

# Inferring Gene Ontology Category Membership via Cross-Experiment Gene Expression Data Analysis

*Ben Goertzel<sup>1</sup>, Izabela Freire Goertzel, Cassio Pennachin,  
Moshe Looks, Murilo Saraiva de Queiroz, Francisco Prosdocimi, Francisco Lobo*

*Biomind LLC  
11160 Veirs Mill Rd., L-15, Suite 161  
Wheaton MD 20902*

---

## 1 Introduction

The rapid emergence of high-throughput methods for acquiring information about the sequences, expressions and functions of genes has provided an abundance of valuable new data. However, the volume and complexity of this data is sufficient that analysis using unaided human intuition, or traditional statistical methods, is not adequate. Advanced mathematical and computational methods are required, along with sophisticated approaches to organizing and querying information.

In order to cope with this situation, researchers have developed a variety of tools, including controlled vocabularies for discussing biological phenomena, and collaborative knowledge bases constructed using these vocabularies. A leading example is the multi-organism Gene Ontology (GO) project, which presents collection of terms divided into three ontologies: cellular component, biological process, and molecular function (Gene Ontology Consortium, 2000). Each ontology is a directed acyclic graph (DAG), in which terms are linked to other terms via specialization and part-whole relationships. Proteins are associated to GO terms according to knowledge of their biological roles. The work we describe in this paper pertains specifically to the GO; however the methodologies described would be analogously applicable to any other carefully constructed and reasonably extensive biological ontology.

The GO is powerful but far from complete, a fact that presents a number of interesting computational problems. While the online GO databases are reasonably extensive, not all genes discussed in the literature have been assigned a position in the GO hierarchy. To address this problem, software systems have been created that automatically scan the research literature, and use computational text analysis to guess the appropriate GO category for a gene mentioned in research papers or abstracts (Raychaudhuri et al, 2002). Also, there is a large number of genes that have not yet been studied in depth, that have not yet been written up in research papers, and that is another strong cause of GO-gene dissociation.

Gene Ontology annotations are useful not only as an easy to understand and navigate source of aggregated biological knowledge. Allocco et al (2004), for instance, developed an interesting way to combine microarray expression data, GO annotations and co-regulation, and quantify the relations between these three kinds of data, based on

---

<sup>1</sup> To whom correspondence should be directed: [ben@biomind.com](mailto:ben@biomind.com).

publicly available Yeast data, showing that GO annotations, inferred or not, can be valuable in the discovery of other kinds of biological knowledge.

High-throughput analytic methods like gene expression microarrays allow us to acquire quantitative data on the whole-genome level. This whole-genome quantitative data – potentially coupled with other kinds of biological knowledge such as gene sequence data, domain conservation profiles or protein-protein interactions – can be used to make educated guesses regarding the correct GO classification for genes that are not yet mentioned in the literature. Intuitively, it seems that if a little-explored gene belongs in a certain GO category, then one should often be able to tell this by comparing data such as:

- Its expression levels, obtained in various cells and experimental conditions;
- The observed protein-protein interactions relating to the protein for which it codes;
- The conserved domains observed in its sequence;
- Homology clues found by sequence alignment;

to comparable data obtained from various genes whose correct GO categories are known. This paper describes preliminary experiments in this line of research. Two novel different methods of gene function inference are presented, along with early experimental results in the context of the budding Yeast genome.

## **2 Related and Previous Work**

Experimental methods have been used for gene function inference. For instance, Hughes et al (2000) report on the creation of a compendium of Yeast expression profiles, through systematic mutations, and its use to identify and characterize genes of previously unknown function via clustering of gene expression values measured against this compendium of profiles. Although this method is precise, it's very laborious and expensive, and requires a comprehensive compendium of profiles to achieve good enough coverage.

Methods that are more computationally intensive yet require far less wet lab work have been devised. Such methods use diverse information sources such as structural similarity, microarray expression data, protein-protein interaction data, etc. The use of both sequence information and microarray expression information can be motivated by two studies which, although not directly concerted with gene function inference, did explore gene similarity metrics based on these kinds of information. Structural similarity information can be derived from protein sequence or tertiary structure. The relation between structural similarity and gene category annotations has been explored by Lord et al (2003). Allico et al (2004) have analyzed the relationship between co-expression and co-regulation, using both microarray and transcription factor binding data. Azuaje and Bodenreider (2004) use microarray gene expression data, and find it to correlate reasonably well with two different measures of Gene Ontology similarity, which are computed on the Gene Ontology DAGs using information-theoretic formulae. Other kinds of information have also been used – for instance, Letovsky and Kasif (2003) have reported automated GO category assignment based on an integration of several different types of protein-protein interaction data. In this report, we concentrate on using microarray gene expression data, sometimes coupled with sequence alignment results obtained with BLAST.

Hvidsten et al (2003) used a learning technique based on rough sets to learn numerous simple rules of GO category membership for genes of unknown function. They used a supervised learning approach, using genes with known GO categories to induce rule models, whose accuracy was then measured using cross-validation on human temporal microarray data. Their results, in terms of accuracy, are comparable to the ones we have obtained. Supervised learning is used in the second approach we will describe in this paper,

although the learning techniques are quite different. Their work requires temporal data, which is not nearly as common as stationary data. Our approach works with both temporal and stationary data, as long as sufficient samples are available.

### 3 Materials and Methods

Our preliminary experiments on gene function inference have been focused on the Yeast genome. Yeast is a good model organism for this kind of work, because the yeast GO annotation is relatively reliable and thorough, and there is a relatively large amount of publicly available, well-studied gene expression data for yeast. From the 6220 ORFs with more than 100 codons in the yeast genome, 3922 have been assigned biological process categories in the GO. The biological challenge is, then, to automatically assign functions to the remaining 2298 genes, based on comparing their gene expression values (and other biological information, if available) to the corresponding information about genes of known GO biological process.

We have chosen to work specifically with the *Biological Process* subset of the GO, due to the higher probability that Biological Process GO categories have reasonably well-correlated gene expression values, and would therefore be easier to predict. Hvidsten et al (2003) shared this intuition, and also focused on the Biological Process ontology. Biological processes in the GO can be broad (e.g. signal transduction, cell growth and maintenance) or narrow (pyrimidine metabolism, alpha-glucoside transport), but they always refer to processes that have multiple steps, yet lack the dynamical subtlety that distinguishes a *pathway* from a *process*. Furthermore, for most of the paper we have restricted attention only to biological process GO categories with 200 or fewer elements. This is because very large, more abstract biological process GO terms are much less likely to share significant gene expression patterns, while very small categories are unlikely to provide us with enough training data for supervised learning.

In these experiments, we use the yeast gene expression data collection described in Spellman et al (1998), which is a combination of data from several different experiments. This data collection, available online at <http://celldcycle-www.stanford.edu>, gives 78 different expression values for each ORF with more than 100 codons in the yeast genome. We used those that are related to a subset of the biological process ontology in GO (which numbers over 1000 categories, and is specified in detail below).

We have not found any single algorithmic approach that solves this problem in a dramatically successful way. Here we will describe two different approaches, with different strengths and weaknesses. We have tested these approaches using standard statistical validation methods – data was split into training and test sets, with an equal distribution of the *positives* (the genes known to belong to the target category) between both sets. Only the genes with known GO annotations were used in the validation process.

Our first approach, *BLAST+MI* is based on a combination of BLAST and mutual information, and according to our tests, it provides highly accurate categorization; however, it only makes a prediction about half of the time, for the other half of genes it remains silent, not having enough information to make a guess.

Our second approach, *BOA*, is based on applying a supervised-categorization algorithm called *combinator-BOA* (Looks et al, 2004) to learn rules that predict membership in a given GO category. This approach provides greater recall: for some GO categories it correctly gets over 90% of the genes in a given GO. However, its precision is not nearly as good as for BLAST+MI; it can often give hundreds of false positives.

In addition to doing statistical validation, we have run the BLAST+MI and BOA algorithms to predict GO biological process categories for 2298 yeast genes of unknown

function according to the Gene Ontology as of March, 2004. These predictions are available online at the supplementary website for the paper.<sup>2</sup>

The algorithms described here may be applied to any organism; we focus on yeast in this paper for reasons of simplicity, and other organisms will be treated in sequel papers.

## 4 GO Category Inference Using Mutual Information and BLAST

The first approach we have found for automatic assignment of GO categories to genes is a relatively simple one, involving the combination of BLAST similarity measurement with a calculation involving mutual information.

### 4.1 GO Category Inference Using BLAST Alone

Our *BLAST+MI* approach is an extension of a simpler and more standard approach in which one does GO category inference using BLAST similarity alone. This is the standard strategy used to make GO category assignments to genes from newly sequenced organisms – in such cases, one categorizes a gene in a newly sequenced organism by assigning it the category of some gene in another better-understood organism, to which it has a high sequence similarity.

In the BLAST-only approach, given a gene  $G$ , and a GO category  $C$ , one first calculates the BLAST e-value between  $G$  and every gene in the GO category. One then does an initial filtering step, in which one discards all e-values bigger than a certain threshold, chosen as  $10^{-5}$ . Then the BLAST similarity from  $G$  to  $C$  is defined as the minimum among the remaining e-values. In many cases the BLAST similarity from  $G$  to  $C$  will be undefined, because no e-value from  $G$  to any gene in  $C$  survives the initial filtering phase.

At this point, we know which genes will have some category predicted for them, and which will not. But we may have a number of predictions for each gene. We may then filter the list of categories predicted for each gene. This is done using a ranking function, via which we select the top  $b$  GO categories predicted for each gene, and consider these as valid predictions. Increasing  $b$  through the values 10-30, we obtain more and more predictions, and also more and more errors.

Figure 1 shows the categorization accuracy achieved via this simple, BLAST-only approach on the test subset of the Spellman dataset. The figure shows positive predictive value (percentage of instances judged positive that actually are positive) and specificity (percentage of instances judged negative that actually are negative) on the test set. For these experiments, we used all subsets of the GO category "Biological Process", with the exception of the categories "Unknown" and "Obsolete Biological Process." Results are shown separately for the set of subsets with 2-200 elements, and the set of subsets with >200 elements.

---

<sup>2</sup> [http://www.biomind.com/papers/inferring\\_go/inferring\\_go.htm](http://www.biomind.com/papers/inferring_go/inferring_go.htm).

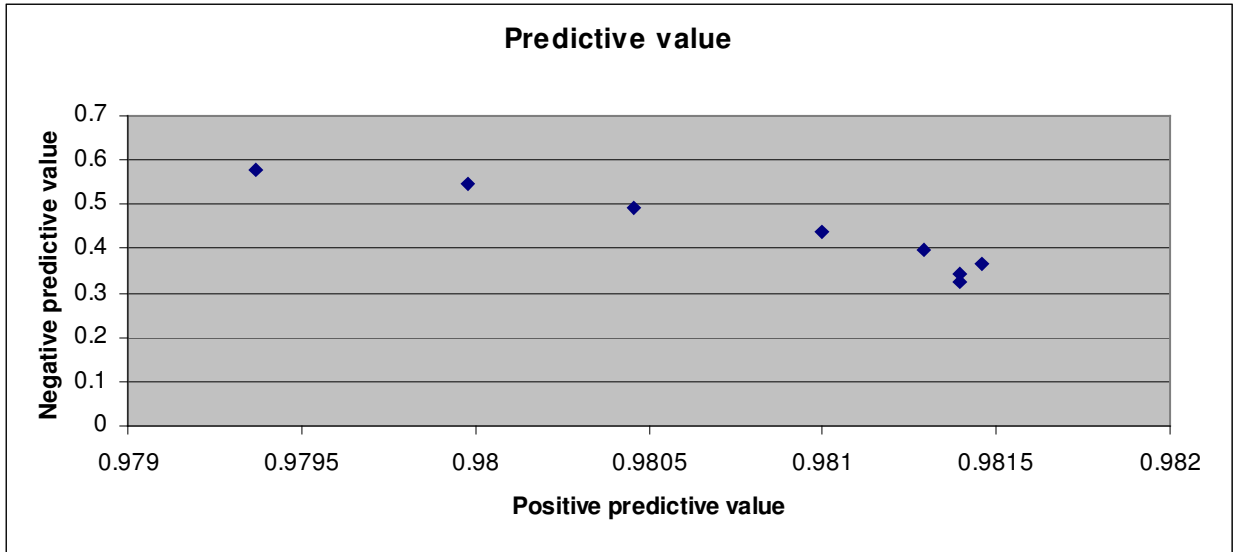


Figure 1a – results for categories with 2-200 elements

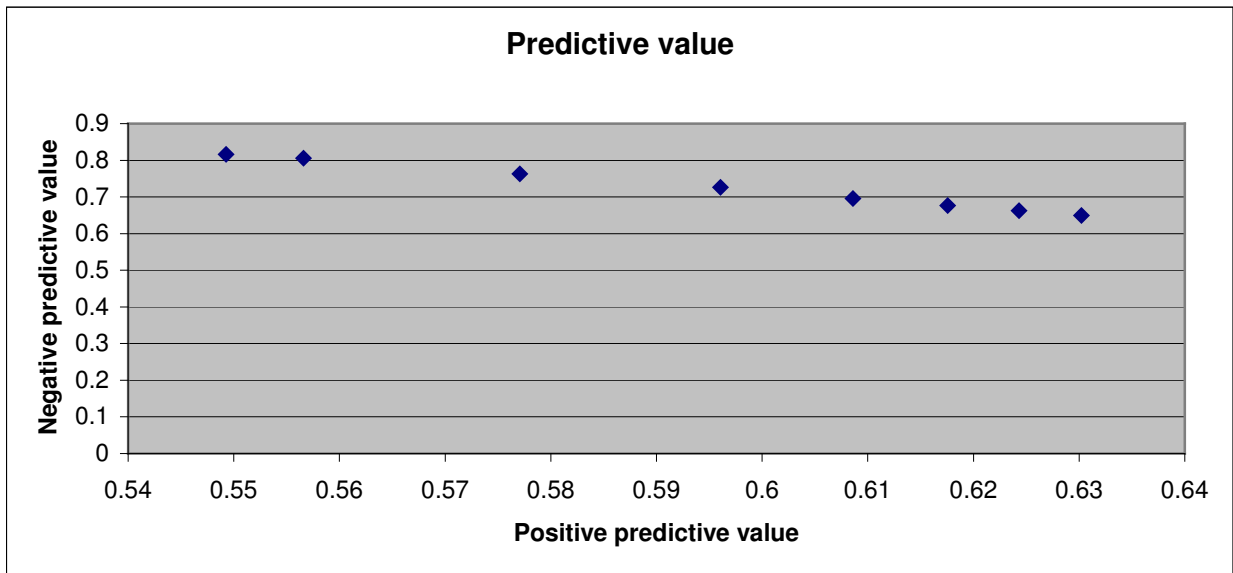


Figure 1b – results for categories with > 200 elements

## 4.2 BLAST+MI

The next step is to improve on these results using gene expression data. In order to do this, we calculate the mutual information (MI; Shannon, 1948) between a gene and each GO category. We do this similarly to with the BLAST similarity discussed above: we calculate the mutual information between the gene and each particular gene in a given GO category, and then take the maximum of these values.

To calculate mutual information, we discretize the gene expression values, replacing each value by one of  $K$  symbols. The symbol replacing an expression value is calculated by first normalizing the values into  $[0,1]$ , then partitioning the interval  $[0,1]$  into  $K$  equally probable partitions. The normalization is done on a per-gene basis. After experimenting with several different values of  $K$ , we chose to use 3 partitions for all our experiments with *BLAST+MI*. We don't use direct mutual information; rather, we use mutual information ranking, to account for differences in both gene and GO-category expression profile similarity.

We then combine the BLAST and MI calculations using a simple algorithm. Two integers are specified,  $b$  and  $m$ . We accept a GO category  $C$  as a prediction for a gene  $G$  if and only if:

- $C$  has rank  $\leq b$  in the list of GO categories sorted by BLAST similarity to  $G$ , and
- $C$  has rank  $\leq m$  in the list of GO categories sorted by MI to  $G$ .

Statistics for this approach, similar to those given above for the BLAST-only approach, are given in Figures 2 and 3. Varying the values of  $b$  and  $m$  varies the number of genes for which predictions are made, and also the number of GO categories predicted for each gene. The range used for both  $b$  and  $m$  was 5-40, with steps of 5 .

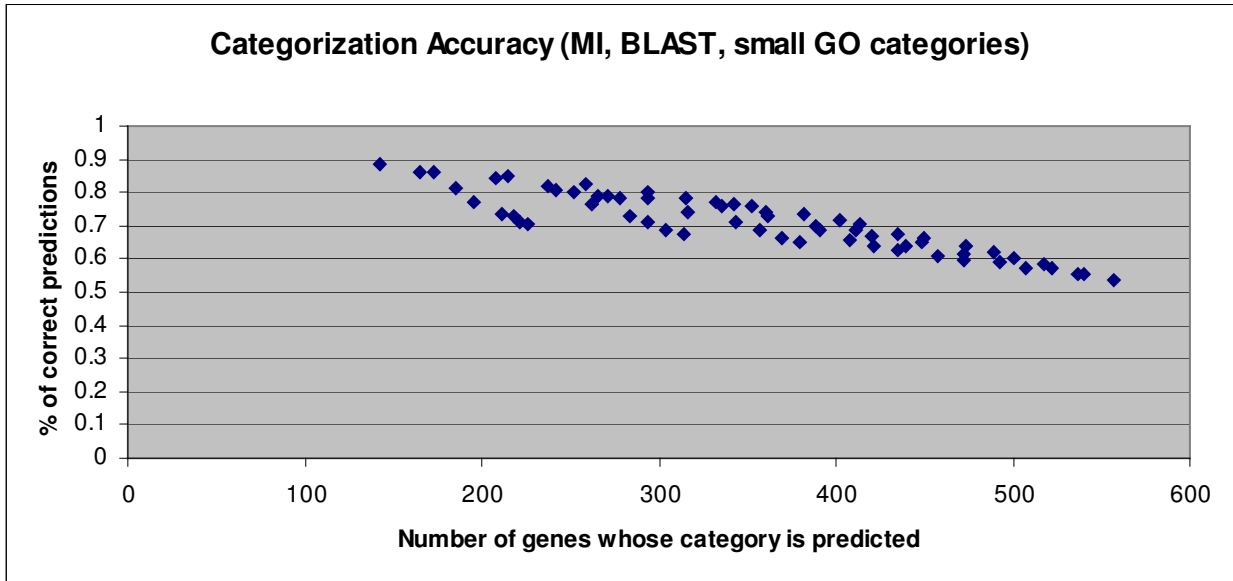


Figure 2a – results for categories with < 200 elements

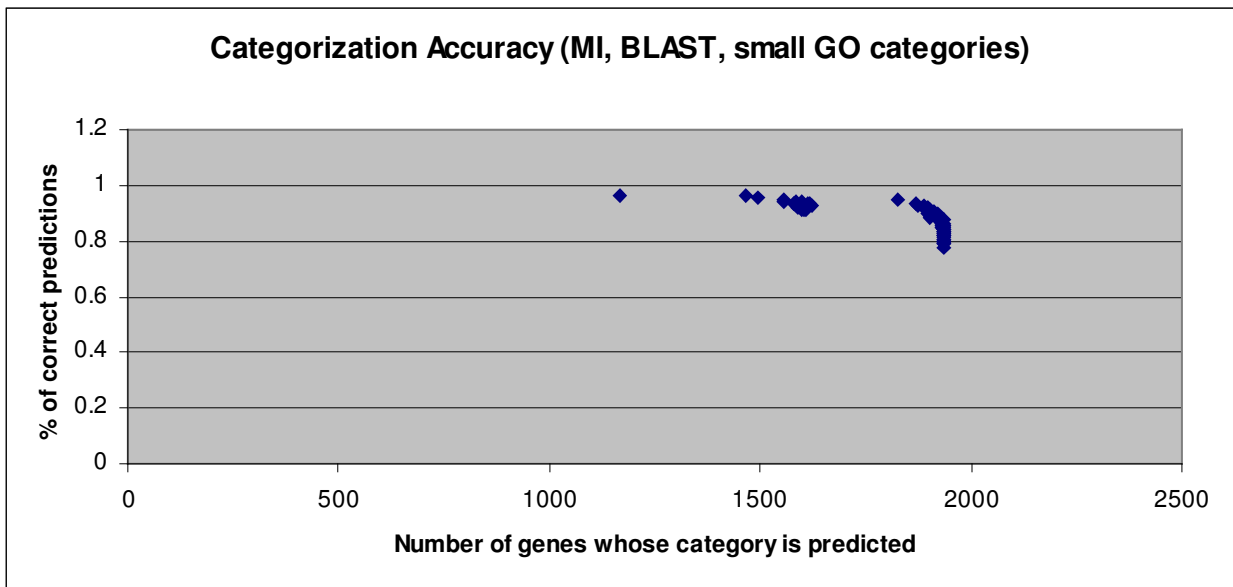


Figure 2b – results for categories with > 200 elements

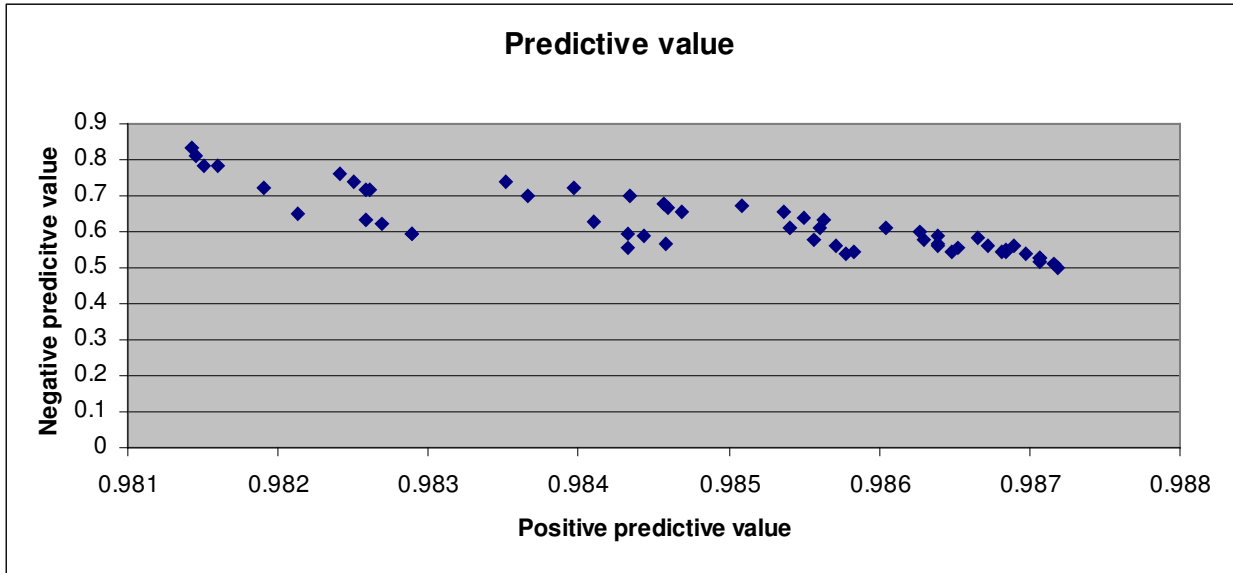


Figure 3a – results for categories with 2-200 elements

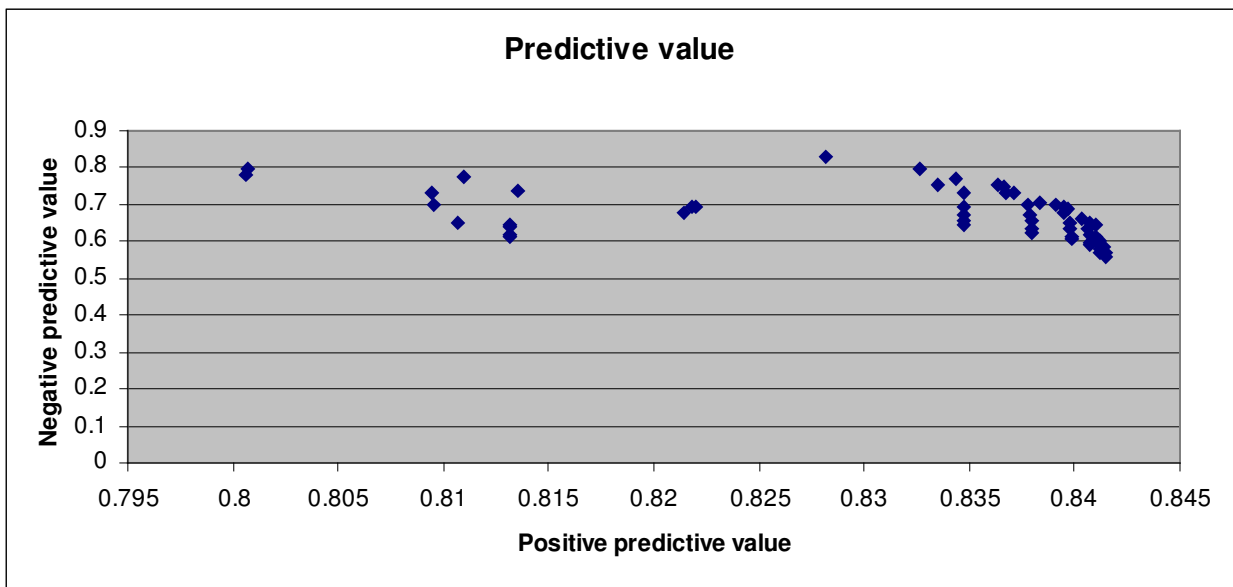


Figure 3b – results for categories with >200 elements

## 5 GO Category Inference Using BOA-Based Supervised Categorization

The approach described in the previous section is powerful, but its dependence on gene sequence comparison renders it intrinsically limited, since for many genes, there are no genes with known GO categorization that are similar enough to be helpful for category assignment. Therefore one would like a method that doesn't require sequence information. This section reports our best efforts in this direction.

Our basic approach, in the absence of useful BLAST information, is to explicitly treat the GO category inference problem as a supervised categorization problem. Given a particular GO category, we try to learn a *model* predicting membership in that category. Mathematically a *model* of a GO category is a function whose input is a gene and whose output is a binary prediction telling whether or not the gene belongs to the category. The model is allowed to internally reference a gene expression data corpus, such as the Spellman dataset used in the work reported here.

We tried four different supervised categorization schemes on this problem:

1. Support vector machines (Cristianini, 2000).
2. Genetic programming (Koza, 1992).
3. A greedy algorithm, designed specifically for the gene function inference problem, and that searches for simple models consisting of conjunctions of experiments, and then does voting among these simple models to make a final prediction.
4. *Combinator-BOA*, an application of the Bayesian Optimization Algorithm (Pelikan, 2002) to data structures called *combinator trees* (Looks et al, 2004).

Of all these, only the third and fourth methods gave results significantly better than random chance. Since the results obtained with *Combinator-BOA* were significantly better than the results obtained with our ad hoc greedy algorithm, we concentrate on the former technique.

The *Combinator-BOA* algorithm is described in detail in (Looks et al, 2004). The approach is somewhat similar to supervised categorization using genetic programming, but with two key changes. The traditional genetic operators of crossover and mutation are replaced by BOA's population-modeling based approach to candidate solution generation. Also, the standard Koza-style function tree is replaced by a combinatory-logic style function tree (Curry and Feys, 1958), which is a binary tree in which all internal nodes represent *function application* and all constant values and variable inputs are contained in leaf nodes.

*Combinator-BOA* is a very general evolutionary learning and optimization technique, but for this application we have restricted it significantly by limiting the function vocabulary that could be used in the learned models. We have tried two different approaches, one using Boolean logical operators and one using arithmetic operators.

Example results obtained applying *Combinator-BOA* classification to GO categories are given in Table 1. More extensive results are given in the online supplementary materials, referenced above. The final column in the table gives the categorization rule used to predict membership in the GO category.

Some results from BLAST+MI are also given in the table. They were obtained by tuning BLAST+MI to maximize the percentage of true negatives. This is probably not the best way to use BLAST+MI – because the strength of BLAST+MI is its ability to maximize the percentage of true *positives*. However, this provides the clearest way to compare BLAST+MI with BOA. The point of the comparison is that, if one is after maximizing the

true negative percentage, then BOA beats BLAST+MI. On the other hand, we do not know how to tune BOA to do what BLAST+MI does best, which is to make a small number of high-accuracy predictions.

GO ID	GO Term	BLAST+MI tuned for large TN		BOA		
		NPV	PPV	NPV	PPV	Model
GO:0000027	Ribosomal large subunit assembly and maintenance	0.75	0.92	0.89	0.85	25 AND !71 AND 12
GO:0000070	Mitotic sister chromatid segregation	0.69	0.76	0.72	0.69	73 AND 77
GO:0000074	Regulation of cell cycle	0.78	0.82	0.65	0.92	!26 AND 7 AND !22 AND !35
GO:0000082	G1/S transition of mitotic cell cycle	0.59	0.66	0.89	0.37	(0.238988 < 19)
GO:0000086	G2/M transition of mitotic cell cycle	0.70	0.83	0.69	0.74	(0.258573 < (69 - 62) / 18)
GO:0000122	Negative regulation of transcription from Pol II promoter	0.66	0.57	0.56	0.71	54 && 21
GO:0000282	Bud site selection	0.80	0.69	0.72	0.63	59 < 30
GO:0000283	Establishment of cell polarity (sensu <i>Saccharomyces</i> )	0.44	0.82	0.74	0.53	11 < 30
GO:0000398	Nuclear mRNA splicing, via spliceosome	0.39	0.78	0.97	0.25	54 OR !33
GO:0000747	Conjugation with cellular fusion	0.69	0.55	0.53	0.67	21 OR !70

**Table 1:** Comparison of BLAST+MI and Combinator-BOA Results on Selected GO Categories  
PPV = positive predictive value, NPV = negative predictive value  
The values in the BLAST+MI column represent values obtained from tuning BLAST+MI to maximize the percentage of true negatives; an odd tuning that is useful only for comparison with BOA..

The classification rules produced by *Combinator-BOA* are satisfyingly simple, both for arithmetic and logical rules. A typical example of a logical rule is the rule for category GO:0000074 shown in the above table. In order to tell whether a gene *G* belongs in the category GO:0000074, it suffices to evaluate the logical rule

$$!26 \text{ AND } 7 \text{ AND } !22$$

which is a shorthand for

$$\text{NOT}(\text{Experiment}_{26}) \text{ AND } \text{Experiment}_{7} \text{ AND } \text{NOT}(\text{Experiment}_{22})$$

To evaluate a rule like this, one first takes the expression values of *G* for all the 78 experiments in the Spellman dataset, and normalizes these values into the interval [0,1]. The normalization is done on a per-gene basis as described above. The values are then discretized as described above, using  $K=2$ , yielding Boolean logical values for each (*gene*, *experiment*) pair. One then evaluates the logical rule corresponding to the category. If the

value that comes out of this is greater than 0.5, then  $G$  belongs to the category modeled by the rule.

Arithmetic rules are calculated similarly: here one does per-gene normalization but no discretization, and then simply plugs the normalized expression values into the arithmetic expression that constitutes the rule. If the arithmetic expression comes out non-negative, then the gene belongs to the corresponding GO category; if it comes out negative, then the gene does not.

The strengths and weaknesses of the supervised categorization approach are well demonstrated by the numbers in Table 1. For example, the model for GO:0000027 is quite successful, predicting category membership correctly 89% of the time, and category non-membership correctly 85% of the time – it has both high specificity and sensitivity. The problem is that a specificity of just 85% results in too many false positives when the data is very skewed, as is the case with GO category membership – the vast majority of genes will *not* belong to a GO category, unless it's an extremely general one. Therefore, while the *Combinator-BOA* approach has very good recall, the precision never gets anywhere near the range obtained by the *BLAST+MI* approach.

## 6 Conclusion

We have reported two different methodologies for inferring gene function (GO biological process category assignment) based on cross-experiment gene expression data. The two methods occupy different points in precision-recall space: one gives high precision but low recall, the other gives low precision but high recall. In both cases there is a fair bit of variation in precision and recall depending on the category.

The algorithms described here were chosen after a process of experimentation with several other approaches. While one never knows for sure, we doubt that these results can be improved upon significantly without using either a much larger gene expression data corpus, or else a corpus containing a variety of types of biological information. The choice of the gene expression corpus to use is a key one. Ideally, one wants a corpus with many experiments, and also that the measurements are taken in normal, or control, conditions, as frequently as possible. The pitfalls of performing gene function inference based on non-control expression profiles are well explored by Birrell et al (2002).

The essential difficulty of doing gene function categorization based solely on gene expression data and sequence data is that, unless one uses BLAST information, one is dealing with a massively underdetermined problem, and it becomes very difficult to decrease the number of false positives below a certain level. However, when BLAST similarity decreases below a certain level of significance, it is effectively useless for gene function categorization. So there is only a certain percentage of uncategorized genes that can be categorized by the more effective, BLAST-incorporating methods. For this reason, one ends up either with a high-precision, low-recall, BLAST-based algorithm, or a lower-precision, higher-recall non-BLAST-based approach.

Our future work in this area will center on the extension of the approach to additional organisms, and the incorporation of additional data types beyond microarray data and sequence data into the analysis. We believe that the *Combinator-BOA* approach has a great deal of potential, and that with the incorporation of a larger quantity of more diverse data into the same algorithmic framework, much higher precision figures can be obtained.

## References

- ALLOCCO, D., Kohane, I., and Butte, A. Quantifying the relationship between co-expression, co-regulation and gene function. *BMC Bioinformatics*, vol. 5, no. 18, 2004.
- AZUAJE, F. and Bodenreider, O. Incorporating ontology-driven similarity knowledge into functional genomics: an exploratory study. *Proceedings of the IEEE Fourth Symposium on Bioinformatics and Bioengineering (BIBE-2004)*, pp. 317-324, 2004.
- BIRRELL, G., Brown, J., Wu, H., Giaever, G., Chu, A., Davis, R., and Brown, J. Transcriptional response of *Saccharomyces cerevisiae* to DNA-damaging agents does not identify the genes that protect against these agents. *PNAS*, vol. 99, pp. 8778-8783, 2002.
- CRISTIANINI, N. and Shaw-Taylor, J. *Support Vector Machines*. Cambridge University Press, 2000.
- CURRY, H., and Feys, R. *Combinatory Logic*. North-Holland, 1958.
- GENE ONTOLOGY CONSORTIUM. Gene Ontology: Tool for the Unification of Biology. *Nature Genetics*, vol. 25, pp. 25-29, 2000.
- HUGHES, T., Marton, M., Jones, A., Roberts, C., Stoughton, R., Armour, C., Bennett, H., Dai, H., He, Y., Kidd, M., King, A., Meyer, M., Slade, D., Lum, P., Stepaniants, S., Shoemaker, D., Gachotte, D., Chakraburttty, K., Simon, J., Bard, M., and Friend, S. Functional discovery via a compendium of expression profiles. *Cell*, vol. 102, pp. 109-126, 2000.
- HVIDSTEN, T., Læg Reid, A., and Komorowski, J. Learning rule-based models of biological process from gene expression time profiles using Gene Ontology. *Bioinformatics*, vol. 19, pp. 1116-1123, 2003.
- KOZA, J. *Genetic Programming*. MIT Press, 1992.
- LÆGREID, A., Hvidsten, T., Midelfart, H., Komorowski, J., and Sandvik, A. Predicting Gene Ontology Biological Process From Temporal Gene Expression Patterns. *Genome Res.*, vol. 13, pp. 965-979, 2003.
- LETOVSKY, S. and Kasif, S. Predicting Protein Function from Protein-Protein Interaction Data: A Probabilistic Approach, *Bioinformatics*, vol. 19 suppl. 1, pp. i197-i204, 2003.
- LOOKS, M., Goertzel, B., and Pennachin, C. Learning Computer Programs with the Bayesian Optimization Algorithm. *In preparation*, 2004.
- LORD, P., Stevens, R., Brass, A., and Goble, C. Investigating semantic similarity measures across the Gene Ontology: the relationship between sequence and annotation. *Bioinformatics*, vol. 19, pp. 1275-83, 2003.

PELIKAN, M. *Bayesian Optimization Algorithm: From Single Level to Hierarchy*, PhD thesis, Computer Science Department, University of Illinois at Urbana-Champaign, 2002.

RAYCHAUDHURI, Soumya, Jeffrey Chang, Patrick Sutphin, Russ Altman. *Genome Research*, 12:2002-214, 2002

SHANNON, C. A mathematical theory of communication, *Bell System Technical Journal*, vol. 27, pp. 379-423 and 623-656, 1948.

SPELLMAN, P., Sherlock, G., Zhang, M., Iyer, V., Anders, K., Eisen, M., Brown, P. Botstein, D., and Futcher, B. Comprehensive identification of cell cycle-regulated genes of the Yeast *Saccharomyces cerevisiae* by microarray hybridization. *Molecular Biology of the Cell* 9, 3273-3297, 1998.