A Weighted \(k\)-Nearest Neighbor Method for Gene Ontology Based Protein Function Prediction

Saket Kharsikar\(^1\), Dale Mugler\(^{1,2}\), Daniel Sheffer\(^1\), Francisco Moore\(^{3,4}\), Zhong-Hui Duan\(^{3,5}\)

\(^1\)Department of Biomedical Engineering, University of Akron, Akron, OH, 44236
\(^2\)Department of Theoretical and Applied Mathematics, University of Akron, Akron, OH 44236
\(^3\)Integrated Bioscience PhD Program, University of Akron, Akron, OH 44236
\(^4\)Department of Biology, University of Akron, Akron, OH 44236
\(^5\)Department of Computer Science, University of Akron, Akron, OH 44236
duan@uakron.edu

Abstract

Numerous genome projects have produced a large and ever increasing amount of genomic sequence data. However, the biological functions of many proteins encoded by the sequences remain unknown. Protein function annotation and prediction become an essential and challenging task of post-genomic research. In this paper, we present an automated protein function prediction system based on a set of proteins of known biological functions. The functions of the proteins are characterized with gene ontology (GO) annotations. The prediction system uses a novel measure to calculate the pair-wise overall similarity between protein sequences. The protein function prediction is performed based on the GO annotations of similar sequences using a weighted \(k\)-nearest neighbor method. We show the prediction accuracies obtained using the model organism yeast (Saccharomyces cerevisiae). The results indicate that the weighted \(k\)-nearest neighbor method significantly outperforms the regular \(k\)-nearest neighbor method for protein molecular function prediction.

1. Introduction

Since the completion of the human genome project, research in genomics has led to a tremendous progress in our comprehensive understanding of biology and medicine. The numerous genome projects have generated vast amounts of sequence data that keeps increasing day by day. However, the biological functions of a large proportion of the sequenced proteins remain unknown. One of the major post-genomic research challenges is to characterize the interrelation between the nucleotide sequences of the genes and the functions of the proteins that they encode and consequently predict the functions of these uncharacterized proteins. With a very high exponential growth in the number of protein sequences obtained over the past decade, it is infeasible to determine the biological functions of each protein by actual biological experimentation. Therefore, it is necessary to develop computational approaches for the functional characterization of the proteins.

Traditionally, the functional annotation of genes has been done manually by the experienced individual curators with the help of advanced searching tools. The function for a protein is assigned based on the characteristics of proteins whose biological functions are known and whose amino acid sequence matches that of the query protein. Such homology methods are commonly used to extend the functional knowledge of proteins to other proteins that are apparently the descendants of some common ancestral protein. Recently, automated annotation systems have been developed to accelerate the protein function annotation process [1-10]. The annotation systems are mainly based on the sequence similarity and/or structure similarity. On the other hand, with the latest biotechnological advances, co-expression relationships obtained from microarray gene expression patterns have also been explored to describe the functional correlations between proteins [11-15]. The assumption behind the approaches is that co-expressed genes might be involved in related biological activities. Therefore, one can expect that proteins tend to have similar biological functions if they exhibit similar expression patterns across diverse conditions.

Since the establishment of gene ontology (GO) [16], many ontology-based sequence annotation
approaches have been developed [14, 17-29]. GO provides a unified vocabulary to describe gene and gene product attributes in any organism. It includes three ontology categories: molecular function, biological processes, and cellular components. A molecular function GO term represents a biological activity involving one or more gene products. A biological process GO term represents a series of biological activities and a cellular component GO term, as the name suggests, represents a component of a cell. The GO terms in each category are organized in a Directed Acyclic Graph (DAG), i.e., a specialized GO term (child) could be associated with one or several less specialized GO terms (parents).

Ontology-based sequence annotation approaches often involve a search of homologous proteins in GO-mapped databases including Genbank and Swiss-Prot. Henning et al’s OntoBlast and Zehetner’s GOBlet present a list of homologues together with their GO terms [22-23]. Martin et al’s GOTcha searches a set of seven model genomes and returns scored matches [24]. Xie et al’s GO Engine combines homology search with text mining [25]. Schug et al developed a rule-based function prediction method based on the intersection of GO terms that contain protein domain at different similarity levels [26]. Abascal et al. presented an automatic annotation method based on the protein family identification [27]. Jensen et al. used neural networks for predictions while Vinayagam et al. used support vector machines [28-29]. More recently, Nariai et al. combined several different data resources and developed a prediction method on a heterogeneous genome wide dataset [30]. The appeal of these approaches is that they can directly assign a biological meaning to an uncharacterized protein sequence.

In this paper, we present a weighted k-nearest neighbor classification system for automated protein function annotations. The complete proteome from the model organism yeast \textit{Saccharomyces cerevisiae} is used. The approach is based on molecular function gene ontology and the protein-sequence similarities. A novel measure of overall similarity between two protein sequences was utilized in the study. The measure combines a set of similarity scores obtained from a local alignment method such as BLAST. We show that the weighted k-nearest neighbor method based on the measure significantly improves the prediction accuracy, compared to the regular k-nearest neighbor method.

### 2. Materials and Method

In this study, we used the complete yeast (\textit{Saccharomyces cerevisiae}) proteome [31]. It includes 6467 protein sequences. A GO definition file was obtained from the Gene Ontology consortium web site [32]. It includes 19094 GO terms, including 9856 biological process terms, 7559 molecular function terms, and 1679 cellular component terms. Among the 6467 protein sequences; 4175 are annotated with 1084 biological process terms, 3317 are annotated with 1060 molecular function terms, and 4735 are annotated with 354 cellular component terms. The final biological process ontology tree consists of 11 levels and 11091 tree nodes (GO groups), of which 903 are unique. The molecular function ontology tree consists of 9 levels and 471 tree nodes of which 362 are unique. The cellular component tree has 7 levels and 1692 tree nodes, of which 284 are unique. Since we are focusing on the protein function prediction, only molecular function groups and the yeast protein sequences annotated with molecular function terms are considered in this study.

The GO terms in the molecular function ontology category were parsed and stored in a tree structure similar to the one used in AmiGO [33] to form a GO tree. Since GO terms are originally organized in a DAG, a GO term may have several parent terms. In this case, the child term appears multiple times on the same level or on different levels of the tree. Protein sequences were then mapped onto the trees. The groups of less than six proteins were removed for statistically meaningful results. Figure 1 shows a protein mapped molecular function ontology tree. The number in the parentheses after each functional group indicates the number of proteins mapped to the group. For example, 218 proteins are annotated with the DNA binding molecular function group.

The weighted k-nearest neighbor classification system developed in the study includes three main computational steps as outlined in Figure 2. We note that all processing scripts for the implementation of the steps were written in Perl and the graphic user interface was written in Java.

The protein sequence similarities are evaluated using a novel measure of overall similarity of two protein sequences [34]. Let \( S_1, ..., S_h \) be scores of a list of best local alignments with certain statistical significance. We use the \( p \)-value:

\[
p = e^{-S} \tag{1}
\]

to measure the probability of finding a pair of protein sequences with a list of scores at least \( S_1, ..., S_h \), where \( S \) is a measure of the overall similarity of two sequences:
\[ S = -\sum_i \ln p_i \]  

(2)

and \( p_i = 1 - e^{-E_i} \) stands for the probability of finding a high-scoring segment pair (HSP) with a local alignment score of at least \( S_i \) and \( E_i \) is the expected number of HSPs of score of at least \( S_i \). The \( E \)-values of \( E_i \) can be obtained directly from an alignment tool such as BLAST. The alignment tool for blasting two sequences (bl2seq) [35] was used in this study. We used version 2.2.11 with default parameters and the substitution matrix BLOSUM62.

The \( K \)-nearest neighbor method is a standard learning method. The training samples of different groups are mapped into a feature space. An uncharacterized sample is assigned to a specific group if the group label is the most frequent label among the \( K \) nearest training samples. \( K \)-nearest neighbor method is one of the most popular methods for classification and has been heavily investigated in the fields of data mining and pattern recognition. However, the accuracy of the method can be severely reduced by the presence of noisy, irrelevant features or inappropriate scaling of the features.

In this study, we develop a weighted \( K \)-nearest neighbor method. The weight of a feature (protein) is calculated based on the overall protein sequence similarity or \( p \)-value measured using Equation (1). Two proteins are considered to be neighbors of each other if the similarity \( p \)-value of the two proteins is less than or equal to a specified threshold. Given an uncharacterized protein \( i \), a list of GO annotations of the \( K \)-nearest neighbor proteins \( \{GO_j\} \) is retrieved. The GO terms are then ranked based the magnitude of the \( p \)-values:

\[ \text{rank}(GO) = \sum_{1 \leq j \leq K} -\log(p_{ij}) \]  

(3)

where protein \( j \) is one of the \( K \)-nearest neighbors and \( p_{ij} \) is the similarity \( p \)-value of the protein pair \( i \) and \( j \). A value \(-300\) is assigned to \( \log(p_{ij}) \) when \( p_{ij} \) is zero.
The GO term with highest ranking will then be assigned to protein $i$. In the case that two or more GO terms are of same ranking score, the number of neighbors $k$ is increased by 1 at a time until the tie is broken. The main advantages of a weighted $k$-nearest neighbor method is that the method is much more stable with the change of $k$ and it significantly reduces the noise that might be introduced by proteins that are not similar to the query protein. To compare the improvement of the weighted $k$-nearest neighbor method, we implemented the regular $k$-nearest neighbor method using Equation (4):

$$\text{rank}(GO_i) = \sum_{1 \leq j \leq k \text{ protein } j \text{ is annotated with } GO_i} 1$$  \hspace{1cm} (4)

where protein $j$ is one of the $k$-nearest neighbors.

3. Results and Discussion

We first performed all-to-all pair-wise protein sequence local alignments using the alignment tool (bl2seq) for comparing two sequences. The $p$-values were then calculated based on Equation (1). The weighted $k$-nearest neighbor method based on sequence alignment was implemented by Equation (3). The regular $k$-nearest neighbor method was implemented using Equation (4). The performances of the methods are examined using the leave-one-out cross-validation scheme. The scheme selects one protein from the original set of 3247 proteins as the validation data and the remaining proteins as the training data. We note that 70 other proteins with molecular function annotation are excluded because they are not similar ($p$-value $\geq 0.1$) to any other proteins in the set. Prediction is performed on the selected protein. The process is repeated for every protein in the dataset. The accuracy is then calculated based on the percentage of correct predictions. The accuracy results are shown in Figure 3-5. As we clearly see, the weighted $k$-nearest neighbor method increases prediction accuracy significantly. Figure 3 shows a 7% improvement for the prediction of the biological functions at the first level of GO tree, when $k = 4$ and the accuracy level is 70% for the regular $k$-nearest neighbor method. The accuracy of the weighted $k$-nearest neighbor method slightly increases from 75% to 76% and is stabilized at this level. On the other hand, when $k \geq 4$, we see a steady decrease of the accuracy obtained from the regular $k$-nearest neighbor method as more irrelevant proteins are included in the prediction. Figure 4 and 5 show the prediction accuracy for the GO terms on the second and third levels of the GO tree. The differences between the weighted and regular $k$-nearest neighbor methods are much more pronounced. For GO terms on the second level of GO tree, the prediction accuracy of the weighted $k$-

![Figure 3. Prediction Accuracy at the 1st Level](image3)

![Figure 4. Prediction Accuracy at the 2nd Level](image4)

![Figure 5. Prediction Accuracy at the 3rd Level](image5)
nearest neighbor method increases steadily to about 60\%. Comparing with regular \( k \)-nearest neighbor method, the accuracy is increased by about 25\% when \( k = 10 \). We do see a slight decrease in accuracy for GO terms on level 3. However, the accuracy of the weighted \( k \)-nearest neighbor method is clearly stable as \( k \) changes. On the other hand, the performance of the regular \( k \)-nearest neighbor method degenerates as \( k \) increases.

In this study, we also explored the protein function prediction accuracy using protein transcription-level expression patterns. Yeast cell cycle data [36] was used. The dataset includes 24 gene expression profiles measuring the mRNA expression levels of the yeast proteins during the growth of yeast cells [36-37]. Each profile includes 6178 related proteins. The expression similarities were measured using the Pearson correlation coefficients between the expression levels of a pair of co-expressed genes. The accuracy of prediction results obtained from the weighted \( k \)-nearest neighbor method based on the gene expression patterns is lower than the ones based on sequence similarity. The lower accuracy might be due to the dataset used in this study which mainly reflects the gene expression pattern changes during the yeast cell cycle. Another possible reason is that the microarray technology is used to measure the transcription levels of the gene expressions, which may not represent the true levels of the protein expressions in the cell.

4. Conclusions

In this study, we developed a weighted \( k \)-nearest neighbor classification system for automated protein function annotations. The classification system was tested on the dataset of a complete yeast proteome. The system identifies homologous proteins using a novel measure of the overall similarity between two protein sequences. The biological functions of proteins are described using GO annotations. The function prediction is performed using a weighted \( k \)-nearest neighbor method. The prediction results indicate that there is a strong correlation between proteins amino acid sequence similarity measured by Equation (1) and their biological function similarity. The weighted \( k \)-nearest neighbor method significantly improves the prediction accuracy comparing with regular \( k \)-nearest neighbor method. Furthermore, we see that the weighted \( k \)-nearest neighbor method is a very stable algorithm with respect to the change of \( k \). We conclude that the method significantly outperformed the regular \( k \)-nearest neighbor method for automated protein function predictions.

5. Acknowledgement

The work is partially supported by NSF DUE 0410727 and UA 2007 faculty research fellowship (ZHD).

6. References


[33] “AmiGO,” http://www.godatabase.org/cgi-bin/amigo/go.cgi
