FUNCTIONAL DISCRIMINATION OF GENE EXPRESSION PATTERNS IN TERMS OF THE GENE ONTOLOGY

LIVIU BADEA[1]
AI Lab, National Institute for Research and Development in Informatics
8-10 Averescu Blvd., Bucharest, Romania, badea@ici.ro

The ever-growing amount of experimental data in molecular biology and genetics requires its automated analysis, by employing sophisticated knowledge discovery tools. We use an Inductive Logic Programming (ILP) learner to induce functional discrimination rules between genes studied using microarrays and found to be differentially expressed in three recently discovered subtypes of adenocarcinoma of the lung. The discrimination rules involve functional annotations from the Proteome HumanPSD database in terms of the Gene Ontology, whose hierarchical structure is essential for this task. While most of the lower levels of gene expression data (pre)processing have been automated, our work can be seen as a step toward automating the higher level functional analysis of the data. We view our application not just as a prototypical example of applying more sophisticated machine learning techniques to the functional analysis of genes, but also as an incentive for developing increasingly more sophisticated functional annotations and ontologies, that can be automatically processed by such learning algorithms.

1 Introduction and motivation

The success of the various whole genome sequencing projects (including the Human Genome Project) has paved the way towards *'functional genomics'*, an undertaking whose main goal is to uncover the functions of the genes and their protein products.

Although painstaking experimental work has revealed essential details on the functions of *individual* genes and proteins, many of their most important effects depend on the orchestration of the activities of entire pathways, comprising numerous genes and proteins.

The major experimental breakthrough that allowed the measurement and analysis of the expression patterns of (tens of) thousands of genes simultaneously is the technology of microarrays and oligonucleotide arrays [1]. It is currently technically feasible to measure the expression levels of the entire set of genes of a living cell and compare such gene expression profiles (patterns) obtained under different experimental conditions. For example, different snapshots of the gene expression patterns in a developing organism can be used to study its development at a global genetic level. In other experiments, different samples of normal and diseased cells can be compared to reveal the genetic abnormalities causing the malady.

Many complex diseases, such as carcinomas, cannot be simply attributed to a single gene or a very small number of genes. They typically show complex disruptions of the gene expression patterns of normal cells. And very frequently, distinguishing between the different subgroups of the disease is essential for its

[1] This work was partially supported by the European Community projects SILK and ILPnet2.

correct treatment, each of the subgroups requiring a different therapy. Unfortunately, currently applied tests are not always capable to distinguish reliably between the various subtypes. The analysis of microarray gene expression data for various tissue samples has enabled researchers to determine gene expression profiles characteristic of the disease subtypes. The groups of genes involved in these genetic profiles are rather large and a deeper understanding of the functional distinction between the disease subtypes might help not only to select highly accurate 'genetic signatures' of the various subtypes, but hopefully also to select potential targets for design drugs.

Most current approaches to microarray data analysis use (supervised or unsupervised) clustering algorithms to deal with the numerical expression data [2]. The functional interpretation of the resulting clusters is however difficult in certain cases, although a large number of *functional annotations* for the genes involved is already available in public databases. This is probably due to the fact that most functional annotations are in free text and thus of little use to an automated analysis program. Despite this, significant efforts are under way to develop a unifying language[2] in terms of which to describe functions of genes, proteins, biological processes and cellular locations. The *Gene Ontology* (GO) [3] is a large, constantly growing hierarchy of molecular biology concepts, which can be used to annotate genes and proteins with functional information. Such annotations are currently available, e.g. in the Proteome HumanPSD database [4].

In this paper, we use Inductive Logic Programming (ILP) [9] to *induce discrimination rules between two recently discovered subtypes of adenocarcinoma* (AC) [5] *in terms of functional knowledge from the GO*. In other words, we would like to explain the differences between two AC subtypes in terms of the functions of the genes that are differentially expressed in these subtypes. As already hinted above, this should enable a deeper understanding of the mechanisms of the two AC subtypes.

2 Microarray data analysis of adenocarcinoma of the lung

Different experimental approaches were developed to study the gene expression profiles of entire genomes under precise conditions. *Microarrays* and *oligonucleotide chips* [1] allow the simultaneous measurement of the concentration of virtually every transcript in a cell, leading to a holistic understanding of cell physiology. In this paper, we are concentrating on the gene expression profiles produced in the study of Garber et al. of adenocarcinoma (AC) of the lung [5]. We have chosen this particular dataset, because unlike other similar diseases (like leukemias [6] or lymphomas [7] – which are more uniform, and thus much easier to analyse), carcinomas of the lung are quite heterogeneous and sometimes hard to distinguish histologically (under the light microscope). Adenocarcinomas (AC)

[2] In the current state of the art, it is more of a *vocabulary* rather than a full-fledged knowledge representation language.

comprise about 30% of all cases, but they are still heterogeneous: Garber et al. have distinguished at least three AC subgroups (AC1-3) with a large difference in survival rate between patients from group AC1 and those from group AC3 (in favour of AC1). In fact, the three subtypes of AC were determined by clustering microarray gene expression profiles (and not by looking at survival rates).

Unlike many diseases which show a rather localized disruption of the genetic profile of normal cells, carcinomas are much more heterogeneous and display more complex disruptions of the gene expression programs of the cells. Tracing these modifications back to the molecular level thus represents a challenging task.

Garber et al. have analysed the expression levels of 23,100 cDNA clones (corresponding to 17,108 unique genes, as defined by Unigene) in 73 different tissue samples (41 AC, 16 SCC, 5 LCLC, 5 SCLC, 5 normal and 1 fetal[3]). After an initial elimination of the low-quality measurements, Garber et al. have selected 918 cDNA clones (representing 835 unique genes) whose expression varied widely among the tissue samples[4] while being most similar within samples originating from the same patient. The resulting dataset taking the form of a 918×73 (gene×sample) matrix was subsequently median centered. Average linkage *hierarchical clustering* of the *samples* was performed using M. Eisen's CLUSTER and TREEVIEW programs [2].

Although hierarchical clustering is an unsupervised method that groups genes exclusively according to the degree of similarity in the patterns of gene expression, it almost perfectly distinguished the known types of lung carcinomas (AC, SCC, LCLC and SCLC) from each other as well as from the normal tissue samples. However, AC did not result as homogeneous cluster, but as 3 distinct sub-clusters: AC1, AC2 and AC3.

Garber et al. have searched for individual genes that were differentially expressed in the 3 AC subgroups and obtained a list of 149 genes (see the supplementary information to [5]). Now, although the distinction between the three AC subgroups may be due to many genes, it is unlikely that a list of 149 genes is the shortest, most comprehensive explanation of the distinction between the AC subgroups. Therefore, Garber et al. tried to subjectively select a subset (of the 149 differentially expressed genes) that would best explain the distinctions between the 3 AC subgroups (Fig.5. of [5]). This selection process consisted in the expert looking at each of the 149 genes and deciding whether it is interesting (as an explanation) or not. While this can be done in this way for one experiment and 149 genes, we argue that processing the ever-growing flood of available microarray data will have to be assisted in a more automatic manner, especially because the functional annotations available are becoming increasingly more sophisticated and hard to manage by a

[3] In this paper, we concentrate on AC – the most heterogeneous, and thus the most interesting group.

[4] A gene whose expression level is constant across all tissue samples is obviously not directly involved in the phenomenon under study.

person’s working memory. For example, Garber et al. used functional annotations in the form of text descriptions, which can be easily obtained from various molecular biology and genetics databases (such as NCBI Entrez, Expasy/SwissProt etc.).

Obviously, such text descriptions cannot be used by an automated system. However, we are fortunate that more sophisticated annotations are already available, for example in the Proteome HumanPSD database [4]. In the following we show how such functional annotations can be used to *automatically induce discrimination rules* between group AC3 (with a significantly lower survival rate) and the other AC subgroups (AC1 and AC2, showing a much better prognosis).

This should help in selecting a functional explanation of the differences between the AC subgroups in a semi-automatic manner (and thereby speed up microarray data analysis for larger datasets).

3 Functional discrimination of genes using ILP

The HumanPSD Proteome database contains, among others, *Gene Ontology* annotations of a large number of human genes (we could find in Proteome 459 of the 918 cDNA clones of Garber et al.). Functional annotations are however of little use without some sort of background knowledge that relates the various concepts involved.

The *Gene Ontology* (GO) [3] is a hierarchy[7] of concepts used in molecular biology and genetics, which can be employed as *background knowledge* for an inductive learner. (The ontology is organized according to *‘Molecular Function’, ‘Biological Process’* and *‘Cellular Component’*, for which we have annotations in the Proteome database.)

Inductive Logic Programming (ILP, or Relational Learning) [9] deals with learning logic programs from positive and negative examples with respect to some existing *background knowledge*. ILP generealizes other machine learning approaches not just by dealing with more complex hypotheses (first order logic programs), but also by taking into account a given background knowledge.

In the following, we show how an ILP learner can be used to induce functional discrimination rules between the genes that correlate well with the AC3-AC1,2 distinction and those that do not. The discrimination rules will involve GO concepts (either GO annotations from the Proteome database, or their generalisations in the GO hierarchy, used as background knowledge).

3.1 Setting up the learning problem: positive and negative examples

We have used the dataset from [5] restricted to the samples classified in AC.

[7] In GO, there are two main types of relationships between concepts: *‘isa’* (inheritance) and *‘part-of’*.

The selection of positive examples (genes correlated with the AC3-AC1,2 distinction) was performed using a nonparametric t-test with a (very strict) P value cutoff of 0.0005, while all genes with $P \geq P_{neg} = 0.3$ were considered negative examples. Finally, only the examples with a counterpart in the HumanPSD database (thus having a GO annotation) were kept. (We thus selected 39 positive and 87 negative examples.)

3.2 The background knowledge

The GO annotations of the examples selected above were used as background knowledge.[8] For instance, the GO annotation of example *'CDKN2A'* (cyclin-dependent kinase inhibitor 2A) is:

```
'Molecular Function'('CDKN2A','cyclin-dependent protein kinase inhibitor').
'Molecular Function'('CDKN2A','tumor suppressor').
'Biological Process'('CDKN2A','cell cycle checkpoint').
'Biological Process'('CDKN2A','regulation of CDK activity').
'Biological Process'('CDKN2A','oncogenesis').
'Biological Process'('CDKN2A','cell cycle arrest').
'Cellular Component'('CDKN2A','nucleus').
```

As many of the annotations are quite specific, they may be unable to cover more than one example. To allow non-trivial generalisations, we added as background knowledge an encoding of the GO hierarchy in the form of Prolog rules of the form

```
go (‘nucleic acid binding’, X) :– go(‘DNA binding’, X).
go(‘DNA binding’, X) :– go(‘DNA replication factor’, X).
```

For example, the first rule states that annotation *X* is in the GO category *'nucleic acid binding'* if it is in category *'DNA binding'*. Of course, the leafs of the GO hierarchy verify a fixpoint clause:

```
go(X, X).
```

3.3 The hypotheses language

Using the ILP learner Progol [8], we looked for hypotheses with (an arbitrary number of) *function* literals in the body (with the second argument instantiated to a constant – in this case a GO annotation), such as:

```
target(Gene) :– function(Gene, ‘calcium binding’), function(Gene, ‘protein binding’).
```

where

```
function(Gene, GOterm) :– ( 'Molecular Function'(Gene,X) ;
```

[8] Unfortunately, the Proteome database contains numerous mistakes, ranging from typos to inconsistencies with the current version of the GO. All of these mistakes were corrected by hand.

```
                                        'Biological Process'(Gene,X)    ;
                                        'Cellular Component'(Gene,X)    ),  go(GOterm, X).
```

3.4 Obtaining all "best" discriminating hypotheses

For each positive example, Progol's heuristic search selects *only one* "best" clause[9] discriminating this positive example from the negative examples, although there might be several clauses with the same covering and the same heuristic estimate. For instance, the positive example *target('S100P')* admits the following alternative "best" discriminating clauses:

```
target(G) :– function(G,'calcium binding'), function(G,'protein binding').
target(G) :– function(G,'calcium binding'), function(G,'cell-cell signaling').
target(G) :– function(G,'calcium binding'), function(G,'cell communication').
target(G) :– function(G,'cell-cell signaling'), function(G,'ligand binding or carrier').
target(G) :– function(G,'protein binding'), function(G,'cell-cell signaling').
target(G) :– function(G,'protein binding'), function(G,'cell communication').
```

Returning just one, as Progol does, may not be enough in our application, where

- the number of negative examples may be too small to invalidate the wrong alternatives (candidate hypotheses), and/or
- alternative viewpoints on the distinction between classes are highly desirable.

Since Progol conducts a *complete* admissible search, it was relatively straightforward to extract *all* alternative hypotheses (having the same covering and the same heuristic estimate as the hypothesis selected by Progol). More precisely, we return for each seed example a list of alternative hypotheses, each with the set of positive examples covered (in general, containing also other examples besides the seed example). For example:

Alternative hypotheses	**Positive examples covered**
target(G) :– function(G,'substrate-bound cell migration')	VEGFC
target(G) :– function(G,'lymph gland development')	VEGFC
target(G) :– function(G,'cell migration')	VEGFC

3.5 Results

Table 1 (on the last page) summarizes the results obtained. Each line represents a discriminating hypothesis and the positive examples covered by it (none of the hypotheses cover negative examples). Lines in the same group represent alternative hypotheses (for example, group 18 contains the alternative hypotheses from the previous section, covering VEGFC). Note that we do not return only the *most*

[9] Best according to a covering-based heuristic.

general alternative hypotheses (such as ‘transcription’ from group 3), since the available example set is small (as well as inherently incomplete) and a general hypothesis like ‘transcription’ may be too general to be informative. (Generalization is also controlled via the negative examples.) Nor do we keep just the *most specific* alternative hypotheses, since these may be too specific. It is ultimately up to the molecular biologist to choose the most significant hypothesis from the set of alternative ones. (The GO hierarchy is “shallow”, thereby making this task easy.)

We found functional discrimination rules for virtually all the genes discussed in Garber et al. [5] in relation to AC [11] and all of them were found to be relevant by an expert molecular biologist.[12] (The descriptions are biologically informative, although a bit too general – probably due to the quality of the initial annotations). The most significant ones are:

- the *'cell cycle control'* genes, especially the cyclin-dependent kinase inhibitor CDKN2A (p16) and the polo-like serine/threonine kinase (PLK),[13] which are both up-regulated in AC3.
- the *'cystein-type endopeptidase'* cathepsin L (CTSL), involved in (extracellular) proteolysis, which is also up-regulated in AC3 with respect to AC1,2.
- the *'signal transduction'* and *'growth factor'* Dickkopf (DKK1), which is known to contribute to neoplastic processes (it may play a key role in the transition from an epithelial to a mesenchymal phenotype - which is significant since AC involve epithelial cells).
- the *'transporter', 'membrane'* solute carrier protein SLC7A5, known to be closely linked to cellular activation and division (the transcripts of the SLC7A5 gene are rapidly induced and degraded, which is unusual for an integral plasma membrane protein and resembles more closely the kinetic seen for protooncogenes and lymphokines in T cells. It is thought to be up-regulated to support the high protein synthesis for cell growth and activation.)
- in the *'tumor antigen'* category, the carcinoembryonic antigen-related cell adhesion molecule CEACAM1 is involved in tumor angiogenesis. (CEACAM1 is down-regulated in AC3 with respect to AC1,2.)

Note that the discrimination rules induced strike a reasonable balance between generality and specificity: they are specific enough to be informative, but also general enough to explain occasional groups of genes that can be distinguished (from the negative examples) by certain *common* functions.

[11] An exception was the thyroid transcription factor TITF1, which did not have a Proteome annotation.

[12] The evaluation of the resulting hypotheses has to be done by human analysis rather than more traditional methods employed in machine learning, mostly due to the small number of genes under study and since we are aiming at automating the last (result interpretation) phase of gene expression analysis, whose validation requires biological experiments.

[13] which is not mentioned in [5] but shows significant correlation with the AC3-AC1,2 distinction.

There was one important functional distinction that could not be discovered due to the incompleteness of the Proteome GO annotation, namely *'angiogenesis'* (involving for example the vascular endothelial growth factor C, VEGFC, which nevertheless was distinguished as being related to *'substrate-bound cell migration'*).

Note that none of the discriminatory descriptions induced by Progol contained general concepts (e.g. ‘oncogenesis’), which are common of all the AC subgroups. Discriminatory descriptions are thus informative: they tell us what distinguishes the genes expressed differentially in AC3 and AC1,2 from the others.

The P_{neg} = 0.3 t-test margin for negative examples may be somewhat arbitrary. Too few negative examples would produce too general and thus useless discriminators. On the other hand, too many negative examples combined with a very coarse grained ontology like GO, may produce very specific discriminators or even none at all.

Still, our experiments showed a surprising *robustness* with respect to the choice of the P_{neg} margin: the discriminators for P_{neg} = 0.3 and 0.1 respectively were very similar (the latter are not shown due to space limitations).

Also, the fact that almost all alternative hypotheses (having the same covering and heuristic estimate as the ‘best’ hypothesis chosen by Progol) *covered the same set of genes* increases our confidence that the distinctions made by the discriminators reflect true functional distinctions rather than contingencies due to a biased initial functional annotation of the genes in the Proteome database. Indeed, for P_{neg} = 0.1 and the seed PLK, Progol learns 3 alternative hypotheses covering different groups of genes:

protein serine/threonine kinase, *biological_process*	CIT,PLK
cell cycle control, enzyme	BUB1B,DUSP4,PLK
cell cycle control, *molecular_function*	BUB1B,CDKN2A,DUSP4,PLK

However 2 of the 3 hypotheses (marked in *italics*) involve two very general GO concepts (‘*biological_process’* and ‘*molecular_function’*) and the distinctions they make are most probably annotation artifacts – in this case some negative examples (which would otherwise be covered) simply happen to lack annotations for ‘*biological_process’* and ‘*molecular_function’*.

The uniformity of the coverings of the other groups of alternative definitions strongly suggests that the distinctions are true functional distinctions and not annotation artifacts (they were also confirmed by an expert).

4 Towards more complex annotations

Most current approaches to functional analysis of experimental data in genetics assign genes to one of a number of typically disjoint functional categories (which can be drawn from the GO, but without using the GO hierarchy). However, while such approaches are useful for obtaining aggregate overviews of whole genomes, they are less appropriate for more fine-grained functional discrimination of groups of genes, for which a predefined set of functional categories may turn out to be too coarse grained.

The hierarchical structure of GO, as well as the presence of negative examples proved to be essential in our application. It is encouraging that the discriminations obtained are biologically sensible – this heavily relies on the GO and the HumanPSD annotations. But this also automatically prompts the question of whether more sophisticated knowledge representation formalisms, such as Description Logics (DL) [13] might allow even more precise functional distinctions to be made.

The discriminatory hypotheses induced typically involve (combinations of) very specific GO properties of the target genes, in order to avoid covering any negative examples. However, the learning algorithm tries not only to avoid negative examples, but also to produce "short" hypotheses. In settings like ours involving complex disruptions of gene networks, we expect to be able to obtain more general descriptions of such disrupted groups of genes (in terms of concepts placed higher in the GO hierarchy). This was the case for example with the '*cell cycle control*' genes, but this *generalization* capability of GO was limited to a certain extent by the *fixed* hierarchy.

A DL may allow an "on-the-fly" construction of concepts, rather than relying on a fixed hierarchy. Thus, we wouldn't need to explicitly record in the ontology all generalizations of existing concepts. For example, the current GO contains not just specific concepts like '*cyclin-dependent protein kinase inhibitor*' or '*transmembrane receptor protein tyrosine kinase activator*', but also their generalization '*kinase regulator*'. On the other hand, a DL may take advantage of the intrinsic composite nature of the concepts above and represent them as $\exists$*inhibits.CDK* and $\exists$*activates.TRPTK*. Their generalization need not be explicitly represented, since it can be computed by taking the *least general generalization* ("least common subsumer" in DL terminology) $\exists$*regulates.kinase* of the two concepts above.

Thus, we think there are two types of possible extensions of the GO that would enhance its utility in functional analyses:

- allowing a more sophisticated knowledge representation language (like DL) for describing GO concepts (this would generalize the hierarchical structure of GO and allow more refined concept generalizations)
- integrating GO with metabolic, regulatory and cell signalling pathway databases (which would allow more precise causal reasoning – for example, determining possible primary causes for complex genetic disruption profiles).

Already the most basic background knowledge on functional annotations, which involves *hierarchies of concepts* (as in GO, where the main type of relational information is in the form of *inheritance* relationships), is not directly treatable by propositional learners like C4.5 [10], but could be dealt with our approach.

However, as the degree of sophistication of functional annotations will increase in the near future, it is important to know whether our approach will still be applicable in such more complex settings. (C4.5 will definitely be out of the question, due to its inherent difficulty of constructing hypotheses involving relational knowledge.) We argue that our approach is indeed applicable in such extended settings, even in the presence of arbitrarily more sophisticated relational annotations, such as:

- *'part-of'* relationships (already present in the Gene Ontology)
- causal relationships between genes/proteins such as *activates(G1,G2)* or *inhibits(G1,G2)* (such types of knowledge exist to a certain degree[14] in genetic regulatory and metabolic pathway databases, such as KEGG).

For example, hypotheses involving *'part-of'*, such as[15]

```
target(Gene)  :–  'Biological Process'(Gene, 'cell cycle control'),
                  'Cellular Component'(Gene, X),
                  active(G1),                                   %gene G1 is active
                  'Cellular Component'(G1, X1), part_of(X, X1).
```

can only be induced using a true relational learner.

5 Discussion

The need for large-scale functional analysis of genomic and proteomic data requires increasingly more sophisticated knowledge discovery tools, able to take advantage of complex functional annotations as well as existing background knowledge on those. The functional discrimination of genes using the GO hierarchy seems a natural exploitation of the knowledge available in GO, but, as far as we know, hasn't been tried before. ([11] also uses the GO, but for inducing signatures of temporal gene expression profiles rather than functional discrimination.)

We view our application not just as a prototypical example of applying more sophisticated machine learning techniques to gene expression analysis, but also as an incentive for developing increasingly more sophisticated functional annotations and ontologies, that can be automatically processed by such learning algorithms.

With the increasing efforts towards more complete functional annotations of genes, proteins and their pathways, the ability to learn complex *relational* descriptions will become even more important, making sophisticated learning techniques, such as ILP, indispensable.

Several data- and knowledge bases of biological networks and pathways, designed with the aim of allowing automated processing, are currently under development. (As opposed to existing databases which were designed mainly for human users.) We argue that a tight feedback should exist between the annotation effort, the experimental conditions [14], but also the knowledge discovery tools to be used on these annotations.

[14] This knowledge is currently very fragmentary and also not well integrated with GO, so we couldn't use it yet in our experiments.

[15] In English: "a gene is a target if it is involved in cell cycle control and it is localized in a sub-component of a cellular component in which another gene is active".

In settings like ours, most of the lower levels of data preprocessing and clustering have been automated. With the ever-growing amount of such expression data available, their high level functional analysis seems to be the major bottleneck, which can be appropriately addressed using more sophisticated machine learning techniques, like ILP, able to deal with complex background knowledge.

The specificities and complexities of functional genomics may require modifying existing learning algorithms, as we did in the case of constructing all alternative "best" hypotheses covering a given seed example. This modification turned out to be crucial in our application (together with the completeness of the hypotheses search).[16]

References

1. Basset D.E., M.B. Eisen, M.S. Boguski. Gene expression informatics – it's all in your mine, Nature Genetics supplement, Vol. 21, Jan. 1999, 51-55.
2. Eisen M.B., P.T. Spellman, P.O. Brown, D. Botstein. Cluster analysis and display of genome-wide expression patterns, Proc.Natl. Acad. Sci., Vol.95, 14863-14868, Dec.1998.
3. Gene Ontology Consortium. Gene Ontology: tool for the unification of biology. Nature Genet. 25:25-29, 2000.
4. http://www.incyte.com/proteome.
5. Garber M.E. et al. Diversity of gene expression in adenocarcinoma of the lung. Proc. Natl. Acad. Sci., Vol. 98, 13784-13789, November 20, 2001.
6. Golub T.R. et al. Molecular classification of cancer: class discovery and class prediction by gene expression monitoring, Science, Vol. 286, 15 October 1999, 531-537.
7. Alizadeh A.A. et al. Distinct types of diffuse large B-cell lymphoma identified by gene expression profiling, Nature, Vol. 403, 503-511, Feb. 2000.
8. Muggleton S. Inverse entailment and Progol. New Gen. Computing J., 13:245-286, 1995.
9. S.H. Nienhuys-Cheng, R. de Wolf. Foundations of Inductive Logic Programming, Vol. 1228 of Lecture Notes in Artificial Intelligence. Springer-Verlag, 1997.
10. Quinlan, J.R. C4.5: Programs for Machine Learning, Morgan Kaufmann, 1993.
11. T.R. Hvidsten, J. Komorowski, A.K. Sandvik, and A. Lægreid. Predicting Gene Function from Gene Expressions and Ontologies, PSB 6:299-310 (2001).
12. Liviu Badea, S.H. Nienhuys-Cheng. A Refinement Operator for Description Logics. Proc. Intl.Conf. on Inductive Logic Programming ILP-2000, LNAI 1866, 40-59, Springer.
13. F. Baader, D. McGuinness, D. Nardi, P. P. Schneider. The Description Logic Handbook, Cambridge University Press, 2003.

[16] Indeed, without it, the learner may return some spurious discriminators without any means of discerning these from the "real" ones. Such spurious discriminators were in fact obtained when we tried a propositional decision tree learner (C4.5rules) on a preprocessed example set taking into account the inheritance relationships in the Gene Ontology. The inappropriateness of C4.5 in this case may also be due to the fact that we are not really dealing with a classification problem, but more with a descriptive induction problem. (C4.5rules also produces rules activated by the *absence* of annotations. However, due to the inherent incompleteness of the GO annotations, we need to restrict ourselves to positive information only and this seems difficult to do using C4.5.)

14. Microarray Gene Expression Data (MGED) Society Ontology Working Group http://www.cbil.upenn.edu/Ontology/MGED_ontology.html.

Discriminatory hypotheses	Positive examples covered
calcium binding, protein binding	S100P
calcium binding, cell-cell signaling	S100P
calcium binding, cell communication	S100P
cell-cell signaling, ligand binding or carrier	S100P
protein binding, cell-cell signaling	S100P
protein binding, cell communication	S100P
integral plasma membrane proteoglycan	ICAM1,PTK7
transcription, from Pol II promoter	IRX5,TRIM29
transcription, DNA-dependent	IRX5,TRIM29
transcription	IRX5,TRIM29
tumor antigen	CEACAM1,STEAP
cell surface antigen	CEACAM1,STEAP
furin	PACE
cell cycle control	BUB1B,CDKN2A,DUSP4,PLK
6-phosphofructokinase	PFKP
glucose metabolism	PFKP
hexose metabolism	PFKP
monosaccharide metabolism	PFKP
carbohydrate metabolism	PFKP
transporter, membrane	ABCC2,AQP8,SLC2A1,SLC7A5,STEAP
RHO small monomeric GTPase	ARHE
actin cytoskeleton reorganization	ARHE
peripheral plasma membrane protein	ARHE
small monomeric GTPase	ARHE
protein kinase cascade	CIT,DUSP4,STAT4
monooxygenase	CYP24
cysteine-type peptidase	CTSH,CTSL
lysosomal cysteine-type endopeptidase	CTSH,CTSL
cysteine-type endopeptidase	CTSH,CTSL
transcription co-repressor	ID3
transcription regulation, from Pol II promoter	ID3,STAT4
transcription regulation	ID3,STAT4
respiration	SFTPC
membrane, ligand	ICAM4,SCYD1
plasma membrane, ligand	ICAM4,SCYD1
RNA helicase	DBY
adenosinetriphosphatase	DBY
helicase	DBY
signal transduction, growth factor	DKK1
substrate-bound cell migration	VEGFC
lymph gland development	VEGFC
cell migration	VEGFC
excretion	UNC13
basement membrane	LAD1

Table 1. Hypotheses discriminating genes differentially expressed in classes AC3 and AC1,2.