A study of statistical methods for function prediction of protein motifs

Tao Tao[1], ChengXiang Zhai[1], Xinghua Lu[2], and Hui Fang[1]

[1]Department of Computer Science
University of Illinois at Urbana-Champaign
Urbana, IL 61801

[2]Department of Biometry and Epidemiology
135 Cannon St., Stuite 303
Medical University of South Carolina
Charleston, SC 29524

Email: {taotao,czhai}@cs.uiuc.edu, lux@musc.edu,hfang@cs.uiuc.edu

Abstract

Automatic discovery of new protein motifs (i.e., amino acid patterns) is one of the major challenges in bioinformatics. Several algorithms have been proposed that can extract statistically significant motif patterns from any set of protein sequences. With these methods, one can generate a large set of candidate motifs that may be biologically meaningful. In this paper, we study several statistical methods to automatically predict the functions of these candidate motifs, including a popularity method, a mutual information method, and statistical translation models. These methods capture, from different perspectives, the correlations between the matched motifs of a protein and its assigned Gene Ontology(GO) terms, which characterize the function of the protein. We evaluate these different methods using the known motifs in the Interpro database. Each method is used to rank candidate terms for each motif. We, then, use mean reciprocal rank(MRR) to evaluate the performance. The results show that in general, all these methods perform well, suggesting that they can all be useful for predicting an unknown motif's function. Among all the methods tested, a statistical translation model with popularity prior performs the best.

1 Introduction

Protein motifs are conserved amino acid sequence patterns that characterize the functions of proteins. In terms of amino acid sequences, a motif can be regarded as a pattern that specifies a typical subsequence of amino acids, often with "gaps" of fixed or various lengths between amino acids. We will use the term *motif* and *pattern* interchangeably in this paper. Knowing the exact pattern and function of a motif is crucial for understanding the structure and function of related proteins. For example, a precise knowledge of motifs of certain

family of proteins would make it much easier to recognize new proteins of the same family, and knowing what motifs match a protein also reveals a lot of information about the function of the protein.

There are many known protein motifs available through several databases such as PROSITE (Bairoch et al., 1996), ProDom (Corpet et al., 1999), CDD (Marchler-Bauer et al., 2002), BLOCKS, PRINTS, PFAM, and InterPro(Apweiler et al., 2001). These databases are usually constructed by studying the set of protein sequences that are known to have certain functions and extracting the conserved motifs (among the sequences) that are believed to be responsible for their functions. However, the number of motifs that can be extracted in this way is quite limited; indeed, these existing databases represent only a small fraction of all the motifs, and it remains a great challenge to identify many undiscovered motifs. The current process for curating these databases is mostly manual and labor-intensive, thus can not scale up to the rapid explosion of data in genomic repositories (Hart et al., 2000).

Several methods have been proposed to automate the process of motif discovery, including MEME (Bailey and Elkan, 1995), the Gibbs Sampler (Lawrence et al., 1993), Pratt (Jonassen et al., 1995), EMOTIF (Huang and Brutlag, 2001), SPLASH (Califano, 2000), and Teiresias (Rigoutsos and Floratos, 1998). See (Brazma et al., 1998) and (Brejova et al.,) for a survey of these techniques. A typical way of discovering protein motifs with these methods is to start with a set of protein sequences that either belong to some known family or simply are similar to each other through alignments, and extract amino acid patterns that are shared by many sequences. This can be called "supervised" discovery – "supervised" in the sense that the "seed" set of proteins are constructed with some prior knowledge about these proteins. Being used in such a supervised way, these algorithms are able to not only successfully identify some known motifs, but also to suggest some more specific (presumably better) patterns (see e.g., (Hart et al., 2000; Rigoutsos et al., 2000)).

A major limitation of supervised pattern discovery is that it is difficult to discover the conserved functional and structural patterns that *cross* protein family boundaries. This is a serious problem, especially because many undiscovered motifs may be of this nature. For this reason, "unsupervised" pattern discovery has been paid much attention recently. The idea is to treat the largest possible database as *an indivisible entity*, and perform pattern discovery on such a database. This can be expected to help discover many more signals that are still conserved at the sequence level (Rigoutsos et al., 2000). If we treat proteins as the biological analog of sentences in natural languages such as English, then any recurrent functional and structural signals whose traces remain at the level of the amino acid sequence should be observable as pattern-words that are being re-used (Rigoutsos et al., 2000). By doing massive pattern mining in this way, we can expect to build a "bio-dictionary" that contains many potentially useful motifs. This is a very promising direction that can potentially help discover many new motifs. An essential task involved in the compilation of such a dictionary is to determine the function (the meaning) of newly identified motifs, which is a problem that we study in this paper.

Our basic idea for predicting a motif's function is based on the observation that the function of a motif is reflected through the function of proteins that match the motif. Specifically, we can expect to infer a motif's function based on the functions of all the matched proteins. For example, in an ideal case when all the proteins match a given motif (M) share the same

particular function (F) which is not shared by any protein that does not match the motif, we clearly can infer that motif M is very much related to function F. In reality, such an ideal case rarely exists, and we propose three statistical methods to measure such correlation and make predictions accordingly based on the popularity of terms, the mutual information between GO terms and motifs, and probabilistic translation models.

We exploit the Gene Ontology (GO) database to obtain the functions of known proteins. The Gene Ontology project (the Gene Ontology Consortium 2001) is a concerted effort by the bioinformatics community to develop a standardized controlled vocabulary (GO terms) and to annotate biological sequences with the vocabulary. For example, the FlyBase gene *18 wheeler* is assigned two function terms: GO:0004888 (*transmembrane receptor*) and GO:0005194 (*cell adhesion molecule*). Both the number of annotated sequences and the number of GO terms associated with individual sequences in the Gene Ontology database are increasing very rapidly. Moreover, natural language processing techniques are also being used to automatically annotate gene products with GO terms (Xie et al., 2002). Thus, it can be foreseen that the annotations of protein sequences in the Gene Ontology database will become more and more detailed, and have a great potential to be used as an enriched knowledge base of proteins.

We evaluated our model using the known motifs from the InterPro database (Apweiler et al., 2001) by comparing the predicted GO terms for each motif with the manually assigned GO terms to the same motif. The results show that in general, all these methods perform well, suggesting that they can all be useful for predicting an unknown motif's function. Among all the methods tested, a statistical translation model with popularity prior performs the best.

The rest of the paper is organized as follows. In Section 2, we describe the data set that we work on. In Section 3, we formulate the problem of motif function prediction as assigning GO terms to a motif, and then present the common basic ideas. We present the popularity method and mutual information method in Section 4 and probabilistic translation models in Section 5. Experiment results and analysis are discussed in Section 6. Section 7 gives our conclusions and future work.

2 The Data Set

We use the February 2003 release of Gene Ontology sequence database [1]. In this database, there are 13,214 protein sequences with annotations of GO terms and matched motifs. The motifs are in different format, such as the PROSITE format, Pfam format, and the InterPro format. We used all the InterPro motifs as the test motifs to evaluate our model, because InterPro database is meant to be an integration of other most commonly used motif databases, so is presumably relatively more complete than other motif databases (Apweiler et al., 2001). For each sequence, we identify the assigned GO terms and the matched InterPro motifs from the GO database. The following is an example.

Example 1 *The FlyBase sequence* `FBgn0026616` *(the "*α*-Man-IIb" gene/protein) is assigned three terms, i.e.,* `GO:0004559` *(*"alpha-mannosidase"*),* `GO:0004572` *(*"mannosyl-oligosaccharide 1,3-1,6-alpha-mannosidase"*), and* `GO:0005794` *(*"Golgi apparatus*"), and*

[1] It is available at http://www.godatabase.org/dev/database/archive/2003-02-01

two InterPro motifs, i.e., `InterPro:IPR000602` *("*Glycoside hydrolase*")*
and `InterPro:IPR001992` *("*Bacterial type II secretion system protein*").*

The Gene Ontology has three types of GO terms describing molecular functions, cellular components, and biology process, respectively. We focus on the functional terms. Thus we excluded sequences with no matched motif in the InterPro database and sequences with no functional GO terms. This gives us a total of 5770 sequences. The annotations involve a total number of 1520 distinct GO terms and 1917 distinct InterPro motifs. The distribution of the number of GO terms per sequence has a mean of 1.285 with a standard deviation of 0.650. The distribution of the number of Interpro motifs per sequence The frequency is similar, and has a mean of 1.768 with a standard deviation of 1.138. Thus in most cases there is just one term and two motifs. However, the maximum number of GO terms assigned to a sequence is 7, and the maximum number of Interpro motifs is 12. We assume that all these 1917 InterPro motifs as our candidate motifs and use all these annotated sequences to predict each motif's function by ranking GO terms for each motif. We use the mappings from InterPro motifs to GO terms in the InterPro database as our "gold standard" to evaluate the accuracy of our prediction method [2]. A majority of the InterPro motif in our annotations (1358 motifs) have been judged in the InterPro database, and they serve as our test motifs.

3 Problem formulation

As discussed in Section 1, many pattern discovery algorithms (e.g., Teiresias) can now be used to mine a large set of protein sequences and generate a set of candidate motif patterns that may be biologically meaningful. Our goal is to *automatically* predict functions of these candidate motifs. The basic idea for predicting a motif's function is to exploit the known functions of the proteins that match the motif. Gene Ontology (GO) provides a standardized way of describing the functions of a protein or motif in terms of the GO terms. A protein is typically assigned several GO terms which characterize the protein biologically, and our goal is to assign appropriate GO terms to a motif based on the GO terms assigned to the proteins matching the motif. To see how this may be possible, consider an ideal scenario where a particular GO term has been assigned to all the proteins matching our motif, but not to any proteins that do not match the motif. In this case, the assignment of this GO term is very strongly correlated with the matching of the motif, so it would be reasonable to infer that the GO term may characterize the function of the motif very well. This example shows that through proteins we can connect the GO terms with motifs, and the correlation between the motifs and GO terms can be exploited to assign appropriate GO terms to a motif. In this section, we formally formulat the problem of motif function prediction as one involving assigning appropriate Gene Ontology (GO) terms to a motif.
Let $\mathcal{M} = \{m_1, ..., m_{|\mathcal{M}|}\}$ be a set of candidate motifs whose functions to be predicted. Let $\mathcal{S} = \{s_1, ..., s_{|\mathcal{S}|}\}$ be a set of protein sequences with known functions that are described using GO terms selected from the set of all GO terms $\mathcal{T} = \{t_1, ..., t_{|\mathcal{T}|}\}$. Given a particular protein sequence s_i in $\mathcal{S}$, there will be a subset of motifs matching the sequence, which we

[2]The mappings are available at `ftp://ftp.ebi/ac.uk/pub/databases/interpro/`

denote by $\mathcal{M}_i$, and there will also be a subset of terms assigned to the protein according to the GO database, which we denote by $\mathcal{T}_i$. For each motif m_i, our goal is to exploit the correlation between $\mathcal{M}_i$ and $\mathcal{T}_i$ and to find a subset of terms $\mathcal{T}(m_i) \subset \mathcal{T}$ that best describe the functions of m_i. This is actually similar to the problem of information retrieval, where a user wants to find a subset of documents that best satisfy the user's information need, which is often expressed in terms of a query. Here, a motif is our "query", and a GO term is a "document".

We further assume that we will obtain $\mathcal{T}(m_i)$ by *ranking* all the terms in $\mathcal{T}$. This is also a strategy often used in information retrieval and can be justified based on statistical decision theory (Robertson, 1977; Zhai, 2002). A natural way of generating a ranking of terms is to compute a "goodness" score for each term and then rank all terms accordingly. Under these assumptions, the problem of predicting motif functions now becomes essentially one to define an appropriate scoring function $f : \mathcal{M} \times \mathcal{T} \rightarrow \Re$ that can generate a score for each term with respect to a motif.

In general, such scoring functions are inferred based on the correlation between $\mathcal{M}_i$'s and $\mathcal{T}_i$'s. We now discuss several different methods to measure such correlation.

4 Popularity method and mutual information method

In this section, we describe briefly two simple methods for measuring the motif-term correlation based on a "popularity count" and mutual information respectively.

The most straightforward method is to use the number of sequences in which a term co-occurs with a motif as a measure of correlation. Terms are thus ranked based on how many times they co-occur with the motif under consideration. We call this method the *popularity method*, since terms with high co-occurrences will be ranked above those with low co-occurrences.

Example 2 *Given $S = \{s_1, s_2\}$, s_1 includes $M_1 = \{m_1, m_2\}$ and $T_1 = \{t_1, t_2\}$ while s_2 includes $M_2 = \{m_1\}$ and $T_2 = \{t_1\}$. Motif m_1 co-occurs with t_1 twice, and with t_2 once. Therefore, for m_1, the ranking of terms are: t_1 t_2.*

One possible deficiency of the popularity method is that a term can have high co-occurrences just because the term is generally popular. This deficiency can be addressed by using the Mutual Information (M.I.), whcih is a commonly used statistic measure to evaluate the correlation between two discrete random variables: X and Y. It is formally defined by $I(X : Y) = \sum_{x,y} p(X = x, Y = y) log \frac{p(X=x,Y=y)}{p(X=x)p(Y=y)}$. A larger mutual information indicates a stronger association between X and Y; $I(X : Y) = 0$ if and only if X and Y are independent. For our purpose, we regard the assignment of a term T to a sequence and the matching of a sequence with a motif M as two binary random variables. The involved probabilities can be estimated based on the number of sequences matching a motif M, the number of sequences assigned a term T, the number of sequences both matching M and assigned T, and the total number of sequences in the database.

5 A probabilistic translation model for motif function prediction

There is another interesting way to look at such a correlation. Both the motifs matching a protein and the terms assigned to a protein can be regarded as a description of the pro-

tein's functions, but in different "languages". Our goal is to figure out how to "translate" a description in terms of the "motif language" to one based on "GO term language" by examining the co-occurrence patterns of motifs and GO terms.
To implement this idea with a probabilistic translation model, let us use two random variables $M \in \mathcal{M}$ (for motif) and $T \in \mathcal{T}$ (for term) to represent the observation of a motif and the assignment of a term in a protein, respectively. The conditional probability $p(M|T)$ indicates the probability that a term T would be translated into a motif M. Thus we would expect $p(M|T)$ to be high if T characterizes some functional aspect of M, and low if otherwise. Given a sequence s_i, we regard its motif set $\mathcal{M}_i$ as the results of applying this translation model $|\mathcal{M}_i|$ times, each time picking a (potentially different) term t from its term set $\mathcal{T}_i$ and "generating" a motif according to $p(M|T = t)$. We thus have a generative probabilistic model for $\mathcal{M}_i$ conditioned on $\mathcal{T}_i$. This allows us to estimate the translation model parameters $p(M|T)$ by maximizing the conditional likelihood of the $\mathcal{M}_i$ given the corresponding $\mathcal{T}_i$, for all the sequence s_i in $\mathcal{S}$. Once the model parameters are estimated, for any motif m_i, we can rank terms based on the posterior probability that a term t has been used to "generate" m_i, i.e., $p(T = t|M = m_i)$. We now describe our probabilistic translation model in more detail.

5.1 Term-motif translation model

Given a sequence $s_i \in \mathcal{S}$, we are interested in defining the conditional probability $p(\mathcal{M}_i|\mathcal{T}_i, s_i)$, i.e., the probability of "generating" all the motifs in $\mathcal{M}_i$ from terms in $\mathcal{T}_i$. We make two assumptions to make the model more tractable. Note that none of the two assumptions holds in reality; we introduce them purely for the sake of simplification.
Assumption 1: Given $\mathcal{T}_i$, each motif in $\mathcal{M}_i$ is generated independently.

With this assumption, we have

$$p(\mathcal{M}_i|\mathcal{T}_i, s_i) = \prod_{m \in \mathcal{M}_i} p(m|\mathcal{T}_i, s_i) \quad (1)$$

$$(2)$$

Assumption 2: Each motif in $\mathcal{M}_i$ is generated using one of the terms in $\mathcal{T}_i$.
With this assumption, we would first pick a term t from $\mathcal{T}_i$ and then generate a motif $m \in \mathcal{M}_i$ according to our translation model $p(m|t)$[3]. This means, $p(m|\mathcal{T}_i, s_i)$ is given by the following mixture model:

$$p(m|\mathcal{T}_i, s_i) = \sum_{t \in \mathcal{T}_i} p(t|s_i)p(m|t) \quad (3)$$

where, $p(t|s_i)$ is the probability of selecting term t to generate a motif for sequence s_i, and $p(m|t)$ is our basic translation model, i.e., the probability of generating motif m given that term t is picked.

[3]To make our presentation more concise, we often omit the random variables in a probability formula. Thus, $p(m|t)$ actually means $p(M = m|T = t)$.

Intuitively, $p(t|s_i)$ is related to our knowledge about the motif set $\mathcal{M}_i$, which has been assumed to be given. For example, if we know (based on whatever prior knowledge we have) that most of those candidate motifs patterns are unlikely to be related with certain functions, then $p(t|s_i)$ should assign low probabilities to those terms that characterize these "unlikely functions". Since we do not assume any such prior knowledge, a reasonable choice for $p(t|s_i)$ is to let it be uniform, i.e., each term in $\mathcal{T}_i$ is equally likely to be selected. Thus, we have

$$p(m|\mathcal{T}_i, s_i) = \frac{1}{|\mathcal{T}_i|} \sum_{t \in \mathcal{T}_i} p(m|t) \tag{4}$$

Thus, the log-likelihood for the whole set of sequences $\mathcal{S}$ is

$$L(\theta|\mathcal{S}) = \sum_{i=1}^{|\mathcal{S}|} \sum_{m \in \mathcal{M}_i} [\log \sum_{t \in \mathcal{T}_i} p(m|t) - \log |\mathcal{T}_i|] \tag{5}$$

where, $\theta = \{p(m_i|t_j)\}$ ($1 \leq i \leq |\mathcal{M}|, 1 \leq j \leq |\mathcal{T}|$) are the translation model parameters, and they satisfy the following constraints:

$$\sum_{i=1}^{|\mathcal{M}|} p(m_i|t_j) = 1, \text{for } j = 1, ..., |\mathcal{T}|$$

This term-motif translation model can also be interpreted as a clustering model. Specifically, we can regard each GO term t as representing one cluster, and treat each motif m as an observed data point. The translation model $p(m|t)$ can thus be interpreted as the (discrete) density function of the distribution of data points in cluster t. Since we allow overlapping clusters, a data point (i.e., a motif) can belong to mulitple clusters. Our goal is to determine what clusters the motif belongs to, i.e., to determine which GO terms should be assigned to the motif.

5.2 Parameter estimation

With the setup given above, we can use the current GO database as our data set (i.e., $\mathcal{S}$), and estimate parameters using the Maximum Likelihood (ML) estimator. That is, our estimate of parameters is given by

$$\hat{\theta} = \arg\max_{\theta} \sum_{i=1}^{|\mathcal{S}|} \sum_{m \in \mathcal{M}_i} [\log \sum_{t \in \mathcal{T}_i} p(m|t) - \log |\mathcal{T}_i|] = \arg\max_{\theta} \sum_{i=1}^{|\mathcal{S}|} \sum_{m \in \mathcal{M}_i} \log \sum_{t \in \mathcal{T}_i} p(m|t) \tag{6}$$

The solution of this maximization problem can not be found analytically, so we rely on numerical algorithms. Since our model is a simple mixture model, we can use the Expectation-Maximization (EM) algorithm (Dempster et al., 1977) to compute $\hat{\theta}$. Specifically, starting from some arbitrary estimate $\theta^(0)$, the EM algorithm iteratively alternates between two steps:

- **E-step:** Given the current estimate of parameters $\theta^{(n)}$, compute the expected complete likelihood, which essentially requires computing the distribution of the hidden variables, i.e., $p(z_{ij}|\theta^{(n)})$ given by

$$p(z_{ij} = t_k|\theta^{(n)}) \quad = \quad \begin{cases} \frac{p(m_j|t_k)}{\sum_{t\in\mathcal{T}_i} p(m_j|t)} & \text{if } t_k \in \mathcal{T}_i \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

- **M-step**: Compute a new estimate of parameters $\theta^{(n+1)}$ by maximizing the expected complete likelihood under the distribution of the hidden variables. The estimation is:

$$p^{(n+1)}(m_j|t_k) \quad = \quad \frac{\sum_{1\le i\le|\mathcal{S}|, m_j\in\mathcal{M}_i} p(z_{ij} = t_k|\theta^{(n)})}{\sum_{j'=1}^{|\mathcal{M}|}\sum_{1\le i\le|\mathcal{S}|, m_{j'}\in\mathcal{M}_i} p(z_{ij'} = t_k|\theta^{(n)})} \tag{8}$$

The computational complexity of each EM iteration is on the order of $O(|\mathcal{S}||\bar{\mathcal{M}}||\bar{\mathcal{T}}|)$, where $|\bar{\mathcal{M}}|$ and $|\bar{\mathcal{T}}|$ are the average size (over all sequences) of $\mathcal{M}_i$ and $\mathcal{T}_i$, respectively.

5.3 Inference (Ranking GO terms for a motif)

Once we have all the parameters in our translation model estimated, we can use the model to infer which GO terms are associated with, and thus should be assigned to a given motif. Specifically, given a motif m, we can rank all GO terms based on $p(t|m)$. Recall that, in Section 3, we have framed the problem of predicting functions of a motif as one to define a scoring function $f : \mathcal{M} \times \mathcal{T} \to \Re$. Our translation model suggests that the most natural way to rank GO terms for a given motif is to define f as $p(t|m)$.
With the translation model, $p(t|m)$ can be computed using Bayes rule as follows:

$$p(T = t|M = m) \quad = \quad \frac{p(T = t)p(M = m|T = t)}{\sum_{t'\in\mathcal{T}} p(T = t')p(M = m|T = t')}$$

This equation requires prior probabilities for every term $p(T)$, which presumably represent our knowledge about which term is more likely to describe any motif's function before we are given any particular motif. There are several different choices, which we illustrate with the following example:

Example 3 *We have two sequences. Sequence 1 is assigned two terms t_1, t_2 and matches three motifs m_1, m_2,m_3. Sequence 2 is assigned two terms t_2, t_3 and matches four motifs m_1, m_2,m_4, m_5.*

1. **Uniform distribution($Prior_{uni}$)**: we consider every term has equal probability. $p(t_i) = \frac{1}{|\mathcal{T}|}$. In Example 3. $p(t_i) = 1/3$
2. **Counts of co-occurred motifs ($Prior_{motif1}$)**: We estimate the priors from the original sequence data set, and let the prior on a term to be proportional to the counts of motifs that have co-occurred with the term. That is, we count every motif-term pair as an entry, and have $2 \times 3 + 2 \times 4 = 14$ entries for Example 3, so $p(t_1) = 3/14$, $p(t_2) = 7/14$, and $p(t_3) = 4/14$.

3. **Counts of *distinct* co-occurred motifs ($Prior_{motif2}$)**: This is similar to $Prior_{motif1}$, but we let the prior to be proportional to the counts of *distinct* motifs that have co-occurred with the term. In our example, t_1 co-occurs with three distinct motifs (i.e., m_1, m_2, m_3), t_2 with five (i.e., m_1, m_2, m_3, m_4, m_5), and t_3 with four (i.e., m_1, m_2, m_4, m_5). Thus, we have $3 + 5 + 4 = 12$ entries, and $p(t_1) = 3/12$, $p(t_2) = 5/12$ and $p(t_3) = 4/12$.

5.4 Comparison with popularity and mutual information

It is instructive to compare this translation model with the other two methods (i.e., the popularity and mutual information methods). In particular, let us take a closer look at what statistics are shared by these methods. First, all methods would favor a term with high co-occurrences, which intuitively makes sense. In fact, this is the only information that the popularity method uses. Mutual information differs from the popularity method in that it would penalize a term with high global frequency. The probabilistic translation model favors terms with high $P(t|m)$, which is proportional to $P(m|t)P(t)$. High co-occurrences clearly contribute to a high $P(m|t)$. However, since $p(m|t)$ must sum to one over all the motifs, a term appearing in many sequences tends to be associated with more motifs, which would reduce the conditional probability for each motif. Thus the "competition" among motifs for a given term essentially achieves the effect of penalizing a common term just as in mutual information. The translation model also introduces another competition among terms assigned to the same sequence, due to the fact that we assume that each motif is generated from precisely one of the terms assigned to the sequence. Intuitively, this causes discounting of the co-occurrences when multiple terms are assigned to a sequence and each term is only given "partial" credit. In summary, the major difference between the translation model and mutual information lies in the compilation among motifs and terms, while the major difference between popularity and the other two methods is the punishment of common terms. A comparison of the empirical performance of these methods would reveal the influence of these factors on the performance.

6 Experiments

We evaluated all three methods using the Gene Ontology database and Interpro database. Specifically, we use the Gene Ontology database to compute a ranking of GO terms for each Interpro motif, and use the known functions of the Interpro motifs as the gold standard to evaluate the performance of a method. Through comparing the results of these different methods, we hope to answer the following questions:

- Does penalizing globally popular terms help improve predicion accuracy?
- Do the "competitions" introduced in the translation model help improve performance?
- How do different priors in the translation model affect the performance?

6.1 Metrics

We use the mean reciprocal rank (MRR) to measure the performance of the prediction results. Given a ranked list of GO terms, MRR is defined as $(\frac{1}{Rank_1} + \frac{1}{Rank_2} + ... + \frac{1}{Rank_k})/k$, where k is the total number of correct GO terms for the motif and $Rank_i$ is the rank of the i^{th} correct GO term in the result list. We take the average over all the motifs and use it as one single value to summarize the total performance.
MRR is bounded between 0 and 1; a higher value shows higher precision. The ranking accuracy of top terms has more influence on the overall MRR performance than that of lowly ranked terms, which is exactly what we want. Intuitively, the inverse of a MRR can be interpreted as the average number of "wrong" terms we must examine before we see a correct term.
Tied ranks (i.e., two or more terms have the same scores) occur frequently in our experiment results. To reduce the sensitivity of the MRR to the random ordering of the tied terms, we extend MRR to use the expected reciprocal of rank $E[1/rank]$, instead of $1/rank$, where E is the expectation operation. For example, if the top two terms are tied, each of them can be ranked as first or second. The expectation is $(1/1 + 1/2)/2 = 0.75$. Both terms are assigned 0.75 instead of 1 or 0.5. We call this new measure expected mean reciprocal rank(eMRR).

6.2 Experiment results

The eMRR results for the popularity method, mutual information method, and translation models with different priors are shown in Table 1.

Table 1. Comparison of eMRR for three methods

Method	Popularity	Mutual Info.	Translation Models		
			$Prior_{uni}$	$Prior_{motif1}$	$Prior_{motif2}$
eMRR	0.8510	0.8428	0.7947	0.8418	0.8367

From Table 1, we first see that all the methods achieved very high eMRR, which means that they can all rank the correct terms at or near the top. Next, we see that different priors do affect the performance of the translation model. The uniform prior is the worst, while the $motif1$ prior performs best. This observation is quite consistent in all our experiments. This shows that the conditional probability $P(m|t)$ alone is not quite accurate for ranking terms; choosing a good prior is necessary for achieving good performance. Since the $motif1$ prior is essentially the popularity method, these results suggest that the number of sequences in which a term and motif co-occur is perhaps the most important factor that helps achieve good performance. This point is amplified by the fact that the simplest popularity method surprisingly outperforms all the other methods.
Comparing the mutual information method with the popularity method, we see that mutual information does not work as well as the popularity method, indicating that penalizing globally common terms is harmful or it has over-penalized common terms. Using priors to

favor popular terms helps the translation model, but even with such help, it is still slightly worse than the mutual information method.

To understand why the popularity works so well, we examined the actual results and judgments for individual motifs. It turns out that our gold standard from Interpro is *highly incomplete*; in many cases, a motif is just assigned one general GO term, even though the known function is more specific. For example, considering the motif $InterPro : IPR000276$, whose function is known to be in the Rhodopsin-like GPCR superfamily, but it is only annotated with one term $GO : 0016021$ (integral to membrane), which means that in our evaluation, all it matters is where this term is ranked. However, the top five terms given by the translation model using motif1 prior, which are shown below, are actually all good GO terms that describe this motif's function. Thus they all should have been treated as being correct.

1. GO:0004984 (olfactory receptor activity)
2. GO:0008188 (neuropeptide receptor activity)
3. GO:0008227 (amine receptor activity)
4. GO:0004995 (tachykinin receptor activity)
5. GO:0004993 (serotonin receptor activity)

Table 2. eMRR of three methods computed using complete judgments

Method	Popularity	Mutual Info.	Translation Models		
			$Prior_{uni}$	$Prior_{motif1}$	$Prior_{motif2}$
eMRR	0.4734	0.4678	0.3899	0.4765	0.4608

This observation immediately brings up the question how reliable our evalution is given that it is based on an extremely incomplete set of judgments. Presumably, a general term is easier to be annotated, and therefore is more popular. Thus this incompleteness may favor popularity method. To test this hypothesis, we extracted the top 5 terms for each motif from all the methods, pooled them together, and had a biologist to manfully judge all of them. We then used these new judgements to re-evaluate the results. The eMRR performance is shown in Table 2

While these new judgments are complete judgments for the top 5 terms given by each method, they are incomplete in some other sense, which is why the figures of eMRR are all much lower than those in Table 1. It is very interesting to see that now the $motif1$ prior outperforms both the mutual information method and the popularity method, though only slightly. This suggests that the strategy of using only the Interpro known motifs as the gold standard is problematic; it is indeed biased toward favoring the popularity method, just as we hypothesized. The results in Table 2 can thus be regarded as more reliable.

The translation model seems to have benefited most from these complete judgments. Since these judgments are on only the top 5 terms, we can envision that with even more complete

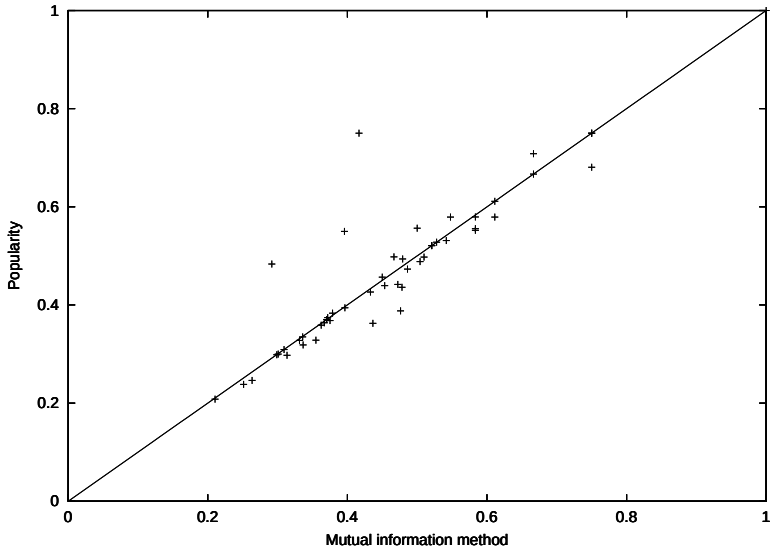

Figure 1. Popularity method vs. mutual information method.

judgements on all terms, the translation model may show even more relative advantage. The mutual information method, on the other hand, still performs worse than the baseline popularity approach, and the motif1 prior is still the best prior. This strongly suggests that it is very important to maintain the influence of the co-occurrence count in ranking the GO terms.

The figures shown in Table 2 are the *average* performance among all motifs. Figure 1 and Figure 2 show the performance comparison between two methods on individual motifs. We see that different methods really "win" at different motifs.

Table 3. eMRR on all sequences vs. sequences with multiple terms

Method	Popularity	Mutual Info.	Translation Models		
			$Prior_{uni}$	$Prior_{motif1}$	$Prior_{motif2}$
eMRR (all)	0.4734	0.4678	0.3899	0.4765	0.4608
eMRR (multiterm only)	0.5147	0.4962	0.4337	0.5308	0.5109

The improvement of the translation model over the popularity method comes from an appropriate combination of the trained translation model $p(m|t)$ with the popularity prior $p(t)$. To further examine to what extent the competitions introduced in the translation model really contribute to performance improvement, we compared all the methods on a subset of sequences that have two or more GO terms assigned. This new set would allow us to see more effect of "term competition" within a sequence. The results are shown in Table 3 together with the results on the complete set for easy comparison. The new results are overall very similar to those obtained from the complete set, but the improvement of the translation model over the popularity method is now more noticeable! This suggests that

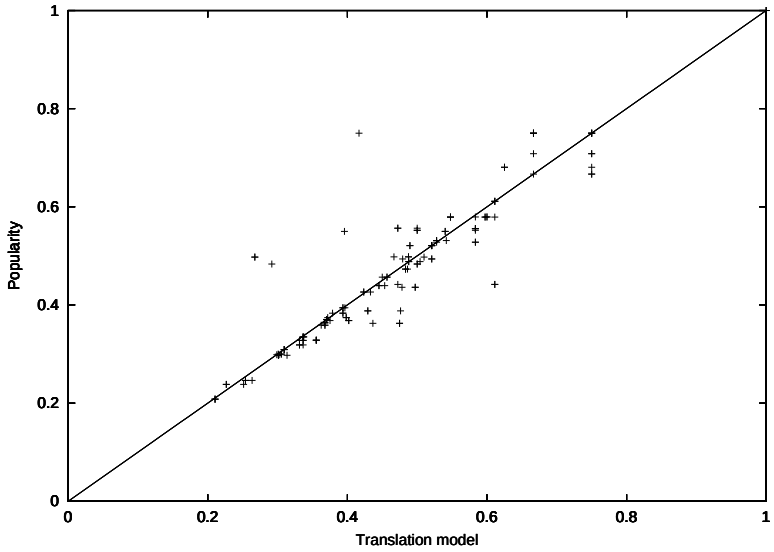

Figure 2. Popularity method vs. translation model.

the translation model is indeed better at handling multiple terms within a sequence and the "competition" among the terms assigned to the same sequence does empirically contribute to performance improvement.

7 Conclusions and future work

In this paper we propose three statistical methods for predicting motif functions by exploiting the correlation between a motif matching a sequence and the Gene Ontology terms assigned to the sequence. Given a motif, all three methods can generate a ranked list of GO terms that may describe the function of the motif. We evaluated these methods using the known motifs in the Interpro database with the Mean Reciprocal Rank (MRR) measure. The results show that

1. Overall, all three methods achieve a high MRR around 0.8.

   Even though the data set is biased due to the annotation procedure and may only reflect the upper bound of the performance, the overall high MRR is still quite encouraging, suggesting the usefulness of our general methodology.

2. Using Interpro judgments as the gold standard for evaluation is problematic because it is highly incomplete.

   It is necessary to have more complete judgments at least on some subset of the results; otherwise, the results may be misleading. As a by product of making additional judgments, we notice that the results of our methods can help the Interpro annotators significantly in providing more complete and precise annotations to these known motifs.

3. The translation model with a popularity prior achieve the best performance.

   This is shown with more complete judgments on some subset of the results. The advantage of the translation model is seen to be amplified as we work on sequences with more terms.

Probabilistic translation models were originally introduced for natural language translation (Brown et al., 1993). We show that these models may also be useful for predicting motif functions. However, it is also clear that a straightforward of applications of an existing model is unlikely successful; careful consideration of the particular problem is necessary. In our study, using an approriate prior has turned out to be important.
A natural direction of future work is to apply these methods to many new candidate motifs whose functions are unknown. We also plan to build a system running on the Web to allow a biologist to find GO terms for any interesting candidate motif. Such a system can also be very useful to assist the Interpro annotators to provide a more complete and precise annotation of motif functions.

References

Apweiler, R., Attwood, T., A.Bairoch, Bateman, A., E.Birney, Biswas, M., P.Bucher, Cerutti, L., Corpet, F., Croning, M., R.Durbin, L.Falquet, Fleischmann, W., Gouzy, J., H.Hermjakob, N.Hulo, I.Jonassen, D.Kahn, Kanapin, A., Karavidopoulou, Y., Lopez, R., Marx, B., Mulder, N., Oinn, T., M.Pagni, F.Servant, C.J.A.sigrist, and Zdobnov, E. (2001). The interpro database, an integrated documentation resource for protein families, domains and functional sites. *Nucleic Acid Research*, 29(1):37–40.

Bailey, T. and Elkan, C. (1995). The value of prior knowledge in discovering motifs with meme. 0(0).

Bairoch, A., P.Bucher, and Hofmann, K. (1996). The prosite database: Its status in 1995. *Nucleic Acid Research*, 24(1):189–196.

Brazma, A., Jonassen, I., Eidhammer, I., and Gilbert, D. (1998). Approaches to the automatic discovery of patterns in biosequences. *Journal of Computational Biology*, 5(2):279–305.

Brejova, B., DiMarco, C., Vinar, T., Hidalgo, S. R., Holguin, G., and Patten, C. Finding patterns in biological sequences.

Brown, P. F., Pietra, S. A. D., Pietra, V. J. D., and Mercer, R. L. (1993). The mathematics of statistical machine translation: Parameter estimation. *Computational linguistics*, 19:263–312.

Califano, A. (2000). Splash: Structural pattern localization algorithm by sequential histograms. *Bioinformatics*, 16(4):341–357.

Corpet, F., Gouzy, J., and Kahn, D. (1999). Recent improvements of the prodom database of protein domain families. *Nucleic Acid Research*, 27(1):263–267.

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the em algorithm. *Journal of Royal Statist. Soc. B*, 39:1–38.

Hart, R. K., Royyuru, A., Stolovitzky, G., and Califano, A. (2000). Systematic and fully automated identification of protein sequence patterns. *Journal of Computational Biology*, 7(3).

Huang, J. Y. and Brutlag, D. (2001). The emotif database. *Nucleic Acid Research*, 29(1):202–204.

Jonassen, I., Collins, J. F., and Higglins, D. (1995). Finding flexible patterns in unaligned protein sequences. *Protein Science*, 4(8):1587–2595.

Lawrence, C. E., Altschul, S., Bogouski, M., Liu, J., Neuwald, A., and Wooten, J. (1993). Detecting subtle sequence signals: A gibbs sampling strategy for multiple alignment. *Science*, 262(0):208–214.

Marchler-Bauer, A., Panchenko, A. R., Shoemaker, B. A., Geer, P. A. T. L. Y., and Bryant, S. H. (2002). Cdd: a database of conserved domain alignments with links to domain three-dimensional structure. *Nucleic Acid Research*, 30(1):281–283.

Rigoutsos, I. and Floratos, A. (1998). Combinatorial pattern discovery in biological sequences: The teiresias algorithm. *Bioinformatics*, 14(1):55–67.

Rigoutsos, I., Floratos, A., Parida, L., Gao, Y., and Platt, D. (2000). The emergence of pattern discovery techniques in computational biology. *Metabolic Engineering*, 2(0):159–177.

Robertson, S. E. (1977). The probability ranking principle in IR. *Journal of Documentation*, 33(4):294–304.

Xie, H., Wasserman, A., Levine, Z., Novik, A., Grebinskiy, V., Shoshan, A., and Minitz, L. (2002). Large-scale protein annotation through gene ontology. *Genome Research*, 12(5):785–794.

Zhai, C. (2002). *Risk Minimization and Language Modeling in Text Retrieval*. PhD thesis, Carnegie Mellon University.