Bioinformatics Advance Access published July 10, 2016

Bioinformatics, 2016, 1–9
doi: 10.1093/bioinformatics/btw366
Advance Access Publication Date: 17 June 2016
Original Paper

OXFORD

Data and text mining

NegGOA: negative GO annotations selection using ontology structure

Guangyuan Fu[1], Jun Wang[1], Bo Yang[2] and Guoxian Yu[1,2,*]

[1]College of Computer and Information Science, Southwest University, Chongqing 400715, China and [2]Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education, Jilin University, Changchun 130012, China

*To whom correspondence should be addressed.

Associate Editor: Jonathan Wren

Received on February 25, 2016; revised on May 7, 2016; accepted on June 1, 2016

Abstract

Motivation: Predicting the biological functions of proteins is one of the key challenges in the post-genomic era. Computational models have demonstrated the utility of applying machine learning methods to predict protein function. Most prediction methods explicitly require a set of *negative examples*—proteins that are known *not* carrying out a particular function. However, Gene Ontology (GO) almost always only provides the knowledge that proteins carry out a particular function, and functional annotations of proteins are incomplete. GO structurally organizes more than tens of thousands GO terms and a protein is annotated with several (or dozens) of these terms. For these reasons, the negative examples of a protein can greatly help distinguishing true positive examples of the protein from such a large candidate GO space.

Results: In this paper, we present a novel approach (called NegGOA) to select negative examples. Specifically, NegGOA takes advantage of the ontology structure, available annotations and potentiality of additional annotations of a protein to choose negative examples of the protein. We compare NegGOA with other negative examples selection algorithms and find that NegGOA produces much fewer false negatives than them. We incorporate the selected negative examples into an efficient function prediction model to predict the functions of proteins in Yeast, Human, Mouse and Fly. NegGOA also demonstrates improved accuracy than these comparing algorithms across various evaluation metrics. In addition, NegGOA is less suffered from incomplete annotations of proteins than these comparing methods.

Availability and Implementation: The Matlab and R codes are available at https://sites.google.com/site/guoxian85/neggoa.

Contact: gxyu@swu.edu.cn

Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

Gene Ontology Consortium (GOC), an international effort to provide controlled structured vocabularies for molecular biology that serve as terminologies, classifications and ontologies to further data integration, analysis and reasoning (Ashburner *et al.*, 2000), over the past decade, has been regularly updating the use of controlled vocabularies to describe GO terms, and also the structure relationship among terms. GOC and partners also provide GO annotations of gene products (i.e. proteins) to describe their biological functions from molecular to organism levels. Recent advances in high-throughput techniques produced various proteomic data, and also accumulated some GO annotations of these data. Yet, the pace of annotating the biological roles of these proteins by wet-lab experiments is far lagging behind the influx of biological data. On the other hand, the knowledge of functional roles of proteins can boost the pace of drug development, understanding the life process and

© The Author 2016. Published by Oxford University Press. All rights reserved. For Permissions, please e-mail: journals.permissions@oup.com

Downloaded from http://bioinformatics.oxfordjournals.org/ at University of California, Los Angeles on August 30, 2016

many other potential applications (Radivojac *et al.*, 2013; Valentini, 2014). Therefore, many groups have been applying machine learning-based methods to take advantage of the available wealth of biological data to automatically predict protein function and show the feasibility (Peña-Castillo *et al.*, 2008; Radivojac *et al.*, 2013).

Despite advances in computational function prediction, few groups take into account the imbalanced nature of protein function prediction and the sparsity of GO annotations (Valentini, 2014; Tao *et al.*, 2007). This is because GO mainly provides the association that a protein is annotated with a GO term, but rarely specifies which terms are not annotated to that protein. The unspecified association between a protein and a term should not be simply treated as a *negative example*—the protein is not annotated with that term. This association may not be experimentally confirmed or appended into GO database by GO curators, which arises due to lack of experimental resources, biologists' research interests (Schnoes *et al.*, 2013). To date, there are more than 50 000 GO terms, each term corresponds to a distinct biological function, but a protein is often annotated with several or dozens of these terms. For these reasons, it is difficult to distinguish true positive examples of a protein from such a large number of candidate terms.

Many function prediction algorithms ask for explicit positive and negative examples to achieve discriminative predictions. However, GO mainly provides the positive examples and rarely records the negative examples of a protein. Some approaches directly utilize available positive annotations and PPI networks (or other data sources, i.e. amino acid sequences and domain families) to rank relevant terms of proteins and predict the most relevant terms as positive annotations (Re *et al.*, 2012; Yu *et al.*, 2013). Some positive-unlabeled learning algorithms (Elkan and Noto, 2008), taking advantage of positive and unlabeled data, are introduced into protein function prediction. For example, Zhao *et al.* (2008) took proteins annotated with the target term as labeled data, while proteins annotated with other terms instead of the target term as unlabeled data. This setting is motivated by the assumption that, proteins currently not annotated with the target term might be annotated with the term as more evidences accumulated. Next, they automatically selected negative examples from the unlabeled data in the learning process.

In this paper, we introduce an approach called NegGOA to select negative examples of proteins. NegGOA takes advantage of a hierarchical semantic similarity between GO terms to exploit the GO hierarchy. To account for the evolvement of annotations, it applies downward random walks with restart (Tong *et al.*, 2008) on the hierarchy, and on the empirical conditional probability that two terms co-annotated to a protein, to model the missing annotations. Next, it integrates the available annotations of a protein and potentiality of missing annotations to select negative examples of the protein. The empirical study on archived GOA files of Yeast, Human, Mouse and Fly genomes shows the proposed NegGOA makes much fewer false negative predictions than recently proposed and related approaches. In addition, by incorporating the negative examples selected by NegGOA (or by these related approaches) into SWSN (Youngs *et al.*, 2013), a Gaussian random field-based algorithm for label propagation with positive examples and selected negative examples, NegGOA also achieves more accurate prediction than these approaches across various evaluation metrics.

2 Related work

Recent approaches explicitly define negative examples, or employ heuristics to select negative examples at first, and then incorporate the selected negative examples to train discriminative classifiers for function prediction (Cesa-Bianchi *et al.*, 2012; Mostafavi and Morris, 2009). Guan *et al.* (2008) presumed all proteins not annotated with a given term as negative examples of that term. Mostafavi and Morris (2009) and Cesa-Bianchi *et al.* (2012) assumed proteins not annotated with sibling terms of the target term as negative examples of that term. This assumption sometimes does not hold, since a protein will be annotated with two (or even more) sibling terms as the cumulation knowledge of protein functions.

Youngs *et al.* (2013) suggested a parametrization Bayesian priors method (called ALBias) to choose negative examples. ALBias approximates the empirical conditional probability that a term annotated to a protein given the protein already annotated with another term, and selects the terms with the smallest probabilities as negative examples of the protein. Next, it incorporates these negative examples into SWSN (Youngs *et al.*, 2013), an improvement of the leading predictor GeneMANIA (Mostafavi *et al.*, 2008) in MouseFunc competition (Peña-Castillo *et al.*, 2008) that can integrate multiple functional association networks into a composite network, for network-based protein function prediction and shows improved accuracy by including the selected negative examples. Youngs *et al.* (2014) suggested another two negative examples selection algorithms, selection of negatives through observed bias (SNOB) and negative examples from topic likelihood (NETL). These two approaches outperform the positive-unlabeled data learning techniques (Zhao *et al.*, 2008) in choosing negative examples. GOC follows the convention to annotate proteins with as many terms as appropriate, as well as with the most specific terms available to reflect what are currently known about the proteins (Blake, 2013; Rhee *et al.*, 2008). A protein annotated with a term implies annotated with its ancestor terms via any path, which is recognized as *true path rule* in GO (Ashburner *et al.*, 2000; Valentini, 2011). ALBias only utilizes *direct* annotations in the GO annotations (GOA) file. SNOB takes advantage of all the annotations, including the appended ancestor annotations of these direct annotations, to approximate the empirical conditional probability between terms, and then to choose negative examples. NTEL takes a protein as a document and all annotations of the protein as words in that document, then it applies Latent Dirichlet Allocation topic model (Blei *et al.*, 2003) to choose negative examples.

Selecting negative examples based on currently available annotations of proteins is heavily biased. Functional annotations of proteins are incomplete and biased by biology research interests, because terms related to areas of scientific interests are expected to be more frequently annotated than other terms (Pesquita *et al.*, 2009; Schnoes *et al.*, 2013; Škunca *et al.*, 2012). The expanded annotations via true path rule on direct annotations and used by Youngs *et al.* (2014) still cannot completely characterize the functional roles of proteins. A large volume of annotations are missing, not yet confirmed by experiments, or not included into GO by GOC. Thomas *et al.* (2012), on behalf of GOC, recently reported that overlooking the incomplete annotations can produce inconsistent conclusions on assessing the similarity among ortholog genes and paralog genes. A protein may should be annotated with a more specific term, but due to lack of experimental support or other causes, the protein is currently annotated with ancestor terms of the term. As we know, a specific (or sparse) term is often annotated to fewer proteins than its ancestor terms, and sparse terms occupy majority of GO. These approaches (i.e. ALBias, SNOB) prefer to select sparse terms as negative examples of a protein, since they only take advantage of incomplete annotations to approximate the empirical

Downloaded from http://bioinformatics.oxfordjournals.org/ at University of California, Los Angeles on August 30, 2016

conditional probability between terms, without resorting to other resources, i.e. ontology structure.

In this paper, we mitigate these aforementioned problems by integrating the ontology structure, the potentiality of additional annotations of proteins and the available annotations of proteins to choose negative examples. Our proposed NegGOA is less affected by the incomplete annotations problem and can more accurately select negative examples than these related techniques. In addition, different from previous approaches that only consider terms currently annotated to proteins of interest, NegGOA takes into account all the terms in GO hierarchy and some of the terms currently not associated with any of these proteins may also be annotated to proteins as time goes by.

3 Methods

Let n be the number of proteins, $\mathcal{T}$ be the set of GO terms, $A \in \mathbb{R}^{n\times|\mathcal{T}|}$ be the available associations between n proteins and $|\mathcal{T}|$ terms. $A(i,t) = 1$ means the ith protein is annotated with $t \in \mathcal{T}$, and $A(i,t) = 0$ means it is unknown whether the protein should be annotated with t or not. Our target is to identify some negative examples from $A(i,t) = 0$, namely updating some $A(i,t) = 0$ as $A(i,t) = -1$, and to improve protein function prediction using these negative examples.

3.1 Hierarchical transitional probability

SNOB (Youngs *et al.*, 2014) and ALBias (Youngs *et al.*, 2013) use the empirical conditional probability of co-occurrence between two terms and available annotations of a protein to choose negative examples of the protein. It is recognized the functional annotations of proteins in most species are far from complete (Blake, 2013; Rhee *et al.*, 2008). Therefore, the empirical conditional probability can only partially reflect the probability of co-occurrence. Indeed, SNOB and NTEL take advantage of ontology hierarchy, by associating ancestor terms of direct annotations of a protein with the same protein, to get a more accurate estimation of the co-occurrence. Similar to Yu *et al.* (2015a), these two methods do not utilize ontology structure. Recent study shows that hierarchical structure among GO terms plays important role in predicting protein function (Wang *et al.*, 2015; Yu *et al.*, 2015b).

We advocate taking advantage of the ontology hierarchy and available annotations of proteins to select negative examples of a protein. To take advantage of the hierarchy, we first compute the semantic similarity between two terms based on a measurement similar to Lin's similarity (Lin, 1998) as follow:

$$\mathrm{sim}_H(t,s) = \frac{2\times \mathrm{IC}(\mathrm{LCA}(t,s))}{\mathrm{IC}(t)+\mathrm{IC}(s)}, \tag{1}$$

where LCA(t, s) means the lowest common ancestor of t and s in the hierarchy, and the ancestor terms of t and s include themselves. IC(t) is the information content of t, and it is defined as:

$$\mathrm{IC}(t) = \left(1 - \frac{\log_2(|\mathrm{desc}(t)|)}{\log_2(|\mathcal{T}|)}\right), \tag{2}$$

$|\mathcal{T}|$ is the number of terms in $\mathcal{T}$, desc(t) is the set including all the descendants of t and itself. Original Lin's similarity uses t's frequency in n proteins to define the information content, and it suffers from the shallow annotation problem and incomplete functional annotations of proteins (Pesquita *et al.*, 2009). In contrast, Eq. (2) employs ontology structure to measure the information content of t and it is irrelevant to functional annotations of proteins. Obviously, a leaf term in the hierarchy has larger information content than its ancestor terms, and this leaf term also describes more specific biological knowledge than its ancestor terms. Eq. (1) is also used by Tao *et al.* (2007) and Yu *et al.* (2015b) to measure the semantic similarity between two terms.

As time goes by, more annotations are appended to a protein. These appended annotations often correspond to descendants of the terms currently associated with the protein. To take advantage of this observation, we apply random walks with restart on the GO hierarchy and take the terms currently associated with a protein as initial walkers. Random walks on a graph is often described by the transitional probabilities between nodes of that graph (Tong *et al.*, 2008). Suppose $G \in \mathbb{R}^{|\mathcal{T}|\times|\mathcal{T}|}$ be the adjacent matrix of the GO direct acyclic graph (DAG). If s is a direct child of t, then $G(t,s) = 1$, otherwise $G(t,s) = 0$. $\mathrm{sim}_H(t,s)$ measures the semantic similarity between any two terms. Similar to Yu *et al.* (2015b), to simulate random walks on the DAG, we initialize the transitional probability matrix $W'_H \in \mathbb{R}^{|\mathcal{T}|\times|\mathcal{T}|}$ on G as follows:

$$W'_H(t,s) = \mathrm{sim}_H(t,s) \times G(t,s), \tag{3}$$

From Eq. (3), we can find that a random walker cannot jump from t to s in the first step if they do not have the parent–child relationship in the DAG. This setting is motivated by our previous study (Yu *et al.*, 2015a) that the likelihood of a missing association (or annotation) between a protein and a term is more accurately estimated by its parental terms than by its other ancestors. Next, we normalize the transitional probability between two terms with respect to other terms as $W_H(t,s) = W'_H(t,s)/\sum_{v\in\mathcal{T}} W'_H(t,v)$.

Suppose a random walker starting at node t and $W^k_H(t,v)$ be the transitional probability from t to $v \in \mathcal{T}$ after k steps. At the same time, the walker also has a probability of staying at t. Intuitively, $W^0_H(t,t) = 1$ and $W^0_H(t,v) = 0$ for any $v \neq t$. $W^{k+1}_H(t,v)$ is defined as follows:

$$W^{k+1}_H(t,v) = \alpha \sum_{s\in\mathrm{desc}(t)} W^k_H(t,s) W_H(s,v) + (1-\alpha) \tag{4}$$

where $\alpha \in [0,1]$ is a scalar parameter to control the restart probability of the random walker.

From Eq. (4), we can see a random walker performs random walks with restart along the direct edges of GO DAG. We want to remind that the walker will not reside on a leaf node of the graph, and the probability a walker moves to a leaf term is smaller than that to its ancestor terms. On the other hand, if $\alpha = 1$, namely the probability for a walker staying at t is zero, then the walker will end at one leaf node. For simplicity and avoiding bias, we just set $\alpha = 0.5$, which means a random walker having equal probability to stay at t and to move to t's descendant terms. The maximum steps from a root term to a leaf term in GO DAG (as of January 2016) is no more than 15. The probability for a walker staying at t is not zero, and the probability the walker jumps to t's direct child terms (if any) is larger than that to t's other descendant terms (if any). In other words, the larger the distance of $v(v \in \mathrm{desc}(t))$ to t, the smaller the transitional probability from t to v is. Given that, we set the number of iterations as 4 in our experimental study, and use $R_H(t,v) = W^4_H(t,v)$ to approximate the *hierarchical transitional probability* from t to v.

3.2 Expanded conditional probability

Eq. (4) only utilizes the ontological structure to measure the transitional probability between two terms, it is independent of any species and irrelevant to the available functional annotations of these n

Downloaded from http://bioinformatics.oxfordjournals.org/ at University of California, Los Angeles on August 30, 2016

proteins. To make use of these annotations of proteins, similar to Youngs *et al.* (2014), we compute the empirical conditional probability between t and s as follows:

$$p(s|t) = \frac{|\mathcal{A}_t \cap \mathcal{A}_s|}{|\mathcal{A}_t|} \tag{5}$$

where $\mathcal{A}_t$ is the set of proteins annotated with term t, $\cap$ is the set intersection operator. SNOB takes advantage of this probability to estimate negative examples of a protein given the protein is annotated with t but not with s. The smaller the conditional probability $p(s|t)$, the larger the likelihood s is chosen as a negative example of the protein.

Both SNOB and ALBias only account for the pairwise conditional probability between t and s. Suppose t and s are co-annotated to some proteins, s and v are co-annotated to some other proteins, but t and v currently are not co-annotated to any of the n proteins, then $p(v|t) = 0$. However, since functional annotations of proteins are incomplete, some proteins may be annotated with both t and v when more experiment evidences are available. ALBias uses parametrization Bayesian priors to alleviate the problem of incomplete annotations. However, similar to SNOB, it also prefers to select v as a negative example of a protein, given the protein is currently annotated with t but not with v. To overcome this limitation, we resort to random walks with restart again as follows:

$$W_C^{k+1}(t,v) = \alpha \sum W_C^k(t,s) W_C(s,v) + (1-\alpha) \tag{6}$$

where $W_C^k(t,v)$ is the transitional probability from t to v after k steps, $W_C(t,s) = p(s|t)/\sum_{v\in\mathcal{T}} p(v|t)$, $W_C^0(t,t) = 1$ and $W_C^0(t,s) = 0 (t \neq s)$.

Here, similar to $R_H(t,v)$, we use $R_C(t,v) = W_C^4(t,v)$ to represent the *expanded conditional probability* between t and v. In fact, Eq. (6) is also motivated by successful applications of label (or function) correlations on boosting the performance of multi-label learning and protein function prediction (Yu *et al.*, 2013; Zhang and Zhang, 2010). If v is missing for a protein and $p(v|t) = 0$, given the protein is already annotated with t, v is less likely being selected as a negative example of the protein by $R_C(t,v)$ than by $p(v|t)$ in Eq. (5).

3.3 Selecting negative examples

Ontology structure has demonstrated its ability to predict protein function and to replenish the missing annotations of incompletely annotated proteins (Cesa-Bianchi *et al.*, 2012; Tao *et al.*, 2007; Valentini, 2014; Yu *et al.*, 2015b). Inspired by these successful applications, we want to exploit the hierarchical transitional probability $R_H(t,v)$, expanded conditional probability $R_C(t,v)$ and available annotations of a protein to predict negative examples of the protein as follow:

$$L_H(i,v) = 1 - \frac{1}{|\mathcal{T}_i|} \sum_{t\in\mathcal{T}_i} R_H(t,v) \tag{7}$$

$$L_C(i,v) = 1 - \frac{1}{|\mathcal{T}_i|} \sum_{t\in\mathcal{T}_i} R_C(t,v) \tag{8}$$

where $\mathcal{T}_i$ is the set of terms currently annotated to the ith protein, including the terms appended by the true path rule. $L_H(i,v)$ is the predicted likelihood of term $v \notin \mathcal{T}_i$ as a negative example of the ith protein from $R_H(t,v)$, it is motivated by the observation that the newly appended annotations of the ith protein correspond to descendant terms of $t \in \mathcal{T}_i$. $L_C(i,v)$ is the predicted likelihood of negative example from $R_C(t,v)$, it is driven by the co-occurrence information of a protein annotated with v given the protein already annotated with t.

One advantage of $L_H(i,v)$ to $L_C(i,v)$ is that, if v currently is *not* annotated to any of the n proteins but it is a direct child (or descendant) of t and t is annotated to protein i, v can be less likely chosen as a negative example of the protein by $L_H(i,v)$ than by $L_C(i,v)$, since $R_H(t,v) > 0$ and $R_C(t,v) = 0$. In other words, $L_C(i,v)$ can only take into account terms currently annotated to these n proteins, whereas $L_H(i,v)$ considers all the terms in GO hierarchy, no matter whether they are currently annotated to these n proteins or not. This advantage suggests that $L_H(i,t)$ suffers less from incomplete and shallow annotations than $L_C(i,t)$. This advantage of $L_H(i,v)$ will be confirmed in the following experimental study.

To this end, we integrate the two predicted likelihoods $L_H(i,v)$ and $L_C(i,v)$ as follow:

$$L(i,v) = \beta L_H(i,v) + (1-\beta) L_C(i,v) \tag{9}$$

where $\beta \in [0,1]$ is a scalar parameter to adjust the contribution of $L_H(i,v)$ and $L_C(i,v)$. We then choose terms corresponding to the largest entries of $L(i,\cdot) \in \mathbb{R}^{|\mathcal{T}|}$ as negative examples of the ith protein.

Yang *et al.* (2012) applied downward random walks on GO hierarchy to improve the semantic similarity between pairwise terms and introduced a measure called integrated similarity measure (ISM) to synthesize the similarity from above and beneath of the terms (Caniza *et al.*, 2014). Particularly, ISM uses a host similarity measure [i.e. Lin's similarity (Lin, 1998)] to measure the similarity above the considered terms. To measure the similarity beneath the terms, it introduces an extra unknown child term for each non-leaf term to model the uncertainty comes from incomplete annotations and ontology structure, and it forces random walkers starting at non-leaf terms always moving downward to leaf terms. Similar to Eq. (8), ISM can also be adopted to select negative examples. However, ISM solely utilizes available annotations to define transitional probability between terms, and thus it only considers terms annotated to at least one protein. In contrast, Eqs. (4) and (6) neither introduce extra unknown terms, nor force random walkers always moving downward to leaf terms. NegGOA not only takes into account all the terms (no matter these terms are annotated to n proteins or not) in GO hierarchy, but also explicitly models the potentiality of missing annotations of proteins. Our following empirical study shows NegGOA can more accurately select negative examples than ISM.

4 Results and discussion

4.1 Datasets

We downloaded the recent GO file (http://geneontology.org/page/download-ontology) (archived date: December 7, 2015). The GO file includes the definition of GO terms and the structure relationship between them. GO organizes terms in three orthogonal branches of biological concepts, namely Biological Process (BP), Molecular Function (MF) and Cellular Component (CC). We downloaded the recent GOA files of Yeast, Human, Mouse and Fly (http://geneontology.org/page/download-annotations) (archived date: December 7, 2015). The GOA file provides GO annotations which associate gene products with GO terms. We excluded the terms labeled as 'obsolete' in the GO file and the annotations with evidence code 'IEA' (Inferred from Electronic Annotation) in the GOA files. We then annotated proteins of Yeast, Human, Mouse and Fly with the recent GOA files, respectively. Particularly, we

Downloaded from http://bioinformatics.oxfordjournals.org/ at University of California, Los Angeles on August 30, 2016

annotated all ancestor terms of direct annotations of a protein to the same protein.

There is no gold standard to evaluate the quality of negative examples selection, since negative examples are rarely available in GOA files. We use the number of False Negative predictions (FNs) as an evaluation criterion. To count the number of FNs, we also downloaded historical GOA files (archived date: May 3, 2013) of these four species from European Bioinformatics Institute (ftp://ftp.ebi.ac.uk/pub/databases/GO/goa/), and annotated these proteins in the same way. Next, NegGOA is employed to predict negative examples based on the annotations from a historical GOA file. If a predicted negative example of a protein is a positive example of the protein in the recent GOA file, then there is one FN. Note, only the proteins having annotations both in the historical GOA file and in the recent GOA file are used to count the number of FNs. Table 1 reveals the GO annotations of proteins and the number of GO terms that annotated to at least one protein. From the table, we can observe that many new annotations were appended to proteins in the past 2 years. In addition, a number of GO terms, which were not annotated to any of the studied proteins in the historical GOA file, now are annotated to some of these proteins in the recent GOA file.

4.2 Methods and evaluation metrics

We compared our approach NegGOA against SNOB (Youngs *et al.*, 2014), ALBias (Youngs *et al.*, 2013) and a baseline approach—Random. SNOB shows better performance than NETL, it also has demonstrated performing better than heuristic techniques (Mostafavi and Morris, 2009) and positive-unlabeled learning approaches (Zhao *et al.*, 2008). Therefore, we do not take those methods as comparing methods. In addition, we introduce two variants (ISM-H and ISM-L) of ISM as another two comparing methods. ISM-L utilizes original Lin's similarity (Lin, 1998) as the host similarity measure to measure the above similarity between pairwise terms, and ISM-H uses a hierarchical structure similarity defined in Eq. (1) as the host measure. Then ISM-H and ISM-L employ an equation similar as Eq. (8) to select negative examples.

For a given term v, ALBias estimates the bias of negative example as follow:

$$\text{bias}(i, v) = 2 * \frac{1}{|\mathcal{D}_i|} \sum_{t \in \mathcal{D}_i} q(v|t) - 1 \tag{10}$$

Table 1. Protein counts, number of direct annotations, number of terms annotated to at least one protein (in parenthesis), number of GO annotations in the historical and recent GOA files for each organism in each branch

	Proteins		Historical-direct	Recent–direct	Historical	Recent
Human		BP	64339(10748)	96221(12691)	635084	928423
	19598	CC	31413(1288)	48128(1599)	208198	302020
		MF	25528(3203)	39560(3742)	104464	156879
Yeast		BP	14065(4242)	16915(4820)	202980	242632
	6381	CC	12204(838)	14551(970)	99730	113807
		MF	9928(2051)	10276(2230)	42054	51268
Mouse		BP	56790(10603)	102281(14056)	529889	981110
	24221	CC	26217(1174)	59442(1659)	150560	315008
		MF	24307(2822)	48651(4011)	89052	178306
Fly		BP	23803(5219)	37487(6250)	256714	421934
	14825	CC	9121(825)	18346(1039)	62226	120241
		MF	7965(1699)	17353(2569)	35318	79227

where $\mathcal{D}_i$ is the *direct* GO annotations of the ith protein, and $q(v|t)$ is the empirical conditional probability approximated as follow:

$$q(v|t) = \frac{n_{tv}^{+}}{n_t^{+} + \lambda e^{\gamma n_t^{+}}} \tag{11}$$

where n_{tv}^{+} is the number of proteins among n proteins directly annotated with both t and v, n_t^{+} is the number of proteins annotated with t, λ and γ are scalar parameters to smooth conditional probability between two extreme assumptions about how missing and currently known annotations are distributed. SNOB approximates the bias of negative example using a non-parametric objective function (with $\lambda = 0$) in Eq. (11). It not only employs the direct annotations, but also the annotations appended by applying true path rule on these direct annotations, to approximate the empirical conditional probability. Random as a baseline algorithm, randomly selects a term $v \notin \mathcal{T}_i$ as a negative example of the i-protein. To reduce the random effect, we repeat the baseline algorithm 100 times for each fixed setting of experiment.

In the following experiments, if extra specified, k and α for NegGOA are fixed as 4 and 0.5, β in Eq. (9) is set to 0.5, respectively. As provided in the codes of ALBias, we set $\lambda = 32$ and $\gamma = 0.0125$ for ALBias. The codes of ALBias and SNOB are available at http://bonneaulab.bio.nyu.edu/software.html, and the codes of ISM are available at http://www.paccanarolab.org/static_content/gosim/.

Besides the number of FNs, we adopt another four evaluation metrics, *AvgAUC*, *Fmax*, *MacroF1* and *RankingLoss* to evaluate the quality of protein function prediction after incorporating the selected negative examples into SWSN. These metrics have been applied to evaluate the performance of function prediction (Radivojac *et al.*, 2013; Yu *et al.*, 2015b), and their formal definitions are provided in the Supplementary File. These metrics capture different aspects of a function prediction algorithm, it is difficult for an algorithm to outperform the others across all the evaluation metrics.

4.3 Performance of negative examples selection methods

In this section, NegGOA first estimates the likelihoods of negative examples for each protein based on annotations from the historical GOA files and results in a likelihood matrix $L \in \mathbb{R}^{n \times |\mathcal{T}|}$. Then, it selects the largest m entries in L as the corresponding selected negative examples. Next, recent GO annotations are used to count the number of FNs made by NegGOA. ISM-L, ISM-H, SNOB, ALBNeg and Random are also trained on the historical GO annotations to select negative examples, and follow the same process as NegGOA to count the number of FNs, respectively.

To further study the difference between NegGOA and these comparing methods, we count the number of FNs in two cases: (i) only the terms annotated to at least one protein in the *historical* GOA file are considered, this case is labeled as $h \geq 1$, and (ii) only the terms annotated to at least one protein in the *recent* GOA file are considered, this case is labeled as $r \geq 1$. Obviously, case $h \geq 1$ obeys the experimental protocol applied in SNOB and ALBias. Case $r \geq 1$ is more realistic and challenging, it involves more terms than case $h \geq 1$. Irrespective of case $h \geq 1$ and case $r \geq 1$, NegGOA utilizes all the terms in GO hierarchy to compute the transitional probability between terms. Youngs *et al.* (2014) combined the annotations in three branches to approximate the conditional probability [see Eq. (11)] for SNOB and ALBias, and then counted FNs in each branch. Since the terms in each GO branch form a DAG by

Downloaded from http://bioinformatics.oxfordjournals.org/ at University of California, Los Angeles on August 30, 2016

themselves, NegGOA computes the hierarchical transitional probability and expanded conditional probability between terms for each branch, and selects negative examples based on the annotations in that branch. The number of FNs with respect to different numbers ($m = 10k, 20k, \ldots, 80k$) of selected negative examples on Human genome is revealed in Table 2 for case $r \geq 1$. Other results on Human, Yeast, Mouse and Fly genomes are provided in Supplementary Tables S1–S7 of the supplementary file.

From these tables, we can draw a conclusion that NegGOA almost always makes much fewer FNs than other comparing methods, irrespective of m, case $h \geq 1$ and the more challenging case $r \geq 1$. Taking Human genome annotated with terms in BP branch for example, NegGOA produces 6 FNs, whereas ISM-H has 58 FNs, ISM-L has 73 FNs, SNOB has 29 FNs, ALBNeg has 54 FNs and Random results in 102.8 FNs for case $r \geq 1$ with $m = 80k$. In fact, SNOB only employs the conditional probability between terms estimated from available annotations [see Eq. (5)] to select negative examples. Although the annotations used by SNOB are already extended by true path rule on direct annotations, and these annotations encode the ontology structure information to some extent, SNOB does not concretely take into account the GO structure. In contrast, NegGOA not only takes advantage of co-occurrence information between terms, but also the hierarchical structure among all the terms of GO hierarchy. Given that, NegGOA produces fewer FNs than SNOB. This observation supports our motivation to use ontology structure for selecting negative examples.

ALBias, similar to SNOB, also employs the approximated conditional probability between terms to predict negative examples of a protein, it does not take into account hierarchical structure as well as SNOB, since it only uses direct annotations of proteins. Furthermore, from Table 1, we can see the number of direct annotations is much smaller than the number of annotations appended by true path rule on these direct annotations. The approximated conditional probability used by ALBias is less reliable than that of SNOB. Therefore, ALBias often produces more FNs than SNOB and NegGOA. ALBias utilizes two additional parameters λ and γ [see Eq. (11)] to account for incomplete annotations, so it sometimes makes fewer FNs than SNOB. This observation suggests the incomplete annotations of proteins should be considered in selecting negative examples. These results also imply the importance of GO hierarchy in selecting negative examples.

Table 2. Number of false negative predictions on *Human* genome under different numbers (m) of selected negative examples ($r \geq 1$)

	M	10k	20k	30k	40k	50k	60k	70k	80k
BP	NegGOA	5	6	6	6	6	6	6	6
	ISM-H	12	20	31	39	45	50	58	58
	ISM-L	7	41	62	62	64	70	71	73
	SNOB	0	4	19	21	22	22	27	29
	ALBias	18	50	50	50	51	54	54	54
	Random	12.8	26.4	38.3	51.9	63.2	75.2	89.7	102.8
CC	NegGOA	0	0	0	0	0	0	0	1
	ISM-H	0	8	10	14	29	31	32	33
	ISM-L	0	0	0	4	6	8	14	15
	SNOB	0	3	3	10	12	14	14	17
	ALBias	1	2	2	9	27	27	47	99
	Random	33.2	67.7	101.3	134.9	170.1	202.3	236.5	270.1
MF	NegGOA	0	0	0	0	0	1	4	5
	ISM-H	0	0	0	0	0	12	14	14
	ISM-L	0	0	1	1	1	2	2	2
	SNOB	0	3	3	3	4	4	4	4
	ALBias	0	0	0	1	1	1	7	12
	Random	7.7	15.4	22.4	30.7	38.6	46.5	53.9	61.6

ISM-H, ISM-L and NegGOA all apply downward random walks on GO hierarchy, but the first two methods produce more FNs than NegGOA. The cause is that ISM only considers the terms annotated to at least one protein. From Table 1, we can find a number of terms, which were not annotated to proteins in historical GOA files, are annotated to these proteins in recent GOA files. Another cause is that ISM uses an additional child term for each non-leaf term to account for uncertainty and it forces all non-leaf terms always moving downward to leaf terms, thus it does not model missing annotations of proteins as well as NegGOA. This observation again supports that missing GO annotations of proteins should be considered in predicting negative examples.

The baseline algorithm Random neither uses the empirical conditional probability between terms, nor the ontology structure, so it is often outperformed by other comparing methods. However, it occasionally makes fewer FNs than ALBias. The cause is that Random does not equally select each term as a candidate negative example of a protein. Instead, it is inclined to less frequent terms. These selected negative examples with respect to less frequent terms are often less likely to be validated in recent GOA files. In practice, both SNOB and ALBias prefer to select less frequent terms as negative examples of a protein. In contrast, NegGOA is not so dependent on the frequency of terms as these comparing methods, thus it suffers less from incomplete annotations of proteins and makes fewer FNs.

To further study the examples of false negative prediction, we list the examples of false negative predictions of NegGOA and SNOB on Human and Yeast genomes in Supplementary Tables S8–S11 of the supplementary file. In addition, we also study the performance of these comparing algorithms on Human and Yeast genomes by using all the available annotations, including the IEA annotations in the GOA files, and report the results in Supplementary Tables S12–S19 of the supplementary file. These results give the same conclusions as in the main text.

GO annotations of proteins are far from complete and comprehensive, it is not so rational to select negative examples of a protein solely based on the approximated conditional probability from currently available annotations. Due to Open-World assumption (Škunca *et al.*, 2012), a protein currently not annotated with a GO term does not means the protein not carrying out the biological function described by the term. Thus, the number of FNs can only partially reflect the ability of these comparing methods in correctly choosing negative examples. More (or less) FNs may be found as functional annotations of proteins becoming more complete. With the improvement and revolution of GO, ontology structure can be and should be employed as an important knowledge source for selecting negative examples. This comparative study corroborates that leveraging ontology structure and available annotations of proteins can more accurately select negative examples than using the available annotations alone.

4.4 Golden set evaluation in yeast: mitochondrial organization

To further explore the reliability of negative example selection methods, we carry out a test on a golden set of annotations of Yeast with respect to BP term 'GO:0007005' (Mitochondrial Organization). 'GO:0007005' is considered as an exhaustively verified function, so all positive and negative examples with respect to 'GO:0007005' are known across the entire Yeast genome (Huttenhower *et al.*, 2009). Here, we do not check the reliability on

Downloaded from http://bioinformatics.oxfordjournals.org/ at University of California, Los Angeles on August 30, 2016

archived GOA files of Yeast. Instead, we directly uses the recent GOA file. Particularly, we randomly select 60% proteins as training proteins and their associations with 'GO:0007005' are known. The associations between the left proteins (40%) and 'GO:0007005' are viewed unknown and to be predicted, these associations are only used to validate the predicted negative examples. Figure 1(a) reveals the Receiver–Operator Characteristic (ROC) curve for each of the comparing algorithms.

From Figure 1(a), we can find that NegGOA initially makes similar true positive rates as ALBias and SNOB, but it achieves larger true positive rates than all the comparing methods, including ISM-H and ISM-L later. In the end Overall, the AUC score for NegGOA is 0.9547, ISM-H is 0.9086, ISM-L is 0.8840, SNOB is 0.9166 and ALBias is 0.8365. Random gets an anticipated score of 0.5, which is much smaller than that of other comparing algorithms. This observation demonstrates the reliability of these methods in selecting negative examples and further corroborates the advantage of NegGOA with respect to these comparing methods.

4.5 Contribution of ontology structure

Here, we conduct experiments to investigate the contributions of random walks, hierarchical transitional probability and expanded conditional probability on selecting negative examples. For this purpose, we vary k (number of iterations of random walks) from 0 to 7, and vary β from 0 to 1 with step-size 0.1. The number of FNs (case $r \geq 1$ with $m = 80$k) under different combinations of these two parameters on Yeast genomes annotated with terms in BP branch are revealed in Figure 1(b). Other results on Yeast genomes are provided in Supplementary Figure S1 of the supplementary file.

From Figures 1(b) and S1 of the supplementary file, we can observe that when β is set to 0 (or 1), the number of FNs is much larger than that when $\beta \in (0, 1)$ and $k \geq 1$. This observation shows that both the hierarchical transitional probability and the expanded conditional probability contribute to negative examples selection. This observation also supports our motivation to integrate the ontology structure and empirical conditional probability for selecting negative examples. The smallest number of FNs is made when $k = 6$ and $\beta \in [0.1, 0.9]$ on Yeast genomes in BP branch. This fact demonstrates that random walks also contribute to negative examples selection. When $\beta = 0$, only the expanded conditional probability is used to select negative examples, and the number of FNs keeps stable when $k \geq 1$. A possible reason is that the empirical conditional probability $p(s|t)$ is computed on any pairwise terms and $W_C^k(t, v)$ keeps relatively stable in the iteration of random walks, and thus the contribution of random walks on empirical conditional probability is neutralized.

An interesting observation is that too many iterations of random walks seem not helpful. That is because a large k results in a large transitional probability between two far away terms in GO hierarchy and the deepest terms in the hierarchy by January 2016 is 15. Therefore, a large k is not desirable. In the previous experiments, we fixed k as 4 to avoid too large k, and set β to 0.5 to avoid preferring the hierarchical transitional probability or expanded conditional probability. From these results, we can find that fewer FNs can be achieved by properly tuning β and k.

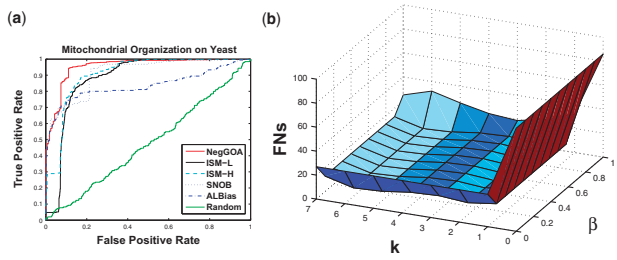

Fig. 1. (**a**) ROC curve for 'GO:0007005' (mitochondrian organization). (**b**) The influence of number of iterations (k) of random walks, relative weight (β) of hierarchical structure similarity and co-occurrence information on Yeast in BP branch

4.6 Negative examples in protein function prediction

To comprehensively study the effect of selected negative examples, we incorporate the selected negative examples into an efficient protein function prediction algorithm—SWSN (Youngs *et al.*, 2013). A brief introduction of SWSN is provided in the supplementary file. Here, we adopt Yeast, Human, Mouse, Fly datasets provided by Mostafavi and Morris (2010) (http://morrislab.med.utoronto.ca/sara/SW/), instead of the genomes extracted from GOA files. Each dataset contains multiple functional association networks. We annotate the proteins in each dataset using the historical GOA files (archived date: May 3, 2013). Since the number of negative examples is supposed to be much larger than that of available annotations, to let the negative examples be self-adaptive, we set the number of selected negative examples as ten times as the number of available annotations of each dataset. The selected negative examples, along with all the available annotations are used by SWSN to predict protein function. Next, we update GO annotations of these proteins using the recent GOA files (archived date: December 8, 2015) and use these updated annotations to validate the predictions made by SWSN. Similar to the experimental protocol of ALBias and SNOB, terms annotated to at least three proteins are considered in the experiments. Table 3 reports the results on Human dataset. The results on other datasets are revealed in Supplementary Tables S20–S22 of the supplementary file. Since the annotations used for training and validation are fixed, these results are reported without standard deviation.

From Table 3, we can easily find that NegGOA achieves better performance than these comparing methods across all the evaluation metrics. In summary, in 48 (4 species × 3 branches × 4 metrics) different configurations, NegGOA outperforms SNOB and ALBias in 93.75 and 97.92% of the cases, loses to them in 6.25 and 2.08% of the cases, respectively. NegGOA always performs better than ISM-L,ISM-H and baseline approach Random. SNOB and ALBias also perform better than Random. SNOB gets better function prediction performance than ALBias, this observation is consistent with the results in Youngs *et al.* (2014). The cause is that the annotations used

Table 3. Performance measures for function prediction on Human dataset

	Branch	NegGOA	ISM-H	ISM-L	SNOB	ALBias	Random
MacroF1	BP	0.7870	0.7130	0.7794	0.7853	0.7733	0.7302
	CC	0.7153	0.6676	0.7137	0.7135	0.7012	0.6721
	MF	0.8440	0.8314	0.8261	0.8419	0.8126	0.7947
AvgAUC	BP	0.9387	0.9167	0.9353	0.9335	0.9217	0.9015
	CC	0.9493	0.9251	0.9331	0.9456	0.9345	0.9049
	MF	0.9555	0.9392	0.9468	0.9525	0.9460	0.9297
1-RankLoss	BP	0.9824	0.8475	0.9571	0.9788	0.9626	0.9535
	CC	0.9860	0.8902	0.9626	0.9830	0.9352	0.9446
	MF	0.9882	0.9448	0.9582	0.9848	0.9567	0.9600
Fmax	BP	0.7623	0.6719	0.7503	0.7553	0.7471	0.7394
	CC	0.7881	0.7699	0.7781	0.7798	0.7716	0.7579
	MF	0.8307	0.8094	0.8234	0.8240	0.8240	0.8158

Downloaded from http://bioinformatics.oxfordjournals.org/ at University of California, Los Angeles on August 30, 2016

by SNOB encode the information of ontology structure to some extent, whereas the direct annotations used by ALBias encode much less information of GO hierarchy. We use Wilcoxon signed rank test (Wilcoxon, 1945) to assess the difference between NegGOA and these comparing algorithms on multiple datasets, and find NegGOA significantly performs better than them with P value smaller than 10^{-6}. From these results, we can draw a conclusion that ontology structure plays important roles in selecting negative examples and hence for function prediction.

We divide the involved GO terms into five groups with different sparsity levels, namely $[3, 10]$, $[11, 30]$, $[31, 100]$, $[101, 300]$ and ≥ 301. For example, group $[3, 10]$ includes the terms annotated to at least 3 proteins and at most 10 proteins and it is the most sparse group among the five groups. We average AUC scores of the terms in each group, and report the results on Human, Yeast, Mouse and Fly in Supplementary Tables S23–S26 of the supplementary file. We use Wilcoxon signed rank test and find that NegGOA gets significantly (with P value smaller than 0.05) larger AUC score than these comparing methods on sparse terms ($[3, 10]$ and $[11, 30]$) and group $[31, 100]$, and it sometimes gets comparable score with these comparing methods on other groups. The reason is that the approximated empirical conditional probability is less reliable on sparse terms than on other terms, and these comparing algorithms are more inclined to select sparse terms as negative examples of a protein than NegGOA. Another cause is that more terms have been annotated to proteins since May 2013. For example, the number of GO terms in BP branch increases from 2928 to 3268, and the number of annotations increases from 201 175 to 240 496 on Human. These appended annotations often describe more detailed biological functions of proteins, and they often correspond to descendants of the terms already annotated to these proteins. These comparing methods favour to select these appended terms as negative examples of proteins, since these terms are not associated with any protein in the historical GOA file. In contrast, NegGOA applies random walks with restart to account for potential missing annotations, so it achieves better performance than them.

GO terms annotated to no more than 30 proteins occupy the majority of GO, these terms often bring much more specific biological information than other terms, and accurately predicting these terms is more challenging (Tao *et al.*, 2007; Valentini, 2014; Yu *et al.*, 2015b). Although the improvement of overall AUC made by NegGOA with respect to these methods are not so large, given such a large number of involved terms and the majority of sparse terms, a small improvement is interesting and prominent.

5 Conclusions and future work

In this paper, we introduce a method called NegGOA to select negative examples of proteins. NegGOA adopts ontology structure, random walks and co-occurrence of terms to model the potentiality of missing annotations of a protein and to select negative examples of the protein. Our extensive experiments show that NegGOA can more accurately select negative examples than the state-of-the-art algorithms. In our future work, we are planning to develop integrative models to utilize other proteomic data sources (i.e. protein–protein interactions) to select negative examples.

Funding

This work is supported by Natural Science Foundation of China (No. 61402378), Natural Science Foundation of CQ CSTC (No. cstc2014jcyj A40031), Fundamental Research Funds for the Central Universities of China (2362015XK07, XDJK2016B009 and XDJK2016D021) and Chongqing Graduate Student Research Innovation Project (No. CYS16070).

Conflict of Interest: none declared.

References

Ashburner,M. *et al.* (2000) Gene Ontology: tool for the unification of biology. *Nat. Genet.*, **25**, 25–29.

Blake,J.A. (2013) Ten quick tips for using the Gene Ontology. *PLoS Comput. Biol.*, **9**, e1003343.

Blei,D.M. *et al.* (2003) Latent Dirichlet allocation. *J. Mach. Learn. Res.*, **3**, 993–1022.

Caniza,H. *et al.* (2014) GOssTO: a stand-alone application and a web tool for calculating semantic similarities on the Gene Ontology. *Bioinformatics*, **30**, 2235–2236.

Cesa-Bianchi,N. *et al.* (2012) Synergy of multi-label hierarchical ensembles, data fusion, and cost-sensitive methods for gene functional inference. *Mach. Learn.*, **88**, 209–241.

Elkan,C. and Noto,K. (2008) Learning classifiers from only positive and unlabeled data. In: *Proceedings of the 14th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, pp. 213–220.

Guan,Y.F. *et al.* (2008) Predicting gene function in a hierarchical context with an ensemble of classifiers. *Genome Biol.*, **9**, S3.

Huttenhower,C. *et al.* (2009) The impact of incomplete knowledge on evaluation: an experimental benchmark for protein function prediction. *Bioinformatics*, **25**, 2404–2410.

Lin,D. (1998) An information-theoretic definition of similarity. In: *Proc. of Intern. Conf. on Machine Learning*, pp. 296–304.

Mostafavi,S. *et al.* (2008) GeneMANIA: a real-time multiple association network integration algorithm for predicting gene function. *Genome Biol.*, **9**, S4.

Mostafavi,S. and Morris,Q. (2009) Using the Gene Ontology hierarchy when predicting gene function. In: *Proceedings of the twenty-fifth Conference on Uncertainty in Artificial Intelligence*, pp. 419–427.

Mostafavi,S. and Morris,Q. (2010) Fast integration of heterogeneous data sources for predicting gene function with limited annotation. *Bioinformatics*, **26**, 1759–1765.

Pesquita,C. *et al.* (2009) Semantic similarity in biomedical ontologies. *PLoS Comput. Biol.*, **5**, e1000443.

Peña-Castillo,L. *et al.* (2008) A critical assessment of *Mus musculus* gene function prediction using integrated genomic evidence. *Genome Biol.*, **9**, S2.

Re,M. *et al.* (2012) A fast ranking algorithm for predicting gene functions in biomolecular networks. *IEEE/ACM Trans. Comput. Biol. Bioinformatics*, **9**, 1812–1818.

Rhee,S.Y. *et al.* (2008) Use and misuse of the gene ontology annotations. *Nat. Rev. Genet.*, **9**, 509–515.

Radivojac,P. *et al.* (2013) A large-scale evaluation of computational protein function prediction. *Nat. Methods*, **10**, 221–227.

Schnoes,M.S. *et al.* (2013) Biases in the experimental annotations of protein function and their effect on our understanding of protein function space. *PLoS Comput. Biol.*, **9**, e1003063.

Škunca,N. *et al.* (2012) Quality of computationally inferred gene ontology annotations. *PLoS Comput. Biol.*, **8**, e1002533.

Tao,Y. *et al.* (2007) Information theory applied to the sparse gene ontology annotation network to predict novel gene function. *Bioinformatics*, **23**, i529–i538.

Thomas,P. *et al.* (2012) On the use of gene ontology annotations to assess functional similarity among orthologs and paralogs: a short report. *PLoS Comput. Biol.*, **8**, e1002386.

Tong,H. *et al.* (2008) Random walk with restart: fast solutions and applications. *Knowl. Inf. Syst.*, **14**, 327–346.

Valentini,G. (2011) True Path Rule hierarchical ensembles for genome-wide gene function prediction. *IEEE/ACM Trans. Comput. Biol. Bioinformatics*, **8**, 832–847.

Valentini,G. (2014) Hierarchical ensemble methods for protein function prediction. *ISRN Bioinformatics*, **2014**, 1–34.

Downloaded from http://bioinformatics.oxfordjournals.org/ at University of California, Los Angeles on August 30, 2016

Wang,S. *et al*. (2015) Exploiting ontology graph for predicting sparsely annotated gene function. *Bioinformatics*, **31**, i357–i364.

Wilcoxon,L. (1945) Individual comparisons by ranking methods. *Biometrics*, **1**, 80–83.

Yang,H. *et al*. (2012) Improving GO semantic similarity measures by exploring ontology beneath the terms and modelling uncertainty. *Bioinformatics*, **28**, 1383–1389.

Youngs,N. *et al*. (2013) Parametric Bayesian priors and better choice of negative examples improve protein function prediction. *Bioinformatics*, **29**, 1190–1198.

Youngs,N. *et al*. (2014) Negative example selection for protein function prediction: the NoGO database. *PLoS Comput. Biol.*, **10**, e1003644.

Yu,G.X. *et al*. (2013) Protein function prediction using multi-label ensemble classification. *IEEE/ACM Trans. Comput. Biol. Bioinformatics*, **10**, 1045–1057.

Yu,G.X. *et al*. (2015a) Predicting protein function using incomplete hierarchical labels. *BMC Bioinformatics*, **16**, 1.

Yu,G.X. *et al*. (2015b) Predicting protein function via downward random walks on a gene ontology. *BMC Bioinformatics*, **16**, 271.

Zhang,M.L. and Zhang,K. (2010) Multi-label learning by exploiting label dependency. In: *Proc. of the 16th ACM SIGKDD Intern. Conf. on Knowledge Discovery and Data Mining(KDD)*, pp. 999–1007.

Zhao,X.M. *et al*. (2008) Gene function prediction using labeled and unlabeled data. *BMC Bioinformatics*, **9**, 57.

Downloaded from http://bioinformatics.oxfordjournals.org/ at University of California, Los Angeles on August 30, 2016