

# Gene function prediction using protein domain probability and hierarchical Gene Ontology information

Jaehee Jung<sup>1</sup> and Michael R. Thon<sup>2</sup>

<sup>1</sup>Department of Computer Science, Texas A&M University, College Station, TX, 77843 USA

<sup>2</sup>Centro Hispano-Luso de Investigaciones Agrarias (CIALE)

Department of Microbiology and Genetics, University of Salamanca, Salamanca, 37185, Spain

jaeheejung@tamu.edu, mthon@usal.es

## Abstract

*The Gene Ontology (GO) is a controlled vocabulary of terms to describe protein functions. It also includes a hierarchical description of the relationships among the terms in the form of a directed acyclic graph (DAG). Several systems have been developed that employ pattern recognition to assign gene function, using a variety of features, including sequence similarity, presence of protein functional domains and gene expression patterns, but most of these approaches have not considered the hierarchical structure of the GO. The DAG represents the functional relationships between the GO terms, thus it should be an important component of an automated annotation system. We propose a Bayesian, multi-label classifier that incorporates the relationships among GO terms found in the GO DAG. A comparative analysis of our method to other previously described annotation systems shows that our method provides improved annotation accuracy when the performance of individual GO terms are compared. More importantly, our method enables the classification of significantly more GO terms to more proteins than were previously possible.*

## 1. Introduction

The development of automated methods for the annotation of predicted gene products (proteins) with functional categories is becoming increasingly important, in order to present genome sequences and genome annotations to biologists in a useful way. Thus, many systems to perform protein functional annotation have been developed that employ various sources of protein information as features, including protein functional sites [5], sequence similarity [6, 8, 13], gene expression

patterns [9], and others. Controlled vocabularies such as the Gene Ontology (GO) are also becoming increasingly important for automated annotation methods.

Often, automated gene annotation methods ignore the hierarchical nature of the controlled vocabulary in order to simplify the classification problem [5, 6, 9, 13]. However, several authors have recently used hierarchical information for gene function annotation [2, 4, 7]. In a comparative analysis, Eisner et al. [4] report that the training set including hierarchical information outperformed similar data in which the hierarchical nature of the data was ignored. In another study, Shahbaba and Neal [12] describe three multinomial logit models although the hierarchy used in this study was a simple tree-like structure in which each node has only one parent. This approach is not sufficient to apply to GO, since GO employs a Directed Acyclic Graph (DAG) in which each node may have multiple parents. King et al. [7] predict gene function based on the relationship between GO annotations using decision trees and a Bayesian network. Barutcuoglu et al. [2] also used a Bayesian network for the purpose of developing a multi-label annotation method, overcoming the shortcoming of inconsistency between the child and parent annotations by preventing child terms from being annotated if the parent is not annotated [3].

Our approach uses the GO DAG scheme to represent a Bayesian network in which the GO terms represent nodes. Using training data consisting of annotated proteins, we calculate the prior probability of each GO term and each linked parent GO term and the conditional probability of each InterPro term at each node. To assign GO terms to unlabeled proteins, we compute the Bayesian probability of each GO term, beginning at the root node of the network. GO terms are assigned recursively by following edges to the child nodes. When the probability of a GO term is below a threshold, we

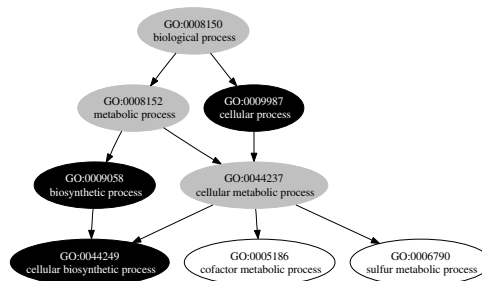
stop testing the child nodes. In the last step, we apply a filter that removes annotations on the basis of the relationship between the GO terms and the feature set. The results of comparative tests show that, compared to non-hierarchical protein domain annotation methods, our method provides more accurate annotations.

## 2. Methods

### 2.1. Data sources

In this study we employ InterPro<sup>1</sup> terms as features, which reflect the presence of conserved functional domains in the proteins. The data set is composed of annotated protein sequences from the UniProt database [1]. Each protein has one or more InterPro terms, since our method employs InterPro terms as features. Each protein has previously been annotated with one or more GO terms using a variety of methods. We removed proteins that lack GO annotations and annotations with the evidence code “Inferred from Electronic Annotation” (IEA) which represent annotations derived from other automated annotation methods. The remaining annotations are those that have been reviewed by subject matter experts. With this data set, two matrices are created, which are  $GO(i, k)$  and  $IPR(i, j)$ , where  $i$  is the number of proteins,  $k$  is the number of GO terms and  $j$  is the number of InterPro terms. The matrix  $IPR$  is the InterPro term matrix, so, if the  $i$ th protein has the  $j$ th InterPro term,  $IPR(i, j)$  is a binary value indicating the presence of the functional domain. The  $GO$  matrix is also built in the same way, that is, the  $i$ th protein has the  $k$ th GO term, and is a binary value representing the presence of the functional category. Since our interest is in the annotation of proteins from fungi, the data set used in this project are proteins from fungi and is composed of 6711 proteins, 3339 InterPro terms, and 3096 GO terms.

Each node in the GO structure represents a protein function which is more specific, but related to the function of the parent nodes to which is connected. Nodes with higher positions in the GO structure represent more general functions, while nodes lower in the structure represent more specific functions, thus, parent GO terms may also be used to describe a protein’s function. For example,  $GO:0044249$  in Fig. 1 has two parent terms,  $GO:0009058$  and  $GO:0044237$ , which have more general functions than  $GO:0044249$ . If a protein is annotated with  $GO:0044249$  then by inference, the protein’s function can also be described by the parent nodes ( $GO:0044237, GO:0008152$  as well as all



**Figure 1. The hierarchical structure of the Gene Ontology**

the linked parent terms. Hence, this protein’s function can be described by the set of all GO terms except for the white colored GO terms in Fig. 1.  $GO:0005186$  and  $GO:0006790$  would not be included in the data set, since they are neither original GO terms nor a GO terms’ parent. Finally,  $k$  in the matrix  $G$  is the list of not only original GO terms that are found in the training data set but also all parent terms of the original GO terms.

For evaluation purposes, we hold out 10% for the validation and the remaining 90% is used for training set. Hence, 6033 proteins are used for training set which calculate the probability of the InterPro term given the true or false GO term, where each feature is independent in each classifier.

### 2.2. Algorithm

The structure of the Bayesian network is determined by the DAG of the Gene Ontology project, which is ultimately determined by subject matter experts. Given the training data, consisting of proteins that have also been annotated with GO terms by subject matter experts, we compute the prior probability of each node. Using the InterProScan application [10], InterPro terms can be automatically assigned to proteins, thus, we can compute the conditional probability of each InterPro term at each node in the network.

To infer GO terms for unannotated proteins, two steps are performed. First, the InterProScan application is used to assign InterPro terms to proteins. Next, the Bayesian probability for each GO term is calculated given the InterPro terms as shown in (1).

$$P(X_{1 \in \{F, T\}}, \dots, X_{i \in \{F, T\}}) = \prod_{i=1}^v P(X_i | Par(X_i)) \quad (1)$$

<sup>1</sup><http://www.ebi.ac.uk/interpro/>

, where  $X_i$  is  $i$  GO term in the network and  $Par$  is the probability of the parent terms.  $P(X_i)$  can be expressed as (2) and the conditional probability of (1) can be inferred from the training set and the probability given the InterPro term.  $M$  is the set InterPro terms determined by InterProScan and  $Z$  is the normalized constant value. Each InterPro term  $I_i$  is independent  $I_j (j \neq i)$ , thus we can get conditional probability by the multiplication.

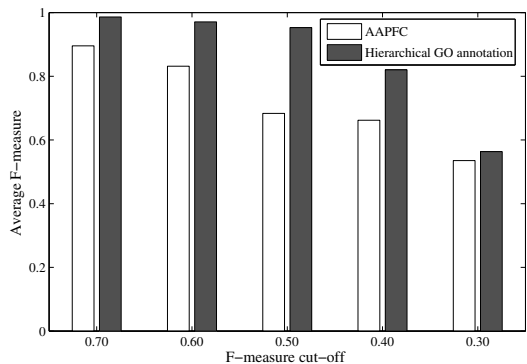
$$P(X_i) = P(G_i | I_1, \dots, I_M) = \frac{P(G_i)P(I_1, \dots, I_M | G_i)}{Z} \quad (2)$$

We calculate the Bayesian probability of the presence of each GO term, in a protein’s annotation, vs. its absence, beginning with the root node. If the probability of the presence of the GO term is larger than the probability of its absence, we assign the GO term and recursively compute the probability of the child terms. If the probability of the absence of the GO term is larger, then we do not assign the GO term to the protein and do not test the child terms.

After the candidate GO terms are determined, a filtering step as previously described [5] is applied. From all the proteins in the training data set we make a list of the occurrence pair between the InterPro terms and GO terms. Then, we examine the candidate GO term and the assigned InterPro term for the protein and determine if a InterPro term-GO term exists in the list. If there is not a InterPro term-GO term pair in the list, then the GO term is removed from the list. This filtering step serves two purposes. 1) It prevents higher level parent GO terms that do not exist in the training set from being assigned to annotated proteins. While such higher level terms may have some utility, depending on the type of downstream sequence analysis, we felt it was important not to allow the annotation system to transitively assign higher level, and less informative annotations to new proteins. 2) Protein functional domains are determinants of function, so we would expect that most of the information present in this data set to exist as positive relationships between InterPro terms and GO terms. As we have shown previously, annotated GO terms that are not associated with an InterPro term in the training set are often false positives [5]. Thus, this filtering step has been shown to improve the performance of the classifier.

### 3. Experiments

In our previous work, we developed a classifier called AAPFC that uses independent Support Vector Machines (SVMs) classifiers for assigning GO terms [5]. The method presented here improves upon our previous work by considering the hierarchical structure of



**Figure 2. Performance of hierarchical GO annotation at various F-measure cutoff values**

**Table 1. The number of GO terms in the classifier and the percentage of proteins that have GO terms at each F-measure cut-off.**

Cut-off F-measure	# of GO		% of protein	
	AAPFC	Sugg.	AAPFC	Sugg.
0.60	15	204	6.5	7.62
0.50	36	292	9.49	10.95
0.40	62	253	17.09	30.47
0.30	105	452	33.85	53.57
0.20	216	629	56.29	75.65

the Gene Ontology. thus, a comparison to the previous method should be helpful to understand of the importance of the use of the GO hierarchy in the model. Fig. 2 shows the results of a comparison of this method to that of AAPFC. For this comparison, performance metrics were estimated with 10-fold cross validation, and the performance is represented as F-measure. F-measure is calculated by  $(2 * Recall * Precision) / (Recall + Precision)$ . Each GO term can be assigned an F-measure, and a cutoff value can be applied to exclude GO terms that have low performance. Even though the same data set is employed in the two methods, the suggested approach results in increased the percentage of annotated proteins. As shown in Table 1, our new approach enables us to train a classifier that can assign significantly more GO terms than our previous method. Considering GO terms with an F-measure higher than 0.30, nearly ten fold more GO terms can be classified using the new method (Table 1).

**Table 2. Performance comparison of 4 different gene function annotation systems.**

Protein	Sugg.	GOcat	IPR2GO	GOtcha
ASSY_YEAST	0.50	0.14	0.13	0.00
EFTU_YEAST	0.50	0.07	0.20	0.30
HSP7F_YEAST	0.46	0.06	0.11	0.00
IME4_YEAST	0.22	0.00	0.00	0.47
KAPA_YEAST	0.31	0.09	0.13	0.39
MCFS2_YEAST	0.32	0.07	0.00	0.16

We performed another comparison by randomly selecting 40 proteins and comparing the annotations derived from four annotation systems which employ different methods and data sources. GOcat [11]<sup>2</sup> is a genetic automatic text categorizer and GOtcha[8] also annotated by the sequence similarity, but IPR2GO is the functional mapping table between InterPro term and GO term. The selected probability cutoff value in Gotcha [8]<sup>3</sup> is 0.50 and others set to default value. The average F-measure value for the prediction in GOcat, Gotcha and IPR2GO is 0.05, 0.22 and 0.04 respectively, but new suggested hierarchical GO annotation is 0.28. Thus, the suggested approach outperformed the other methods. The table 2 shows the F-measure value of six of the 40 test proteins.

## 4. Conclusions

In this paper, we propose a method for assigning GO terms to proteins using InterPro terms as features and learning a Bayesian network and then scanning the GO hierarchy from the root node to leaf nodes. This approach provides improved performance when the F-measure of individual GO terms are compared, but more importantly, it enables us to include more GO terms in the classifier which in turn enables many more proteins to be annotated. One shortcoming of this strategy is that the annotation model uses only conserved functional domains in the form of InterPro terms as features. The combination of various other heterogeneous features can make a more robust annotation. Our long term plan is to develop a system for assigning GO terms using multiple heterogeneous feature types including biochemical properties, phylogenetic profile, sequence similarity, and others with hierarchical information.

<sup>2</sup><http://eagl.unige.ch/GOCat/>

<sup>3</sup><http://www.compbio.dundee.ac.uk/gotcha/gotcha.php>

## 5. Acknowledgements

This research was supported by funds from the United States Department of Agriculture (grant number 2007-35600-17829) and from the Programa Ramón y Cajal, Ministerio de Educación y Ciencia, Spain.

## References

- [1] A. Bairoch, R. Apweiler, C. H. Wu, W. C. Barker, B. Boeckmann, S. Ferro, E. Gasteiger, H. Huang, R. Lopez, M. Magrane, M. J. Martin, D. A. Natale, C. O'Donovan, N. Redaschi, and L.-S. L. Yeh. The universal protein resource (UniProt). *Nucleic Acids Research*, 33:D154–159, 2005.
- [2] Z. Barutcuoglu, R. E. Schapire, and O. G. Troyanskaya. Hierarchical multi-label prediction of gene function. *Bioinformatics*, 22(7):830–836, 2006.
- [3] N. Cesa-Bianchi, A. Conconi, and C. Gentile. Regret Bounds for Hierarchical Classification with Linear-Threshold Functions. *LNCS*, 3120:93–108, 2004.
- [4] R. Eisner, B. Poulin, D. Szafron, P. Lu, and R. Greiner. Improving protein function prediction using the hierarchical structure of the gene ontology. In *CIBCB*, pages 354–363, 2005.
- [5] J. Jung and M. R. Thon. Automatic annotation of protein functional class from sparse and imbalanced data sets. *LNCS*, 4316:65–77, 2006.
- [6] S. Khan, G. Situ, K. Decker, and C. Schmidt. GoFigure: Automated Gene Ontology Annotation. *Bioinformatics*, 19(18):2484–2485, 2003.
- [7] O. D. King, R. E. Foulger, S. S. Dwight, J. V. White, and F. P. Roth. Predicting Gene Function From Patterns of Annotation. *Genome Research*, 13(5):896–904, 2003.
- [8] D. Martin, M. Berriman, and G. Barton. GOtcha: a New Method for Prediction of Protein Function Assessed by the Annotation of Seven Genomes. *BMC Bioinformatics*, 5(1):178, 2004.
- [9] P. Pavlidis, J. Weston, J. Cai, and W. S. Noble. Learning gene functional classifications from multiple data types. *Journal of Computational Biology*, 9(2):401–411, 2002.
- [10] E. Quevillon, V. Silventoinen, S. Pillai, N. Harte, N. Mulder, R. Apweiler, and R. Lopez. Interproscan: protein domains identifier. *Nucleic Acids Research*, 33:W116–120, 2005.
- [11] P. Ruch. Automatic assignment of biomedical categories: toward a generic approach. *Bioinformatics*, 22(6):658–664, 2006.
- [12] B. Shahbaba and R. Neal. Gene function classification using bayesian models with hierarchy-based priors. *BMC Bioinformatics*, 7(1):448, 2006.
- [13] A. Vinayagam, R. Konig, J. Moormann, F. Schubert, R. Eils, K. Glatting, and S. Suhai. Applying support vector machine for gene ontology based gene function prediction. *BMC Bioinformatics*, 5:116, 2004.