

# Knowledge-Guided Analysis of Gene Expression Data Using Genetic Programming, Support Vector Machines and the Gene Ontology and PIR Databases

Cassio Pennachin, Ben Goertzel<sup>1</sup>, Lucio Coelho, Izabela Freire Goertzel, Murilo Saraiva de Queiroz, Francisco Prosdocimi, Francisco Lobo

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

---

## 1 Introduction

Microarray technology, which allows the collection of large-scale gene expression data, has matured significantly in the last few years. But the analysis of microarray data is still somewhat problematic. Experimental noise is still a common issue, the number of samples studied is not always adequate to ensure the desired levels of statistical significance; and most crucially, one is dealing with a large volume of quantitative data possessing complex patterns on various levels.

The data-analysis challenge posed by microarrays has two aspects. Firstly, one wants to achieve mathematically and statistically valid results describing aspects of microarray data. Secondly, one wants to obtain analytical results that aid the scientist's mind in understanding the data, and relating the data to existing knowledge and future experiments. Sometimes a statistically valid analysis is all that's required, and conceptual transparency to humans is irrelevant – this may be the case, for instance, if one's goal is to learn a diagnostic rule that predicts if a person has a certain disease based on a complex combination of gene expression values derived from their blood samples. Often, however, the role of microarray data analysis is to provide information to be integrated by biologists into their future research; in these cases, comprehensibility of results is equally important as mathematical accuracy.

Here we approach the problem of microarray data analysis in a manner that incorporates both familiar and novel aspects. We take a supervised-categorization based approach; that is, we deal with the problem of learning *models* (mathematical combinations of gene expression values) that distinguish one category of gene expression profiles from another. We utilize two well-known machine learning algorithms, genetic programming and support vector machines. However, we apply these algorithms in an unusual way – not merely giving them gene expression values directly, but also giving them additional inputs, which are derived from gene expression values using knowledge resources such as the Gene Ontology (*GO*; Gene Ontology Consortium, 2000), and the Protein Information Resource (*PIR*; Wu et al, 2003). This approach allows highly effective gene expression data categorization, and it also has the benefit of addressing the problem of conceptual transparency, mentioned above. It often produces classification models that have clear biological meaning and are useful for directing the scientist's mind toward an understanding of the issues under study.

---

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

We describe our methods in detail, and then we report results obtained by applying them to three standard datasets. Our methods provide outstanding accuracy results in two of the three datasets, with no classification errors. In the third dataset, however, we only obtained good results when performing standard gene expression based categorization.

## 1.1 Supervised Categorization of Gene Expression Data

A number of prior researchers have applied supervised categorization technology to microarray gene expression data. This work has used a variety of algorithms and has yielded some quality results; however, the present technique overcomes some significant limitations of the methods described in the literature.

We must emphasize the difference between supervised categorization and *clustering*, a form of unsupervised learning (Eisen et al, 1998) that has frequently been applied to microarray data. Clustering uses the expression profiles as its sole input, and identifies groups of genes with similar expression patterns. Several different clustering algorithms have been applied to microarray expression data, providing useful insights into gene expression patterns under many different situations.

Categorization, unlike clustering, requires that a subset of the gene expression samples collected using microarrays be labeled. Labels can indicate the presence of a disease, good or bad prognostics, etc, and identify the different classes in the data at hand. Expression profiles are usually called *feature vectors* in this context. A feature vector corresponds to the profile of one patient, or sample, in the dataset at hand, and each gene's expression value corresponds to one feature. Labeled feature vectors are used in the learning of *models*, or *classifiers*, which can then hopefully be used to accurately classify unlabeled samples. Usually, statistical validation is used with sets of unlabeled samples, to measure the accuracy and generalization power of the learned models.

As hinted in the discussion above, the purpose of performing supervised classification on microarray data is twofold. Most explored is the fact that one is looking for both models that are highly accurate, and can therefore be used to develop diagnostics. But an equally important goal, and of more immediate consequences, is the search for models that are not only accurate, but also *compact and understandable*, as such models can be used as guidelines for experimental work, in the context of exploratory or discovery-oriented research projects.

Since the earliest days of microarray data classification (Brown, 2000), numerous techniques have been used for supervised learning, among them genetic programming (Koza, 1992), k-nearest neighbors, decision trees (Quinlan, 1993), support vector machines (Cristianini, 2000), and ensemble methods that combine multiple models to generate a more powerful one (Dietterich, 2000; Tan and Gilbert, 2003). Dudoit et al (2002) provide a review and comparison of multiple learning techniques in several microarray datasets containing data on different kinds of tumors. Lately, support vector machines and ensemble methods, especially boosting, seem to be the most prevalent ones for microarray data categorization, among the standard machine learning tools.

The techniques in this diverse set have different advantages and problems. A common pattern seems to emerge, though. The techniques that are most informative, like genetic programming and decision trees, seem to provide poor classification accuracy, and to be more prone to overfitting to the training data (thus generalizing poorly). Support vector machines and ensemble methods, on the other hand, tend to provide excellent accuracy and generalization, but generate models that can't be easily understood. The latter problem comes in varying degrees of severity. Some data patterns are not immediately comprehensible but become conceptually transparent after a few minutes of scrutiny and database-surfing. Other data patterns, however, may simply be too

complicated for the researcher to fully grasp their underlying why and wherefore, at least without truly tremendous amounts of study.

To overcome this disadvantage of SVMs and ensemble methods, it is common to use some technique to generate a small list of important genes. One approach is to perform numerous runs of supervised learning, and to extract from these runs a list of genes that are prevalent in the learned models (Cho et al, 2004). Another approach is to perform aggressive feature elimination, in order to finally learn models that use only a small subset of the features in the original dataset (Guyon et al, 2002; Wang et al, 2003). Both techniques can alleviate the understandability problem, yet they don't fully solve it – it is important not only to know which genes have a differentiating role in the data at hand, but also how these roles relate to each other in the construction of a classifier. In other words, a list of genes isn't nearly as informative as a simple model involving those genes.

As prior researchers have shown, the machine learning approach to microarray data analysis is a powerful one, based on rigorous statistical methodology and advanced mathematical pattern-search algorithms. However, the available machine learning techniques have significant shortcomings. Here we describe an innovative approach to overcoming these shortcomings of straightforward machine learning by utilizing knowledge from GO, PIR and other knowledge resources. This approach involves constructing structures we call *enhanced feature vectors* – numerical vectors, used as inputs to machine learning algorithms, that incorporate numbers specially computed via the combination of gene expression values with information from ontologies and databases. We demonstrate, with results from three different datasets, that our approach can provide models that are accurate, compact, and also understandable.

## **2 Generating Enhanced Feature Vectors with Background Knowledge Integration**

One approach to improving supervised categorization of microarray data would be to develop new and better classification algorithms. This is a valid and interesting area of research. However, we felt it might be more rewarding to take a different tack, and to focus on developing techniques for *guiding existing learning techniques in the direction of patterns that are likely to be biologically relevant*. This is what is achieved by the enhanced feature vectors approach.

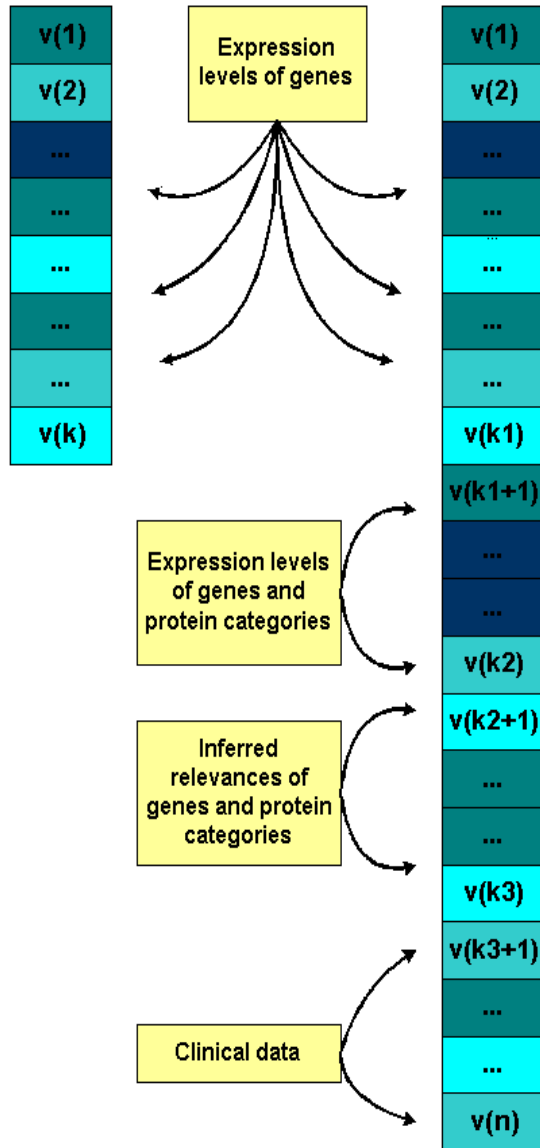
While the implementation of our approach is rooted in bioinformatics and mathematics, much of its inspiration is drawn from the discipline of cognitive science. The human brain is excellent at pattern recognition, but not because it is a fully general pattern-recognizing machine; rather, because it combines a decent capacity for general pattern-recognition with excellent, evolved tuning for the patterns it habitually encounters. Thus, for instance, the human visual cortex is better at doing edge detection in nature scenes than in purely artificial pictures of a similar mathematical complexity -- because it has an in-built *inductive bias* (Baum, 2004) tuned by millennia of experience detecting edges in retinal images derived from the natural environment. Similarly, in the enhanced feature vectors approach, one seeks to give machine learning algorithms an *inductive bias* guided by knowledge of biology and the types of patterns that are likely to occur in biological systems.

The methodology of enhanced feature vectors, which we propose here, provides a simple and intuitive way of guiding machine learning algorithms toward patterns of biological significance and human comprehensibility. The method is not a panacea, but we believe it is a valuable addition to the microarray analysis toolkit.

In some cases enhanced feature vectors allows higher-accuracy categorization than straightforward machine learning; in other cases it does not. In nearly all cases, however, it produces more easily biologically comprehensible classification rules. This is particularly

important when one of the main goals of classification analysis is to gain biological understanding. In most cases, there are a lot of classification rules solving a given classification problem with roughly equal accuracy. Using the enhanced feature vectors approach, a single classification rule may encapsulate a large amount of biological information, making interpretation simpler.

The basic idea of the enhanced feature vectors approach, as the name suggests, is to produce feature vectors containing additional entries besides the usual (normalized, transformed) gene expression values. These additional features consist of biologically-relevant aggregates of gene expression values, or of additional information regarding entities under study, such as questionnaire answers, or other clinical data. We have experimented with multiple ways to produce these additional feature values, and in this paper we will report in detail only on the simplest of these; future publications will cover more advanced approaches.



**Figure 1.** Abstract representation of enhanced feature vectors.

The left side shows an ordinary feature vector of gene expression values, and the right side shows an enhanced feature vector. In this example clinical data is included along with expression levels for GO and PIR categories; and the case where the membership of genes in GO categories is automatically inferred rather than given by the GO database is also covered.

The simplest way to produce additional features is to use the Gene Ontology, which contains human-created groupings of genes into categories. The GO is not a complete or flawless resource – it will contain some errors of commission, and quite possibly a great number of errors of omission (i.e. there may be very many cases in which a certain gene is involved in a certain biological process, but the GO does not acknowledge this). However, it represents fairly solid biological knowledge agreed-up on and utilized by a wide section of the biology research community.

For each entity whose gene expression profile is under study, we may create a single feature value corresponding to each GO category. In the simplest approach, these GO-derived feature values may be computed by averaging. Let  $G$  be the set of genes whose expression values have been measured (i.e., the original feature set), and  $GO$  the set of gene ontology categories. If we have a GO category  $C \in GO$ , and an entity  $E$ , then the value of the feature corresponding to  $C$  in the feature vector of entity  $E$  may be defined as the average expression in  $E$  of all the genes  $g_j \in G$  annotated to belong to  $C$ .

More formally, assume we have an entity (e.g. an organism or a tissue sample, evaluated at a single time point)  $E_i$ , and let  $gene\_exp_{ij}$  denote the (perhaps normalized and transformed) expression level of gene  $g_j$  in entity  $E_i$ . Let  $GO_k$  denote the  $k$ 'th GO category under consideration. Let  $G_k$  denote the set of genes contained in  $GO_k$ .

We may consider a number  $w_{jk}$   $[0,1]$  associated with each element  $g_j$  in  $G_k$ , which is the confidence with which it is known that  $g_j \in G_k$ . As a default this may be set close to 1 (e.g. 0.99), but in cases where GO category membership is determined via automated learning, the confidence may be significantly lower. In the results reported here we set all  $w_{jk}$  constant, but we have done other work to be reported elsewhere, in which the  $w_{jk}$  vary significantly because some gene-GO assignments are made by fairly unreliable machine learning methods.

Given all these preliminaries, we now define the amalgamated expression value of the GO category  $GO_k$ , as

$$GO\_exp_{ik} = \sum_{j: g_j \in G_k} [ w_{jk} \ gene\_exp_{ij} ]$$

This may be useful for all GO categories, but it is particularly valuable for GO categories that are in the *Biological Process* subset of the GO, and even more so for categories that are fairly far down in the GO hierarchy, so that they refer to a fairly specific biological process rather than a highly general process. More specific processes will tend to have more highly correlated gene expression patterns, whereas in general processes the averaging formula might cancel out relevant patterns.

Using this approach, one may obtain, for each entity being categorized, an extended feature vector of length

$$|gene\_set| + |GO\_cat\_set|$$

where  $|gene\_set|$  is the number of genes measured by the microarrays under use, and  $|GO\_cat\_set|$  is the number of gene ontology categories being utilized. Alternately, one may choose to utilize only the GO-based feature vector entries, thus obtaining a feature vector of length  $|GO\_cat\_set|$ .

This basic methodology can be expanded quite naturally by adding biological data sources used beyond the GO. The main additional source we have experimented with is the PIR (Protein Information Resource), which groups proteins into families and superfamilies. In this case we may associate an additional feature with each protein family or superfamily, and define the feature value for an entity according to an unweighted analogue of the GO formula given above

Another natural data source is pathway diagrams, such as those available on KEGG (Kanehisa, 1997) or the signal transduction pathway databases. One may create a feature corresponding to each pathway, and define the value of a pathway-feature as the average value of the genes involved in the pathway. In this case, however, averaging is less likely to be a good choice, due to the different roles played by each gene in the pathway. A more informed formula would probably be desirable, but that would require encoding of pathway roles for each gene.

Regardless of the details, which may be various as the above discussion makes clear, the essence of the enhanced feature vector approach is simple. Rather than simply associating each entity being classified with a vector of gene expression values, one associates it with additional values as well. These values represent biologically meaningful aggregates of gene expression values (or, in some cases, clinical data); that is, they are derived not only using the gene expression values themselves, but also using *biological background knowledge* (which we sometimes abbreviate BI, for Background Information). One may then use these aggregate values as input to machine learning algorithms. One can append them to the traditional feature vectors, or ignore the original gene expression values and utilize only the aggregated values.

Enhanced feature vectors may lead to greater classification accuracy, a consequence of the inability of machine learning algorithms to scan the space of all possible combinations of feature values. In principle, any data pattern expressed in terms of enhanced feature vectors, could also be expressed in terms of ordinary feature vectors – since the additional entries in enhanced feature vectors are simply combinations of the entries in ordinary feature vectors. However, it may not always be possible for a machine learning algorithm, running in realistic amounts of compute time, to find these patterns as combinations of gene expression values – whereas the patterns may be quite simple and easy to find when expressed in terms of knowledgeably aggregated gene expression values. What we have found, through our experimentation with microarray data analysis, is that in some cases enhanced feature vectors lead to greater classification accuracy, but in others they do not. So, where classification accuracy is concerned, we regard enhanced feature vectors not as a universal solution, but rather as another valuable trick to be added to the toolkit.

However, the greater human comprehensibility of the models achieved using enhanced feature vectors will be made clear in the following section. The degree of comprehensibility of classification models depends largely on the classification algorithm in use. We have experienced extensively with genetic programming and support vector machines.

In the case of support vector machines, the models are largely opaque to the human eye regardless of whether enhanced feature vectors are used or not. However, one can deduce from an SVM model that features are most important for the model; and in this case, the enhanced feature vectors pay a dividend, because the list of *important features* turns out to be a list of important biological processes or cellular components (the GO case) or protein families or pathways (if other databases are in use). It can be very valuable for a biologist to find out, for instance, which biological processes are most important for accurately separating two categories of organisms under study. This list of *most useful processes* may well be different from the list of biological processes whose aggregate expression values most clearly distinguish the two categories in question.

In the case of genetic programming, models tend to be relatively simple and to involve a small number of features. This is a context in which the enhanced feature vector approach really shines. One often finds classification rules that are highly compact and involve a simple logical combination of a small number of biological processes. Such rules transparently and immediately give scientific insight into the biological processes that fundamentally distinguish the categories being studied. We will see examples of this in the following section.

## **3 Experimental Results**

### **3.1 Biological Data**

We have experimented with three publicly available datasets:

1. **ALL/AML.** Acute lymphoblastic leukemia (ALL) and acute myeloid leukemia (AML), originally published by Golub et al (1999). The goal with this dataset is cancer subtype classification based on expression patterns of bone marrow samples. The data is originally split in a training set with 38 samples (27 ALL and 11 AML) and a test set with 34 samples (20 ALL and 14 AML).
2. **Lung cancer.** Originally published by Gordon et al (2002), this dataset has expression data on samples of lung malignant pleural mesothelioma (MPM) and adenocarcinoma (ADCA), the goal being to classify between the two tumor types. The training data has 32 samples, equally split between the two classes. The test set has 134 ADCA samples and 15 MPM samples.
3. **Prostate cancer.** (Singh et al, 2002) has published this dataset containing both cancer and control samples. The training dataset has 52 cancer samples and 50 control samples, while the test dataset has 25 cancer samples and 9 normal ones.

The ALL/AML dataset is well-known and has been analyzed in a number of studies (Cho et al., 2004; Dudoit et al., 2002; Guyon et al., 2002; Tan and Gilbert, 2003; Wang et al., 2003). Classification accuracy values between 91% and 100% on the test set have been reported. The other two datasets have been analyzed in (Tan and Gilbert, 2003), and ensembles of decision trees obtained test set accuracies of 93.29% for Lung cancer dataset and 73.53% for the Prostate cancer dataset.

## 3.2 Methods

We tried many combinations of data treatments and categorization parameters in order to achieve the results shown in the next section. First of all, the datasets underwent a normalization based on log-transform and Z score, in such a way that, after normalization, all features had mean zero and variance 1.

Two well-known categorization methods – Genetic Programming (GP) and Support Vector Machines (SVMs) – were used. The details of GP and SVM usage during the tests are listed below:

- Regarding Background Information (BI) usage, one of the following combinations of features was chosen:
  - Original features plus BI (Background Information) features. BI features are the extra features added by the enhanced feature vectors approach, measurements of the relations between the genes in the dataset and GO (Gene Ontology) and PIR (Protein Information Resource) categories.
  - BI features only.
- For SVM tests, the kernel parameter was varied among three different types: Linear, Gaussian and Polynomial. The kernel basically defines what kind of function is being used to compute a hypersurface that separates the two categories in the n-dimensional feature space.
- For GP tests, the fitness function was varied among six different types:
  - Euclidean: fitness for a given automaton is calculated as the inverse of the Euclidean distance between the desired and computed outputs of the automaton for each patient.
  - Hit Based: fitness is calculated as the number of hits (correct classifications) of an automaton for all patients.

- Hit Based with penalty against constant automata: similar to hit based fitness; however, automata that produce the same classification for all inputs are penalized with fitness zero.
- Linear with penalty against constant automata: fitness is the inverse of the sum of absolute errors between expected and computed outputs; however, automata that produce the same output for all inputs receive fitness zero.
- Category Skew fitness: similar to hit based, but hits are weighted in inverse proportion to the abundance of elements in the category being hit; that is, hits in categories with fewer elements are heavier than the ones for categories with many elements.
- F-Measure Fitness: fitness for a given automaton is calculated as a modified F-measure of the classification results obtained.
- The operator sets used in GP test were arithmetic, arithmetic reduced (+ and – operations only) and logic/arithmetic.
- For both GP and SVM tests, some form of feature selection (or no feature selection at all) was used. A given test could use one of the following feature selection types:
  - None: i.e., all features in the dataset are used.
  - Most Expressed Features: selects the  $n$  features that have higher average values in each category.
  - Most Variant Features: selects the  $n$  features that have higher differences among the elements belonging to a given category and the ones that do not belong to it.
- For all feature selection processes, the number of features being selected was chosen between 10 and 1000. Feature selection acted indiscriminately over all types of features -- gene expression and BI -- with no artificial emphasis on any specific type. Therefore, feature numerical significance alone played a role during feature selection.
- All GP parameters not specified above were set to the following default values:
  - Population Size: 1000;
  - Maximum automaton depth: 5;
  - Number of generations: 50;
  - Mutation rate: 0.005;
  - Elitism on;
  - Tournament-based selection with a tournament size of 2.
- All SVM parameters not specified above were set to the following default values:
  - Epsilon: 0.001
  - Tolerance: 0.001
  - Standard deviation for the Gaussian kernel: 1.0
  - $r$ ,  $s$  and  $d$  parameters for the Polynomial kernel: 1.0

Separate training and test sets were used for statistical validation, in order to allow comparison with previously published results.

### 3.3 Results

Table 1 summarizes the accuracy we obtained in our tests using enhanced feature vectors. We compare these results with previous reports on the same dataset. Accuracy values are for the test sets only.

Dataset	Method	Accuracy	Accuracy in Literature
Lung Cancer	SVM	100.00%	93.29%
	GP	91.30%	
Prostate Tumor	SVM	100.00%	73.53%
	GP	85.30%	
ALL/AML	SVM	70.60%	100%
	GP	73.50%	

Table 1. Test set accuracies

One can see that enhanced feature vectors provide outstanding accuracy when used with SVMs, and also pretty high accuracy (comparable to or better than literature) when used with Genetic Programming, in both the Lung and Prostate cancer datasets. Unfortunately, enhanced feature vectors offer rather poor performance in the ALL/AML dataset – the causes for this performance are currently being studied and will be reported in a sequel to this paper. Using SVMs without background information on that dataset, we obtained an accuracy of 94.12%, which shows that the learning techniques we applied are able to learn accurate and general models in that dataset, but are somehow misled by the presence of the extra features. Results on that dataset were poor when we used both the original and extra features, and when we used only the extra features.

The models learned using genetic programming are compact and informative. Below, we show the best models found for each dataset using genetic programming, in algebraic form. Figures 1 through 3 show graphical representations of these models.

- **Lung cancer:**

$$\frac{(NM\_005110 + NM\_001614)}{(NM\_002230 * 0.099310) * (FAM0002476 + NM\_006994)} / \frac{(NM\_005139 / 0.002459) * NM\_003311 / GO:0006776 * SF001194 * 0.021784 * (GO:0050770 + SF002561) / NM\_002997}{}$$

- **Prostate Tumor**

$$GO:0015263 / 0.214165 * GO:0050549 * 0.209606 * 0.466819 * GO:0047450 * 0.172400 * 0.086049 * (FAM0009488 * GO:0001560 - 0.843054 / GO:0047584) * GO:0030511 / 0.053761 * GO:0005081 / 0.014583$$

- **ALL/AML**

$$GO:0015043 - GO:0009073 - GO:0015046 + GO:0006558 + GO:0015542 - GO:0016319 + GO:0004128 - GO:0015304 + GO:0046469 - GO:0045154 + GO:0015518 + GO:0006926 - GO:0046469 - GO:0008632 + GO:0030900 - GO:0005366$$

Tables 2 through 4 display the most important features used by the models learned using enhanced feature vectors, also providing some potential insight into the underlying biological phenomena. More detailed analysis of these genes and categories is underway, and will be published in subsequent reports on both the Lung and Prostate cancer cases.

Type	Name	Description
GENE	NM_001614	actin, gamma 1
GENE	NM_001740	calbindin 2, 29kDa (calretinin)
GENE	NM_002010	fibroblast growth factor 9 (glia-activating factor) (FGF9)
GENE	NM_000295	serine (or cysteine) proteinase inhibitor, clade A (alpha-1 antiproteinase, antitrypsin), member 1 (SERPINA1), transcript variant 1
GENE	NM_002537	ornithine decarboxylase antizyme 2 (OAZ2)
BI	FAM0033404	
BI	GO:0030172	troponin C binding [molecular function]
BI	SF001509	eukaryotic type I DNA topoisomerase
GENE	NM_005110	glutamine-fructose-6-phosphate transaminase 2 (GFPT2)
GENE	NM_005139	annexin A3 (ANXA3)
GENE	NM_005775	vinexin beta (SH3-containing adaptor molecule-1) (SCAM-1)
GENE	NM_000604	fibroblast growth factor receptor 1 (fms-related tyrosine kinase 2, Pfeiffer syndrome) (FGFR1), transcript variant 1
BI	GO:0005519	cytoskeletal regulatory protein binding [molecular function]
BI	FAM0033857	
GENE	NM_006404	protein C receptor, endothelial (EPCR) (PROCR)
BI	FAM0020295	
BI	GO:0042573	retinoic acid metabolism [biological process]
BI	SF000628	basic fibroblast growth factor receptor 1
BI	FAM0002508	
BI	SF001720	

*Table 2. Useful features for Lung Cancer best model*

Type	Name	Description
GENE	NM_001280	cold inducible RNA binding protein (CIRBP)
GENE	NM_000995	ribosomal protein L34 (RPL34), transcript variant 1
BI	SF002233	<b>rat</b> ribosomal protein L34
GENE	NM_002056	glutamine-fructose-6-phosphate transaminase 1 (GFPT1)
BI	FAM0000573	
BI	GO:0030288	periplasmic space (sensu Gram-negative Bacteria) [cellular component]
GENE	NM_004839	homer homolog 2 (Drosophila) (HOMER2), transcript variant 1
GENE	NM_012248	selenophosphate synthetase 2 (SEPHS2)
BI	GO:0016260	selenocysteine biosynthesis [biological process]
BI	GO:0001300	chronological cell aging [biological process]
BI	GO:0001301	progressive alteration of chromatin during cell aging [biological process]
BI	GO:0001302	replicative cell aging [biological process]
BI	GO:0007098	centrosome cycle [biological process]
BI	GO:0007100	centrosome separation [biological process]
BI	GO:0007571	age-dependent general metabolic decline [biological process]
BI	GO:0007576	nucleolar fragmentation [biological process]
BI	GO:0007099	centriole replication [biological process]
BI	GO:0007101	male meiosis centrosome cycle [biological process]
BI	GO:0046605	regulation of centrosome cycle [biological process]
GENE	NM_002520	nucleophosmin (nucleolar phosphoprotein B23, numatrin) (NPM1)

*Table 3. Useful features for Prostate Tumor best model*

## 4 Conclusion

We have described a novel approach to categorizing gene expression data derived from microarrays, using machine learning algorithms applied to enhanced feature vectors, produced by means of GO and PIR. The classification accuracy obtained by this approach is competitive with that of ordinary machine learning, and in some cases significantly superior. And, in the case of genetic programming, the classification models produced are much easier to understand and interpret than the models produced by machine learning algorithms acting on gene expression data directly. This conceptual transparency is very important in cases where microarray analysis is being used as part of a larger biological research effort, so that the results of categorization are intended to be used by scientists to aid in their interpretation of biological phenomena.

Type	Name	Description
BI	SF000735	nucleoside diphosphate kinase
BI	GO:0005366	myo-inositol:hydrogen symporter activity [molecular function]
BI	GO:0015304	glucose uniporter activity [molecular function]
BI	GO:0005351	sugar porter activity [molecular function]
BI	GO:0008506	sucrose:hydrogen symporter activity [molecular function]
BI	GO:0015519	D-xylose:hydrogen symporter activity [molecular function]
BI	GO:0015043	leghemoglobin reductase activity [molecular function]
BI	GO:0015046	rubredoxin reductase activity [molecular function]
BI	GO:0008937	ferredoxin reductase activity [molecular function]
BI	GO:0030586	(methionine synthase) reductase activity [molecular function]
BI	GO:0030917	midbrain-hindbrain boundary development [biological process]
BI	GO:0016319	mushroom body development [biological process]
BI	GO:0006558	L-phenylalanine metabolism [biological process]
BI	GO:0008632	apoptotic program [biological process]
BI	GO:0006927	programmed cell death, transformed cells [biological process]
BI	GO:0009164	nucleoside catabolism [biological process]
BI	GO:0004128	cytochrome-b5 reductase activity [molecular function]
BI	GO:0015043	leghemoglobin reductase activity [molecular function]
BI	GO:0015517	galactose:hydrogen symporter activity [molecular function]
BI	GO:0015774	polysaccharide transport [biological process]
BI	GO:0015760	glucose-6-phosphate transport [biological process]
BI	GO:0048036	central complex development [biological process]
BI	GO:0045476	nurse cell apoptosis [biological process]

Table 4. Useful features for ALL/AML Leukemia best model

## 4.1 Future Work

Finally we mention a few further extensions that we have experimented with, but that are not included in the results presented in this paper. Results using these further extensions will be presented in sequel papers.

As suggested in Figure 1 above and the related discussion, there is a valuable intersection between enhanced feature vectors and the process of gene function inference (Hvidsten et al, 2003; Goertzel et al, 2004). The weighted GO formula is particularly useful when one couples the enhanced feature vector approach with a process of gene function inference, in which one uses automated techniques to estimate the GO categories to which a gene belongs. GO category assignments derived via automated inference will generally be assigned lower confidence than those assigned by human biologists, and this lower confidence can be reflected in the weightings. A sequel paper will present microarray categorization results obtained using enhanced feature vectors derived from gene function inference. This is particularly valuable in the cases of organisms without a very fully developed Gene Ontology database. In these cases the most powerful source of GO category assignments for genes is often automated cross-organism inference.

Also, it can be useful to create features that are based on Boolean combinations of GO categories or PIR families/superfamilies, or other categories, rather than individual categories. While using random Boolean combinations is not worthwhile, there are techniques that search in an unsupervised way for *interesting* Boolean combinations – combinations that occur much more, or less, frequently than one would expect according to probabilistic independence assumptions. We have developed one such technique based on the *combinator-BOA* function learning algorithm (Looks et al, 2004).

We have also experimented with including some features not derived from gene expression data at all, but rather derived from *clinical data*. This clinical data may be

medical in nature (e.g. the measured level of various substances in the blood), or it may derive from the answers human patients have given on questionnaires relevant to their medical conditions. In any case, these clinical-data values may be appended to the feature vectors just like the aggregated values discussed above. This data may present special problems for some machine learning algorithms because in some cases it may be discrete rather than continuous, so that by combining it with gene expression values one is creating a combination of continuous and discrete features. But this is unproblematic for the GP, and easily addressed for the SVM.

## References

BAUM, E. *What is Thought?* MIT Press, 2004

BROWN, M., Grundy, W., Lin, D., Cristianini, N., Sugnet, C. Knowledge-based analysis of microarray gene expression data by using support vector machines. *Proc. Natl. Acad. Sci. USA*, vol. 97, pp. 262-267, 2000.

CHO, J., Lee, D., Park, J., and Lee, I. Gene selection and classification from microarray data using kernel machine. *FEBS Letters*, vol. 571, pp. 93-98, 2004.

CRISTIANINI, N. and Shaw-Taylor, J. *Support Vector Machines*. Cambridge University Press, 2000.

DIETTERICH, T. An experimental comparison of three methods for constructing ensembles of decision trees: bagging, boosting, and randomization. *Machine Learning*, vol. 40, pp. 139-157, 2000.

DUDOIT, S., Fridlyand, J., and Speed, T. Comparison of discrimination methods for the classification of tumors using gene expression data. *J Am Stat Assoc*, vol. 97, pp. 77-87, 2002.

EISEN, M., Spellman, P., Brown, P., and Botstein, D. Cluster analysis and display of genome-wide expression patterns. *Proc. Natl. Acad. Sci. USA*, vol. 95, pp. 14863-14868, 1998.

GENE ONTOLOGY CONSORTIUM. Gene Ontology: Tool for the Unification of Biology. *Nature Genetics*, vol. 25, pp. 25-29, 2000.

GOERTZEL, B., Goertzel, I., Pennachin, C., Looks, M., Queiroz, M., Prosdociami, F., and Lobo, F. Inferring Gene Ontology Category Membership via Cross-Experiment Gene Expression Data Analysis. *In preparation*, 2004.

GOLUB, T., Slonim, D., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J., Coller, H., Loh, M., Downing, J., Caligiuri, M., Bloomfield, C., and Lander, S. Molecular Classification of Cancer: Class Discovery and Class Prediction by Gene Expression Monitoring. *Science*, vol. 286, pp. 531-537, 1999.

GORDON, G., Jensen, R., Hsiao, L., Gullans, S., Bluemnstock, J., Ramaswamy, S., Richard, W., Sugarbaker, D., and Bueno, R. Translation of microarray data into clinically relevant cancer diagnostic tests using gene expression ratios in lung cancer and mesothelioma. *Cancer Res.*, vol 62, pp. 4963-4967, 2002.

- GUYON, I., Weston, J., Barnhill, S., and Vapnik, V. Gene selection for cancer classification using support vector machines. *Machine Learning*, vol. 46, pp. 389-422, 2002.
- HVIDSTEN, T., Lægreid, 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.
- KANEHISA, M. A database for post-genome analysis. *Trends Genet*, vol. 13, pp. 375-376, 1997.
- KOZA, J. *Genetic Programming*. MIT Press, 1992.
- LOOKS, M., Goertzel, B., and Pennachin, C. Learning Computer Programs with the Bayesian Optimization Algorithm. *In preparation*, 2004.
- QUINLAN, J. *C4.5: Programs for Machine Learning*. Morgan-Kaufmann, 1993.
- SINGH, D., Febbo, P., Ross, K., Jackson, D., Manola, J., Ladd, C., Tamayo, P., Renshaw, A., D'Amico, A., Richie, J., Lander, E., Loda, M., Kantoff, P., Golub, T., and Sellers, W. Gene expression correlates of clinical prostate cancer behavior. *Cancer Cell*, vol. 1, pp. 203-209, 2002.
- TAN, A., and Gilbert, D. Ensemble machine learning on gene expression data for cancer classification. *Applied Bioinformatics*, vol. 2, pp. S75-S83, 2003.
- WANG, J., Bø, T., Jonassen, I., Myklebost, O., and Hovig, E. Tumor classification and marker gene prediction by feature selection and fuzzy c-means clustering using microarray data. *BMC Bioinformatics*, vol. 4:60, 2003.
- WU, C., Yeh, L., Huang, H., Arminski, L., Castro-Alvear, J., Chen, Y., Hu, Z., Ledley, R., Kourtesis, P., Suzek, B., Vinayaka, C., Zhang, J., and Barker, W. The Protein Information Resource. *Nucleic Acids Research*, vol. 31, pp. 345-347, 2003
- .

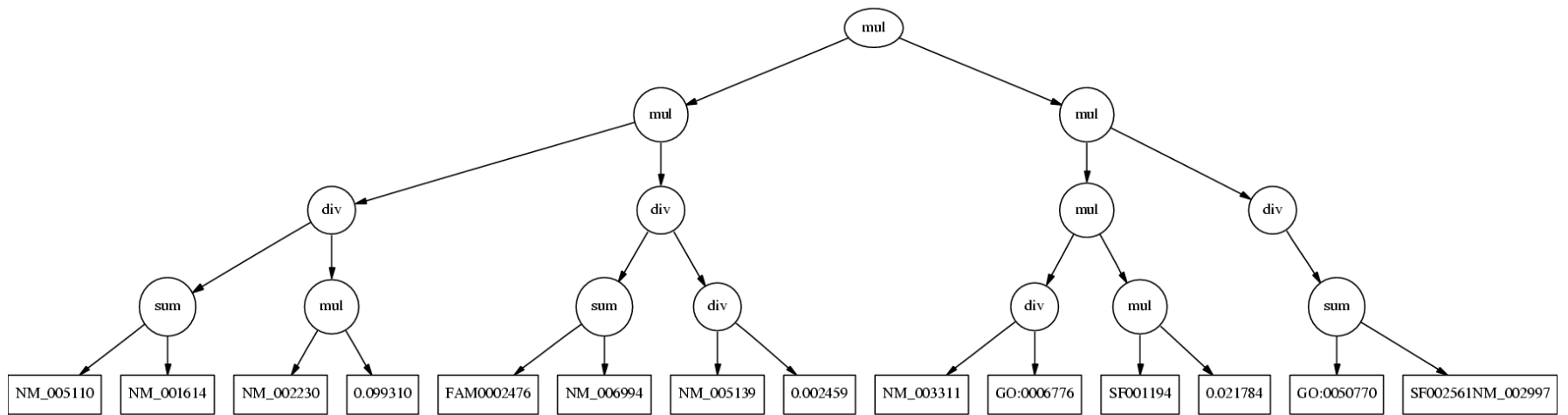


Figure 1. Best GP model for Lung Cancer

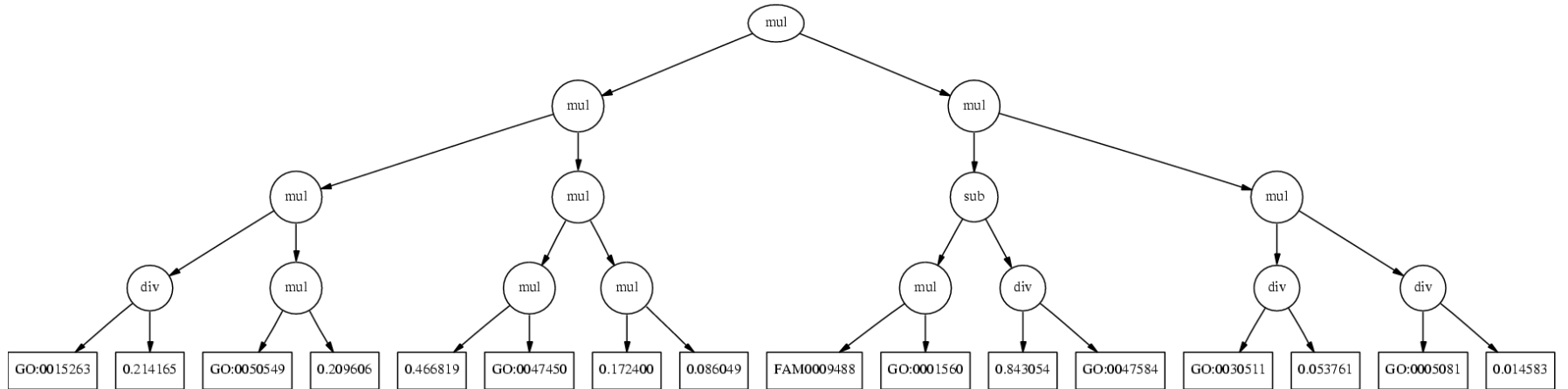


Figure 2. Best GP model for Prostate Cancer

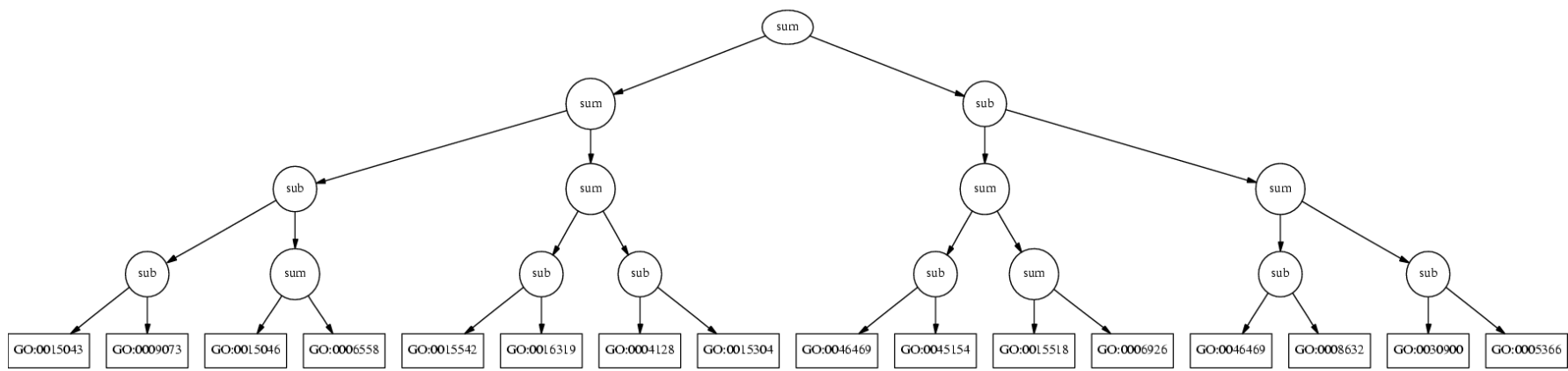


Figure 3. Best GP model for AML/ALL classification