GOstruct 2.0: Automated Protein Function Prediction for Annotated Proteins

Indika Kahanda
Montana State University
357 Barnard Hall, Montana State University
Bozeman, MT 59717
indika.kahanda@montana.edu

Asa Ben-Hur
Colorado State University
Computer Science Department, 1873 Campus Delivery
Fort Collins, CO 80523-1873
asa@cs.colostate.edu

ABSTRACT

Automated Protein Function Prediction is the task of automatically predicting functional annotations for a protein based on gold-standard annotations derived from experimental assays. These experiment-based annotations accumulate over time: proteins without annotations get annotated, and new functions of already annotated proteins are discovered. Therefore, function prediction can be considered a combination of two sub-tasks: making predictions on annotated proteins and making predictions on previously unannotated proteins. In previous work, we analyzed the performance of several protein function prediction methods in these two scenarios. Our results showed that GOstruct, which is based on the structured output framework, had lower accuracy in the task of predicting annotations for proteins with existing annotations, while its performance on un-annotated proteins was similar to the performance in cross-validation. In this work, we present GOstruct 2.0 which includes improvements that allow the model to make use of information of a protein's current annotations to better handle the task of predicting novel annotations for previously annotated proteins. This is highly important for model organisms where most proteins have some level of annotations. Experimental results on human data show that GOstruct 2.0 outperforms the original GOstruct in this task, demonstrating the effectiveness of the proposed improvements. This is the first study that focuses on adapting the structured output framework for applications in which labels are incomplete by nature.

CCS CONCEPTS

•Theory of computation →Structured prediction; •Computing methodologies →Supervised learning by classification; Structured outputs; Support vector machines; *Cross-validation;* **•Applied computing →Computational proteomics; Bioinformatics;**

KEYWORDS

Automated Protein Function Prediction, Hierarchical Multilabel Classification, Structured Support Vector Machines, incomplete labels.

1 INTRODUCTION

Identifying the functions of the multitude of proteins encoded by a genome is a key problem in bioinformatics [15, 27]. The Gene Ontology (GO) [1] has emerged as the standard for describing protein function. It is a structured vocabulary with thousands of terms which describes different aspects of protein function using a hierarchy of keywords. It is composed of three independent subontologies for annotating the molecular functions of proteins, the biological processes they participate in, and the cellular components in which these occur. Examples of GO categories in these subontologies include RNA binding (molecular function), chromosome segregation (biological process) and nucleus (cellular component).

Despite the difficulty of experimentally determining function, scientists have been able to annotate a large number of proteins with GO categories through various wet-lab experiments. Currently there are more than 100,000 GO annotations for a large number of proteins across many species. But as the cost of sequencing continues to decline rapidly [24] there is an increasingly larger gap between sequences with and without annotations; therefore it is not realistic to use experimental determination as the sole means of annotation, and computational methods, enabled by the existing data, have emerged as a viable alternative.

In recent years, the bioinformatics community has developed many computational methods for function prediction [27]. Many of the earliest methods performed "transfer of annotation" using amino acid sequence similarity to proteins with known functions [9]. Over the years, discriminative algorithms such as SVMs [12, 25] and decision trees [13, 34], as well as probabilistic models which perform label propagation on graphs [5, 21, 22] have been developed. More recently, structured output methods have demonstrated high accuracy in GO term prediction [33], enzyme function prediction [2], and phenotype ontology term prediction [17]. The success in computationally determining the functions of proteins using a variety of data sources — protein sequence or structure, and various biological network data [8, 27, 33, 36] — has led to the establishment of automated function prediction (AFP) as one of the most important bioinformatics challenges in the last decade. However, according to a community based competition for evaluating automated function prediction methods (CAFA, see below), there is still significant room for improvement in the accuracy of existing methods [27].

Due to the emergence of a multitude of computational methods for GO term prediction, the community has realized the need for a systematic and organized means of comparing the performance of these methods in order to assess how far the area has progressed.

ACM-BCB'17, August 20-23, 2017, Boston, MA, USA.
© 2017 Copyright held by the owner/author(s). 978-1-4503-4722-8/17/08…$15.00
DOI: http://dx.doi.org/10.1145/3107411.3107417

Taking note from critical assessment efforts such as CASP (Critical Assessment of protein Structure Prediction) [23] and CAPRI (Critical Assessment of Prediction of Interactions) [14], the AFP community decided to hold its own competition: CAFA (Critical Assessment of Function Annotation) [27]. The main objective of CAFA is to gather all AFP researchers in one place to fairly assess and compare the latest computational methods using a centralized and independent assessment. In the first CAFA (CAFA1) the participants were provided with a list of protein targets that didn't have any previous GO annotations and were asked to submit computational predictions using their own AFP methods [27]. Once the predictions were submitted, the organizers collected the experimentally validated GO annotations acquired for those target proteins over a period of six months. Finally, the computational predictions were compared against those annotations to compute the accuracy of each AFP method.

The recent CAFA2 challenge had exactly the same setup, except that the list of 100,000 target proteins consisted of both annotated and unannotated proteins [15]. The added requirement of making predictions on currently annotated proteins makes CAFA2 a more realistic representation of the function prediction problem, as it better models the accumulation of annotations over time.

The AFP problem posed in CAFA is more challenging than the typical machine learning problem, as the usual assumption in machine learning is that the distribution of examples in the training set is reflective of that in the test set. In the CAFA AFP problem this assumption likely does not hold because the training is performed on an older set of annotations while testing is performed on newer annotations; and it is known that distribution of GO categories changes over time due to biases in the annotation process [29]. Furthermore, the annotations acquired for annotated proteins and the annotations acquired for unannotated proteins can be expected to be different in frequency and specificity: an annotated protein can be expected to acquire more specific GO categories than an unannotated protein, and perhaps more of them, as the biology community tends to study proteins that are already characterized.

In previous work we identified the CAFA2 requirements as a combination of two subtasks: making predictions on annotated proteins and making predictions on unannotated proteins [18]. In the task of making predictions on annotated proteins, methods are trained using the set of annotations acquired on or before a specific time-stamp t_1, and tested on the set of annotations gathered on the same set of proteins after t_1. In other words, the same set of proteins are used for training and testing, but the training labels are annotations that were available at t_1, while testing labels are annotations made available after t_1. In the task of making predictions on unannotated proteins, methods are trained using the set of annotations acquired on or before t_1 while they are tested on the annotations acquired for proteins that were not annotated on or before t_1. In this setup, the proteins used for training and the proteins used for testing are disjoint sets. This distinction is illustrated in Figure 1.

It turns out that the distinction between these two AFP tasks is an important one. In recent work we have compared the performance of GOstruct, binary support vector machines and guilt-by-association methods on these two tasks with their performance in cross-validation, which is typically used to assess and compare AFP methods. We observed that making predictions for already annotated proteins is challenging for all three methods compared to the task of making predictions on unannotated proteins [18]. This observation can be understood given that none of the methods take into account the incompleteness of existing annotations in their learning and inference procedures.

When it comes to the task of making predictions for annotated proteins, several recent methods use only existing annotations for making predictions. In other words, given the labels associated with a protein, these methods predict additional labels based only on its existing annotations. These methods are based on singular value decomposition [7], autoencoder neural networks [6], probabilistic latent semantic analysis [20] and Latent Dirichlet Allocation [26]. However, all these methods ignore the wealth of information available in all other types of genomic data sources such as sequence and protein-protein interactions. Therefore, it is worth exploring how both existing annotations and other data can be used together for this task; in methods like GOstruct and label reconciliation methods such as described in Guan et al. [12], information on existing annotations can be encoded in the learning or inference procedure itself. In this work, we propose novel improvements to the GOstruct method by taking into consideration that existing annotations are incomplete. We observe that GOstruct 2.0, the new version of GOstruct, has improved performance compared to original in the task of making predictions for annotated proteins, suggesting that better modeling of the problem leads to better performance. To the best of our knowledge, this is the first study that focuses on modeling the incompleteness of the labels in a structured output framework, and may be of use in other applications as well.

2 METHODOLOGY

2.1 Approach

The problem of GO term prediction is a hierarchical multilabel classification problem (HMC) [4], as a given protein can be annotated with multiple labels, and the set of labels have a hierarchy associated with them. Whereas the standard approach is to use multiple classifiers, one for each GO category, GOstruct takes the approach of using a single classifier that learns a direct mapping from inputs to the space of hierarchically consistent labels; this is achieved using structured prediction, which is a framework for learning a mapping from inputs to a label space that has a structure associated with it [35]. This framework can capture information from the inter-relationships between labels and allows the prediction of a set of labels that are hierarchically consistent. It eliminates the need for multiple classifiers, and the need for establishing hierarchical consistency between the predictions.

The key component of GOstruct's structured SVM formulation is the compatibility function, which computes the compatibility between a given protein and a label (i.e., a set of GO annotations). GOstruct learns this compatability by maximizing the margin between the correct label and all incorrect labels [33]. However, while it does allow for the possibility of mis-annotations, it does not explicitly take into account that the existing annotations of a protein are partial, i.e., are only a subset of its true annotations.

Our results from the previous work [18] demonstrate that the AFP task of making predictions on annotated proteins is much

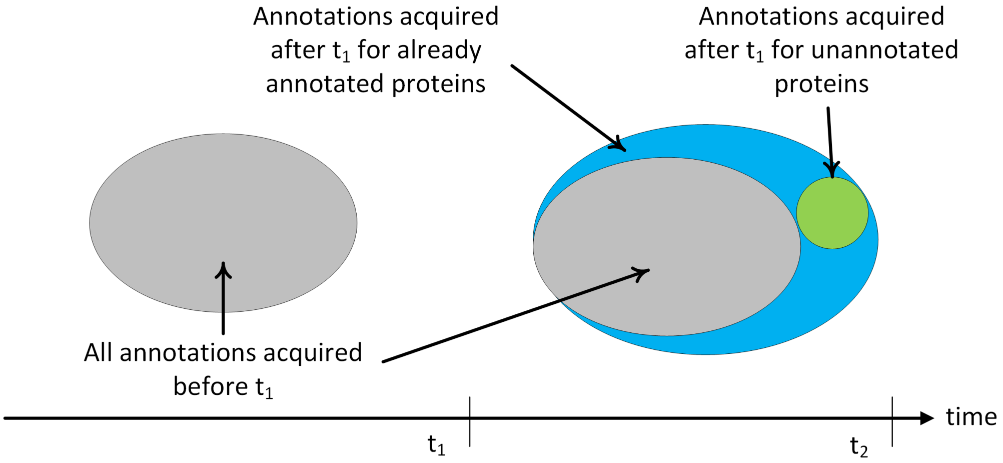
Annotations acquired after t_1 for already annotated proteins
Annotations acquired after t_1 for unannotated proteins
All annotations acquired before t_1
time
t_1
t_2

Figure 1: Overview of the two AFP subtasks. We distinguish between three sets of annotations that are used to define the train/test set with respect to the two tasks: making prediction for annotated proteins and making prediction for unannotated proteins. Annotations accumulate until time t_1 and form the set in grey, which is the training set for the task of making prediction for annotated proteins. The set of annotations acquired for those proteins after t_1 form the set in blue, which is the test set in the task of making prediction for annotated proteins. The set of annotations acquired after t_1 for proteins that were un-annotated before t_1 is denoted by the set in green, and is used as the test set in that task.

more difficult than performing well on un-annotated proteins or in cross-validation. This suggested that the task of predicting annotations for already annotated proteins could benefit from algorithms that explicitly leverage existing annotations to better rank novel predicted annotations.

GOstruct 2.0 is a modification of the original structured SVM formulation that reduces the penalty for margin violations involving examples for which the second best candidate label is an extension of its actual label. This allows the model to be more flexible with candidate labels that are extensions of the actual labels. This models the process of annotation whereby a protein might accumulate annotations with increasing level of specificity.

2.2 Models

In this section we present the proposed structured SVM formulation of GOstruct 2.0. Let $\mathcal{X}$ be the input space where proteins are represented, and let $\mathcal{Y}$ be the space of labels (GO categories). The training set consists of labeled training examples $\{(x_i, y_i)\}_{i=1}^{n}$ where $x_i \in \mathcal{X}$ and $y_i \in \mathcal{Y}$. The set of GO categories annotated to a given protein is collectively referred to as its (structured) label. $\mathcal{Y}$ represents each GO subontology in a vector space where component j represents category j, and a label vector $y \in \{0, 1\}^d$, where d is the number of GO categories has $y_j = 1$ if the corresponding protein is associated with category j, and 0 otherwise. In constructing the label vectors we assume the so-called *true-path rule*: whenever a protein is annotated with a given category, it is also associated with all its ancestors in the hierarchy. This is also known as *hierarchical consistency*. The *compatibility* function $f : \mathcal{X} \times \mathcal{Y} \mapsto \mathbb{R}$ maps input-output pairs to a score that indicates the strength of the association of an input to a set of GO categories. This function is expressed as a linear function in a feature space representing the labels and inputs, i.e. $f(x, y) = w^T \phi(x, y)$, where $\phi(x, y)$ is the joint input-output feature map. The predicted label $\hat{y}$ for an input x can then be obtained using the argmax operator as $\hat{y} = \arg\max_{y \in \mathcal{Y}_c} f(x, y)$ where $\mathcal{Y}_c \subset \mathcal{Y}$ is the set of all candidate labels. GOstruct uses the combinations of all GO categories present in the training set as the set of candidate labels $\mathcal{Y}_c$.

In order to obtain correct classification, the compatibility value of the true label (correct set of GO annotations) of an input protein needs to be higher than that of any other candidate label. This is captured by the following large-margin formulation [35]:

$$\min_{w,\epsilon} \frac{1}{2} ||w||_2^2 + \frac{C}{n} \sum_{i=1}^{n} \epsilon_i$$

subject to :

$$f(x_i, y_i) - \max_{y \in \mathcal{Y}_c} f(x_i, y) \geq 1 - \epsilon_i \qquad i = 1, \ldots, n \tag{1}$$

$$\epsilon_i \geq 0 \qquad i = 1, \ldots, n, \tag{2}$$

where w is the weight vector, C is a user-specified soft-margin constant, $\mathcal{Y}_c$ is the set of candidate labels, ϵ_i are the slack variables which allow margin violations, and $|| \cdot ||_2$ is the L^2 norm. The first constraint, Equation (1), ensures that the compatibility score for the actual label of a protein is higher than all other candidate labels, and the use of slack variables allow flexibility in satisfying this constraint. This optimization problem is solved in the dual with a kernel function that is a product of a linear input space kernel and a linear output space kernel as described elsewhere [33].

In this work we propose a modification to the above model in order to help GOstruct better handle the task of making predictions for annotated proteins. For that purpose, we define a label y' as an *extension* of a label y if whenever $y_i = 1$ then $y'_i = 1$. In other words, y' is consistent with y, and might include additional and more specific terms than y. Figure 2 illustrates this concept.

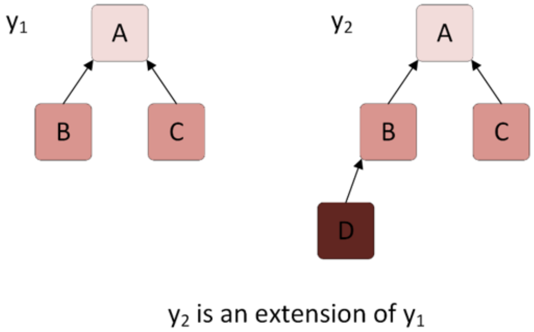

Figure 2: Label extensions: the label y_2 in this toy example is an extension of y_1 because y_2 includes all terms represented by y_1.

The original GOstruct formulation does not take into account that the existing labels represent a snapshot of our knowledge of protein function, which is incomplete [3]. For example, in our dataset, the fraction of human proteins that had new molecular function, biological process and cellular component annotations between January 2009 and December 2013 are approximately 14%, 35% and 28%, respectively. In order to model the fact that an existing label can be incomplete, any second best candidate label that is an extension of the existing annotations should be penalized to a lesser extent. This can be achieved by using a different (and lower) soft-margin constant for the subset of examples for which the second best candidate label is a more specific label. To do this, we partition the examples into two non-overlapping sets based on whether their corresponding second best candidate label are extensions of the known annotations and associate different soft-margin constants with them. We denote these two sets by A^+ and A^-, and their associated soft-margin constants by C^+ and C^-, respectively; A^+ is the set of examples for which the second best candidate label is an extension of its known label, A^- are the rest of the examples. We propose the following modification of the original structured SVM formulation as follows:

$$\min_{w,\epsilon^1,\epsilon^2} \frac{1}{2}||w||_2^2 + \frac{C^+}{n}\sum_{i\in A^+}\epsilon_i^1 + \frac{C^-}{n}\sum_{i\in A^-}\epsilon_i^2$$

subject to:

$$f(x_i, y_i) - \max_{y\in\mathcal{Y}_c} f(x_i, y) \geq 1 - \epsilon_i^1 \qquad \text{for } i \in A^+ \tag{3}$$

$$f(x_i, y_i) - \max_{y\in\mathcal{Y}_c} f(x_i, y) \geq 1 - \epsilon_i^2 \qquad \text{for } i \in A^- \tag{4}$$

$$\epsilon_i^1 \geq 0 \qquad \text{for } i \in A^+ \tag{5}$$

$$\epsilon_i^2 \geq 0 \qquad \text{for } i \in A^-, \tag{6}$$

where ϵ_i^1 and ϵ_i^2 are the slack variables, and $\mathcal{Y}_c$ is the candidate label set. The soft-margin constants C^+ and C^- are hyperparameters specified by the user and should satisfy $C^- \geq C^+$ for a lower penalty on annotations that are more specific than the known ones. The new constraints (inequalities 3 and 4) allow the model to be more flexible towards picking candidate labels that are extensions of the current label. This allows it to use the information of the existing annotations to model the accumulation of annotations over time.

2.3 Experimental setup

We compare the performance of GOstruct 2.0 against the original GOstruct (denoted as 1.0) with respect to the three tasks (a) cross-validation (CV), (b) making predictions on annotated proteins (denoted as NA) and (c) making predictions on unannotated proteins (denoted as NP) on human proteins. Each method was trained/tested using the same set of features and labels.

We extracted GO annotations from the Gene Ontology and UniProt databases on 12/01/2013. We removed all annotations not originating from an experimental assay, and ignored GO categories that were annotated with less than 10 proteins. The number of proteins/annotations in the train/test sets with respect to the three setups are given in Table 1.

We generated three types of sequence features (localization signals, low complexity regions and transmembrane data). BLAST scores were represented using a simpler version of a score used in [8]. We extracted PPI and other functional association data (co-expression, co-occurrence, etc.) from BioGRID 3.2.106, STRING 9.1 and GeneMANIA 3.1.2. To take advantage of the information found in the biomedical literature, an NLP pipeline was utilized to extract the co-occurrence of protein names and GO terms both at the sentence and paragraph level from all full-text publications available in PubMed; the datasets are publicly available [16], and additional details are provided elsewhere [18].

We use 5-fold cross validation for evaluating the performance in the CV task. Here, the folds are generated by partitioning the complete set of annotations randomly without regard to their time-stamps (i.e. all existing annotations of a protein are used as the gold standard). In the NA task, methods are trained using the set of annotations acquired on or before the year 2009, and tested on the set of annotations gathered on the same set of proteins after 2009. In the NP task, methods are trained using the set of annotations acquired on or before the year 2009 while they are tested on the annotations acquired for proteins that were not annotated on or before 2009. Note that the NA and NP tasks share the same training data; for example, in the molecular function subontology, there are 4, 305 proteins that had at least one annotation before 2009 (see Table 1) In our experiments we used $C^+ = 1$ and $C^- = 10$ for all our experiments; the value of C^- is the default value of the GOstruct soft-margin parameter on the basis of experiments in other datasets; we have not optimized the value of C^+.

We use *term-centric* AUC [27] as our primary evaluation measure for reporting results. In order to perform a fair comparison across setups we first identified the GO subgraph that consists of only the GO categories common to all three setups (CV, NA and NP). Then we computed the AUC only on this subgraph.

3 RESULTS AND DISCUSSION

Figure 3 and Table 2 show CV, NA, and NP results in human for all three GO namespaces: molecular function, biological process and

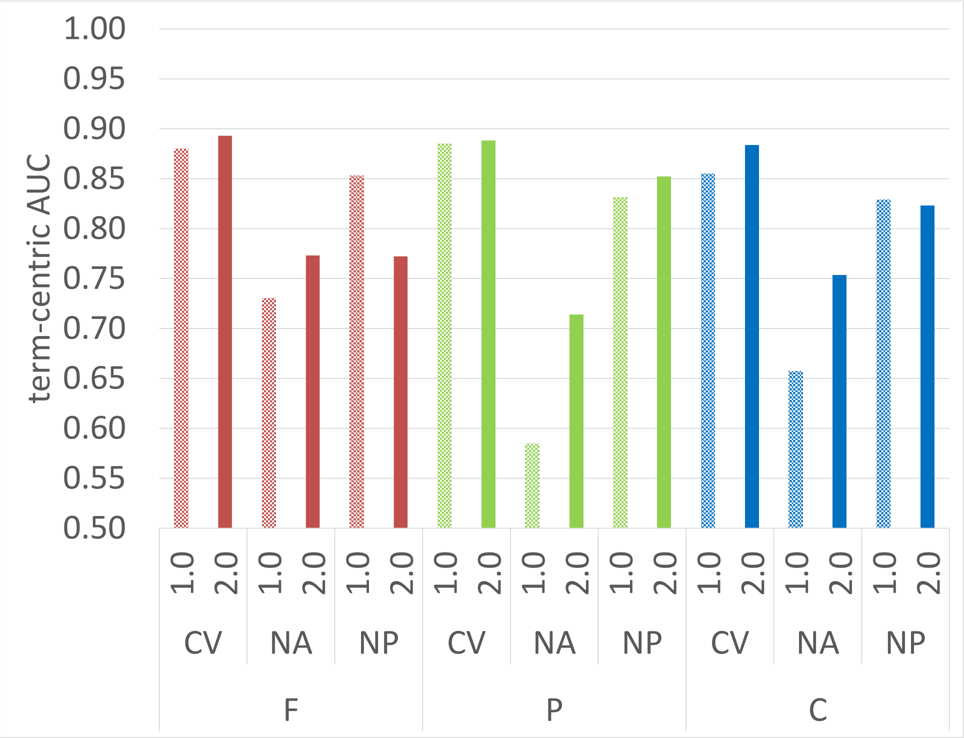

Figure 3: Performance comparison between GOstruct 2.0 and the original GOstruct in CV, NA and NP for human. GOstruct 2.0 (solid fill) and the original GOstruct (denoted as 1.0 and shown in patterned fill) are evaluated in CV (cross-validation), NA (novel-annotation) and NP (novel-proteins) on human. Performance is reported using the term-centric AUC in the molecular function (F), biological process (P) and cellular component (C) subontologies (depicted using red, green, and blue bars, respectively). Higher values indicate better performance.

Table 1: The number of proteins and the number of annotations in the training and test sets for the three setups: CV (cross-validation), NA (novel-annotations) and NP (novel-proteins) in human. Numbers are separately provided for the three subontologies: molecular function (F), biological process (P) and cellular component (C) subontologies. For the CV setup, numbers represent average values computed across train/test folds (5-fold cross-validation).

		Training set		Test set	
Ontology	Setup	Proteins	Annots.	Proteins	Annots.
F	CV	4532	8467	1133	2116
	NA	4305	6898	799	1343
	NP	4305	6898	1344	2174
P	CV	7533	31794	1883	7948
	NA	5824	12196	3301	13192
	NP	5824	12196	3574	12973
C	CV	8440	19196	2110	4799
	NA	5082	8185	2966	5511
	NP	5082	8185	5468	10200

cellular component. First, as noted in our previous work [18], performance for the NA task is much lower. We observe that GOstruct 2.0 outperforms GOstruct 1.0 in the NA task in all subontologies. We also note that it performs as well as or better than GOstruct 1.0 in the molecular function and biological process subontologies in yeast (data not shown). This demonstrates the ability of the new constraints to model the accumulation of annotations over time for improving performance in the NA task.

The largest improvement is in the biological process subontology (a jump from 0.58 to 0.71, see Figure 3). This observation can be attributed to the fact that it is by far the deepest subontology [11] as well as it being the subontology with the highest rate of increase in the number of categories over the years [10]. This makes biological process annotations more incomplete than the other two, thereby providing for more room for improvement. It is also important to note that GOstruct 2.0 produced AUCs above 0.7 for all three subontologies, which is a major improvement over the original GOstruct, which had low values of 0.58 and 0.66 in NA for the BP and CC subontologies, respectively. On unannotated proteins the results are mixed. This not surprising, since the 2.0 constraints are specifically designed for the NA scenario; however, it is always possible to apply the original version for proteins without annotations. In fact,

Table 2: Performance comparison between GOstruct 2.0 and the original GOstruct in CV, NA and NP for human. GOstruct 2.0 and original GOstruct (denoted as 1.0) are evaluated in CV (cross-validation), NA (novel-annotation) and NP (novel-proteins). Performance is reported using the term-centric AUC and the corresponding p-values computed using paired t-tests in the molecular function (F), biological process (P) and cellular component (C) subontologies. P-values < 0.05 are in bold.

Ontology	Setup	1.0	2.0	P-value
F	CV	0.88	0.89	**9.90E-03**
	NA	0.73	0.77	**6.58E-07**
	NP	0.85	0.77	**2.89E-12**
P	CV	0.89	0.89	3.06E-01
	NA	0.58	0.71	**8.10E-200**
	NP	0.83	0.85	**1.54E-11**
C	CV	0.85	0.88	**1.15E-04**
	NA	0.66	0.75	**5.54E-21**
	NP	0.83	0.82	5.74E-01

based on these results, the latest release of the GOstruct library[1] defaults to the 1.0 behavior for the NP task and applies 2.0 behavior for the NA task automatically based on the input annotations.

4 CONCLUSIONS AND FUTURE WORK

In this work we developed GOstruct 2.0, which includes enhancements to the underlying structured SVM formulation of the original GOstruct, allowing it to model the incompleteness of GO annotations. Using human data we showed that GOstruct 2.0 outperforms the original GOstruct, especially for the task of making predictions on annotated proteins. To the best of our knowledge, this is the first study that incorporates explicit modeling of label incompleteness in the context of the structured output framework, and demonstrated the importance of modeling label incompleteness. This idea can be applied in other settings. Several large-scale biological ontologies such as the Human Phenotype Ontology (HPO) [28], the Mammalian Phenotype Ontology (MPO) [31, 32] and the Disease Ontology (DO) [19, 30], are manually curated, and are also known to be incomplete due to the lag between the large volume of information in the published literature and the time-consuming process of manual curation. Therefore, the methodology developed here is directly applicable for the tasks of predicting novel annotations in these ontologies.

Several issues related to this work remain to be solved. An extensive model selection procedure for selecting the optimal soft-margin parameters with respect to the newly introduced constraints could further improve performance. This work only looked at modifying the learning procedure of the original GOstruct to account for incompleteness of labels. Another aspect worth exploring is introducing modifications to the inference rule used in the original GOstruct. GOstruct uses all combinations of labels present in the training set as the candidate label set when making predictions. However, for the task of making predictions on annotated proteins it may be beneficial to restrict the candidate label set to a subset of labels that are consistent to the current label, and further investigation is required to explore how to best incorporate this idea into the framework.

ACKNOWLEDGMENTS

The authors would like to thank Dr. Karin Verspoor and Dr. Christopher Funk for providing the literature features used with GOstruct method. This work was supported by the NSF Advances in Biological Informatics program through grant number 0965768.

REFERENCES

[1] Michael Ashburner, Catherine A. Ball, Judith A. Blake, David Botstein, Heather Butler, J. Michael Cherry, Allan P. Davis, Kara Dolinski, Selina S. Dwight, Janan T. Eppig, Midori A. Harris, David P. Hill, Laurie Issel-Tarver, Andrew Kasarskis, Suzanna Lewis, John C. Matese, Joel E. Richardson, Martin Ringwald, Gerald M. Rubin, and Gavin Sherlock. 2000. Gene Ontology: tool for the unification of biology. *Nature Genetics* 25, 1 (May 2000), 25–29. DOI: http://dx.doi.org/10.1038/75556

[2] Katja Astikainen, Liisa Holm, Esa Pitkänen, Sandor Szedmak, and Juho Rousu. 2008. Towards structured output prediction of enzyme function. In *BMC proceedings*, Vol. 2. BioMed Central, S2.

[3] William A. Baumgartner, K. Bretonnel Cohen, Lynne M. Fox, George Acquaah-Mensah, and Lawrence Hunter. 2007. Manual curation is not sufficient for annotation of genomic databases. *Bioinformatics* 23, 13 (2007), i41–i48. DOI: http://dx.doi.org/10.1093/bioinformatics/btm229

[4] Wei Bi and James T. Kwok. 2011. Multi-Label Classification on Tree- and DAG-Structured Hierarchies. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, Lise Getoor and Tobias Scheffer (Eds.). ACM, New York, NY, USA, 17–24.

[5] Yu Chen and Dong Xu. 2004. Global protein function annotation through mining genome-scale data in yeast Saccharomyces cerevisiae. *Nucleic acids research* 32, 21 (2004), 6414–6424.

[6] Davide Chicco, Peter Sadowski, and Pierre Baldi. 2014. Deep Autoencoder Neural Networks for Gene Ontology Annotation Predictions. In *Proceedings of the 5th ACM Conference on Bioinformatics, Computational Biology, and Health Informatics (BCB '14)*. ACM, New York, NY, USA, 533–540. DOI: http://dx.doi.org/10.1145/2649387.2649442

[7] Davide Chicco, Marco Tagliasacchi, and Marco Masseroli. 2012. Genomic Annotation Prediction Based on Integrated Information. In *Computational Intelligence Methods for Bioinformatics and Biostatistics*, Elia Biganzoli, Alfredo Vellido, Federico Ambrogi, and Roberto Tagliaferri (Eds.). Lecture Notes in Computer Science, Vol. 7548. Springer Berlin Heidelberg, 238–252. DOI: http://dx.doi.org/10.1007/978-3-642-35686-5_20

[8] Wyatt T. Clark and Predrag Radivojac. 2011. Analysis of protein function and its prediction from amino acid sequence. *Proteins: Structure, Function, and Bioinformatics* 79, 7 (2011), 2086–2096. DOI: http://dx.doi.org/10.1002/prot.23029

[9] Ana Conesa, Stefan Götz, Juan Miguel García-Gómez, Javier Terol, Manuel Talón, and Montserrat Robles. 2005. Blast2GO: a universal tool for annotation, visualization and analysis in functional genomics research. *Bioinformatics* 21, 18 (2005), 3674–3676.

[10] The Gene Ontology Consortium. 2015. Gene Ontology Consortium: going forward. *Nucleic Acids Research* 43, D1 (2015), D1049–D1056. DOI: http://dx.doi.org/10.1093/nar/gku1179 arXiv:http://nar.oxfordjournals.org/content/43/D1/D1049.full.pdf+html

[11] Olivier Dameron, Charles Bettembourg, and Nolwenn Le Meur. 2013. Measuring the evolution of ontology complexity: the gene ontology case study. *PLoS One* 8, 10 (2013), e75993.

[12] Yuanfang Guan, Chad L Myers, David C Hess, Zafer Barutcuoglu, A Caudy, and O Troyanskaya. 2008. Predicting gene function in a hierarchical context with an ensemble of classifiers. *Genome biology* 9, Suppl 1 (2008), S3.

[13] Boris Hayete and Jadwiga R Bienkowska. 2005. GOtrees: predicting go associations from protein domain composition using decision trees.. In *Pacific Symposium on Biocomputing*, Vol. 10. World Scientific, 127–138.

[14] Jol Janin, Kim Henrick, John Moult, Lynn Ten Eyck, Michael J. E. Sternberg, Sandor Vajda, Ilya Vakser, and Shoshana J. Wodak. 2003. CAPRI: A Critical Assessment of PRedicted Interactions. *Proteins: Structure, Function, and Bioinformatics* 52, 1 (2003), 2–9. DOI: http://dx.doi.org/10.1002/prot.10381

[15] Yuxiang Jiang, Tal Ronnen Oron, Wyatt T Clark, Asma R Bankapur, Daniel DfiAndrea, Rosalba Lepore, Christopher S Funk, Indika Kahanda, Karin M Verspoor, Asa Ben-Hur, and others. 2016. An expanded evaluation of protein function

[1] http://sourceforge.net/projects/strut/

prediction methods shows an improvement in accuracy. *Genome biology* 17, 1 (2016), 184.

[16] Indika Kahanda, Chris Funk, Fahad Ullah, Karin Verspoor, and Asa Ben-Hur. 2015. Supporting data for "A close look at protein function prediction evaluation protocols". *GigaScience Database* (2015). DOI:http://dx.doi.org/10.1002/prot.23029

[17] Indika Kahanda, Christopher Funk, Karin Verspoor, and Asa Ben-Hur. 2015. PHENOstruct: Prediction of human phenotype ontology terms using heterogeneous data sources. *F1000Research* 4 (2015).

[18] Indika Kahanda, Christopher S. Funk, Fahad Ullah, Karin M. Verspoor, and Asa Ben-Hur. 2015. A close look at protein function prediction evaluation protocols. *GigaScience* 4, 1 (2015), 1–10. DOI:http://dx.doi.org/10.1186/s13742-015-0082-5

[19] Warren A Kibbe, Cesar Arze, Victor Felix, Elvira Mitraka, Evan Bolton, Gang Fu, Christopher J Mungall, Janos X Binder, James Malone, Drashtti Vasant, and others. 2014. Disease Ontology 2015 update: an expanded and updated database of human diseases for linking biomedical knowledge through disease data. *Nucleic acids research* (2014), gku1011.

[20] M. Masseroli, D. Chicco, and P. Pinoli. 2012. Probabilistic Latent Semantic Analysis for prediction of Gene Ontology annotations. In *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*. 1–8. DOI:http://dx.doi.org/10.1109/IJCNN.2012.6252767

[21] Jason McDermott, Roger Bumgarner, and Ram Samudrala. 2005. Functional annotation from predicted protein interaction networks. *Bioinformatics* 21, 15 (2005), 3217–3226.

[22] Sara Mostafavi, Debajyoti Ray, David Warde-Farley, Chris Grouios, Quaid Morris, and others. 2008. GeneMANIA: a real-time multiple association network integration algorithm for predicting gene function. *Genome Biology* 9, Suppl 1 (2008), S4.

[23] John Moult, Jan T. Pedersen, Richard Judson, and Krzysztof Fidelis. 1995. A large-scale experiment to assess protein structure prediction methods. *Proteins: Structure, Function, and Bioinformatics* 23, 3 (1995), ii–iv. DOI:http://dx.doi.org/10.1002/prot.340230303

[24] Paul Muir, Shantao Li, Shaoke Lou, Daifeng Wang, Daniel J. Spakowicz, Leonidas Salichos, Jing Zhang, George M. Weinstock, Farren Isaacs, Joel Rozowsky, and Mark Gerstein. 2016. The real cost of sequencing: scaling computation to keep pace with data generation. *Genome Biology* 17, 1 (2016), 1–9. DOI:http://dx.doi.org/10.1186/s13059-016-0917-0

[25] Guillaume Obozinski, Gert Lanckriet, Charles Grant, Michael I Jordan, and William Stafford Noble. 2008. Consistent probabilistic outputs for protein function prediction. *Genome Biology* 9, Suppl 1 (2008), S6.

[26] P. Pinoli, D. Chicco, and M. Masseroli. 2014. Latent Dirichlet Allocation based on Gibbs Sampling for gene function prediction. In *Proceedings of the IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology*. 1–8. DOI:http://dx.doi.org/10.1109/CIBCB.2014.6845514

[27] Predrag Radivojac, Wyatt T. Clark, Iddo Friedberg, and others. 2013. A large-scale evaluation of computational protein function prediction. *Nat Meth* 10, 3 (Mar 2013), 221–227. DOI:http://dx.doi.org/10.1038/nmeth.2340

[28] Peter N. Robinson. 2012. Deep phenotyping for precision medicine. *Human Mutation* 33, 5 (2012), 777–780. DOI:http://dx.doi.org/10.1002/humu.22080

[29] AM Schnoes, DC Ream, AW Thorman, PC Babbitt, and I Friedberg. 2012. Biases in the experimental annotations of protein function and their effect on our understanding of protein function space. *PLoS computational biology* 9, 5 (2012), e1003063–e1003063.

[30] Lynn Marie Schriml, Cesar Arze, Suvarna Nadendla, Yu-Wei Wayne Chang, Mark Mazaitis, Victor Felix, Gang Feng, and Warren Alden Kibbe. 2012. Disease Ontology: a backbone for disease semantic integration. *Nucleic acids research* 40, D1 (2012), D940–D946.

[31] Cynthia L Smith and Janan T Eppig. 2009. The mammalian phenotype ontology: enabling robust annotation and comparative analysis. *Wiley Interdisciplinary Reviews: Systems Biology and Medicine* 1, 3 (2009), 390–399.

[32] Cynthia L Smith, Carroll-Ann W Goldsmith, and Janan T Eppig. 2004. The Mammalian Phenotype Ontology as a tool for annotating, analyzing and comparing phenotypic information. *Genome biology* 6, 1 (2004), R7.

[33] Artem Sokolov and Asa Ben-Hur. 2010. Hierarchical classification of Gene Ontology Terms Using the GOstruct Method. *J. Bioinformatics and Computational Biology* 8, 2 (2010), 357–376.

[34] Weidong Tian, Lan V Zhang, Murat Tasan, Francis D Gibbons, Oliver D King, Julie Park, Zeba Wunderlich, J Michael Cherry, and Frederick P Roth. 2008. Combining guilt-by-association and guilt-by-profiling to predict Saccharomyces cerevisiae gene function. *Genome Biology* 9, Suppl 1 (2008), S7.

[35] Ioannis Tsochantaridis, Thorsten Joachims, Thomas Hofmann, and Yasemin Altun. 2005. Large Margin Methods for Structured and Interdependent Output Variables. *J. Mach. Learn. Res.* 6 (Dec. 2005), 1453–1484.

[36] David Warde-Farley, Sylva L. Donaldson, Ovi Comes, Khalid Zuberi, Rashad Badrawi, Pauline Chao, Max Franz, Chris Grouios, Farzana Kazi, Christian Tannus Lopes, Anson Maitland, Sara Mostafavi, Jason Montojo, Quentin Shao, George Wright, Gary D. Bader, and Quaid Morris. 2010. The GeneMANIA prediction server: biological network integration for gene prioritization and predicting gene function. *Nucleic Acids Research* 38, suppl 2 (2010), W214–W220. DOI:http://dx.doi.org/10.1093/nar/gkq537