Pacific Symposium on Biocomputing 6:299-310 (2001)

PREDICTING GENE FUNCTION FROM GENE EXPRESSIONS AND ONTOLOGIES

T.R. HVIDSTEN, J. KOMOROWSKI[a]
Knowledge Systems Group, Department of Information and Computer Science, Norwegian University of Science and Technology, 7491 Trondheim, Norway

A.K. SANDVIK, A. LÆGREID
Department of Physiology and Biomedical Engineering, Norwegian University of Science and Technology, 7489 Trondheim, Norway

We introduce a methodology for inducing predictive rule models for functional classification of gene expressions from microarray hybridisation experiments. The basic learning method is the rough set framework for rule induction. The methodology is different from the commonly used unsupervised clustering approaches in that it exploits background knowledge of gene function in a supervised manner. Genes are annotated using Ashburner's Gene Ontology and the functional classes used for learning are mined from these annotations. From the original expression data, we extract a set of biologically meaningful features that are used for learning. A rule model is induced from the data described in terms of these features. Its predictive quality is fine-tuned via cross-validation on subsets of the known genes prior to classification of unknown genes. The predictive and descriptive quality of such a rule model is demonstrated on the fibroblast serum response data previously analysed by Iyer et. al. Our analysis shows that the rules are capable of representing the complex relationship between gene expressions and function, and that it is possible to put forward high quality hypotheses about the function of unknown genes.

1 Introduction

Functional genomics studies gene function on a large scale by conducting parallel analysis of gene expression for a large number of genes [1]. This research is a natural successor to the ongoing genome sequencing efforts such as, for example, the Human Genome Project, and is made possible by the microarray technology [2] that gives a view into the organisation of molecular cellular life through quantitative measurements of gene expression levels. The complexity of molecular biology is reflected by the huge data sets generated from microarray experiments. This complexity enforces an extensive use of computers to store and analyse expression data.

Automated gene expression analysis is based on the assumption that genes with similar functions have similar expression profiles in cells [3]. This is utilised by inductive learning methods that predict the function of genes that have

[a] Corresponding author

an unknown function (*unknown genes*), from their expression-similarity with genes with a known function (*known genes*). Both *unsupervised* and *supervised* inductive learning may be used for this purpose. As of today, it seems that the former dominates the state of the art (e.g. Eisen et. al.[3] and Iyer et. al.[4]), even though applications of the latter do exist (e.g. Brown et. al.[5]). Unsupervised methods are exemplified by clustering algorithms, with hierarchical clustering[6] and self-organising maps[7] being the most popular ones in functional genomics. These methods use a similarity measure to cluster genes with similar expression profiles. Existing biological knowledge, in terms of known gene–function relations, is then used to validate these clusters and, if justified, to put forward hypotheses about the function of unknown genes.

Clustering analysis is purely syntactical in the sense that it does not take advantage of the existing knowledge in the learning process. Instead, this knowledge is introduced after the induction step (i.e. in the validation step) and then often in a highly manual fashion. There are few or no objective criteria that may be used to evaluate the predictive strength of clusters. Consequently, there are limited possibilities to compare the results of different clustering approaches or to estimate the quality of the classification of unknown genes. Supervised learning is an alternative to unsupervised learning that takes a fundamentally different approach to the use of existing knowledge. Given a set of known genes and their functions, a supervised learning algorithm automatically learns a definition of the functions exemplified by the known genes. This class definition is called a *predictive model*, since it may be used to predict the function of genes and, most importantly, the function of unknown genes. Its predictive quality can be tested on a small subset of the known genes, while its descriptive value can be inspected by biological experts.

In this paper we present a methodology for inducing predictive models for functional genomics. Our approach is fully implemented in the ROSETTA system[8,9], a publicly available toolkit for data mining and knowledge discovery[10] using rough sets[11], and is tested on the fibroblast serum response data[12] previously analysed by Iyer et. al.[4]. Our method basically consists of four steps. Genes are first annotated using Ashburner's Gene Ontology[13] and the functional classes which we want to learn are mined from these annotations. From the original expression data, we then extract a set of biologically meaningful features that are used for learning. A rule model is induced from the expression data described in terms of these features using the rough set framework for rule induction. The rule model is evaluated and finally applied to classify the unknown genes.

2 Method

The basic vehicles for data representation in rough set theory[11] are *information systems* and *decision systems*. An information system is a pair $\mathcal{A} = (U, A)$ where U is a non-empty finite set of objects called the universe and A is a non-empty finite set of attributes such that $a : U \to V_A$ for every $a \in A$. In our context, an information system constitutes a table where each row represents a gene and each column a measurement of this gene's expression. A *decision system* is any information system of the form $\mathcal{A} = (U, A \cup \{d\})$, where $d \notin A$ is the decision attribute. In our context, the functions of the known genes constitute the decision attribute.

2.1 Annotating genes and mining functional classes from ontologies

Determining the functional classes from which we want to learn is not entirely straightforward. To understand this we should have in mind the obvious fact that all genes in principle carry out a unique function. On the other hand, one could also argue that all genes carry out the same function; the synthesis of proteins. Consequently, we need to select a certain generality level in which to view the biological system, that is, we cannot learn from classes with only one member and there is no point in learning only one class. The problem may be solved by viewing functions as a hierarchical structure, a *gene ontology*. A gene ontology may be seen as a tree, where parent nodes give a more general description of a gene than their children. The leaf nodes give an accurate description of each gene. The information associated with a gene as a result of its location in the ontology is what we will refer to as a gene *annotation*.

Each known gene in a microarray experiment can be annotated by finding one or more nodes in the ontology that best represent the existing knowledge about its function(s). From the annotations of all the known genes we then find a set of functional classes such that each class is as specific as possible without including too few training examples (genes). The functional classes possessing this property may easily be retrieved from the gene ontology by traversing the tree bottom-up: starting from the leaf nodes and stopping when reaching nodes that within their subtrees contain at least λ genes. These genes are now labelled with the annotation corresponding to the root nodes of the subtrees and constitute functional classes.

Given an information system $\mathcal{M} = (U, A)$ containing microarray experiment measurements, the annotations retrieved from the ontology extend this information system into a decision system $\mathcal{M} = (U^*, A \cup \{d\})$. The cardinality of the image $d(U) = \{k \mid d(x) = k \text{ and } x \in U\}$ is called the rank of d and is de-

noted $r(d)$. The decision attribute d determines a partition $\{X_{\mathcal{A}}^1, X_{\mathcal{A}}^2, ..., X_{\mathcal{A}}^{r(d)}\}$ of the universe, where $X_{\mathcal{A}}^k = \{x \in U^* \mid d(x) = v_d^k\}$ for $1 \leq k \leq r(d)$ are the functional classes that we want to learn. Note that U^* is an extension of U such that each gene $x \in U$ is represented in U^* by one object for each annotation of x.

2.2 Extracting useful features for learning

Essential to the quality of the predictive model is, of course, not only the quality of the measured expressions and the collected annotations, but also the representation of the training examples. Even if the measurements and annotations had been perfect, one would still have to expect a poor classification result should the training examples be represented with the wrong features.

We propose a preprocessing strategy for time series that consider only significant changes in expression levels over time sub-intervals. This shift of focus from a quantitative representation to a qualitative one has several attractive features. Semantically, the qualitative representation holds the important features that could indicate whether two genes are related or not, i.e. significant changes in expression levels over sub-intervals. At the same time it is sufficiently general so as not to be significantly affected by noise. Furthermore, we represent each gene in the decision system $\mathcal{M}$ relative to its annotation by emphasising properties which are common to other genes with the same annotation. Consequently, we escape the problem of having identical objects belonging to different classes; this is otherwise a serious problem when learning gene functions since most genes are involved in more than one function. Genes showing no similarity to other genes with the same annotation are discarded under the assumption that they have misleading or irrelevant annotations.

Given a decision system of expression data and annotations $\mathcal{M} = (U^*, A \cup \{d\})$, we define significant change in expression levels by means of a set of templates T. A template $t \in T$ is a prototypical pattern of expression level and can be matched with a gene $x \in U^*$ over all possible sub-intervals $I = \{(a_i, a_j) \mid 1 \leq i < j \leq |A| \text{ and } a_i, a_j \in A\}$. A cluster C_t^i includes all genes matching template t in sub-interval i such that $C_t^i = \{x \in U^* \mid i \in I \text{ and } t \in T \text{ and } \text{match}(x, t, i)\}$. On the basis of these definitions we define a new decision system of learning examples $\mathcal{L} = (U^\dagger, I \cup \{d\})$, where $U^\dagger = \{x \in U^* \mid \exists i \in I \text{ and } i(x) \neq \emptyset\}$ and $i(x) = \{t \in T \mid x \in C_t^i \text{ and } |C_t^i|_{d(x)} > \varepsilon * |U^*|_{d(x)}\}$. Here, ε is a value in the interval $[0, 1]$, and $|C_t^i|_{d(x)}$ and $|U^*|_{d(x)}$ are the number of genes with the same annotation as x in cluster C_t^i and in universe U^*, respectively.

The strategy described above deals with the problems of multiple annotations per gene and annotations that are correct according to the literature,

but which may not be relevant in a specific biological setting. Together with the strict requirements of supervised learning for a structural representation of background (validation) knowledge, these two existing problems may be the main reasons why almost every approach to computational functional genomics is unsupervised rather than supervised. That is, unsupervised methods escape these problems since they do not require a strict formalisation of background knowledge, and since knowledge is applied manually after the induction step.

2.3 Inducing and testing a predictive model

Pawlak's rough set theory [11] constitutes a mathematically sound framework for inducing minimal decision rules from data. We use this framework to induce a predictive rule model from the decision system $\mathcal{L}$.

The rough set framework for rule induction Central to the notion of rough sets is the concept of *indiscernibility*. Given a decision system $\mathcal{L} = (U^\dagger, I \cup \{d\})$, we define a relation $IND_\mathcal{L}(I, x, d) = \{y \in U^\dagger \mid (d(x) = d(y))$ or $(\forall i \in I\ (i(x) = i(y)$ or $i(x) = \emptyset))\}$ called the *indiscernibility relation*. It holds all objects (genes in our case) which either have the same annotation as x or for all sub-intervals are members of the same clusters as x. *Generalised decision* $\delta_I(x) = \{i \mid \exists y \in U^\dagger\ y \in IND_\mathcal{L}(I, x, d)$ and $d(y) = i\}$ defines all annotations associated with genes being indiscernible from x.

From the definition of indiscernibility we derive for each gene $x \in U^\dagger$ the set of *reducts* $RED_\mathcal{L}(x, d)$ to be the minimal sets of attributes $B \subseteq I$ such that $IND_\mathcal{L}(B, x, d) = IND_\mathcal{L}(I, x, d)$. Finding the set of minimal reducts is NP-hard [14], however, there are heuristics that compute sufficiently many reducts in an acceptable time. Since real-world data almost always is polluted with noise and since it only takes one noisy object to alter the indiscernibility relation, methods finding approximate reducts that reveal the underlying, general pattern in the data have been developed. Two such approaches are *dynamic reducts* [15] and *α-reducts* [16].

Reducts serve the purpose of synthesising minimal decision rules of the form $\alpha \rightarrow \beta$. The most fundamental building block for assembling rules is called a *descriptor*. A descriptor is an expression $i = i(x)$, where $i \in I$. Descriptors may be combined in a recursive manner in order to form more complex formulae such as $F_I(x) = \bigwedge_{i \in I}(i = i(x))$ and $G_I(x) = \bigvee_{j \in \delta_I(x)}(d = j)$. Minimum decision rules from the decision system $\mathcal{L}$ constitute the set $RUL_\mathcal{L} = \bigcup_{x \in U^\dagger} \{F_B(x) \rightarrow G_B(x) \mid B \in RED_\mathcal{L}(x, d)\}$. For a detailed introduction to rough sets see Komorowski et. al. [17].

Testing strategy A classifier is evaluated by dividing the set of available training examples into a *training set* and a *test set*. A systematic approach to testing is *k-fold cross validation* where the set of training examples is divided into k subsets in which each of these subsets is used once for testing and $k-1$ times for training. Since the training examples in the decision system $\mathcal{L}$ are represented relative to its decision class (functional class), we need to transform each subsystem of $\mathcal{L}$ acting as a test set such that all its genes are represented by all matching templates in all possible sub-intervals. This ensures unbiased test sets and hence the classification quality measured on these test sets can be directly used as an estimate of the quality of the classification of the unknown genes.

Model evaluation The induced rules constitute a classifier or a predictive model denoted κ. When applied to a gene $x \in U^{\dagger}$ in a test set, this classifier assigns a classification $\hat{d}_{\kappa}(x)$ to x. $d(x)$ is assumed to be the true actual classification of x. We will only consider binary classifiers in which $\hat{d}_{\kappa}(x)$ takes the form $\hat{d} : U \stackrel{\Phi}{\rightarrow} [0,1] \stackrel{\theta_\tau}{\rightarrow} \{0,1\}$. In most cases, when we are faced with more than two functional classes, we choose one fixed class at a time and classify unseen objects as either belonging to this class or to one of the other classes. Hence, $\Phi(x)$ is the certainty of a rule model that x belongs to the fixed class, while $\theta_\tau(x)$ is a simple threshold function that evaluates to 0 if $\theta_\tau(x) < \tau$, and 1 otherwise. $\Phi(x)$ is realised by a voting procedure that lets each matching rule cast a number of votes in favour of the decision class the rule indicates.

A frequently used graphical representation of classifier performance is the *receiver operating characteristic* (ROC) curve[18]. This curve results from plotting the true positive rate (sensitivity) against the false positive rate (1 - specificity) while letting τ vary across the full spectrum of possible values in $[0,1]$. The ROC curve is commonly collapsed into one value by computing the *area under the ROC curve* denoted AUC. This value is attractive since it, unlike for example accuracy, is independent of both error costs and prevalence of classes.

2.4 Predicting the function of unknown genes

When classifying unknown genes one needs to select a fixed threshold τ for each fixed functional class. This value may be selected by minimising the function $v * (1 - \text{specificity}(\tau)) + (1 - \text{sensitivity}(\tau))$ over the test set. Hence, if $v > 1$ the cost of false positives is weighted higher than the cost of false negatives. This makes sense, since we want as few wrong hypothesis about gene function as possible when classifying unknown genes.

The unknown genes are predicted using the rules $RUL_{\mathcal{L}}$ induced from the

whole set of training examples in $\mathcal{L}$. Let $\Phi_i(x)$ be the certainty of a rule model that the unknown gene x belongs to functional class i. The set of predicted functions is now $PRED_{RUL_{\mathcal{L}}}(x) = \{i \mid \Phi_i(x) > \tau_i\}$. τ_i is the threshold selected for the functional class i as described above.

3 Results

Iyer et al. [4] studied the human fibroblast's response to serum. These cells have a pivotal structural role in connective tissue and in important processes such as wound healing. The temporal changes in mRNA level of 8613 human genes were measured at 12 time points in the time period between 0 minutes and 24 hours after serum stimulation. A subset of 517 genes whose expression changed substantially in response to serum was selected for further analysis.

Ashburner's Gene Ontology [13] is under development by experts on the biology of fruit fly (*Drosophila melanogaster*), yeast (*Saccharomyces cerevisiae*) and mouse (*Mus musculus*). Since these organisms have a large number of genes which are similar, homologous, to humans, this ontology can in large parts be used to annotate human genes. The ontology divides gene function into three top-level categories: cell compartment, function and process. In our analysis of the fibroblast serum response data [12] we concentrated on process, since this is the only aspect of gene function where one may expect a significant correlation between annotations and temporal gene expression profiles [5].

From the 517 genes in the fibroblast serum response data [12], 300 could be annotated to one or more processes in Ashburner's Gene Ontology [13] by the use of knowledge extracted manually from literature and databases such as SWISS-PROT [19]. The total number of annotations was 647, hence each gene seems to be involved in over 2 processes on average. Requiring 10 genes in each functional class ($\lambda = 10$) resulted in a training set containing 16 processes including 209 genes and 335 annotations. Requiring 20 genes ($\lambda = 20$) resulted in a training set containing 10 processes including 215 genes and 332 annotations.

We used two templates in our analysis, one that defined an increase and one that defined a decrease in an expression level. In order to match one of these templates, a gene needed to increase/decrease at least 0.8 over at least three time points. Temporary changes in the opposite direction were allowed, but not by more than 0.2 from one time point to the next. The values of 0.8 and 0.2 were informally derived from the selection criteria Iyer et. al. [4] used to single out the 517 genes. Additionally, the data passed through two transformations before we applied the templates. First, the initially logarithmic data was made linear by the simple inverse logarithmic transformation 2^t. Then moving

average transformation $t_i = \frac{t_i - t_{i-1}}{2}$ was used to smooth out spikes.

The above two templates were used to construct a final training set as described in Sec. 2.2. $\lambda = 10$ and $\varepsilon = \frac{1}{3}$ resulted in 166 genes and 246 annotations. $\lambda = 20$ and $\varepsilon = \frac{1}{3}$ resulted in 156 genes and 223 annotations. Consequently, we were left with approximately half of the known genes for training.

The set of training examples was divided into a training set (2/3) and a test set (1/3). The training set was then further divided in a 10-fold cross validation setting in order to select the best algorithm and to fine-tune its parameters. Finally, the test set was included in a $5 * 3$-fold cross validation. A genetic algorithm computing α-reducts ($\alpha = 90$) was used (for details see Vinterbo and Øhrn [20]). The results for $\lambda = 10$ (Sec. 2.1), $\varepsilon = 1/3$ (Sec. 2.2) and $v = 2$ (Sec. 2.4) are shown in Tab. 1. The corresponding average AUC-values for $\lambda = 20$ are 0.81 both in the 10-fold and in the $5 * 3$-fold cross validation setting.

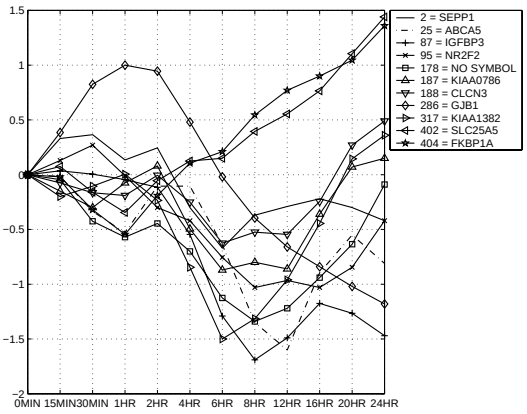

Figure 1: Expression graphs for the 11 genes associated with process *transport* (1.1.2).

Applying the rule model to the unknown genes using the threshold values in Tab. 1 resulted in process predictions for 191 of the 217 unknown genes. A total number of 545 predictions were made, corresponding on average to almost 3 hypotheses about the function of each unknown gene. The most frequently predicted processes were *transcription regulation from Pol II promoter* and *apoptosis*, while *chemotaxis* and *blood coagulation* constituted very few predictions.

We have shown how the predictive model is capable of putting forward hypotheses about the function of unknown genes, and how the quality of these hypotheses can be estimated in a cross-validation testing on the known genes. Furthermore, our predictive model can be visually inspected in order to evaluate its descriptive quality. Fig. 1 shows the expression graphs of the 11 genes in the training examples associated with process *transport* (1.1.2). Three rules were induced for this process:

1. 2H - 6H(Decreasing) AND 12H - 20H(Increasing) → Process(1.1.2)

2. *2H - 6H(Decreasing)* $\rightarrow$ *Process(1.1.2)*
3. *12H - 20H(Increasing)* $\rightarrow$ *Process(1.1.2)*

As one can see, the rules perfectly match our visual impression of the important changes in the expression levels in this process. The total number of rules induced was 85 for $\lambda = 10$ and 50 for $\lambda = 20$.

A full overview of the results including rules, predictions and biological interpretations will be published in a journal article.

4 Discussion

A predictive model may be evaluated along two axes: predictive quality and descriptive quality. Tab. 1 shows that the predictive quality is significantly better than a random classifier (AUC = 0.50). Moreover, it shows that by weighting the cost of false positives higher than false negatives, we can obtain high quality hypotheses about the function of unknown genes. The descriptive value of our simple language of templates over sub-intervals is illustrated in Sec. 3 with process *transport*. Furthermore, biological experts have studied several of the processes and found that the induced rules very much describe the complex molecular biological events during the fibroblast response.

Iyer et. al. [4] applied an agglomerative implementation of the hierarchical clustering algorithm to cluster 452 of the 517 genes into 10 groups on the basis of their similarity in expression level over the entire period of 24 hours. However, as visually shown by Iyer et. al. [4], the similarities in expression levels for genes involved in the same process are often only revealed in shorter time frames than 24 hours. This is, among other factors, due to the fact that several genes are active in more than one process. This point is illustrated in Fig. 1. The numbers given on the side of the gene names in the figure are taken from the dendrogram built by the hierarchical clustering algorithm used by Iyer et. al. [4], and reflect the similarity as defined by this algorithm. Clearly, process *transport* is not well described by this approach. We believe that our language of templates over sub-intervals is more suitable for describing the complex relationships between gene expressions and processes.

5 Conclusions

We have described a supervised learning methodology for discovering gene functions from expression data. We have shown how gene ontologies can be used to determine the functional classes that we want to learn and we have also shown how to deal with inconsistency and irrelevance in the annotations. The methodology is demonstrated on the fibroblast serum response data [12]

and greatly reduces the number of functions for unknown genes that have to be further investigated in the wet lab.

We believe that the inclusion of domain knowledge is important in order to predict gene function from expression data; we do not believe that a syntactical analysis such as clustering utilises this resource well enough. As more genes become known and more of reliable annotations are well formalised in terms of ontologies, the strength of supervised methods will become even more evident. With a large part of the knowledge of gene function formalised, it will be possible to fully automate the task of annotation and hence greatly reduce the overall time needed to learn predictive models from gene expressions.

Acknowledgements

We would like to thank Tor-Kristian Jenssen and Dyre Tjeldvoll for helpful discussions and remarks; a special thank to Dyre Tjeldvoll for his work on semi-automatic gene annotations from Ashburner's Gene Ontology. Thanks also to Aleksander Øhrn for his help with the ROSETTA system.

References

1. Strachan S and Read AP, *Human Molecular Genetics*, BIOS Scientific Publishers Ltd, 1999.
2. Schena M, Shalon D, Davis R and Brown PO, Quantitative monitoring of gene expression patterns with a Complementary DNA microarray, *Science*, 270(5235):467–470, 1995.
3. Eisen M, Spellman P, Brown P and Botstein D, Cluster analysis and display of genome-wide expression pattern, *Proc. Natl. Acad. Sci. USA*, 95(25):14863–14868, 1998.
4. Iyer VR, Eisen MB, Ross DT, Schuler G, Moore T, Lee JCF, Trent JM, Staudt LM, Dudson Jr. J, Boguski MS, Lashkari D, Shalon D, Botstein D and Brown PO, The transcriptional program in the response of human fibroblasts to serum, *Science*, 283(5398):83–87, 1999.
5. Brown MPS, Grundy WN, Cristianini N, Sugnet CW, Furey TS, Ares M and Haussler D, Knowledge-based analysis of microarray gene expression data by using support vector machines, *Proc. Natl. Acad. Sci. USA*, 97(1):262–267, 1999.
6. Schalkoff R, *Pattern Recognition - Statistical, structural and neural approaches*, John Wiley & Sons, Inc., 1992.
7. Kohonen T, The Self-Organizing Map, *Proceedings of the IEEE*, 78(9):1464–1480, 1990.

8. Komorowski J, Skowron A and Øhrn A, The ROSETTA system, in Klösgen W and Zytkow J, *Handbook of Data Mining and Knowledge Discovery*, Oxford University Press, 2000.
9. The ROSETTA homepage: `http://www.idi.ntnu.no/~aleks/rosetta`
10. Fayyad U, Piatetsky-Shapiro G and Smyth P, From Data Mining to Knowledge Discovery in Databases, *AI magazine*, 17(3):37–54, 1996.
11. Pawlak Z, Rough Sets, *International Journal of Computer and Information Sciences*, 11:341–356, 1982.
12. The fibroblast serum response data on the WEB: `http://genome-www.stanford.edu/serum/`.
13. The homepage of Ashburner's Gene Ontology: `http://genome-www.stanford.edu/GO/`
14. Skowron A and Rauszer C, The Discernibility Matrices and Functions in Information Systems, pages 331–362 in Slowinski, *Intelligent Decision Support - Handbook of Applications and Advances of the Rough Sets Theory*, Dordrecht, Kluwer Academic Publishers, 1992.
15. Bazan JG, A comparison of dynamic and non-dynamic rough set methods for extracting laws from decision tables, chapter 17, pages 321–365 in Polkowski L and Skowron A, *Rough Sets in Knowledge Discovery 2: Methodology and Applications*, volume 18 of *Studies in Fuzziness and Soft Computing*, Physica-Verlag, Heidelberg, Germany, 1998.
16. Skowron A and Nguyen HS, Boolean reasoning scheme with some applications in data mining, pages 107–115 in Zytkow and Rauch, *Proceedings of the Third European Symposium on Principles and Practice of Knowledge Discovery in Databases* (PKDD'99), volume 1704 of *Lecture Notes in Artificial Intelligence*, Springer-Verlag, Prague, Czech Republic, 1999.
17. Komorowski J, Pawlak Z, Polkowski L and Skowron A, Rough Sets: A tutorial, pages 3 – 98 in Pal SK and Skowron A, *Rough-Fuzzy Hybridization - A New Trend in Decision Making*, Springer-Verlag Singapore Pte Ltd, 1999.
18. Hanley JA and McNeil BJ, The meaning and use of the area under a receiver operating characteristic (ROC) curve, *Radiology*, 143:29–36, 1982.
19. The SWISS-PROT homepage: `http://www.expasy.ch/sprot/`
20. Vinterbo S and Øhrn A, Minimal approximate hitting sets and rule templates, *International Journal of Approximate Reasoning*, 25(2):123-143, 2000.

Table 1: Results classifying 16 processes in the fibroblast serum response data. The first column lists the process with the Ashburner's Gene Ontology address in brackets. Then AUC-values for the 10-fold cross validation are given, followed by AUC, standard error for AUC, sensitivity, specificity, accuracy and thresholds for the $5 * 3$-fold cross validation. Sensitivity, specificity and accuracy are specific to the given threshold obtained by minimising the expression $2 * (1 - \text{specificity}(\tau)) + (1 - \text{sensitivity}(\tau))$.

PROCESS	10-**folds CV** **AUC**	$5 * 3$-**fold CV** **AUC**	**SE**	**SENS.**	**SPEC.**	**ACC.**	**THR.**
transcription regulation from Pol II promoter (1.1.1.4.11.1.4.2)	0.78	0.78	0.08	0.76	0.75	0.75	0.13
protein metabolism and modification (1.1.1.5)	0.74	0.78	0.13	0.61	0.85	0.84	0.11
steroid metabolism (1.1.1.9.6)	0.86	0.74	0.13	0.63	0.81	0.79	0.08
defense (immune) response (1.1.13.2)	0.79	0.69	0.10	0.36	0.93	0.86	0.25
cytoskeleton organization and biogenesis (1.1.14.4)	0.76	0.82	0.06	0.81	0.75	0.77	0.15
positive control of cell proliferation (1.1.16.1)	0.76	0.84	0.09	0.84	0.82	0.82	0.11
cell cycle control (1.1.17.6)	0.97	0.81	0.09	0.78	0.77	0.77	0.07
transport (1.1.2)	0.72	0.68	0.10	0.33	0.88	0.81	0.15
apoptosis (1.1.8.1)	0.73	0.77	0.10	0.65	0.81	0.78	0.16
substrate-bound cell migration (1.1.9.1)	0.84	0.82	0.13	0.89	0.78	0.78	0.01
chemotaxis (1.1.9.2)	0.69	0.76	0.10	0.43	0.95	0.89	0.18
cell-cell matrix adhesion (1.2.1.3)	0.90	0.79	0.06	0.64	0.83	0.78	0.24
cell surface receptor linked signal transduction (1.2.3.1)	0.82	0.73	0.07	0.50	0.84	0.75	0.16
intracellular signalling cascade (1.2.3.2)	0.68	0.82	0.10	0.66	0.85	0.84	0.15
embryogenesis and morphogenesis (1.3.5)	0.66	0.92	0.10	0.91	0.88	0.88	0.15
blood coagulation (1.4.9)	0.81	0.90	0.10	0.85	0.88	0.88	0.17
AVERAGE	**0.78**	**0.79**	**0.10**	**0.67**	**0.84**	**0.81**	

Pacific Symposium on Biocomputing 6:299-310 (2001)