Roe, and J. L. Faulon, "Predicting protein-protein interactions using signature products," *Bioinf.*, vol. 21, no. 2, pp. 218–226, 2005.
[20] A. Ben-Hur and W. S. Noble, "Kernel methods for predicting protein–protein interactions," *Bioinf.*, vol. 21, suppl. 1, pp. i38–i46, 2005.
[21] J. Shen, J. Zhang, X. Luo, W. Zhu, K. Yu, K. Chen, Y. Li, and H. Jiang, "Predicting protein–protein interactions based only on sequences information," *Proc. Nat. Acad. Sci. USA*, vol. 104, no. 11, pp. 4337–4341, 2007.
[22] S. Pitre, F. Dehne, A. Chan, J. Cheetham, A. Duong, A. Emili, M. Gebbia, J. Greenblatt, M. Jessulat, N. Krogan, X. Luo, and A. Golshani, "PIPE: A protein-protein interaction prediction engine based on the re-occurring short polypeptide sequences between known interacting protein pairs," *BMC Bioinf.*, vol. 7, pp. 365, 2006.
[23] P. Due, B. Holstein, R. Lund, J. Modvig, and K. Avlund, "Social relations: Network, support and relational strain," *Soc. Sci. Med.*, vol. 48, no. 5, pp. 661–673, 1999.
[24] E. M. Marcotte, M. Pellegrini, M. J. Thompson, T. O. Yeates, and D. Eisenberg, "A combined algorithm for genome-wide prediction of protein function," *Nature*, vol. 402, no. 6757, pp. 83–86, 1999.
[25] F. Pazos and A. Valencia, "Similarity of phylogenetic trees as indicator of protein-protein interaction," *Protein Eng.*, vol. 14, no. 9, pp. 609–614, 2001.
[26] M. A. Mahdavi and Y. Lin, "Prediction of protein-protein interactions using protein signature profiling," *Genomics, Proteomics Bioinf.*, vol. 5, no. 3, pp. 177–186, 2007.
[27] M. Deng, S. Mehta, F. Sun, and T. Chen, "Inferring domain-domain interactions from protein-protein interactions," *Genome Res.*, vol. 12, no. 10, pp. 1540–1548, 2002.
[28] K. C. C. Chan, A. K. C. Wong, and D. K. Y. Chiu, "Learning sequential patterns for probabilistic inductive prediction," *IEEE Trans. Syst., Man Cybern.*, vol. 24, pp. 1532–1547, Oct. 1994.
[29] J. A. Suykens and J. Vandewalle, "Least squares support vector machine classifiers," *Neural Process Lett.*, vol. 9, no. 3, pp. 293–300, 1999.
[30] Y. Park, "Critical assessment of sequence-based protein-protein interaction prediction methods that do not require homologous protein sequences," *BMC Bioinf.*, vol. 10, no. 1, p. 419, 2009.
[31] I. Xenarios, L. Salwínski, X. J. Duan, P. Higney, S. Kim, and D. Eisenberg, "DIP, the database of interacting proteins: A research tool for studying cellular networks of protein interactions," *Nucleic Acids Res.*, vol. 30, no. 1, pp. 303–305, 2002.
[32] T. K. Prasad, R. Goel, K. Kandasamy, S. Keerthikumar, S. Kumar, S. Mathivanan, D. Telikicherla, R. Raju, B. Shafreen, and A. Venugopal, "Human protein reference database-2009 update," *Nucleic Acids Res.*,vol. 37, suppl. 1, pp. D767–D772, 2009.
[33] D. L. Wheeler, T. Barrett, D. A. Benson, S. H. Bryant, K. Canese, V. Chetvernin, D. M. Church, M. DiCuccio, R. Edgar, and S. Federhen, "Database resources of the national center for biotechnology information," *Nucleic Acids Res.*, vol. 35, suppl. 1, pp. D5–D12, 2007.
[34] C. Chang and C. Lin, "LIBSVM: A library for support vector machines," *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 3, p. 27, 2011.
[35] A. Hu and K. Chan, "Utilizing both topological and attribute information for protein complex identification in PPI networks," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 10, no. 3, pp. 780–792, May 2013.
[36] A. Valencia and F. Pazos, "Prediction of protein-protein interactions from evolutionary information," *Methods Biochemical Anal.*, vol. 44, pp. 411–426, 2003.
[37] S. Chakrabarti and A. R. Panchenko, "Coevolution in defining the functional specificity," *Proteins: Struct., Function, Bioinf.*, vol. 75, no. 1, pp. 231–240, 2009.
[38] G. B. Gloor, L. C. Martin, L. M. Wahl, and S. D. Dunn, "Mutual information in protein multiple sequence alignments reveals two classes of coevolving positions," *Biochemistry*, vol. 44, no. 19, pp. 7156–7165, 2005.
[39] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.
[40] N. Friedman, D. Geiger, and M. Goldszmidt, "Bayesian network classifiers," *Mach. Learn.*, vol. 29, nos. 2-3, pp. 131–163, 1997.
[41] J. Yu, M. Guo, C. J. Needham, Y. Huang, L. Cai, and D. R. Westhead, "Simple sequence-based kernels do not predict protein–protein interactions," *Bioinf.*, vol. 26, no. 20, pp. 2610–2614, 2010.
[42] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The WEKA data mining software: an update," *ACM SIGKDD Explorations Newslett.*, vol. 11, no. 1, pp. 10–18, 2009.
[43] D. B. Smith and S. J. Kevin, "Single-step purification of polypeptides expressed in Escherichia coli as fusions with glutathione S-transferase," *Gene*, vol. 67, no. 1, pp. 31–40, 1988.
[44] G. Rigaut, A. Shevchenko, B. Rutz, M. Wilm, M. Mann, and B. Séraphin, "A generic protein purification method for protein complex characterization and proteome exploration," *Nature Biotechnol.*, vol. 17, no. 10, pp. 1030–1032, 1999.
[45] J. Zahiri, O. Yaghoubi, M. Mohammad-Noori, R. Ebrahimpour, and A. Masoudi-Nejad, "*PPIevo*: Protein–protein interaction prediction from PSSM based evolutionary information," *Genomics*, vol. 102, no. 4, pp. 237–242, 2013.
[46] S. R. Maetschke, M. Simonsen, M. J. Davis, and M. A. Ragan, "Gene ontology-driven inference of protein–protein interactions using inducers," *Bioinf.*, vol. 28, no. 1, pp. 69–75, 2012.
[47] A. Franceschini, D. Szklarczyk, S. Frankild, M. Kuhn, M. Simonovic, A. Roth, J. Lin, et al., "STRING v9. 1: Protein-protein interaction networks, with increased coverage and integration," *Nucleic Acids Res.*, vol. 41, no. D1, pp. D808–D815, 2013.
[48] J. Zahiri, M. Mohammad-Noori, R. Ebrahimpour, S. Saadat, J. H. Bozorgmehr, T. Goldberg, and A. Masoudi-Nejad, "LocFuse: Human protein–protein interaction prediction via classifier fusion using protein localization information," *Genomics*, vol. 104, no. 6, pp. 496–503, 2014.

Lun Hu received the BEng degree from the Department of Control Science and Engineering, Huazhong University of Science and Technology, Wuhan, China, in 2006 and the MSc and PhD degrees from the Department of Computing, The Hong Kong Polytechnic University, Hong Kong, China, in 2008 and 2015, respectively. He is currently an assistant professor with the School of Computer Science and Technology, Wuhan University of Technology, Wuhan, Hubei, China. His research interests include data mining algorithms and applications to graph clustering and bioinformatics

Keith C. C. Chan received the BMath (Hons.) degree in computer science and statistics and the MASc and PhD degrees in systems design engineering from the University of Waterloo, Canada. He joined IBM Canada Laboratory soon after graduation; and since 1994, he has been with The Hong Kong Polytechnic University, where he is currently a professor in the Department of Computing. His research interest is in bioinformatics, data mining, computational intelligence, and software engineering. He has over 250 publications in referred journals and conferences in these areas.