Predicting Protein Function using Text Data from the Biomedical Literature

Andrew Wong[1]*, Hagit Shatkay[1,2,3,]*

[1]School of Computing, Queen's University, Kingston, ON, K7L 3N6, Canada
[2]Dept. of Computer and Information Sciences, University of Delaware, Newark, DE, 19716, US
[3]Delaware Biotechnology Institute, University of Delaware, Newark, DE, 19711, US

*To whom correspondence should be addressed: 3aw14@queensu.ca, shatkay@cis.udel.edu

1. INTRODUCTION

The past decade has been marked by a rapid growth in the number of newly discovered genes and proteins. While their sequences are available, their function remains unknown. Thus, a central goal of computational biology is to develop methods that can accurately predict gene and protein function; this is the motivation behind the CAFA (Critical Assessment of Function Annotations) Challenge. This abstract presents a text-based protein function prediction system we developed and submitted to the CAFA Challenge.

Most existing prediction systems are based on features derived from protein sequence, structure, protein-protein interaction data, or an integration of such features (1). In contrast, our system uses text-based features, derived from the biomedical literature, to characterize proteins. The main idea behind our method is that we can extract from abstracts associated with proteins, key-terms that are correlated with different aspects of a protein, which can be used to represent it. Given a set of proteins whose function is known, a classifier based on this protein-text-representation can be trained to predict function. The motivation of using text-based features lies in the abundance of literature discussing proteins, and the readily understandable semantics of text-based features. In an earlier work (2) our team showed that text data can be integrated with sequence data to accurately predict protein subcellular location. We adopted the same text-based classification framework, and modified it to predict protein functions.

2. METHODS

To adapt the classification framework used originally in the protein localization system EpiLoc (2) to predict protein function, we modified the feature selection process and implemented a different classifier. We established our training sets for the *Biological Process (BP)* and *Molecular Function (MF)* functions in GO *separately* by identifying in Uniprot proteins with a *BP or MF* annotation. To train our classifier only on proteins whose function is confirmed (experimentally), we removed from the set proteins with computational evidence codes: ISS, ISO, ISA, ISM, IGC, RCA, IEA and NAS. The resulting dataset contains 62,022 *BP* proteins and 30,921 *MF* proteins. To build the body of text associated with each protein, we compiled a collection of 68,337 abstracts by extracting all the PubMed identifiers appearing in the respective UniProt entries and downloading the abstracts from PubMed. We then extracted key-terms from the collection of text through feature selection as described next.

The feature selection step identifies distinguishing terms whose distribution in text is statistically different between different function classes. In contrast to our location prediction system (2) that focused on a relatively small number of organelles as classes (at most 13), in the CAFA challenge the potential *function classes* are about 20,000 *BP* terms and about 9,000 *MF terms* in GO. This large number of classes makes it difficult to identify key-terms that uniquely differentiate between pairs of classes. Moreover, as GO forms a hierarchy, the different function classes are not independent of each other. As the publications associated with proteins in offspring-classes may share the same characteristic vocabularies with those associated with proteins in their parents' class, informative key-terms may appear to be non-distinguishing.

To address both issues, we iteratively collapsed descendant classes into their respective parent classes, starting at the leaves and continuing up to the second level of GO's *BP* and *MF* ontology. After merging offspring classes, we are left with 10 classes in *MF* and 24 classes in *BP*. Based on the set of merged classes, we then selected characteristic key-terms using the Z-score criterion and represented proteins as a vector of the key-terms frequencies as done earlier in the EpiLoc system (2).

For our classifier, we chose to use the k-Nearest Neighbors (kNN), as opposed to support vector machines used before (2), because it is simple to implement and to modify (3). In compliance with the CAFA Challenge requirements, we modified the kNN algorithm to include multi-class classification and assigned a confidence score to each predicted function. The modified kNN, with *k = 10*, finds for each protein, represented as a feature vector, its 10 nearest neighbors in the training set, as measured by the Euclidean distance. If three or more of the 10 nearest neighbors have the same function(s), this function(s) is associated with the protein along with a confidence score. For a protein, *p*, and a predicted function, *f*, the confidence score $C_f(p)$ is calculated as:

$$C_f(p) = 1 - \frac{\sum_{i=1}^{|N^f|} d_f\left(N_i^f\ ,p\right)}{|N^f|}\ ,\qquad 3 \leq |N^f| \leq 10,$$

where, out of the 10 nearest neighbors of the query protien p, $|N^f|$ is the number of nearest neighbors with function *f*, $d_f(N_i^f, p)$ is the normalized distance between p and its *i*'th nearest neighbor with function *f*, N_i^f ; normalization is done by dividing the distance between N_i^f and *p* by the maximum distance between any two proteins in the training set. The average normalized distance over the neighbors with function *f*, is subtracted from 1 so that a shorter distance corresponds to a higher confidence.

3. RESULTS

Prior to submitting the official results on the CAFA dataset, we evaluated our kNN classifier using 5-fold cross validation and compared it to a baseline classifier (denoted as *Rand*) that assigns classes at random based on the distribution of proteins in GO function classes within the training set. The results are measured in terms of average precision (AP) and average recall (AR). The AP and AR are calculated by dividing the sum of all classes' precision and recall, respectively, by the number of proteins in the training set, where the summands are weighted by the number of proteins in each class. The results are shown in Table 1. The function class with highest predicted accuracy in *MF*, '*binding'*, and has a precision of 0.64 a recall of 0.96; it contains 20,097 proteins. In contrast, the BP function class '*response to stimulus'* with the best performance has a precision of 0.23, a recall of 0.19; 6,573 proteins are in this class. For half of the *MF* classes, which have more than 1000 proteins, the precision ranges from 0.3-0.6, up to 0.3 better than the baseline. However, for classes with fewer than 1000 proteins, there is only little improvement in performance. Three *MF* and three *BP* classes, have fewer than 100 proteins, and the *kNN* classifier makes no predictions for those.

	AP (kNN)	**AP (Rand)**	**AR (kNN)**	**AR (Rand)**
Molecular Function	0.54	0.44	0.64	0.45
Biological Process	0.17	0.11	0.17	0.13

Table 1. Performance of our classifier (kNN) compared to the random (Rand) classifier

4. CONCLUSION AND OUTLOOK

Our classifier gives more accurate predictions for the *MF* than the *BP* ontology. This is probably because we merged more than twice as many classes in *BP* than in *MF*. Also, since a single *BP* function involves many different proteins with different molecular functions, specifically selecting distinguishing key-terms that are related to protein-protein interactions and biological processes, as opposed to sequence and structure, may improve classification for *BP*. For both *MF* and *BP*, our classifier performs better when classes have more than 1000 proteins. As for the small classes for which no predictions were made, it may prove beneficial to combine them into a single category during classification and later refine the classification using sequence data. We plan to investigate the proteins and the associated text from these small classes to explain the lack of predictions.

We also plan to combine text data with other types of data to improve prediction performance, in particular, for rare and small classes. Moreover, it is likely that a multi-resolution classification scheme will enable high-level predictions with high confidence scores to be refined to classes at lower levels of the GO.

5. REFERENCES

1. Friedberg I. 2006. Automated protein function prediction – the genomic challenge. *Briefing in Bioinformatics*. 7b:225-242
2. Shatkay H., Hoglund A., Brady S., Blum T., Donnes P., and Kohlbacher O. 2007. SherLoc: High-Accuracy Prediction of Protein Subcellular Localization by Integrating Text and Protein Sequence Data. *Bioinformatics.* 23:1410-1417
3. Yao Z. and Ruzzo W. L. 2006. A regression-based K nearest neighbor algorithm for gene function prediction from heterogeneous data. *BMC Bioinformatics*. 7(Suppl 1): 11