FINDING LINEAR MOTIF PAIRS FROM PROTEIN INTERACTION NETWORKS: A PROBABILISTIC APPROACH

Henry C.M. Leung *, M. H. Siu, S.M. Yiu and Francis Y.L. Chin

Department of Computer Science, The University of Hong Kong
Pokfulam Road, Hong Kong
{cmleung2, mhsiu, smyiu, chin}@cs.hku.hk

Ken W.K. Sung

Department of Computer Science, National University of Singapore
Singapore
ksung@comp.nus.edu.sg

Abstract. Finding motif pairs from a set of protein sequences based on the protein-protein interaction data is a challenging computational problem. Existing effective approaches usually rely on additional information such as some prior knowledge on protein groupings based on protein domains. In reality, this kind of knowledge is not always available. Novel approaches without using this knowledge is much desirable. Recently, Tan *et al.* [10] proposed such an approach. However, there are two problems with their approach. The scoring function (using χ^2 testing) used in their approach is not adequate. Random motif pairs may have higher scores than the correct ones. Their approach is also not scalable. It may take days to process a set of 5000 protein sequences with about 20,000 interactions. In this paper, our contribution is two-fold. We first introduce a new scoring method, which is shown to be more accurate than the χ-score used in [10]. Then, we present two efficient algorithms, one exact algorithm and a heuristic version of it, to solve the problem of finding motif pairs. Based on experiments on real datasets, we show that our algorithms are efficient and can accurately locate the motif pairs. We have also evaluated the sensitivity and efficiency of our heuristics algorithm using simulated datasets, the results show that the algorithm is very efficient with reasonably high sensitivity.

1. INTRODUCTION

In the cell of all organisms, protein-protein interactions occur in the structure of sub-cellular organelles, the transport machinery across different membranes, packaging of chromatin, the network of sub-membrane filaments, signal transduction and regulation of gene expression, etc. Aberrant protein-protein interactions have been linked to a number of neurological disorders such as Alzheimer's disease [12], Muscular Dystrophy [4] and Huntington's disease [2]. Because of their importance, much research has been performed in order to understand the mechanism of protein-protein interactions.

Protein is a sequence of amino acids with 3D structure. Some subsequences of a protein will form sub-structures, called domains, on the surface of the 3D structure. These domains characterize the functions of each protein by controlling what kind of molecules this protein will bind to. When two proteins bind together (*interact*), their domains will exchange charges to form bonds which stabilize the protein-protein complex. For example, proteins with Src homology 3 (SH3) domain (GxxPxNY) usually bind to proteins with polyproline type II helical structure (PxxP). We call GxxPxNY and PxxP a *binding motif pair* or motif pair in short. Discovering motif pairs helps us understand many protein-involved mechanisms and predict the functions of a protein. Biological experiments such as site-directed mutagenesis [3] and phage display [5] are available for discovering motif pairs. However, these experiments are both laborious and expensive.

If we are given the protein-protein interaction data (e.g. the DIP (Database of Interacting Proteins) database [14]), discovering the motif pair is more difficult than discovering the motif of the binding sites of co-regulated DNA sequences, which has been well studied in the literature[1,6-8], because of the following issues:

* Corresponding author.

1. Protein sequences are composed of 20 amino acids whereas DNA sequences are comprised of 4 nucleotides. Therefore, it is more computationally involved when we work with protein sequences.
2. For co-regulated DNA sequences, it is assumed that the given DNA sequences contain over-represented subsequences of a similar pattern (motif). However, we can only identify the pair of similar patterns (motif pair) if the set of relevant protein-protein interactions are isolated. This might not be easy as there are many possible subsets of interactions. Even when the set of n interactions can be isolated, there are still 2^{n-1} ways of grouping the protein sequences to identify the motif pair.
3. Usually there is more information available for discovering DNA motif since, for example, the sequences without the binding sites (control set) can provide extra information for solving the problem. However, missing interactions between two protein sequences in the database do not imply the non-existence of motif pairs because the missing protein-protein interaction data might be due to the lack of experiments between these pair of proteins.

A naive approach is to fix a particular protein and the group of proteins that are known to bind to this protein. Then, identify the motif from the group of protein sequences. This can be done using standard motif discovery tools such as MEME [1] or Weeder [8]. And then also find a motif pattern that can uniquely identify the particular protein to form the motif pair. However, this method works only when the number of protein sequences that bind to the same protein is large, say > 4. In practice, it is usually not the case. In fact, even when a particular protein can bind to a group of many proteins, those bindings might be due to many factors, not just a single motif pair. The problem of finding a motif pair is then reduced to finding a sequence pattern that can uniquely identify that particular protein and also an over-represented sequence pattern in a *subset* (not necessary all) of proteins in the group. Since that particular protein might be uniquely identified by more than one pattern and there can be many subsets of proteins in the group whose sequences have similar patterns, the number of possible motif pairs can be many. So, it is impossible to determine the correct motif pair, if it exists, which initiates the interactions.

To handle the above problem, [9] proposed to take advantage of prior knowledge of protein groupings according to protein domains. Instead of considering the bindings between a particular sequence and a group of proteins, they consider the bindings between a group of protein sequences and another group of sequences with a particular domain so as to increase the number of sequences in the instance. A modified Gibbs sampling algorithm was developed to identify the motif pair. This method will work if we already know one of the motifs in the motif pair; otherwise, how to isolate two groups of proteins that are related to the same motif pair from the interaction database is non-trivial.

Recently, Tan *et al*. [10] introduced a method to discover motif pairs without knowledge of motifs participating in the motif pair and without any prior knowledge on the protein groupings. The basic idea of their approach is as follows. Based on the input sequences, they generated all possible substring pairs of a certain length from any two interacting sequences. For each possible motif pair, they identified the two groups of proteins that contain an instance of the motifs. They compare the number of observed interactions between these two groups and the expected number of interactions using χ^2 testing. The motif pairs with the highest χ-score (implying the observed number of interactions is much larger than the expected number) were reported. They developed two algorithms, D-MOTIF and D-STAR, for discovering binding motif pairs based on this idea. D-MOTIF can discover the motif pair with the highest χ-score while D-STAR is a heuristics algorithm.

There are several problems with this approach. We found out that the χ-score is not an adequate measure for motif pairs. Since the expected number of interactions decreases with the number of sequences that contain the two motifs, algorithms using χ^2 testing tend to discover binding motif pairs that occur only in a few sequences. For example, when there is only one sequence with motif M_1 and only one sequence with motif M_2, if there is an interaction between these two sequences, the χ-score will be high. However, M_1 and M_2 are not statistically significant as they occur in one sequence

only. This is also the reason why their approach requires two input minimum thresholds for both the number of sequences containing each motif and the number of interactions between these sequences. Moreover, they assume that the interactions are uniformly distributed in the input sequences, which may not be correct since the interaction experiments may be biased to some sequences depending on the choices of the researchers. As a result, χ-score may not be a good measure in this type of studies[†].

Both algorithms, D-MOTIF and D-STAR, are not practical for large data sets. For D-MOTIF, as the minimum number of interactions of at least 3 is assumed, all possible motif pairs from each interaction triplet (that is, any three pairs of sequences that are known to bind) are considered. Based on these motif pairs, they isolate the two subsets of proteins that contain the motif for calculating the χ-score. In the worst case, D-MOTIF runs in $O(m^3(n\ (|\Sigma| - 1)^d)^6)$ time where m is the number of interactions, n is the length of a sequence, and Σ is the alphabet (assuming that l and d are small). D-MOTIF takes a long running time and requires a huge amount of memory even for a data set of about a hundred sequences. On the other hand, D-STAR is a heuristic version of D-MOTIF and does not consider all possible motif pairs. In their study, they adopted the *mismatch* (l, d)-motif model (that is, the motif is of length l and the instances differ from the motif by d mismatches). Based on the observation that if we considered all sequences containing a substring y with at most $2d$ mismatches from an instance x of a real motif M, we include all instances of M. They, therefore, only consider the substrings that occur in the input sequences to isolate the two subsets of proteins. However, they might include a lot of noisy instances in the subsets. Although D-STAR can discover binding motif pairs from a data set with a hundred sequences in minutes, it takes days when the number of sequences is increased to a thousand. D-STAR runs in $O(m^2n^2 + mtn^2)$ time where t is the number of sequences.

Our contributions: In this paper, we introduce a new scoring method by calculating the probability that the observed number of interactions is generated under a null hypothesis. If this probability is small, the null hypothesis is incorrect for the pair of motifs and these two motifs are likely to be a motif pair. This scoring method tends to discover motif pairs that occur in many sequences (instead of one sequence) with a large number of observed interactions. The scoring function resolves the problems of χ-score and does not require any pre-set thresholds. Experimental results on real biological data show that our scoring method can model binding motif pairs better than χ^2 testing.

We propose to use the *wildcard* motif (l, d)-model and have developed an exact algorithm, FindMotif(l, d, r), to find the top r motif pairs with the highest scores and a heuristic algorithm, MotifHeuristics(l, d, r), to find r motif pairs which are guaranteed to be local optimal solutions. The exact algorithm runs in $O(m^2n^2)$ time while the heuristic version runs in $O(rm^2n)$ time where r is the number of random seeds used in the algorithm. Usually l and d are small and r is around 200, so both our exact and heuristic algorithms run faster than D-MOTIF and D-STAR, respectively. In practice, MotifHeuristics can discover motif pairs for more than 5000 protein sequences only in about 20 minutes. For the exact algorithm, it only takes about half an hour to handle a data set of about 150 sequences and 230 interactions. We have also evaluated MotifHeuristics using simulated datasets, the results show that MotifHeuristics is efficient with reasonably high sensitivity.

This paper is organized as follows. We will describe our scoring method and the formal problem definition in Section 2. In Section 3, we will describe our algorithms for discovering motif pairs. Experimental results on real biological data will be shown in Section 4. Section 5 concludes the paper.

2. PROBLEM DEFINITION

2.1. Motif Representation

Protein is a sequence of amino acids which can be represented by a sequence of symbols in Σ =

[†] As an example, based on the same SH3 domains dataset used in [1], which consists of about 150 protein sequences and 230 interactions, instead of using their heuristics algorithm, we exhausted all possible substring pairs using the same set of parameters in [1]. We computed the χ-score of each pair. We found that all motif pairs that are similar to the ones reported by [1] are ranked 90 or below. In other words, the correct motif pair can only be found in rank 90 or below.

{'A'..'Z'} – {'B', 'J', 'O', 'U', 'X', 'Z'}. Proteins with similar function usually contain similar substrings. These substrings can be modeled by an abstract representation called *motif*. In this paper, we define (*l*, *d*)-motif to be a length-*l* string with *d* wildcard symbols, denoted by 'x', and $l - d$ symbols in Σ. For example, "GxxPxNY" is a (7,3)-motif. A (*l*,*d*)-motif *M* represents those length-*l* substrings σ which are the same as *M* but with each wildcard symbol 'x' replaced by a symbol in Σ. Each string σ is called an *instance* of motif *M*. For example, "GACPQNY" is an instance of the motif "GxxPxNY".

2.2. (*l*, *d*)-Motif Pair Finding Problem

Given a set *S* of *t* protein sequences and a set $I \subseteq \{\{s_i, s_j\} \mid s_i, s_j \in S\}$ of *m known* interactions between sequences in *S*, we want to discover a pair of motifs M_1 and M_2 such that the set of instances of M_1 (denoted as $S(M_1)$) interact with the set of instances of M_2 (denoted as $S(M_2)$). Note that since many interactions between sequences in *S* are still unknown and there are no interactions between some pairs of sequences, there may not be interactions between all sequences in $S(M_1)$ and $S(M_2)$.

2.2.1. *χ-score*

Tan *et al*. [10] discovered the motif pair by comparing the observed number of interactions with the expected number of interactions for every motif pair. Given a motif pair M_1 and M_2, they calculated the expected number $E(M_1,M_2)$ of interactions between sequences in $S(M_1)$ and $S(M_2)$ by assuming the *m* known interactions are uniformly distributed in the *t* input sequences. They compared $E(M_1,M_2)$ with the observed number $O(M_1,M_2)$ of interactions between sequences in $S(M_1)$ and $S(M_2)$ by χ^2 testing.

$$\chi\text{-score} = \frac{(O(M_1,M_2) - E(M_1,M_2))^2}{E(M_1,M_2)}$$

Tan *et al*. [10] discovered motif pairs by considering those pairs with high χ-scores. However, using χ-score has two main weaknesses.

1. **Large χ-score when $S(M_1)$ and $S(M_2)$ are small.** If the sizes of $S(M_1)$ and $S(M_2)$ are small, since the value of $E(M_1,M_2)$ is extremely small, even when there is only one interaction between sequences in $S(M_1)$ and $S(M_2)$, the χ-score can be very large. For example, in a database with 5000 sequences and 20000 interactions, the value of $E(M_1,M_2)$ for a motif pair M_1 and M_2 with one sequence in $S(M_1)$ and $S(M_2)$ respectively is 0.0016. If there is an interaction between these two sequences, the χ-score will be 623. Tan *et al*. try to solve this problem by limiting the sizes of $S(M_1)$ and $S(M_2)$ to be bigger than a threshold. However, different thresholds should be used for different input data and they tend to discover motif pairs with the size of $S(M_1)$ and $S(M_2)$ being the same as the threshold.
2. **Number of interactions is not uniformly distributed.** Since biologists usually perform experiments on some special proteins, these proteins participate in more known interactions than other proteins, e.g. YBL063W protein participates in 283 known interactions while YJR091C protein participates in 1 known interaction only. Therefore, the assumption that the *m* known interactions are uniformly distributed in the *t* input sequences is incorrect. The method by Tan *et al*. might discover the wrong motif pairs and at the same time might miss the correct ones.

2.2.2. *p-score*

Instead of using χ-score as the scoring function, we calculate the probability that there are $O(M_1,M_2)$ or more interactions between sequences in $S(M_1)$ and $S(M_2)$ based on a null hypothesis (described below). If this probability is small, the null hypothesis cannot model the motif pair M_1 and M_2 and the motif pair is statistically significant.

Given a motif M_1, let $I(M_1) \subseteq I$ be the interactions involving sequences in $S(M_1)$ and $T(M_1)$ ($T(M_1)$ be the set of sequences that interact with sequences in $S(M_1)$. As the number of ways of distributing *x* objects onto *y* boxes ("onto" means at least one object per box) is $\binom{x-1}{y-1}$ and the null hypothesis assumes that the $|I(M_1)|$ interactions are uniformly distributed onto $T(M_1)$, the conditional probability that there are exactly *i* interactions between sequences in $S(M_1)$ and $S(M_2)$ given $I(M_1)$ and $T(M_1)$ can be calculated as follows (to be precise, only the value of $|T(M_1)|$, $|T(M_1)$ ($S(M_2)|$, and

$|T(M_1) - S(M_2)|$ are needed in the calculation). The conditional probability

$$p_i = P(O(M_1,M_2) = i \mid T(M_1), I(M_1), S(M_1), S(M_2)) = \frac{\binom{i-1}{|T(M_1) \cap S(M_2)-1}\binom{|I(M_1)|-i-1}{|T(M_1)-S(M_2)|-1}}{\binom{|I(M_1)|-1}{|T(M_1)|-1}}$$

Note that in the numerator of the equation, the i interactions are distributed uniformly onto $T(M_1) \cap S(M_2)$ and the remaining $|I(M_1)| - i$ interactions onto $T(M_1) - S(M_2)$. The conditional probability that there are $O(M_1,M_2)$ or more interactions between sequences in $S(M_1)$ and $S(M_2)$ given $I(M_1)$ and $T(M_1)$ can be calculated by summing up all possible i from $O(M_1,M_2)$ to $U = \min\{I(M_1) - |T(M_1) - S(M_2)|, |S(M_1)| \times |T(M_1) \cap S(M_2)|\}$, where i is upper bounded by two cases: (1) when each sequence in $T(M_1) - S(M_2)$ is involved in exactly one interaction, and (2) the maximum number of interactions between $S(M_1)$ and $T(M_1) \cap S(M_2)$. We have $P_1 = P(O(M_1, M_2) \geq i \mid S(M_1), T(M_1), I(M_1), S(M_2)) = \sum_{i=O(M_1,M_2)}^{U} p_i$.

Similarly, we can calculate the conditional probability $P_2 = P(O(M_1, M_2) \geq i \mid S(M_2), T(M_2), I(M_2), S(M_1))$ that there are i or more interactions between sequences in $S(M_1)$ and $S(M_2)$ given that the sequences in $S(M_2)$ participate in $|I(M_2)|$ interactions with sequences in $T(M_2)$. The p-score of a motif pair $\{M_1, M_2\}$ can be represented by the conditional probability $P_0 = P(O(M_1, M_2) \geq i \mid S(M_1), T(M_1), I(M_1), S(M_2), T(M_2), I(M_2))$. However, P_0 cannot be calculated easily. We estimate the value of P_0 by the following equation:

$$P_0 \sim P_1 \bullet \frac{P(T(M_2), I(M_2) \mid O(M_1,M_2) \geq i, S(M_2), S(M_1))}{P(T(M_2), I(M_2) \mid S(M_2), S(M_1))}$$

When the values of $P(T(M_2), I(M_2) \mid O(M_1, M_2) \geq i, S(M_2), S(M_1))$ and $P(T(M_2), I(M_2) \mid S(M_2) , S(M_1))$ are the same, P_0 will be exactly the same as P_1. Let $\hat{O}(M_2) = (|I(M_2)|/ |T(M_2)|) |T(M_2) \cap S(M_1)|$ be the expected number of interactions between $S(M_2)$ and $T(M_2) \cap S(M_1)$. When $i / \hat{O}(M_2)$ is small, the difference between $P(T(M_2), I(M_2) \mid O(M_1, M_2) \geq i, S(M_2), S(M_1))$ and $P(T(M_2), I(M_2) \mid S(M_2), S(M_1))$ is small. Therefore, P_0 can be approximated by P_1 when $i / \hat{O}(M_2)$ is small. Similarly, P_0 can be approximated by P_2 when $i / \hat{O}(M_1)$ is small. Thus we compare the values of $i / \hat{O}(M_2)$ and $i / \hat{O}(M_1)$. We use P_1 (P_2) to approximate the p-score of (M_1, M_2) when $i / \hat{O}(M_2)$ ($i / \hat{O}(M_1)$) is smaller.

A motif pair M_1 and M_2 with small p-score means that $M_1(M_2)$ has unexpectedly large number of interactions with $M_2(M_1)$. Therefore, we discover motif pairs by searching for pairs of motifs M_1 and M_2 with small p-scores.

Using p-score as the scoring function overcomes the two weaknesses of χ-score. Firstly, motif pair M_1 and M_2 with small size usually has a large p-score, e.g. when both $S(M_1)$ and $S(M_2)$ contain one sequence and there is an interaction between them, p-score$(M_1,M_2) = 1$. Therefore, by using p-score as the scoring function, we tend to discover motif pair M_1 and M_2 when there are an unexpectedly large number of interactions between large sets sequences $S(M_1)$ and $S(M_2)$. Secondly, by using $T(M_1)$, $T(M_2)$, $S(M_1)$, $S(M_2)$ in calculating the p-score, we do not rely on the assumption that the interactions are uniformly distributed among *all* input sequences.

3. ALGORITHM

3.1. Exact Algorithm

In this section, we propose an exact algorithm, FindMotif(l, d, r), to identify the top r (l, d)-motif pairs with the lowest p-scores. The algorithm is based on the idea of voting. Based on the scoring function we proposed in Section 2, we require the following information to calculate the score of each possible motif pairs. In the following, we assume that l and d are small, so $\binom{l}{d}$ is a constant.

1. For each motif M,
   - $N_S[M]$: the number of sequences containing an instance of M, $|S(M)|$;
   - $N_I[M]$: the total number of interactions for sequences in $S(M)$, $|I(M)|$;
   - $N_T[M]$: the number of sequences interacting with the sequences in $S(M)$, $|T(M)|$.
2. For each pair of motifs M_1 and M_2,
   - $N_O[M_1, M_2]$: the total number of interactions between sequences in $S(M_1)$ and those in $S(M_2)$, that is, the observed interactions, $O(M_1, M_2)$;
   - $C[M_1, M_2]$: the number of sequences in $S(M_2)$ that interact with sequences in $S(M_1)$, $|T(M_1) \cap S(M_2)|$.

For each length-l substring s in the input sequence, we add a vote to $N_S[M]$ for which s is an instance of the

motif M. Note that there are $\binom{l}{d}$ such M's. Each $N_S[M]$ can get at most one vote from each input sequence. $N_S[M]$ can be computed in $O(tn)$ time where t is the number of sequences and n is the length of each sequence.

To compute N_I, N_T, N_O, and C, we do the voting as follows. For each interaction $\{s_i, s_j\} \in I$, for every length-l substring x in s_i, we add a vote to $N_I(M)$ and a vote to $N_T(M)$ where x is an instance of M. Each $N_I(M)$ can get at most one vote from *each interaction*. Each $N_T(M)$ can get at most one vote for *every sequence* s_j. If s_i and s_j are two different sequences, for every length-l substring y in s_j, we add a vote to $N_I(M)$ if $N_I(M)$ has not yet received a vote from the same interaction. And we add a vote to $N_T(M)$ where y is an instance of M if $N_T(M)$ has not yet received a vote from s_i. For every two length-l substrings x and y (in s_i and s_j, respectively), we add a vote to $N_O[M_1, M_2]$ and a vote to $C[M_1, M_2]$ and $C[M_2, M_1]$ where x and y are instances of M_1 and M_2 respectively. Each $N_O[M_1, M_2]$ can get at most one vote from *each interaction*. Each $C[M_1, M_2]$ can get at most one vote from each sequence s_j. Each $C[M_1, M_2]$ can get at most one vote from each sequence s_i. This step takes $O(mn^2)$ time where m is the total number of interactions.

Finally, we can pre-compute all possible values for $\binom{i}{j}$ for different values of i, j to be used in the calculation of the score of a motif pair. The maximum value for i is the largest possible number of interactions, r, for a motif and can be obtained from the $N_I[M]$ table, which is usually a lot smaller than m. Computing all these values takes only $O(r^2)$ time. Then, computing the score for a motif pair takes $O(m)$ time as it involves finding the sum of at most $O(m)$ terms, each can be computed in constant time using the pre-computed $\binom{i}{j}$ values. There are altogether $O(\binom{l}{d}^2 \cdot |\Sigma|^{2(l-d)})$ possible motif pairs where Σ is the alphabet for amino acids . However, some of them may not have any instance in the input sequences, so we only need to compute the score for at most $O(mn^2)$ pairs since we only need to consider those pairs of sequences which have an interaction. The overall time complexity is $O(m^2n^2)$. The space complexity is $O(\binom{l}{d}^2 \cdot |\Sigma|^{2(l-d)})$. As l and d are usually small, they can be treated as constants. Theorem 1 follows.

Theorem 1: The time and space complexities of FindMotif(l, d, r) are $O(m^2n^2)$ and $O(|\Sigma|^{2(l-d)})$, respectively.

For the SH3 domain dataset of 146 protein sequences and 233 interactions, the algorithm takes about half an hour with $l = 8$ and $d = 5$, which is a lot faster than D-MOTIF. We also develop a heuristics algorithm that can run faster than FindMotif(l, d, r) which can handle the whole yeast dataset of about 5000 sequences with over 20,000 interactions in about 20 minutes.

Remarks: In practice, if the space requirement of $O(\binom{l}{d}^2 \cdot |\Sigma|^{2(l-d)})$ is too large, we apply some simple tricks to reduce the space requirement. For example, we first fix a set of positions for the wildcard characters and process the motifs with these positions as wildcards. The space requirement can be reduced to $O(|\Sigma|^{2(l-d)})$. Then, repeat the procedure for $\binom{l}{d}$ times. Another trick is to fix the first character, say 'A', of one motif and process the motifs starting with that character, the space requirement will then be reduced to $O(|\Sigma|^{2(l-d)-1})$.

3.2. Heuristics Algorithm

FindMotif finds the p-score for all motif pairs in the sequences that interact. As the motif pair space is very large and computing the p-score is time-consuming, FindMotif has a long running time when dealing with a moderately large dataset or when l is large. The heuristics algorithm MotifHeuristics improves the running time of FindMotif by considering less motif pairs, thus reducing the number of times the p-score is calculated.

Instead of finding the p-score for all motif pairs that interact, MotifHeuristics starts with r random seed (l, d)-motifs that are found in the sequences. For each seed x, we can find its optimal partner motif y such that motif pair $\{x, y\}$ has the lowest p-score by voting, similar to FindMotif. The optimal motif partners of the seeds then becomes the new seeds. In the next step, we go on to find the optimal partner motif for all the new seeds. By doing this, we can obtain motif pairs with lower p-score. This process is repeated until the p-score does not improve. The resulting motif pairs are local optimal motif pairs. It is likely that at least one of the randomly generated seeds will converge to the optimal motif pair.

Table 1. Experimental results on SH3 domain dataset using FindMotif

M_1	M_2	$S(M_1)$, S(M_2)	$O(M_1, M_2)$	p-score
PxNxVxxx	LxxLxxSx	22, 69	80	4.78×10^{-14}
LxPxxTxx	GxxPxxYx	29, 17	54	1.70×10^{-13}
LSxSxxxx	PxNxVxxx	57, 22	68	1.78×10^{-13}
LLxxLxxx	PxNxVxxx	60, 22	83	2.65×10^{-13}
PxNxVxxx	SxSxIxxx	22, 58	82	4.40×10^{-13}
SLxxKxxx	PxNxVxxx	47, 22	67	6.66×10^{-13}
PxxPxRxx	GxxPxxYx	28, 17	57	2.24×10^{-12}
SxIDxxxx	GxxPxxYx	36, 17	63	3.34×10^{-12}
LxPxxTxx	AxxSxGxx	29, 23	52	4.46×10^{-12}
GxxPxxYx	LxxLxxSx	17, 69	80	5.66×10^{-12}

Top 10 motif pairs reported by FindMotif on the SH3 dataset with $l = 8$, $d = 5$, $r = 200$.

Table 2. Experimental results on SH3 domain dataset using MotifHeuristics

M_1	M_2	$S(M_1)$, S(M_2)	$O(M_1, M_2)$	p-score
LxxLxxSx	PxNxVxxx	69, 22	80	4.78×10^{-14}
LxPxxTxx	GxxPxxYx	29, 17	54	1.70×10^{-13}
GxxPxxYx	PxxPxRxx	17, 28	57	2.24×10^{-12}
GxxPxNxx	PxxPxRxx	18, 28	54	1.04×10^{-11}
LxxSxKxx	TxxGxVxx	38, 15	38	1.12×10^{-11}
QSxxSxxx	SxxQxxIx	36, 24	40	1.36×10^{-11}
SxxSxSxx	ATxPxxxx	79, 18	76	1.87×10^{-11}
IxxTTxxx	KxxPExxx	21, 18	27	8.50×10^{-11}
PSxLxxxx	YxxDYxxx	47, 10	42	1.40×10^{-10}
SxPxPxxx	AxAxYxxx	37, 12	44	3.20×10^{-10}

Top 10 motif pairs reported by MotifHeuristics on the SH3 dataset with $l = 8$, $d = 5$, $r = 200$.

By setting an appropriate the value for r, the algorithm can finish within reasonable time with high accuracy.

The time complexity for MotifHeuristics(l, d, r): Let the algorithm run for k iterations for each seed. In each iteration, at most r seeds are given. The algorithm finds the optimal partner of the r seeds. Similar to the exact algorithm, this step is done by voting N_S, N_I, N_T, N_O and C. However, it takes $O(mnr)$ time only as one of the motif (the seed) is known. As there are r seeds and $O(mn)$ possible optimal partner for each seed, it takes $O(rm^2n)$ time to calculate the scores of all pairs. The overall time complexity is $O(krm^2n)$, as there are k iterations. In practice, the algorithm halts after around 10 iterations, or we can stop the execution after 10 iterations. Therefore, k can be neglected. Again, we treat l and d as constants. Theorem 2 follows.

Table 3. Experimental results on yeast dataset

M_1	M_2	$S(M_1)$, S(M_2)	$O(M_1, M_2)$	p-score
GxxPxNxV	PxLPxRxx	32, 39	78	4.25×10^{-35}
PPxPxRxx	GxxPxNYx	27, 19	72	9.12×10^{-35}
RRxDxxQx	SSPxKxxx	11, 107	56	4.20×10^{-32}
GCxxAExx	SSxxSxxS	16, 508	128	1.17×10^{-31}
LVxxFLxx	LxxSPxKx	77, 68	35	6.86×10^{25}
DTxGQxxx	LxxYIxIx	43, 31	38	1.38×10^{-20}
GDGTxxxx	IWDxRxxx	40, 16	23	3.39×10^{19}
GSTGxxxx	AxxLxNSx	46, 70	38	4.28×10^{-19}
IGxAIxxx	GxKTxKxx	59, 50	39	1.62×10^{-18}
FGxxTxNx	ALRxLxxx	16, 103	43	1.75×10^{-18}

Top 10 motif pairs reported by MotifHeuristics on the yeast dataset with $l = 8$, $d = 5$, $r = 200$.

Theorem 2: The time and space complexities of MotifHeuristics (l, d, r) are $O(rm^2n)$ and $O(r \cdot |\Sigma|^{l-d})$, respectively.

4. EXPERIMENTS

We have performed experiments to evaluate our scoring function and the performance of MotifHeuristics. We ran our program on real biological data and verified the results with those obtained from biological experiments. We have also compared the performance our algorithm with the heuristics algorithm D-STAR proposed by Tan *et al*. [10]. All the experiments were performed on a standalone computer with 2.4GHz Intel CPU and 4GB memory. In each experiment, we used 200 seeds for MotifHeuristics and each seed was refined at most 10 times.

4.1. SH3 Domains Dataset

SH3 domains are similar amino acid segments that are found to bind motifs "PxxP", "PxxPx[RK]" and "[RK]xxPxxP" [12]. It has been experimentally determined that the binding is due to the presence of the pattern "GxxPxNY" in SH3 domain (PDB ID:1AVZ). We tested whether FindMotif and MotifHeuristics are able to recover this motif pair "GxxPxNY" and "PxxP". The dataset was obtained from Biomolecular Object Network Databank [13]. It contains 146 yeast proteins, including 23 that contain the SH3 domain, and 233 protein-protein interactions.

Top 10 motif pairs reported by FindMotif on the SH3 dataset with $l = 8$, $d = 5$. We tested FindMotif and MotifHeuristics with $l = 8$ and $d = 5$. The results are shown in Table 1 and 2. FindMotif discovered the motif pair "**G**xx**P**xx**Y**" and **P**xx**P**x**R** at rank 7 and "**G**xx**P**xx**Y**" and "**P**xL**P**" at rank 13. MotifHeuristics also discovered the similar motif pair at rank 3. Therefore, both algorithms could discover the known motif pair.

Tan *et al*. stated that the heuristeric D-STAR algorithm was able to discover the known motif pairs [10]. We exhausted all possible motif pairs using their model and parameters. We found the first motif pair similar to "GxxPxNY" and "PxxP" was ranked 90 and there were 89 motif pairs having higher χ-score than the known motif pair. This result suggested that D-STAR might miss out motif pairs with good scores (the top 89 pairs). On the other hand, our scoring function was more robust than the χ-score proposed by Tan *et al.* as FindMotif has considered all possible motif pairs and the known motif pair is ranked within the top ten results.

4.2. Yeast Dataset

To measure the efficiency and further verify the correctness of MotifHeuristics on large dataset, we ran the algorithm on the yeast dataset. The yeast dataset is obtained by merging data from the MIPS and DIP databases. It includes 5246 yeast proteins and 21225 interactions. Since this dataset contains most protein sequences and interactions in the SH3 dataset, we expect our algorithms can also discover motif pair similar to "GxxPxNY" and "PxxP". We used MotifHeuristics to discover motif pairs in this dataset with parameters $l = 8$, $d = 4$ and $r = 200$. The results were shown in Table 3. MotifHeuristics reported the motif pair "**G**xx**P**x**N**xV" and "**P**xL**P**x**R**xx" at rank 1. This shows that our scoring function perform well in both small and large datasets.

4.3. Running Time Comparison

The running time of the three algorithms on the above datasets are shown in Table 4. For the SH3 dataset, all algorithms could discover a motif pair similar to the known pair "GxxPxNY" and "PxxP". Since FindMotif guaranteed finding the motif pair with the lowest *p*-score, it took the longest time (54 minutes) to finish. For the two heuristics algorithms, D-STAR took 14 minutes while MotifHeuristics took 51 seconds only. On the other hand, MotifHeuristics had the shortest running time. For the yeast dataset, since the number of protein sequences and interactions were large, FindMotif and the heuristics algorithm, D-STAR, could not finish in 5 days. On the other hand, MotifHeuristics was able to discover the correct motif pair in 44 minutes only. So, MotifHeuristics is more scalable and efficient.

Table 4. Comparison of the algorithms' running time

Algorithm \ Dataset	SH3 dataset	Yeast dataset
FindMotif	54 min ($l = 8$, $d = 5$)	-
MotifHeuristics	51 s ($l = 8$, $d = 5$, $r = 200$)	44 min ($l = 8$, $d = 4$, $r = 200$)
D-STAR	14 min	-

'-' means the algorithm did not finish in 5 days

4.4. Simulated Data

We have further evaluated the sensitivity and efficiency of MotifHeuristics using simulated data. We generated 146 (same size as the SH3 dataset) length-668 random protein sequences (each nucleotide has equal probability to occur). We randomly picked a (8,5)-motif pair and planted s instances (i.e., $|S(M_1)| = |S(M_2)| = s$) of each motif in the sequences. 200 interactions are randomly assigned to the 146 sequences. An additional i (i.e., $|O(M_1, M_1)|$) interactions are assigned to the instances of the planted motif pair. We have tried two settings: $s = 10$ and $s = 20$. For each setting, we study the running time and the success rate of MotifHeuristics using different values of i. For each set of parameters s and i, 50 different data sets were generated.

Note that we only consider the algorithm to be successful if the planted motif pair appears in the output as rank 1. The results are shown in Figure 1. When the number of interactions increases to about 45 and 75, respectively, for the cases of 10 and 20 planted motif instances, the success rate increases to more than 80%. We found that with fewer interactions, the *p*-value of the planted motif pair is usually larger than 1×10^{-11} making it difficult to be distinguished from noise. The results are consistent with the real dataset (see Table 2). The

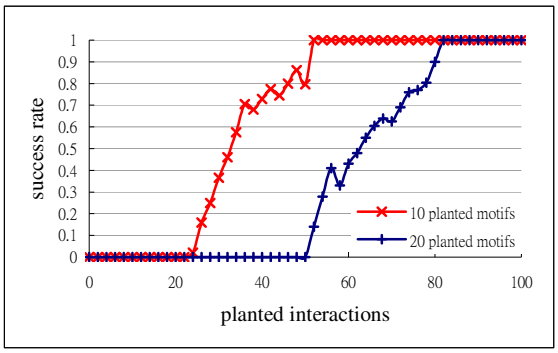

Fig. 1. Success rate against number of planted interactions ($l = 8$, $d = 5$, $r = 200$).

average running time for each dataset is about 1 minute. Overall speaking, MofitHeuristics is fast with reasonably high sensitivity.

5. CONCLUSION

In this paper, we have proposed a new scoring function to evaluate whether a motif pair is significant based on the given protein-protein interaction data. We also developed an exact algorithm and a heuristic algorithm to solve the problem. We show that our scoring function is more accurate than the one used in [10] and our algorithms are more efficient and scalable than existing algorithms. Possible future directions include the followings. The scoring function we proposed is an approximation to the conditional probability we defined in Section 2. Whether a more accurate scoring function exists is an interesting and important question. Although our algorithms can process a large dataset within reasonable amount of time, a more efficient (in terms of time and space) algorithm is always desirable. Whether the current approach can be effectively applied to locate motif triplets is also a challenging extension.

Acknowledgments

This paper is supported by the RGC grant HKU 7120/06E.

References

1. Bailey, T., Elkan, C. Fitting a mixture model by expectation maximization to discover motifs in biopolymers. *ISMB* 1994; 28-36.
2. Goehler, H. et al. Can we infer peptide recognition specifying mediated by SH3 domains? *FEBS Lett* 2002; **513(1)**: 38-44.
3. Hans, J., Brandt, W., Vogt, T. Site-directed mutagenesis and protein 3D-homology modelling suggest a catalytic mechanism for UDP-glucose-dependent betanidin 5-O-glucosyltransferase from Dorotheanthus bellidiformis. *Plant J.* 2004; **39(3)**: 319-333.
4. Hu, H. et al. A map of WW domain family interactions. *Proteomics* 2004; **4(3)**: 643-655.
5. Karkkainen, S, Hiipakka, M, Wang, JH, Kleino, I, Vaha-Jaakkola, M, Renkema, GH, Liss, M, Wagner, R, Saksela, K. Identification of preferred protein interactions by phage-display of the human Src homology-3 proteome. *EMBO Rep* 2006; **7(2)**: 186-191.
6. Leung, H., Chin, F. Discovering Motifs with Transcription Factor Domain Knowledge. *PSB* 2007; **12**: 472-483.
7. Leung, H., Chin, F. Finding motifs from all sequences with and without binding sites. *Bioinformatics* 2007; **22(18)**: 2217-2223.
8. Pavesi, G., Mereghetti, P., Zambelli, F., Stefani, M., Mauri, G., Pesole, G. MoD Tools: regulatory motif discovery in nucleotide sequences from co-regulated or homologous genes. *Nucleic Acids Res* 2006; **34**: W566-W570.
9. Reiss, D. J., Schwikowski, B. Combinatorial pattern discovery in biological sequences: The TEIRESIAS algorithm. *Bioinformatics* 2004; **14(1)**: 55-67.
10. Tan, Soon-Heng, Hugo, Willy, Sung, Wing-Kin, Ng, See-Kiong. A correlated motif approach for finding short linear motifs from protein interaction networks. *BMC Bioinformatics* 2006; **7**: 502.
11. Tong, A. H. Y. et al. A combined experimental and computational strategy to define protein interaction networks for peptide recognition modules. *Science* 2002; **295(5553)**: 321-324.
12. Weidemann, A., Konig, G., Bunke, D., Fischer, P., Salbaum, M., Masters, C., Beyreuther, K. Identification, biogenesis, and localization of precursors of Alzheimer's disease A4 amyloid protein. *Cell* 1989; **57**: 115-126.
13. Biomolecular Object Network Databank. http://bond.unleashedinformatics.com
14. Database of Interacting Proteins. http://dip.doe-mbi.ucla.edu/