

The Use of Gene Ontology Evidence Codes in Preventing Classifier Assessment Bias

Mark F. Rogers Asa Ben-Hur

March 12, 2009

Abstract

Motivation: The biological community’s reliance on computational annotations of protein function makes correct assessment of function prediction methods an issue of great importance. The fact that a large fraction of the annotations in current biological databases are based on computational methods can lead to bias in estimating the accuracy of function prediction methods. This can happen since predicting an annotation that was derived computationally in the first place is likely easier than predicting annotations that were derived experimentally, leading to over-optimistic classifier performance estimates.

Results: We illustrate this phenomenon in a set of controlled experiments using a nearest-neighbor classifier that uses PSI-BLAST similarity scores. Our results demonstrate that the source of Gene Ontology (GO) annotations used to assess a protein function predictor can have a highly significant influence on classifier accuracy: the average accuracy over four species and over GO terms in the biological process namespace increased from 0.72 to 0.87 when the classifier was given access to annotations that are assigned evidence codes that indicate a possible computational source, instead of experimentally determined annotations. Slightly smaller increases were observed in the other namespaces. In these comparisons the total number of annotations and their distribution across GO terms were kept the same.

Conclusion: In conclusion, taking into account GO evidence codes is required for reporting accuracy statistics that do not overestimate a model’s performance, and is of particular importance for a fair comparison of classifiers that rely on different information sources.

1 Introduction

Biologists rely extensively on annotations of protein function when conducting their research. Protein annotations provide information such as a protein’s molecular functions, the processes in which it participates, and the cellular locations where it is found. The Gene Ontology (GO) was developed as a comprehensive taxonomy for describing gene product characteristics (Ashburner *et al.*, 2001). The ontology is comprised of three hierarchical

namespaces that contain 22,000 terms for describing a gene product's different functional aspects. Researchers commonly annotate proteins with GO terms as a result of running laboratory experiments, performing database searches or using computational models to classify proteins.

The most accurate annotations are derived from laboratory experiments, which are often time-consuming and expensive. However, the overwhelming amount of protein sequence data that is generated by genome sequencing makes it impossible to annotate all newly sequenced genomes experimentally. To obtain annotations quickly and cheaply, researchers use tools such as the Basic Local Alignment Search Tool (BLAST) and its successors (Altschul *et al.*, 1990, 1997) to help them identify homologs with known functions. In addition, the machine learning community has developed a variety of classifiers that annotate proteins based on data such as domain composition, protein interaction data, and microarray gene expression data, (see e.g., Zhou *et al.* (2002); Letovsky and Kasif (2003); Deng *et al.* (2003); Mostafavi *et al.* (2008); Tian *et al.* (2008)).

Computational methods provide a way to annotate novel proteins efficiently, but they may introduce errors into protein databases. As early as 1996, researchers identified automated functional assignment errors as a major problem (Bork and Bairoch, 1996), and those errors may propagate when annotations are transferred between similar proteins (Bork and Bairoch, 1996; Gilks *et al.*, 2002). One of the sources of error in homology-based methods is transfer of annotations associated with one domain when sequence similarity is based on other domains (Marcotte and Marcotte, 2002). Another is that the level of sequence similarity required to make confident predictions varies (Tian and Skolnick, 2003). Not only can automated tools introduce errors, but different annotation methods may assign different, possibly conflicting annotations to the same gene (Brenner, 1999). Despite efforts to address these issues, errors remain in current databases. In 2007 Jones *et al.* estimated that the error rate for some computationally derived GO annotations may reach 49% (Jones *et al.*, 2007).

In this paper we explore a related issue that arises when assessing the performance of a protein function predictor. One would expect a classifier that uses sequence similarity to be successful in predicting annotations that were assigned on this basis. And in general, this can lead to over-optimistic classifier performance estimates for a classifier that uses a method similar to that used in the annotation process, and may obscure a classifier's propensity for propagating annotation errors. While some authors have noted this issue (e.g., L. Peña-Castillo *et al.* (2008)), no other paper explicitly demonstrates the bias that results in using computationally derived annotations in assessing classifier performance as we do here.

To address this issue, researchers must be able to identify the kind of procedure that was used to assign a given annotation. The Gene Ontology provides *evidence codes* that characterize the method used to annotate a gene. The evidence codes distinguish, for example, between annotations based on laboratory experiments and those based on computational predictions (GO Consortium, 2008). In this paper we demonstrate the impact of using annotations that may have been derived computationally, leading to significantly higher accuracy than is obtained when using only experimentally determined annotations. In doing this we wish to alert researchers to this pitfall in working with GO annotations, and to provide

Code	Description	Class
IDA	wet-lab assay (direct assay)	<i>N</i>
IEP	wet-lab assay (expression pattern)	<i>N</i>
IGC	inferred from genomic context	<i>N</i>
IGI	wet-lab assay (genomic interaction)	<i>N</i>
IMP	wet-lab assay (mutant phenotype)	<i>N</i>
IPI	wet-lab assay (physical interaction)	<i>N</i>
IEA	generated computationally or transferred from another database	<i>B</i>
ISS	inferred from sequence or structural similarity	<i>B</i>
RCA	result of a reviewed computational analysis	<i>B</i>
NAS	non-traceable author statement	<i>E</i>
TAS	traceable author statement	<i>E</i>
ND	no data available	<i>E</i>
IC	inferred by curator	<i>N/B/E</i>

Table 1: A classification of GO evidence codes. We classify evidence codes by their potential to bias classifiers that use sequence similarity. *Nonbiasing* codes are evidence codes that are associated with annotations that are not based on sequence similarity, and hence shouldn't bias a classifier that uses this information. *Biasing* codes are associated with annotations that may be based on sequence similarity, and may therefore create bias. The ISS evidence code was recently augmented with three sub-category codes which were used only 11 times in our dataset. *IC*-coded annotations are inferred from other annotations, and therefore inherit the classification of their reference annotations. (*N=Nonbiasing*, *B=Biasing*, *E=Excluded*.)

guidelines for good experimental design. The data and source for generating the results are available at <http://www.cs.colostate.edu/~asa/supplements/bias-package.tar.gz>.

2 Methods

In this paper we conduct a set of experiments that demonstrate the bias that results from using annotations that were derived from sequence similarity to assess the performance of a classifier that uses such information. We used GO evidence codes to identify annotations that can potentially lead to classifier assessment bias. We define *biasing* evidence codes as those that denote annotations that may have been derived from sequence similarity (see Table 1). We classified an evidence code as *nonbiasing* if it is not based on sequence or structural similarity. The *nonbiasing* codes include all those codes that are associated with annotations that were derived from a wet lab assay. We also included the IGC code (inferred from genomic context) as *nonbiasing*. The IGC code signifies that an annotation is inferred based on some aspect of a protein's genomic context such as synteny, operon structure or phylogenetic analysis. Sequence similarity may be used to establish the genomic context, but

it only has an indirect role. No annotations with this evidence code appear in the data we downloaded on May 20, 2008. By extension, *biasing/nonbiasing* annotations are annotations that are associated with *biasing/nonbiasing* evidence codes, respectively.

Some evidence codes are associated with annotations whose original source is ambiguous, so those were excluded from our analysis. For example the TAS and NAS evidence codes are based on a traceable/non-traceable author statement which could have been based on sequence similarity (GO Consortium, 2008). Considering the TAS and NAS evidence codes as *biasing* resulted in no substantive change in the results.

Some annotations are *Inferred by Curator* (IC), which means that an expert used one annotation as evidence for another. We assign each annotation denoted as IC the evidence code of the annotation from which it was derived.

We illustrate the bias in using computationally derived annotations by comparing the accuracy of classifiers that use *biasing* annotations with classifiers that use *nonbiasing* annotations, making sure that each classifier has access to the same number of annotations, with a similar distribution across GO terms. The classifier is a one-nearest-neighbor classifier that uses PSI-BLAST scores to determine if a protein should be annotated with a given GO term. We collected annotations and sequences from four simple, well-annotated eukaryotes that each had a large number of *nonbiasing* annotations and computed the accuracy of the classifier using a form of cross-validation called Leave-One-Species-Out (Vinayagam *et al.*, 2004) that mimics the scenario of annotating newly sequenced genomes by iterating over the species and annotating each one on the basis of annotations in the other species.

2.1 Experimental Setup

To detect the impact that *biasing* annotations may have on classifier accuracy, we trained and tested two classifiers for each GO term in our experiments: one using *nonbiasing* data and the other using *biasing* data. We then computed each classifier’s balanced accuracy for comparison (the definition of balanced accuracy is provided below). We conducted Leave-One-Species-Out experiments by selecting each of the four species in turn to be the “left-out” species excluded from the training data.

To assess a classifier, we used annotated proteins from the left-out species for testing and annotated proteins from the remaining four species for training. For a GO term t we further split the testing and training data sets into positive and negative subsets that depended on t ’s position in the GO hierarchy. For positive examples, we selected proteins explicitly annotated with term t . Note that although proteins annotated with a descendent term of t can be validly annotated with t , such proteins were not included in the positive examples for t since we wanted to create datasets that were independent of each other. We required that the testing and training sets in both *nonbiasing* and *biasing* populations contain at least 10 positive examples each. GO terms represented by fewer examples than these were not included in our experiments.

For choosing negative examples, we applied the strategy proposed in Qiu *et al.* (2007). For each GO term t , we followed all paths from t to the GO hierarchy root and removed from consideration proteins annotated with terms along these paths, since examples labeled

with ancestors of t could also be legitimate examples of t . In addition, we removed from consideration examples labeled with descendants of t : annotations for which t was on a path from the root. Proteins not removed from consideration were then candidates for negative examples. Following the procedure described in Qiu *et al.* (2007), we randomly sampled from these candidates to achieve a 3-to-1 ratio of negative-to-positive examples in all training and test sets.

We identified two potential sources of bias that could influence our results: the number of examples in each data set, and the proportions of GO terms in each data set. Since a classifier’s performance is likely to improve as the amount of training data increases, we ensured that both the *nonbiasing* and *biasing* classifiers for a GO term used the same number of training and test examples. For the four species we studied, the relative proportions of GO terms were often different between *nonbiasing* and *biasing* annotation populations. To eliminate potential bias from proportion differences, we used the training species’ *nonbiasing* annotations to establish a baseline distribution of GO terms. We then ensured that the negative training and test populations for both *nonbiasing* and *biasing* experiments followed this distribution.

In our experiments we used the GO-*slims* ontology, a high-level subset of 133 terms from the full GO hierarchy. Annotations from the full ontology can therefore be converted to GO-*slims* annotations by mapping them to the deepest ancestors which belong to the GO-*slims* ontology. Focusing on these higher-level terms helped ensure enough annotations for training and testing a classifier, and also allowed us to keep the distribution of negative examples across GO terms similar for both *biasing* and *nonbiasing* datasets. We then trained classifiers for each GO-*slims* term for which there were at least 10 *nonbiasing* examples and 10 *biasing* examples available for both training and testing.

2.2 Classifier

For our experiments we used a k -nearest-neighbor classifier that uses PSI-BLAST scores as a measure of sequence similarity. Each protein was characterized by its PSI-BLAST scores with respect to proteins in the four other species. More specifically, we used the negative of the logarithm of the PSI-BLAST E-values to quantify similarity to ensure that a larger score denotes higher similarity. We used `blastpgp` version 2.2.14 running on Linux, and ran PSI-BLAST for two rounds using all annotated proteins within the four species as both the query set and the database set. For PSI-BLAST comparisons, we used an E-value threshold of 1000 and a multipass E-value threshold of 0.0001.

To ensure accurate classifier predictions, we established a classifier E-value threshold of 10^{-10} , and made predictions only when E-values were below this threshold. Our classifier made a binary decision of whether or not a protein should be annotated with a given GO term. The 1-nearest-neighbor classifier made a prediction when a protein’s nearest neighbor had an E-value below the threshold and predicted the protein to be associated with a given GO term if the nearest neighbor belonged to the set of positive examples (proteins annotated with the given term). For $k > 1$ nearest neighbors, the classifier made predictions when at least k proteins could be found with E-values below the threshold, predicting the term if at

least half of them were positive examples for the term.

Whenever the proportion of positive and negative examples is not balanced, accuracy, i.e. the proportion of correctly classified examples, is not a good measure of classifier performance. We therefore used *balanced accuracy* which takes into account the number of examples in each class (Guyon *et al.*, 2006). It is computed as follows. For each GO term t we keep track of the number of true positive (TP_t), true negative (TN_t), false positive (FP_t) and false negative (FN_t) predictions. Balanced accuracy is now computed as the average of the prediction accuracy for positive and negative examples:

$$B_t = \frac{1}{2} \left[\frac{TP_t}{TP_t + FN_t} + \frac{TN_t}{TN_t + FP_t} \right]. \quad (1)$$

Compare this with the standard measure of accuracy $(TP_t + TN_t)/(2 * n)$, where n is the number of examples.

2.3 Data

For our experiments we selected simple eukaryotes that have a large number of *nonbiasing* annotations (see Table 3) to ensure that classifier accuracy is computed on the basis of a large number annotations. We note that in most species the number of *nonbiasing* annotations is very small, a result of the fact that except for a few model organisms, there hasn't been sufficient experimental work to determine the function of more than a handful of genes (or the existing work is not sufficiently represented in GO annotations). This led us to choose the yeast species *S. cerevisiae* and *S. pombe*, the fruit fly *D. melanogaster* and the nematode *C. elegans*.

3 Results

For each GO term in the GO-slims list of terms we performed two leave-one-species out experiments—one with *biasing* data, and one with *nonbiasing* data. For each left-out species and GO term we required at least 10 examples for the training and test sets, and computed each classifier's accuracy in predicting proteins with that GO term. Results obtained for terms in the GO *biological process* namespace are shown in Figure 1. For each species and GO term we plot the accuracy of a classifier that uses *biasing* annotations against the accuracy of a classifier that uses *nonbiasing* annotations. In most cases these accuracy results appear above the diagonal, clearly indicating the bias phenomenon. In Table 2 we compare the average accuracy obtained in each namespace and each species. It illustrates that the accuracy obtained using *biasing* data is significantly higher than that obtained using *nonbiasing* data across species and namespaces, with the highest significance in the *biological process* namespace; statistical significance was assessed using the Wilcoxon signed-rank statistic (Walpole and Myers, 1978). When pooling the results across species the results for each namespace are highly statistically significant.

Classifier Performance in <i>biological process</i>				
Species	Slims Terms	p -value	Balanced Accuracy	
			<i>nonbiasing</i>	<i>biasing</i>
<i>C.elegans</i>	21	$< 10^{-4}$	0.67	0.90
<i>D.melanogaster</i>	26	$< 10^{-5}$	0.68	0.90
<i>S.cerevisiae</i>	25	$< 10^{-4}$	0.74	0.85
<i>S.pombe</i>	25	0.003	0.79	0.82
<i>All Species</i>	30	0*	0.72	0.87

Classifier Performance in <i>cellular component</i>				
Species	Slims Terms	p -value	Balanced Accuracy	
			<i>nonbiasing</i>	<i>biasing</i>
<i>C.elegans</i>	7	0.018	0.71	0.94
<i>D.melanogaster</i>	16	$< 10^{-3}$	0.68	0.89
<i>S.cerevisiae</i>	15	0.036	0.76	0.81
<i>S.pombe</i>	15	0.111	0.75	0.80
<i>All Species</i>	20	$< 10^{-7}$	0.73	0.85

Classifier Performance in <i>molecular function</i>				
Species	Slims Terms	p -value	Balanced Accuracy	
			<i>nonbiasing</i>	<i>biasing</i>
<i>C.elegans</i>	7	0.018	0.84	0.95
<i>D.melanogaster</i>	17	0.002	0.83	0.92
<i>S.cerevisiae</i>	19	0.004	0.83	0.91
<i>S.pombe</i>	13	0.006	0.84	0.92
<i>All Species</i>	23	$< 10^{-7}$	0.83	0.92

Table 2: Comparison of classifier performance using *nonbiasing* and *biasing* evidence codes. In each species we trained classifiers for each GO slim term that had enough *nonbiasing* and *biasing* annotation examples for training and testing. We report for each species and namespace the number of slim terms included, the average balanced accuracy across GO terms and the Wilcoxon signed-rank p -value, indicating the level of significance of the difference between the performance using *nonbiasing* and *biasing* codes. *(The p -value for all species in the *biological process* namespace was smaller than the precision limit on our machines.)

Species Name	Namespace	<i>N</i>	<i>B</i>	<i>E</i>	Total
<i>C. elegans</i>	<i>BP</i>	26692	12663	57	39412
	<i>CC</i>	2252	7623	106	9981
	<i>MF</i>	3902	25057	70	29029
	Total	32846	45343	233	78422
<i>D. melanogaster</i>	<i>BP</i>	13772	8162	5002	26936
	<i>CC</i>	3064	6589	3427	13080
	<i>MF</i>	2604	15912	4426	22942
	Total	19440	30663	12855	62958
<i>S. cerevisiae</i>	<i>BP</i>	12298	10402	1332	24032
	<i>CC</i>	10537	15521	854	26912
	<i>MF</i>	5688	16143	2047	23878
	Total	28523	42066	4233	74822
<i>S. pombe</i>	<i>BP</i>	4358	4580	1304	10242
	<i>CC</i>	8872	4602	379	13853
	<i>MF</i>	1939	5246	1575	8760
	Total	15169	14428	3258	32855

Table 3: Statistics for the GO annotations for the species studied in this paper. For each species we show the numbers of *nonbiasing*, *biasing* and *excluded* codes within each GO namespace. (Namespaces are abbreviated as follows: *BP*=*biological process*, *CC*=*cellular component*, and *MF*=*molecular function*. Evidence code classes are abbreviated *N*=*nonbiasing*, *B*=*biasing*, and *E*=*excluded*.) Annotation files were downloaded from the Gene Ontology website (www.geneontology.org).

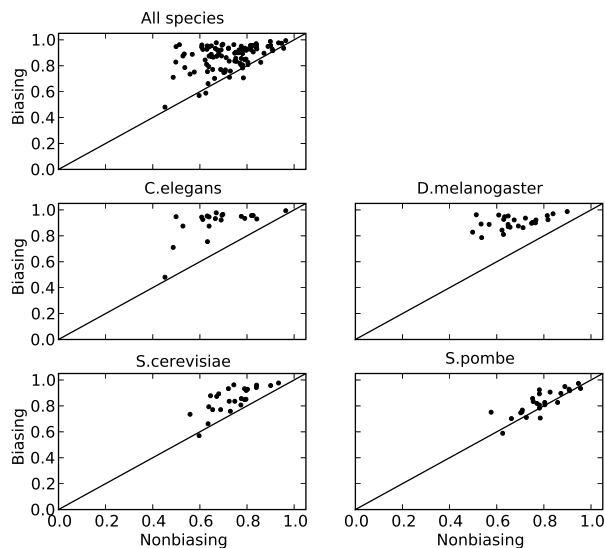


Figure 1: Comparison of accuracy between *nonbiasing* and *biasing* terms in the *biological process* namespace. Each point represents the comparative performance for a single GO term. Points above the diagonal line reveal terms for which our classifier had higher accuracy when trained with *biasing* terms than with *nonbiasing* terms.

In our experiments we found that a 1-NN classifier performed slightly better than 3-NN or 5-NN classifiers. For this reason, the presented results are for the 1-NN classifier; statistical significance was also observed for the 3-NN and 5-NN classifiers (results shown in the supplementary material). In our analysis we excluded the TAS and NAS evidence codes since it is not clear whether annotations with these evidence codes are based on sequence similarity. However, we obtained similar results when TAS and NAS were included as *biasing* (results not shown).

The most significant results were obtained for the *biological process* namespace, followed by *cellular component* and *molecular function*. Statistical significance is impacted by the number of terms that were tested as that increases the statistical power. The largest number of tested terms was in the *biological process* namespace (21-26) followed by *molecular function* (7-19) and *cellular component* (7-16) (recall that a GO term was tested in a particular species if there was sufficient data, which is why the number of terms varies from species to species). Statistical significance in the *cellular component* namespace was higher than in the *molecular function* namespace; therefore the number of terms tested does not by itself explain the differences in p-values. The differences in significance are also related to the relative performance across namespaces. *Molecular function* annotations are the easiest to predict from sequence, with an average *nonbiasing* balanced accuracy of 0.83, compared to 0.72 and 0.73 in *biological process* and *cellular component*, respectively. We believe that since *molecular function* is more directly predictable from sequence (see also Lee *et al.* (2007)) the addition of the *biasing* annotations doesn't provide as big an advantage. We note that overall there are more *biological process* terms and many more *biological process nonbiasing* annotations

that improve the Wilcoxon test’s sensitivity. We also note that because *molecular function* is more easily predicted from sequence, the number of *biasing molecular function* annotations is the highest (see Table 3). In the supplementary material we plot the GO-slits hierarchy, and indicate which terms had sufficient data to be represented in our experiments.

4 Discussion

Our results have implications beyond classifiers that are based on sequence similarity. Current protein function prediction methods use a large variety of data other than sequence similarity: protein-protein interactions and gene expression for example (see e.g., L. Peña-Castillo et al. (2008)). Such a classifier may be biased by a different set of codes than a classifier that uses sequence similarity. For example, a classifier that relies on protein interaction data might be biased by terms associated with the IGI evidence code (*Inferred from Genetic Interaction*) or the IPI evidence code (*Inferred from Physical Interaction*), but it may not be influenced by terms with the ISS evidence code (*Inferred from Structural or Sequence Similarity*).

Taking into account the origin of GO annotations is of particular importance when comparing methods that leverage different kinds of information since each method may be biased by different sets of evidence codes. A fair comparison would use the *nonbiasing* evidence codes common to both models, though this filtering would restrict the amount of annotation data.

Our work does not imply that a classifier shouldn’t use all available information—even annotations derived from *biasing* annotations—when classifying novel proteins. However, our results should caution researchers against using *biasing* annotations when validating or comparing classifiers.

The bias phenomenon is not limited to prediction of protein function. GO annotations are predictive of protein-protein interactions since proteins that participate in similar processes, or are localized in the same cellular compartment are more likely to interact. Therefore they are often used as features in methods for predicting protein-protein interactions (see e.g., Ben-Hur and Noble (2005)). In this case, the classifier would be biased by annotations with the IPI code. Therefore care needs to be taken whenever using GO annotations as features for a classifier.

There is another caveat in using computationally derived predictions as training data for a classifier. It has been noted that computationally predicted annotations have a high error rate: Jones et al. estimate that the error rate among GO annotations with the ISS evidence code (*Inferred by Sequence or Structural Similarity*) may be as high as 49% (Jones et al., 2007), and the error rate in experimentally determined annotations at around 18%. It is well known that “transfer of annotation” can then lead to propagation of errors (Karp, 1998; Gilks et al., 2002; Valencia, 2005). Any classifier may propagate errors, but training and testing a classifier on *biasing* data can make it appear to be more accurate than it really is, possibly masking its propensity to transfer errors. Therefore, when training a classifier to predict the functions of novel proteins using annotations that have a variety of evidence

codes, it may be advisable to take into account their varying reliability levels (see Pal and Eisenberg (2005) and (Buza *et al.*, 2008) for rankings of the perceived reliability of different evidence codes).

5 Conclusion

The relative paucity of experimentally derived annotations of protein function has led researchers to rely on computationally predicted annotations when assessing the performance of methods for protein function prediction. But predicting annotations that were derived computationally using similar information should be easier than predicting experimentally based annotations, which would lead to over-optimistic estimates of classifier accuracy. In this paper we illustrated this bias in the context of function prediction on the basis of sequence similarity; our results show a highly significant difference between classifiers that have access to computationally predicted annotations and classifiers that have access only to annotations derived from biological experiments.

References

- Altschul, S., Gish, W., Miller, W., Myers, E., and Lipman, D. (1990). Basic local alignment search tool. *J Mol Biol*, **215**(3), 403–410.
- Altschul, S., Madden, T., Schaffer, A., Zhang, J., Zhang, Z., Miller, W., and Lipman, D. (1997). Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Research*, **25**, 3389–3402.
- Ashburner, M., Ball, C., Blake, J., Butler, H., Cherry, J., Corradi, J., Dolinski, K., Eppig, J., Harris, M., Hill, D., *et al.* (2001). Creating the gene ontology resource: design and implementation. *Genome Res*, **11**(8), 1425–1433.
- Ben-Hur, A. and Noble, W. (2005). Kernel methods for predicting protein-protein interactions. *Bioinformatics*, **21 suppl 1**, i38–i46.
- Bork, P. and Bairoch, A. (1996). Go hunting in sequence databases but watch out for the traps. *Trends in Genetics*, **12**(10), 425–427.
- Brenner, S. (1999). Errors in genome annotation. *Trends in Genetics*, **15**(4), 132–133.
- Buza, T., McCarthy, F., Wang, N., Bridges, S., and Burgess, S. (2008). Gene Ontology annotation quality analysis in model eukaryotes. *Nucleic Acids Research*.
- Deng, M., Chen, T., and Sun, F. (2003). An integrated probabilistic model for functional prediction of proteins. In *RECOMB '03: Proceedings of the seventh annual international conference on Research in computational molecular biology*, pages 95–103, Berlin, Germany. ACM Press.
- Gilks, W., Audit, B., Angelis, D. D., Tsoka, S., and Ouzounis, C. A. (2002). Modeling the percolation of annotation errors in a database of protein sequences. *Bioinformatics*, **18**(12), 1641–1649.
- GO Consortium (2008). Guide to GO Evidence Codes. <http://www.geneontology.org/GO.evidence.shtml>.
- Guyon, I., Gunn, S., Hur, A. B., and Dror, G. (2006). Design and analysis of the NIPS2003 challenge. In I. Guyon, S. Gunn, M. Nikravesh, and L. Zadeh, editors, *Feature extraction, foundations and applications*. Springer Verlag.
- Jones, C., Brown, A., and Baumann, U. (2007). Estimating the annotation error rate of curated GO database sequence annotations. *BMC Bioinformatics*, **8**, 170.
- Karp, P. (1998). What we do not know about sequence analysis and sequence databases. *Bioinformatics*, **14**(9), 753–754.
- L. Peña-Castillo *et al.* (2008). A critical assessment of *Mus musculus* gene function prediction using integrated genomic evidence. *Genome Biology*, **9 Suppl 1**(S2).

- Lee, D., Redfern, O., and Orengo, C. (2007). Predicting protein function from sequence and structure. *Nature Reviews Molecular Cell Biology*, **8**(12), 995–1005.
- Letovsky, S. and Kasif, S. (2003). Predicting protein function from protein/protein interaction data: a probabilistic approach. *Bioinformatics*, **19** (Suppl. 1), i197–i204.
- Marcotte, C. and Marcotte, E. (2002). Predicting functional linkages from gene fusions with confidence. *Applied Bioinformatics*, **1**, 93–100.
- Mostafavi, S., Ray, D., Warde-Farley, D., Grouios, C., and Morris, Q. (2008). GeneMANIA: a real-time multiple association network integration algorithm for predicting gene function. *Genome Biology*, **9** Suppl 1(S4).
- Pal, D. and Eisenberg, D. (2005). Inference of Protein Function from Protein Structure. *Structure*, **13**(1), 121–130.
- Qiu, J., Hue, M., Ben-Hur, A., Vert, J., and Noble, W. (2007). A structural alignment kernel for protein structures. *Bioinformatics*, **23**(9), 1090–1098.
- Tian, W. and Skolnick, J. (2003). How well is enzyme function conserved as a function of pairwise sequence identity? *Journal of Molecular Biology*, **333**(4), 863–882.
- Tian, W., Zhang, L., Taşan, M., Gibbons, F., King, O., Park, J., Wunderlich, Z., Cherry, J., and Roth, F. (2008). Combining guilt-by-association and guilt-by-profiling to predict *Saccharomyces cerevisiae* gene function. *Genome Biology*, **9** Suppl 1(S7).
- Valencia, A. (2005). Automatic annotation of protein function. *Current Opinion in Structural Biology*, **15**(3), 267–274.
- Vinayagam, A., König, R., Moormann, J., Schubert, F., Eils, R., Glatting, K., and Suhai, S. (2004). Applying Support Vector Machines for Gene ontology based gene function prediction. *BMC Bioinformatics*, **5**(1), 116.
- Walpole, R. and Myers, R. (1978). *Probability and Statistics for Engineers and Scientists*, 2Ed. MacMillan Publishing Co., Inc., New York, NY.
- Zhou, Y., Huang, G., and Wei, L. (2002). UniBLAST: a system to filter, cluster, and display BLAST results and assign unique gene annotation. *Bioinformatics*, **18**(9), 1268.