Incorporating Functional Inter-relationships into Algorithms for Protein Function Prediction

Gaurav Pandey*, Vipin Kumar
Department of Computer Science and Engineering, University of Minnesota, Twin Cities
200, Union Street SE, Minneapolis, MN 55414, USA
*To whom correspondence should be addressed: gaurav@cs.umn.edu

1. INTRODUCTION

A variety of recently available high throughput data sets, such as protein-protein interaction networks, microarray data and genome sequences, offer important insights into the mechanisms leading to the accomplishment of a protein's function. However, the complexity of analyzing these data sets manually has motivated the development of numerous computational approaches for predicting protein function (1). Several of these approaches use data mining and machine learning techniques for this task, and have produced very encouraging results. For a recent comprehensive survey on this topic, see reference (2).

Commonly used data mining techniques for the task of protein function prediction consider the functional classes to be used for annotation as independent of each other. However, it is well known that a protein may perform multiple functions, which may further have significant inter-relationships when viewed as concepts in a widely accepted hierarchical organization of functional classes such as Gene Ontology (3). Traditional techniques do not handle such inter-relationships, hence by incorporating them, the performance of protein function prediction algorithms could be improved.

In this paper, we use the similarity measure defined by Lin (4) as a measure of the similarity between two functional classes, and modify the traditional *k*-nearest neighbor classification algorithm to take this similarity into account. Evaluation of the algorithm on functional classification of gene expression data indicates that the use of inter-relationships between functional classes indeed substantially improves the accuracy of the hypotheses generated by protein function prediction algorithms.

2. PROPOSED APPROACH

The traditional k-NN classifier determines the annotations of a protein by finding all abundant functional classes in its neighborhood, which is the set of k proteins nearest to p in the data set, using the formula:

$$classes(p) = \{c \mid (\sum_{p' \in nbd(p)} sim(feature(p), feature(p')) * [c \in classes(p')]) > threshold_1\}$$

Kuramochi *et al* (5) showed that this simple algorithm performed comparably to more powerful classification algorithms such as SVMs for functional classification of gene expression data. We modified the above formula as follows, to take the similarity between functional classes into account:

$$classes(p) = \{c \mid (\sum_{p' \in nbd(p)} sim(feature(p), feature(p')) * \max_{c' \in classes(p')} \{sim(c,c')\}) > threshold_2\}$$

Thus, if a protein p is strongly expected to belong to class c, but its neighborhood does not contain enough evidence for this annotation, then the above formula enables other proteins in the neighborhood to contribute to this evidence, in proportion to the similarity of its most similar class to c. This incorporation of class similarities is expected to have the advantage of improving the predictions for proteins that do not have enough evidence for annotation by a certain class as per the original function prediction hypothesis, by enabling the transfer of annotations from close proteins annotated with similar classes. Thus, this method makes the hypothesis underlying automated function prediction more flexible by using the similarity of features of two proteins to indicate functional *similarity* (6) instead of functional *equivalence*, as assumed by most current techniques.

We use the hierarchical organization of functional classes in the Gene Ontology to model the similarity of two classes (nodes) in GO using Lin's measure (4): $sim(c_1,c_2) = \frac{2 \times [\ln p_{ms}(c_1,c_2))]}{\ln p(c_1) + \ln p(c_2)}$. Here, c_1 and c_2 are functional classes in one of the GO hierarchies, $p(c)$ is the probability of a protein being annotated with class c, and $p_{ms}(c_1,c_2) = \min_{c \in S(c_1,c_2)} \{p(c)\}$, where $S(c_1,c_2)$ is the set of common ancestors of c_1 and c_2. This

measure evaluates the similarity of two nodes in a hierarchy in terms of the population of the *least* common ancestor, and is normalized to have a value in the range of [0,1]. The use of Gene Ontology to identify inter-relationships between functional classes, and the use of the above similarity measure to quantify these inter-relationships enables us to incorporate biologically significant knowledge into our function prediction algorithm, and improve the performance of previous algorithms, as detailed in the following section.

Fig 1: Original class similarity matrix derived using Lin's measure

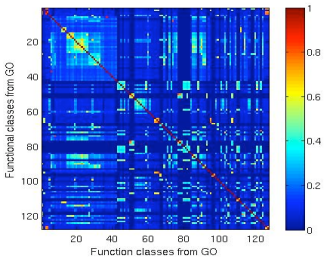


Fig 2: Optimally filtered class similarity matrix producing best AUC performance

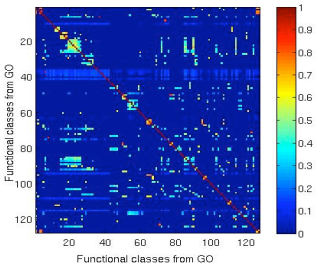
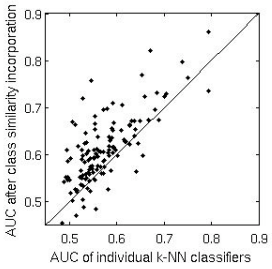


Fig 3: Improvement in AUC score of each class using class similarity



3. RESULTS AND DISCUSSION

We used Mnaimneh *et al*'s gene expression data set (7) to test the effectiveness of incorporating functional class similarities into a functional classification algorithm. This data set measures the expression of 6306 genes from *S. cerevisiae* under a set of 215 titration experiments. A set of 127 functional classes chosen were from the *biological process* ontology of GO, each having at least 10 members, and had been suggested by Myers *et al* to be testable in a wet lab (8). Figure 1 graphically shows the matrix of pair-wise similarities between these classes calculated using Lin's similarity measure. The density of this matrix suggests that it is likely to contain spurious similarities due to factors such as the abundance of classes from the *cellular process* ontology in the target set. Hence, we used an appropriate similarity filtering threshold for each class, which was done as follows. The original data set was split into two halves, and the first half was used as input to the label similarity-incorporated k-NN classifier discussed above. Through a five-fold cross validation procedure, the classification performance was evaluated for each class in terms of the AUC measure for several thresholds, and the one producing the best performance was chosen as the best filtering threshold for each class. Figure 2 shows the resultant filtered class similarity matrix.

Now, this similarity matrix is used for the predicting the functions of the genes in the other half of the original data set through a five-fold cross validation procedure, which is run multiple times in order to obtain robust estimates of the AUC scores for several classes. In addition, the basic k-NN classifier (5) is also used for predicting the functional classes of all the genes in the dataset through a five-fold cross validation procedure, and the AUC score of each class is computed. Figure 3 shows the comparison of the AUC scores of the 127 classes obtained using both the class similarity-equipped (y-axis) and the basic k-NN algorithms (x-axis). This plot shows that the performance of 102 classes is improved by considering similarities between classes, with the improvement being very significant for several classes, while the performance for the other 25 classes is only slightly deteriorated. These results show the utility of modeling similarities between functional classes as a way of incorporating the knowledge embodied in Gene Ontology, and thus producing more accurate predictions for proteins. Generalization of this concept for use with other classification methods, such as SVM, and other types of biological data, such as protein-protein interaction networks, is in progress.

4. REFERENCES

1. Marcotte, E. M., 2000, Computational genetics: finding protein function by nonhomology methods, Curr Opin Struct Biol. 10, 3, 359–365
2. Pandey, G., Kumar, V. and Steinbach, M., 2006. Computational Approaches for Protein Function Prediction: A Survey, TR 06-028, Dept of Comp Science and Engineering, University of Minnesota, Twin Cities
3. Ashburner, M. *et al*., 2000. Gene Ontology: tool for the unification of biology, *Nat. Genet.*, 25(1), 25–29.
4. Lin D., 1998, An Information-Theoretic Definition of Similarity, Proc. Intl. Conf. Machine Learning, pp 296-304
5. Kuramochi, M. *et al*, 2005. Gene Classification Using Expression Profiles: A Feasibility Study, Intl. J. Artificial Intelligence Tools, 14(4): 641-660
6. Lord PW *et al*, 2003, Semantic similarity measures as tools for exploring the gene ontology, in PSB, pp 601-612
7. Mnaimneh, S. et al, 2004, Exploration of essential gene functions via titrable promoter alleles, Cell, 118(1), 31-44
8. Myers C.L. *et al*, 2006, Finding function: evaluation methods for functional genomic data, BMC Genomics, 7:187