AUTOMATIC CANCER DIAGNOSTIC DECISION SUPPORT SYSTEM FOR GENE EXPRESSION DOMAIN

By

Alexander Statnikov

Thesis

Submitted to the Faculty of the

Graduate School of Vanderbilt University

in partial fulfillment of the requirements

for the degree of

MASTER OF SCIENCE

in

Biomedical Informatics

August, 2005

Nashville, Tennessee

Approved,

Constantin F. Aliferis

Shawn Levy

Douglas P. Hardin

Ioannis Tsamardinos

To my beloved and infinitely supportive wife, Kristina,

and to my parents: my mother, Elena Feofilova, and my father, Roman Statnikov

ACKNOWLEDGEMENTS

First of all, I would like to acknowledge my academic advisors, Dr. Constantin F. Aliferis and Dr. Ioannis Tsamardinos, for their contribution to my Master's project. In addition, I am grateful to Dr. Aliferis for introducing me to the fields of Biomedical Informatics and Machine Learning; training me both a researcher and as a professional scientific programmer; and setting up a framework for my future research and career development. I am also indebted to Dr. Tsamardinos for providing me with excellent technical and scientific ideas that are relevant for my professional and scientific growth.

I would also like to thank everybody with whom I had pleasure to work during this project. In particular, I would to express my gratitude to Dr. Douglas P. Hardin and Dr. Shawn Levy who being members of my Master's committee contributed not only to this project, but also to the journal manuscript based on this work. I would like to acknowledge all Biomedical Informatics faculty and graduate students for their countless contributions to the success of this project.

Finally, I am forever indebted to my wife, Kristina Statnikova, for her understanding, endless patience and encouragement. This project would not be possible without her support.

TABLE OF CONTENTS

Page

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

INTRODUCTION

Development of cancer diagnostic models and discovery from DNA microarray data is of great interest in bioinformatics and medicine. Diagnostic models from gene expression data go beyond traditional histopathology and provide accurate, resource-efficient, and replicable diagnosis [Golub1999]. Furthermore, biomarker discovery in high-dimensional microarray data facilitates learning about the biology of cancer [Balmain2003]. Currently, building of cancer diagnostic models from microarray gene expression data has three challenging components: collection of samples, assaying, and statistical analysis. A typical statistical analysis process takes from a few weeks to several months and involves many specialists: clinical researchers, statisticians, bioinformaticians, and programmers. As a result, statistical analysis is a serious bottleneck in the development of cancer diagnostic models, and its enhancement by an automated or semi-automated system will benefit research significantly. Our goal is thus to build a system that takes microarray data as input and outputs a high-quality cancer diagnostic model, produces a reliable performance estimate, allows application of this model to unseen patients, and enables biomarker discovery. In order for the system to be clinically successful, it should implement the best known methodologies applicable to this domain and use sound techniques for model selection and performance estimation in an automated fashion. An ideal system should achieve the same or better quality than human analysts and complete the entire process within minutes or a few hours requiring minimal human effort.

First, to inform development of such a system, we address the following questions by conducting a comprehensive algorithmic evaluation in the domain of cancer gene expression data: (a) Which one among the many powerful classifiers currently available for gene expression diagnosis performs the best *across many cancer types*? (b) How classifiers interact with existing gene selection methods in datasets with varying sample size, number of genes, and cancer types? (c) It is possible to increase diagnostic performance further using meta-learning in the form of ensemble classification? (d) How to parameterize the classifiers and gene selection procedures so as to *avoid overfitting*? Next, based on conclusions of the algorithmic evaluation, we develop a software system *GEMS* (Gene Expression Model Selector) for

classification and biomarker discovery in microarray gene expression data and conduct a preliminary evaluation of the system.

This thesis is organized as follows: Chapter II describes existing software systems for cancer diagnosis from microarray gene expression data as well as prior research in this field. Chapter III summarizes methodology used for the algorithmic evaluation. The results of evaluation are presented in Chapter IV. The discussion and limitations of evaluation are provided in Chapter V, and conclusions are drawn in Chapter VI. Chapter VII introduces the system *GEMS*. Chapter VIII describes an evaluation of *GEMS* by applying the system to cancer microarray datasets (not used in algorithmic evaluation) and by assessing performance of developed diagnostic models using microarray datasets from different laboratories. Chapter IX provides directions for future research and outlines current limitations of *GEMS*. The thesis concludes with Chapter X.

CHAPTER II

RELATED WORK

Existing software systems

Currently, there exist many dozens of software systems designed for microarray gene expression data analysis ([Causton2003] and [Parmigiani2003]). Since prior research has demonstrated superiority of supervised classification methods for cancer diagnosis over unsupervised techniques [Simon2003], we focused only on systems implementing supervised classification algorithms. Using this criterion, we identified 16 software systems (Table 1): 6 are commercial (names are shown with boldface in Table 1) and 10 can be used free of charge for non-profit research. All systems have several of the following limitations. First, the performance quality of the learning algorithms selected for inclusion into the systems is unknown. Typically, the algorithmic palette reflects the authors' preferences and their prior publication history; there is often limited evidence that these algorithms are indeed appropriate for this domain and equally important, that they are among the best performing ones. Second, many classification algorithms implemented in the software systems are not able to handle multicategory diagnosis, despite that most diagnostic tasks involve several diseases and that powerful multicategory classification methods do exist in machine learning. Third, none of the systems automatically optimizes the parameters and the choice of both classification and gene selection algorithms (also known as model selection) while simultaneously avoiding overfitting[1]. The user of these systems is left with two choices: either to avoid rigorous model selection and possibly discover a suboptimal model, or to experiment with many different parameters and algorithms and select the model with the highest cross-validation performance. The latter is subject to overfitting primarily due to multiple-testing, since parameters and algorithms are selected after all the testing sets in cross-validation have been seen by the algorithms. We aim to address all these problems in the proposed software system for classification and biomarker discovery from microarray gene expression data.

[1] Only one commercial software system, Partek Predict by Partek Inc., attempts to automatically conduct a rigorous optimization of the parameters and the choice of algorithms while providing unbiased performance estimates. Unfortunately, the current version 6.0 of Partek Predict does not completely implement this methodology, since it does not allow optimization of the choice of gene selection algorithms.

Table 1: Software systems for gene expression-based cancer diagnosis (supervised classification only).

Name	Version	Developer	Supervised classification	Cross-validation for performance estimation	Automatic model selection for classifier and gene selection methods	URL
ArrayMiner ClassMarker	5.2	Optimal Design, Belgium	· K-Nearest Neighbors; · Voting	**Yes**	**No**	http://www.optimaldesign.com/ArrayMiner
Avadis Prophetic	3.3	Strand Genomics, USA	· Decision Trees; · Neural Networks; · Support Vector Machines.	**Yes**	**No**	http://avadis.strandgenomics.com/
BRB ArrayTools	3.2 Beta	National Cancer Institute, USA	· Compound Covariate Predictor; · Diagonal Linear Discriminant Analysis; · Nearest Centroid; · K-Nearest Neighbors; · Support Vector Machines.	**Yes**	**No**	http://linus.nci.nih.gov/BRB-ArrayTools.html
caGEDA	(accessed 10/2004)	University of Pittsburgh and University of Pittsburgh Medical Center, USA	· Nearest Neighbors methods; · Naïve Bayes Classifier.	**Yes**	**No**	http://bioinformatics.upmc.edu/GE2/GEDA.html
Cleaver	1.0 (accessed 10/2004)	Stanford University, USA	· Linear Discriminant Analysis	**Yes**	**No**	http://classify.stanford.edu
GeneCluster2	2.1.7	Broad Institute, Massachusetts Institute of Technology, USA	· Weighted Voting; · K-Nearest Neighbors.	**Yes**	**No**	http://www.broad.mit.edu/cancer/software
GeneLinker Platinum	4.5	Predictive Patterns Software, Canada	· Neural Networks; · Support Vector Machines; · Linear Discriminant Analysis; · Quadratic Discriminant Analysis; · Uniform/Gaussian Discriminant Analysis.	**Yes**	**No**	http://www.predictivepatterns.com/
GeneMaths XT	1.02	Applied Maths, Belgium	· Neural Networks; · K-Nearest Neighbors; · Support Vector Machines.	**Yes**	**No**	http://www.applied-maths.com/genemaths/genemaths.htm
GenePattern	1.2.1	Broad Institute, Massachusetts Institute of Technology, USA	· Weighted Voting; · K-Nearest Neighbors; · Support Vector Machines.	**Yes**	**No**	http://www.broad.mit.edu/cancer/software
Genesis	1.5.0	Graz University of Technology, Austria	· Support Vector Machines	**No**	**No**	http://genome.tugraz.at/Software/Genesis/Genesis.html
GeneSpring	7	Silicon Genetics, USA	· K-Nearest Neighbors; · Support Vector Machines.	**Yes**	**No**	http://www.silicongenetics.com

Table 1 (continued): Software systems for gene expression-based cancer diagnosis (supervised classification only).

Name	Version	Developer	Supervised classification	Cross-validation for performance estimation	Automatic model selection for classifier and gene selection methods	URL
GEPAS	1.1 (accessed 10/2004)	National Center for Cancer Research (CNIO), Spain	· K-Nearest Neighbors; · Support Vector Machines; · Diagonal Linear Discriminant Analysis.	**Yes**	**Limited** (for number of genes)	http://gepas.bioinfo.cnio.es/tools.html
MultiExperiment Viewer	3.0.3	The Institute for Genomic Research, USA	· K-Nearest Neighbors; · Support Vector Machines.	**Yes**	**No**	http://www.tigr.org/software/tm4/mev.html
PAM	1.21a	Stanford University, USA	· Nearest Shrunken Centroids	**Yes**	**Limited** (for a single parameter of the classifier)	http://www-stat.stanford.edu/~tibs/PAM/
Partek Predict	6.0	Partek, USA	· K-Nearest Neighbors; · Nearest Centroid Classifier; · Discriminant Analysis.	**Yes**	**Limited** (does not allow optimization of the choice of gene selection algorithms)	http://www.partek.com/
Weka Explorer	3.4.3	University of Waikato, New Zeland	· K-Nearest Neighbors; · Decision Trees; · Rule Sets; · Bayesian Classifiers; · Support Vector Machines; · Multi-Layer Perceptron; · Linear Regression; · Logistic Regression; · Meta-Learning Techniques (Boosting, Bagging).	**Yes**	**No**	http://www.cs.waikato.ac.nz/ml/weka/

Prior methodological studies

Previous studies in cancer diagnosis model creation from gene expression data provide limited evidence for selecting the best performing learning techniques. We identified 193 primary gene expression-based cancer diagnosis studies using the ONCOMINE Cancer Microarray Database [Rhodes2004a], the UPITT Cancer Gene Expression Data Set Link Database [UPMC2004], and the Stanford Microarray Database [Gollub2003]. A review of these studies and publications that reanalyzed publicly available datasets (identified by querying ISI Web of Science Cited Reference Search and PubMed Central Citation Search for citations of primary microarray studies) revealed the following:

- A typical study applies only a few (usually, 2-3) classification algorithms to a single cancer microarray dataset;
- The majority of diagnostic tasks pursued by the studies are binary (i.e. with two possible outcomes), whereas real-life diagnostic problems are generally multicategory;
- Researchers often apply parametric classifiers without rigorous optimization of their parameters;
- Different computational experimental designs employed by the studies (e.g., *N*-fold cross-validation, leave-one-out cross-validation, hold-out cross-validation, bootstrapping, etc) make the findings incomparable.

We also located two meta-analyses covering the scope of our research: [Ntazni2003] and [Rhodes2004b]. According to [Ntzani2003], only 26% of studies in this domain attempted independent validation or cross-validation of their findings. This questions whether published results will generalize well to unseen patients. Unfortunately, neither of these two meta-analyses is aimed at the identification of the best performing methodologies, nor can be used to do so. The meta-analysis by Ntzani and Ioannidis [Ntzani2003] examined the predictive performance of DNA microarrays for cancer diagnosis and prognosis in general, without resorting to specific algorithms. The meta-analysis by Rhodes *et al.* [Rhodes2004b] is geared toward biomarker assessment across 40 studies and uses a single simplistic biomarker discovery method and an equally simplistic and a non-standard classifier.

In addition, two recent bioinformatics studies ([Berrar2003] and [Romualdi2003]) performed comparative analyses of multicategory classification algorithms in the cancer gene expression domain. However, the results of these evaluations cannot serve as a basis for the development of cancer diagnostic

decision support system for the following two reasons: first, both evaluations are limited only to two microarray datasets and second, neither study optimized parameters of the classifiers in all datasets, which is likely to result in suboptimal application of diagnostic methods.

For the above reasons, and given the plethora of classification algorithms applicable to gene expression-based cancer diagnostic problems, it was unclear what constitutes a small subset of methods that perform optimally across many datasets and cancer types. Therefore, we decided to conduct such an evaluation *de novo* in order to base our system on the currently best techniques for the chosen task and domain. The methods and results of this evaluation are discussed in the next chapters.

CHAPTER III

METHODS USED FOR ALGORITHMIC EVALUATION

Support Vector Machine-based classification methods

Support Vector Machines (SVMs) [Vapnik1998] are arguably the single most important development in supervised classification of recent years. SVMs often achieve superior classification performance compared to other learning algorithms across most domains and tasks; they are fairly insensitive to the *curse of dimensionality* and are efficient enough to handle very large-scale classification in both sample and variables. In clinical bioinformatics they have allowed construction of powerful experimental cancer diagnostic models based on gene expression data with thousands of variables and as little as few dozens samples (e.g., [Furey2000], [Guyon2002], and [Aliferis2003a]). Moreover, several efficient and high-quality implementations of SVM algorithms (e.g., [Joachims1999] and [Chang2003]) facilitate application of these techniques in practice. The first generation of SVMs could only be applied to binary classification tasks. Yet, most real-life diagnostic tasks are not binary. Moreover, all other things being equal, multicategory classification is significantly harder than binary classification [Mukherjee2003]. Fortunately, several algorithms have emerged during the last few years that allow multicategory classification with SVMs. The preliminary experimental evidence currently available suggests that some multicategory SVMs (MC-SVMs) perform well in isolated gene expression-based cancer diagnostic experiments ([Yeo2001], [Su2001], [Ramaswamy2001], [Yeang2001], and [Lee2003]).

Below we outline the principles behind SVM algorithms used in the study. Full technical descriptions can be found in the references provided in text. A detailed review of binary SVMs, exact mathematical formulations of both binary and multiclass SVM algorithms, and an illustration of MC-SVMs methods via a solution of example cancer diagnostic problem are presented in Appendices A, B, and C, respectively [Statnikov2005]. In the description of methods below, k is the number of classes or distinct diagnostic categories, and n is the number of samples or patients in the training dataset.

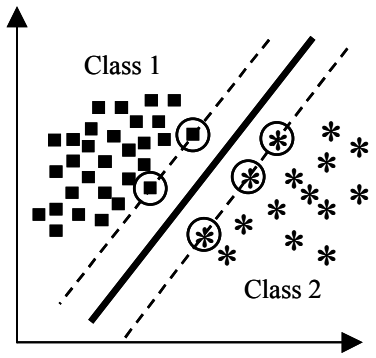

Figure 1: A binary SVM selects a hyperplane (bold line) that maximizes the width of the "gap" (margin) between the two classes. The hyperplane is specified by "boundary" training instances, called "support vectors" shown with circles. New cases are classified according to the side of the hyperplane they fall into.

Binary SVMs

The main idea of binary SVMs is to implicitly map data to a higher dimensional space via a kernel function and then solve an optimization problem to identify the maximum-margin hyperplane that separates training instances [Vapnik1998]. The hyperplane is based on a set of boundary training instances, called *support vectors*. New instances are classified according to the side of the hyperplane they fall into (Figure 1). The optimization problem is most often formulated in a way that allows for non-separable data by penalizing misclassifications.

Multiclass SVMs: One-Versus-Rest (OVR)

This is conceptually the simplest multiclass SVM method (see [Kressel1999] for details). Here we construct k binary SVM classifiers: class 1 (positive) versus all other classes (negative), class 2 versus all other classes, … , class k versus all other classes (Figure 2a). The combined OVR decision function chooses the class of a sample that corresponds to the maximum value of k binary decision functions specified by the furthest "positive" hyperplane. By doing so, the decision hyperplanes calculated by k SVMs "shift", which questions the optimality of the multicategory classification.

This approach is computationally expensive, since we need to solve k quadratic programming (QP) optimization problems of size n. Moreover, this technique does not currently have theoretical justification such as the analysis of generalization, which is a relevant property of a robust learning algorithm.

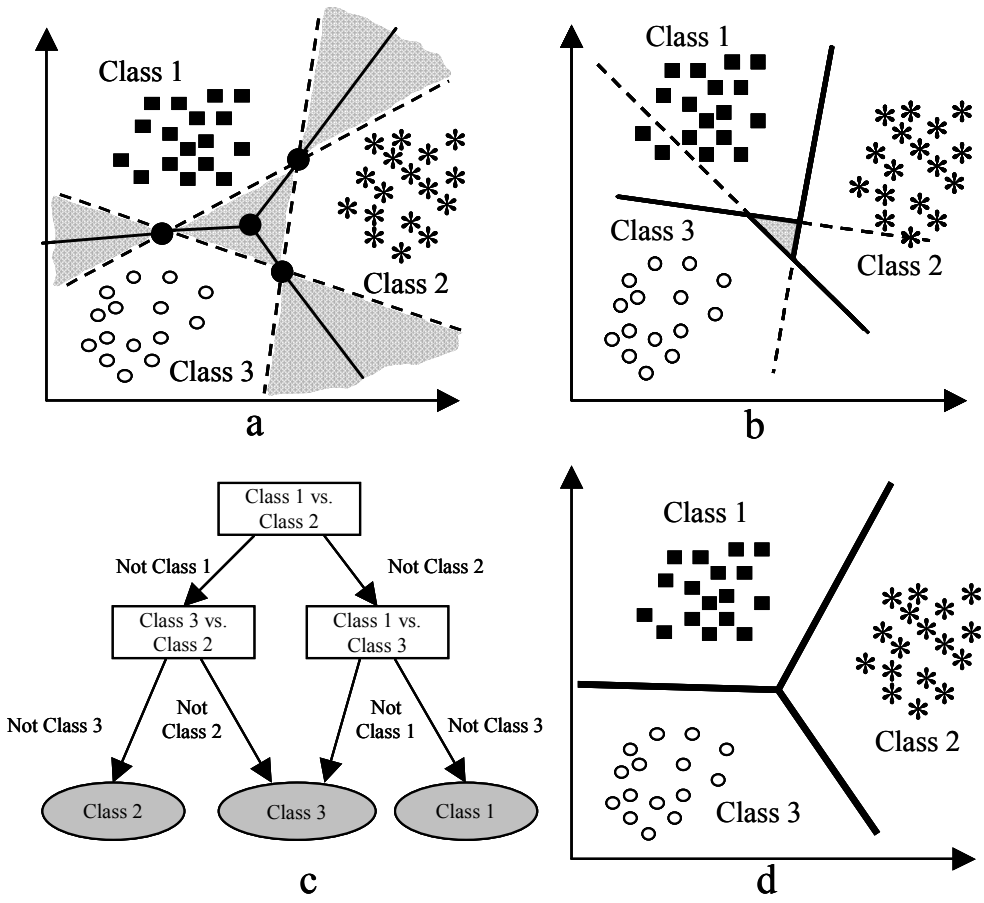

Figure 2: MC-SVM algorithms applied to a three-class diagnostic problem. (a) MC-SVM One-Versus-Rest constructs 3 classifiers: (1) class 1 vs classes 2 and 3, (2) 2 vs 1 and 3, and (3) 3 vs 1 and 2. (b) MC-SVM One-Versus-One constructs 3 classifiers: (1) class 1 vs class 2, (2) 2 vs 3, and (3) 1 vs 3. (c) MC-SVM DAGSVM constructs a decision tree on the basis of One-Versus-One SVM classifiers. (d) MC-SVM methods by Weston and Watkins and by Crammer and Singer construct a single classifier by maximizing margin between all classes simultaneously.

Multiclass SVMs: One-Versus-One (OVO)

This method involves construction of binary SVM classifiers for all pairs of classes; in total there are $\binom{k}{2} = \frac{k(k-1)}{2}$ pairs (see Figure 2b and [Kressel1999]). In other words, for every pair of classes, a binary SVM problem is solved (with the underlying optimization problem to maximize the margin between two classes). The decision function assigns an instance to a class which has the largest number of votes, so-called *Max Wins strategy* [Friedman1996]. If ties still occur, a sample will be assigned a label based on the classification provided by the furthest hyperplane.

One of the benefits of this approach is that for every pair of classes we deal with a much smaller optimization problem, and in total we need to solve *k(k-1)/2* QP problems of size *smaller than n*. Given that QP optimization algorithms used for SVMs are polynomial to the problem size, such a reduction can yield substantial savings in the total computational time. Moreover, some researchers postulate that even if the entire multicategory problem is non-separable, while some of the binary sub-problems are separable, then OVO can lead to improvement of classification compared to OVR [Kressel1999]. Unlike the OVR approach, here tie-breaking plays only a minor role and does not affect the decision boundaries significantly. On the other hand, similarly to OVR, OVO does not currently have established bounds on the generalization error.

Multiclass SVMs: DAGSVM

The training phase of this algorithm is similar to the OVO approach using multiple binary SVM classifiers; however the testing phase of DAGSVM requires construction of a rooted binary decision directed acyclic graph (DDAG) using $\binom{k}{2}$ classifiers (see Figure 2c and [Platt2000]). Each node of this graph is a binary SVM for a pair of classes, say (p, q). On the topologically lowest level there are k leaves corresponding to k classification decisions. Every non-leaf node (p, q) has two edges – the left edge corresponds to decision "not p" and the right one corresponds to "not q". The choice of the class order in the DDAG list can be arbitrary as shown empirically in [Platt2000].

In addition to inherited advantages from the OVO method, DAGSVM is characterized by a bound on the generalization error.

Multiclass SVMs: Method by Weston and Watkins (WW)

This approach to multiclass SVMs is viewed by some researchers as a natural extension of the binary SVM classification problem (see Figure 2d, [Hsu2002] and [Weston1999]). Here, in the k-class case one has to solve a single quadratic optimization problem of size *(k-1)n* which is identical to binary SVMs for the case k=2. In a slightly different formulation of QP problem, a bounded formulation, decomposition techniques can provide a significant speed-up in the solution of the optimization problem ([Hsu2002],

[Platt1999]). This method does not have an established bound on the generalization error, and its optimality is not currently proved.

Multiclass SVMs: Method by Crammer and Singer (CS)

This technique is similar to WW (see Figure 2d, [Hsu2002] and [Crammer2000]). It requires solution of a single QP problem of size *(k-1)n*, however uses less slack variables in the constraints of the optimization problem, and hence it is cheaper computationally. Similarly to WW, the use of decompositions can provide a significant speed-up in the solution of the optimization problem [Hsu2002]. Unfortunately, the optimality of CS, as well as the bounds on generalization has not been demonstrated yet.

Non-SVM classification methods

In addition to five MC-SVM methods, three popular classifiers, K-Nearest Neighbors (KNN), Backpropagation Neural Networks (NN), and Probabilistic Neural Networks (PNN) were also used in this study. These learning methods have been extensively and successfully applied to gene expression based cancer diagnosis (e.g., [Khan2001], [Ramaswamy2001], [Pomeroy2002], [Nutt2003], [Signh2002], and [Berrar2003]).

K-Nearest Neighbors (KNN)

The main idea of KNN is that it treats all samples as points in the *m*-dimensional space (where *m* is the number of variables) and given an unseen sample *x*, the algorithm classifies it by a vote of K nearest training instances as determined by some distance metric, typically Euclidian distance [Mitchell1997].

Backpropagation Neural Networks (NN)

Backpropagation Neural Networks are feed-forward neural networks with signals propagated only forward through the layers of units. These networks are composed of (I) an input layer of units, which we feed with gene expression data, (II) hidden layer(s) of units, and (III) an output layer of units, one for each diagnostic category, so-called *1-of-n encoding* (see Figure 3a and [Mitchell1997]). The connections among units have weights and are adjusted during the training phase (epochs of a neural network) by

backpropagation learning algorithm. This algorithm adjusts weights by propagating the error between network outputs and true diagnoses backward through the network and employs gradient descent optimization to minimize the error function. This process is repeated until we find a vector of weights that best fits the training data. When training of a neural network is complete, unseen data instances are fed to the input units, propagated forward through the network, and the network outputs classifications.

Probabilistic Neural Networks (PNN)

Probabilistic Neural Networks belong to the family of Radial Basis Function (RBF) neural networks [Mitchell1997]. RBF networks are feed-forward neural networks with only one hidden layer. The primary difference between a backpropagation neural network with one hidden layer and an RBF network is that for the latter one, the inputs are passed directly to the hidden layer *without weights*. The Gaussian density function is used in a hidden layer as an activation function. The weights for the connections among the hidden and the output layer are optimized via a least squares optimization algorithm. A key advantage of RBF networks is that they are trained much more efficiently than backpropagation neural networks.

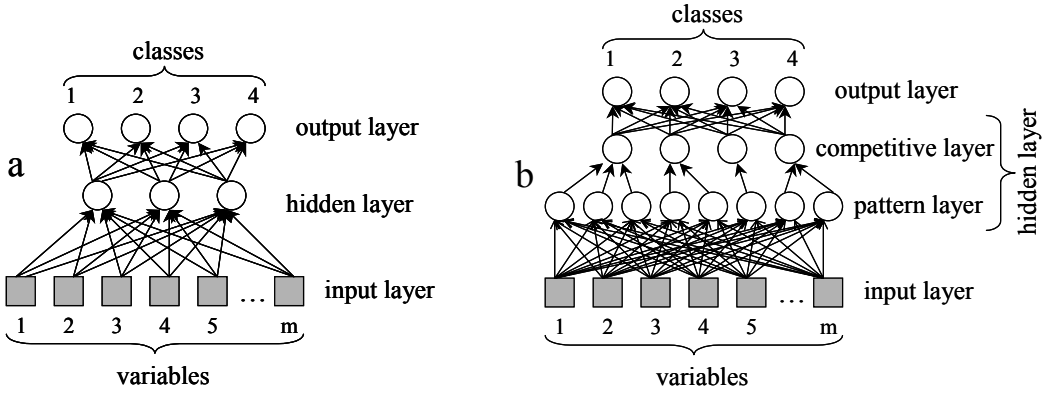

Figure 3: Simplified illustration of the design of neural networks for a 4-category diagnostic problem with m-dimensional samples of variables (genes) and training set containing N samples. (a) Backpropagation Neural Network contains inputs for m variables (genes); hidden layer contains 3 units (this number is usually determined by cross-validation); and output layer contains a unit for each diagnostic category (1-of-n encoding scheme). (b) Probabilistic Neural Network contains inputs for m variables (genes); pattern layer contains N units (a unit for each training instance); competitive layer contains 4 units (a unit for each diagnostic category) and receive inputs only from pattern units that are associated with the category to which the training instance belongs; and output layer contains a unit for each diagnostic category.

Probabilistic Neural Networks are made up of (I) an input layer, (II) a hidden layer consisting of a pattern layer and a competitive layer, and (III) an output layer (see Figure 3b, [Demuth2001] and [Specht1990]). The pattern layer contains one unit for each sample in the training dataset. Given an unseen training sample x, each unit in the pattern layer computes a distance from x to a specific training instance and applies a Gaussian density activation function. The competitive layer contains one unit for each diagnostic category, and these units receive inputs only from pattern units that are associated with the category to which the training instance belongs. Each of unit in the competitive layer sums over the outputs of the pattern layer and computes a probability of x belonging to a specific diagnostic category. Finally, the output unit corresponding to maximum of these probabilities outputs 1, while those remaining output 0.

Ensemble classification methods

Given that learners used in this study are different in a sense that they give preference to different models, the final classification performance may be improved via use of algorithms that combine outputs of individual classifiers, so-called *ensembles of classifiers*. This idea has received much attention in machine learning literature (e.g., [Ho1994] and [Sharkey1996]) and has been recently applied to the gene expression domain ([Dudoit2002] and [Valentini2003]). Learning how to combine classifiers to further improve performance is an additional meta-learning problem. Since there is no consensus on which methods are the best in ensembling classifiers, we considered a number of techniques: the most common approach by majority voting [Freund1995] and more complex approaches, Decision Trees (DT) [Murthy1997] and MC-SVM methods (OVR, OVO, DAGSVM). When algorithms were applied for ensembling of classifiers, the input dataset consisted of attributes corresponding to the outputs of classifiers (either SVM or both SVM and non-SVM algorithms) and the original class labels. Combining classifiers by DT or MC-SVM methods could yield majority voting for some cases, but DT or MC-SVMs allow many more ways to construct ensemble of classifiers.

Parameters for the classification algorithms

Parameters for the classification algorithms were chosen by nested cross-validation procedures to optimize performance while avoiding overfitting as described in the experimental design subsection.

For all five MC-SVM methods we used a polynomial kernel $K(x,y) = (\gamma \cdot x^T y + r)^p$, where x and y are samples with gene expression values and p, γ, r are kernel parameters. We performed classifier optimization over the set of values of cost C (the penalty parameter of SVMs) = {0.0001,0.01,1,100} and p = {1,2,3}. The kernel parameters γ and r were set to default values as in [Chang2003]: γ = 1/number of variables and r = 0. For Backpropagation Neural Networks, we performed optimization by implementing early stopping regularization techniques following [Goodman1996] on top of the Matlab Neural Network toolbox with parameter selection in a nested cross-validation fashion in order to avoid overfitting. In particular, we used feed-forward NN with one hidden layer and the number of units chosen from the set {2,5,10,30,50} based on cross-validation error. We employed gradient descent with adaptive learning rate backpropagation, mean squared error performance goal set to 10^{-8} (an arbitrary value very close to zero), fixed momentum of 10^{-3}, and an optimal number of epochs in the range [100, 10000] based on the early stopping criterion of [Goodman1996]. For Probabilistic Neural Networks, we optimized the smoothing factor σ, a parameter of the Gaussian density function, over 100 different values ranging from 0.01 to 1.00. The parameter σ was set the same for all diagnostic categories. Similarly, we performed a thorough optimization of the KNN classifier over all possible numbers of neighbors K ranging from 1 to the total number of instances in the training dataset based on cross-validation error.

Datasets and data preparatory steps

The datasets used in this work are described in Table 2. In addition to nine multicategory datasets which were most of the multicategory cancer diagnosis datasets in humans found in the public domain at the time the present study was initiated, two binary datasets (i.e. with two diagnoses), *DLBCL* and *Prostate_Tumor*, were also included to empirically confirm that the employed MC-SVM learners behave well in binary classification tasks as theoretically expected.

The studied datasets were produced primarily by oligonucleotide-based technology. Specifically, in all datasets except for *SRBCT*, RNA was hybridized to high-density oligonucleotide Affymetrix arrays HG-U95 or Hu6800, and expression values (average difference units) were computed using Affymetrix GENECHIP analysis software. The *SRBCT* dataset was obtained by use of two-color cDNA platform with

consecutive image analysis performed by DeArray Software and filtering for a minimal level of expression [Khan2001].

The genes or oligonucleotides with "absent" calls in all samples were excluded from the analysis to reduce the amount of noise in the datasets ([Lu2002] and [Wouters2003]), and if this was the case, the number of genes is listed in bold in Table 2. While setting up datasets for experiments, we took advantage of all available documentation in order to increase the number of categories or diagnoses for the outcome variable. For example, the original *Brain_Tumor1* data analysis had only two categories – glioblastomas and anaplastic oligodendrogliomas. Instead of a binary classification problem, we solved a diagnostic problem with four outcomes: classic glioblastomas, non-classic glioblastomas, classic anaplastic oligodendrogliomas, and non-classic anaplastic oligodendrogliomas.

In summary, the 11 datasets had 2-26 distinct diagnostic categories, 50-308 samples (patients), and 2308-15009 variables (genes) after the data preparatory steps outlined above. All datasets are available for download from www.gems-system.org.

We note that no new methods to preprocess gene expression data were invented. We relied instead on standard normalization and data preparatory steps performed by the authors of the primary dataset studies. In addition to that, we performed a simple rescaling of gene expression values to be between 0 and 1 for speeding up SVM training. The rescaling was performed based on the training set in order to avoid overfitting.

Table 2: Cancer-related human gene expression datasets used in this study. In addition to 9 multicategory datasets, 2 datasets with two diagnoses were included to empirically confirm that MC-SVM methods behave as well as binary SVMs in binary classification tasks as theoretically expected. The column "Max. prior" indicates the prior probability of the dominant diagnostic category.

Dataset name	Diagnostic Task	Number of				Max. prior	Reference
		Sam-ples	Variables (genes)	Cate-gories	Variables / Samples		
11_Tumors	11 various human tumor types	174	12533	11	72	15.5%	[Su2001]
14_Tumors	14 various human tumor types and 12 normal tissue types	308	**15009**	26	49	9.7%	[Ramaswamy2001]
9_Tumors	9 various human tumor types	60	**5726**	9	95	15.0%	[Staunton2001]
Brain_Tumor1	5 human brain tumor types	90	**5920**	5	66	66.7%	[Pomeroy2002]
Brain_Tumor2	4 malignant glioma types	50	**10367**	4	207	30.0%	[Nutt2003]
Leukemia1	Acute myelogenous leukemia (AML), acute lympboblastic leukemia (ALL) B-cell, and ALL T-cell	72	**5327**	3	74	52.8%	[Golub1999]
Leukemia2	AML, ALL, and mixed-lineage leukemia (MLL)	72	**11225**	3	156	38.9%	[Armstrong2002]
Lung_Cancer	4 lung cancer types and normal tissues	203	12600	5	62	68.5%	[Bhattacherjee2001]
SRBCT	Small, round blue cell tumors (SRBCT) of childhood	83	2308	4	28	34.9%	[Khan2001]
Prostate_Tumor	Prostate tumor and normal tissues	102	**10509**	2	103	51.0%	[Singh2002]
DLBCL	Diffuse large b-cell lymphomas (DLBCL) and follicular lymphomas	77	**5469**	2	71	75.3%	[Shipp2002]

Experimental design for model selection and evaluation

Two experimental designs were employed to obtain reliable performance estimates and avoid overfitting. Both experimental designs are based on two loops. The inner loop is used to determine the best parameters of the classifier (i.e. values of parameters yielding the best performance on the validation dataset). The outer loop is used for estimating the performance of the classifier built using the previously found best parameters by testing on an *independent set of patients*. Design I uses a stratified 10-fold cross-validation in the outer loop and a stratified 9-fold cross-validation in the inner loop [Weiss1991]. It is often referred to as *nested stratified 10-fold cross-validation*. Design II uses leave-one-out cross-validation (LOOCV) in the outer loop and a stratified 10-fold cross-validation in the inner loop. We chose to employ both designs because there exists contradictory evidence in the machine learning literature regarding whether *N*-fold cross-validation provides more accurate performance estimates than LOOCV and vice-versa for zero-one loss classification [Kohavi1995].

Building of the final diagnostic model involves: (a) finding the best parameters for the classification algorithm using a single loop of cross-validation analogously to the inner loop in Designs I and II; (b) building the classifier on all data using the previously found best parameters; and (c) estimating a conservative bound on the classifier's future accuracy by running either Design I or II.

Nested cross-validation procedure

The *nested cross-validation* procedure ([Scheffer1999] and [Dudoit2003]) allows the simultaneous optimal selection of parameters (tuning) of a classifier and the unbiased (non-overfitted) estimation of the performance of the final diagnostic model. Cross-validation is a method for providing an estimate of the performance of a diagnostic model produced by a learning procedure A on available data D. First, one partitions the data D into N non-overlapping and balanced subsets of cases. Then, the following is repeated N times: A is trained on the N-1 subsets (training set) and tested on the hold-out subset (testing set). Finally, the average performance ρ of A over the N testing sets is reported. This methodology produces an unbiased performance estimate ρ of the model produced by A by training on all the available data D (i.e., all N subsets comprise the training set). The pseudo-code of the procedure referred to as *cross-validation for*

performance estimation is shown in Figure 4, where for simplicity A is a classifier with a fixed parameter α.

Cross-validation for performance estimation:

1. Repeat N times:
   - *Training set* ← *N-1* subsets;
   - *Testing set* ← remaining subset;
   - Train the classifier A on the *training set* using parameter α;
   - Test it on the *testing set*.
2. Return ρ, the average performance of A over N *testing sets*.

Figure 4: Cross-validation for performance estimation.

Cross-validation for model selection:

1. Repeat for $i = 1, \ldots, m$:
   a. Repeat N times:
      - *Training set* ← *N-1* subsets;
      - *Testing set* ← remaining subset;
      - Train the classifier A on the *training set* using parameter α_i;
      - Test it on the *testing set*.
   b. Record $P(i)$, the average performance of A over N *testing sets*.
2. Determine α_j, where j = argmax $P(i)$ for $i = 1, \ldots, m$;
3. Train the classifier A on the entire data D using parameter α_j and return the resulting classification model.

Figure 5: Cross-validation for model selection.

Nested cross-validation:

1. Repeat N times:
   - *Training set* ← *N-1* subsets;
   - *Testing set* ← remaining subset;

   1.1. Repeat for $i = 1, \ldots, m$:
      a. Repeat *N-1* times (for samples only in the *training set*):
         - *Training_validation* set ← *N-2* subsets;
         - *Testing_validation set* ← remaining subset;
         - Train the classifier A on the *training_validation set* using parameter α_i;
         - Test it on the *testing_validation set*.
      b. Record $P(i)$, the average performance of A over *N-1 testing_validation sets*.
   1.2. Determine α_j, where j = argmax $P(i)$ for $i = 1, \ldots, m$;
   1.3. Train the classifier A on the *training set* using parameter α_j.

   - Test the classifier obtained in step 1.3 on the *testing set*.
2. Return ρ, the average performance of A over N *testing sets*.

Figure 6: Nested cross-validation for performance estimation in the outer loop and model selection in the inner loop (dashed box).

Typically however, a classifier used for learning is parametric and the optimal value of parameters should be estimated and used to produce the final model. Let us assume that the classifier can be applied with parameter α taking m values = $\{\alpha_1, \alpha_2, \alpha_3, \ldots, \alpha_{m-1}, \alpha_m\}$, where α_i is a vector with the following parameters:

- Choice of classification algorithms (e.g., K-Nearest Neighbors, Support Vector Machines);
- Parameters of the specific classification algorithms (e.g., number of neighbors K for K-Nearest Neighbors, penalty parameter C for Support Vector Machines);
- Choice of algorithms applied prior to classification, such as gene selection, normalization, imputation, and others (e.g., gene selection by signal-to-noise ratio, gene selection by ANOVA);
- Parameters of algorithms applied prior to classification (e.g., number of genes to be used for classification).

To estimate the optimal value of the parameter α, cross-validation is used again. The performance $P(i)$ of learner A trained with parameter α_i is estimated for $i = 1, \ldots, m$ by cross-validation. The final model is built by training A on all available data D using the parameter α_j, where j = argmax $P(i)$ for $i = 1, \ldots, m$ (Figure 5). Notice that in Figure 5 cross-validation is used only for model selection and it does not provide an unbiased performance estimate for the final model and so we call this procedure *cross-validation for model selection*.

In order to combine optimal model selection and unbiased performance estimation, the *cross-validation for model selection* is "nested" inside the *cross-validation for performance estimation* to obtain the *nested cross-validation* procedure (Figure 6). The dashed box in Figure 6 corresponds to *cross-validation for model selection* (steps 1.1, 1.2, and 1.3) "nested" into the steps 1 and 2 belonging to *cross-validation for performance estimation*. Since the optimized classifier is each time evaluated on a testing set not used for learning, the resulting performance estimate ρ is unbiased.

The algorithm in Figure 6 avoids the following common pitfall in estimating the performance of a diagnostic model produced by a parametric classifier: Quite often, the procedure in Figure 5 is used to identify the best parameter values and to build the final model; however, the best cross-validation performance $P(j)$, where j = argmax $P(i)$ for $i = 1, \ldots, m$ is often reported as an estimate of performance of

the final model, instead of applying a second cross-validation loop over the whole model selection procedure as in Figure 6. For a sufficiently large number of attempted parameter values, one is likely to be found that by chance alone provides a high estimate of cross-validation performance. The less the available sample is, and the more complex models the classifier can build, the more acute becomes the problem. In contrast, the described *nested cross-validation* protocol will be able to identify whether the model selection procedure is selecting values that by accident produce models that perform well on the test sets, or indeed they generalize well to unseen cases.

Gene selection

To study how dimensionality reduction can improve classification performance, we applied all classifiers with subsets of 25, 50, 100, 500, and 1000 top-ranked genes, following the example set by [Furey2000]. Genes were selected according to four gene selection methods/metrics: (1) ratio of genes between-categories to within-category sums of squares (BW) [Dudoit2002]; (2-3) signal-to-noise (S2N) scores [Golub1999] applied in a one-versus-rest (S2N-OVR) and one-versus-one (S2N-OVO) fashion; and (4) Kruskal-Wallis nonparametric one-way ANOVA (KW) [Jones1997]. The ranking of the genes was performed based on the training set of samples to avoid overfitting.

Performance metrics

We used two classification performance metrics. The first metric is accuracy since we wanted to compare our results with the previously published studies that also used this performance metric. Accuracy is easy to interpret and simplifies statistical testing. On the other hand, accuracy is sensitive to the prior class probabilities and does not fully describe the actual difficulty of the decision problem for highly unbalanced distributions. For example, it is more difficult to achieve an accuracy of 50% for a 26-class dataset *14_Tumors* with prior probability of the major class = 9.7% compared to an accuracy of 75% for a binary dataset *DLBCL* with prior of the major class = 75.3%.

The second metric is *relative classifier information* (RCI), which corrects for differences in prior probabilities of the diagnostic categories, as well as the number of categories. RCI is an entropy-based

measure that quantifies *how much the uncertainty of a decision problem is reduced by a classifier relative to classifying using the priors* [Sindwani2001].

Overall research design

To maintain the feasibility of this study, we pursued a *staged factorial design*: in stage I, we conducted a fully factorial design involving datasets and classifiers without gene selection; in stage II, we focused on the datasets for which the full gene sets yielded poor performance and applied gene selection in a factorial fashion. In addition, we optimized algorithms using accuracy only and limited the possible cardinalities of selected gene sets to only five choices as described in the subsection on gene selection.

While the above choices restricted the number of models generated, *the resulting analyses still generated more than* $2.6 \cdot 10^6$ *diagnostic models*. The total time required was 4 single-CPU months using Intel Xeon 2.4 GHz platform. Out of this set of models, only one model was selected for each combination of algorithm and dataset.

Notice that, despite the very large number of examined models, the final performance estimates are not overfitted. This is because only one model is selected per split for the estimation of the final performance and it is applied to previously *unseen cases*. Thus, regardless of how much performance is overestimated in the inner loop (which, in the worst case, may result in not choosing the best possible parameters' combination), the outer loop guarantees proper estimation of performance.

Statistical comparison among classifiers

To test that differences in accuracy between the best method (i.e. one with the largest average accuracy) and all remaining algorithms are non-random, we need a statistical comparison of observed differences in accuracies.

In machine learning, the major study about comparison of supervised classification learning algorithms is that of Dietterich which suggests using *N*-fold cross-validated paired *t*-test for comparison of *N*-fold accuracy estimates for a single dataset [Dietterich1998]. However, the author clearly admits that this test violates independence and, even more importantly, does not address how this procedure is applied to a multitude of datasets. That is why we decided to use random permutation testing which does not rely on

independence assumptions and can be straightforwardly applied to several datasets [Good2000]. For every algorithm X, other than the best algorithm Y, we performed the following steps: (I) We defined the null hypothesis H_0 to be: classification algorithm X is as good as Y, i.e. the accuracy of the best algorithm Y minus the accuracy of algorithm X is zero. (II) We obtained the permutation distribution of Δ_{XY}, the estimator of the true unknown difference between accuracies of the two algorithms, by repeatedly rearranging the outcomes of X and Y at random. (III) We computed the cumulative probability (p-value) of Δ_{XY} being greater than or equal to observed difference $\hat{\Delta}_{XY}$ over 10,000 permutations. If the p-value was smaller than 0.05, we rejected H_0 and concluded that the data support that algorithm X is not as good as Y in terms of classification accuracy, and this difference is not due to sampling error. In order to increase the resolution of simulated sampling distribution, we computed a single value of accuracy over all samples from all datasets. In other words, we treated classifier's predictions from all 11 datasets as if we had one large dataset with samples from all individual datasets.

Implementations of algorithms

We used the MC-SVM algorithms implemented by the LibSVM team [Chang2003], since they use state-of-the-art optimization methods SMO [Platt1999] and TRON [Lin1999] for the solution of MC-SVM problems. The implementation of NN and PNN classifiers was based on the Matlab Neural Networks toolbox [Demuth2001]. We applied Matlab R13 implementation of the CART algorithm [Murthy1997] for DT, and we used our own implementations of KNN, ensemble classification, gene selection, as well as statistical comparison algorithms.

CHAPTER IV

RESULTS OF ALGORITHMIC EVALUATION

Classification without gene selection

The performance results of experiments without gene selection obtained using Design I (nested stratified 10-fold cross-validation) with accuracy and RCI as a performance metric are shown in Tables 3 and 4, respectively. Results for Design II are almost identical and are provided only in Appendix E, section 1 [Statnikov2005]. The fact that we obtained similar results with two different experimental designs is evidence in favor of the reliability of performance estimation procedures.

Table 3: Performance results (accuracies) without gene selection obtained using a nested stratified 10-fold cross-validation design (Design I). These results are further improved by gene selection (see Figure 7). The last column in the bottom table reports average performance computed over datasets.

Multicategory classification

Method		9_Tumors	11_Tumors	14_Tumors	Brain_Tumor1	Brain_Tumor2	Leukemia1
MC-SVM	OVR	65.10%	94.68%	74.98%	**91.67%**	77.00%	**97.50%**
	OVO	58.57%	90.36%	47.07%	90.56%	**77.83%**	97.32%
	DAGSVM	60.24%	90.36%	47.35%	90.56%	**77.83%**	96.07%
	WW	62.24%	94.68%	69.07%	90.56%	73.33%	**97.50%**
	CS	**65.33%**	**95.30%**	**76.60%**	90.56%	72.83%	**97.50%**
non-SVM	KNN	43.90%	78.51%	50.40%	87.94%	68.67%	83.57%
	NN	19.38%	54.14%	11.12%	84.72%	60.33%	76.61%
	PNN	34.00%	77.21%	49.09%	79.61%	62.83%	85.00%

		Multicategory classification			*Binary classifcation*		**Averages**
Method		Leukemia2	Lung_Cancer	SRBCT	Prostate_Tumor	DLBCL	
MC-SVM	OVR	**97.32%**	96.05%	**100.00%**	**92.00%**	**97.50%**	**89.44%**
	OVO	95.89%	95.59%	**100.00%**	**92.00%**	**97.50%**	85.70%
	DAGSVM	95.89%	95.59%	**100.00%**	**92.00%**	**97.50%**	85.76%
	WW	95.89%	95.55%	**100.00%**	**92.00%**	**97.50%**	88.03%
	CS	95.89%	**96.55%**	**100.00%**	**92.00%**	**97.50%**	89.10%
non-SVM	KNN	87.14%	89.64%	86.90%	85.09%	86.96%	77.16%
	NN	91.03%	87.80%	91.03%	79.18%	89.64%	67.73%
	PNN	83.21%	85.66%	79.50%	79.18%	80.89%	72.38%

Table 4: Performance results (RCI) without gene selection obtained using a nested stratified 10-fold cross-validation design (Design I). These results are further improved by gene selection (see Figure 8). The last column in the bottom table reports average performance computed over datasets.

Multicategory classification

Method		9_Tumors	11_Tumors	14_Tumors	Brain_Tumor1	Brain_Tumor2	Leukemia1
MC-SVM	OVR	77.00%	95.80%	90.53%	**82.31%**	77.49%	**93.90%**
	OVO	78.24%	92.24%	64.99%	80.77%	**80.27%**	93.05%
	DAGSVM	**78.67%**	92.24%	65.64%	80.77%	**80.27%**	90.16%
	WW	76.22%	95.80%	86.30%	80.77%	74.75%	**93.90%**
	CS	77.25%	**96.20%**	**90.96%**	80.77%	74.44%	**93.90%**
non-SVM	KNN	63.38%	83.93%	82.73%	67.86%	64.48%	64.45%
	NN	65.57%	67.80%	16.24%	61.42%	62.49%	53.06%
	PNN	55.59%	81.39%	81.40%	43.86%	61.73%	68.85%

		Multicategory classification			***Binary classifcation***		**Averages**
Method		Leukemia2	Lung_Cancer	SRBCT	Prostate_Tumor	DLBCL	
MC-SVM	OVR	**94.42%**	89.45%	**100.00%**	**71.14%**	**90.91%**	**87.54%**
	OVO	92.35%	87.95%	**100.00%**	**71.14%**	**90.91%**	84.72%
	DAGSVM	92.35%	87.95%	**100.00%**	**71.14%**	**90.91%**	84.55%
	WW	91.90%	87.46%	**100.00%**	**71.14%**	**90.91%**	86.29%
	CS	91.90%	**91.40%**	**100.00%**	**71.14%**	**90.91%**	87.17%
non-SVM	KNN	76.95%	68.48%	80.71%	51.09%	63.08%	69.74%
	NN	78.02%	64.97%	87.50%	33.25%	58.36%	58.97%
	PNN	73.51%	59.72%	68.92%	39.22%	38.23%	61.13%

Notably, RCI performance metric revealed different results compared to accuracy. For example, the best RCI for *14_Tumors* dataset is 90.96% and for *Prostate_Tumor* is 71.14%. In contrast, when accuracy was employed, we obtained 76.60% in *14_Tumors* and 92% in *Prostate_Tumor*. The difference can be explained by the difficulties of the classification problems - *14_Tumors* is much harder (it has 26 classes with prior of the most frequent class 9.7%, see Table 2) than *Prostate_Tumor* (it is a binary problem with prior 51%, see Table 2).

According to Table 3, in 8 out of 11 datasets, MC-SVMs perform cancer diagnoses with accuracies > 90%. The results for RCI performance metric are similar (Table 4): in 7 out of 11 datasets, MC-SVMs yield diagnostic performance with RCI > 90%. Overall, all MC-SVMs outperform KNN, NN, and PNN significantly. The only exception is KNN and PNN applied to *14_Tumors* dataset which outperformed OVO and DAGSVM, but still were unable to perform better than more robust MC-SVM techniques, OVR,

WW, and CS. The superior classification performance of the SVM-based methods compared to KNN, NN, and PNN reflects that SVMs are less sensitive to the *curse of dimensionality* and more robust to a small number of high-dimensional gene expression samples than other non-SVM techniques [Aliferis2003b]. A more detailed explanation of this matter follows in the next subsection on classification results with gene selection.

Among MC-SVMs, OVR, WW, and CS yield the best results and are not statistically significant from each other at the 0.05 level (Appendix E, section 2 [Statnikov2005]). On the other hand, OVO, DAGSVM, KNN, PNN, and NN have poorer performance than the above methods to a statistically significant degree. OVO and DAGSVM perform very similar, which is due to the fact that both MC-SVM methods use the same binary SVM classifiers. We conjecture that OVO and DAGSVM perform worse than other MC-SVM methods because both algorithms are based on one-versus-one binary classifiers that use only a fraction of total training samples at a time (samples that belong to two classes) and ignore information about distribution of the remaining examples which may be significant for the classification. In case of large sample sizes, we expect MC-SVMs OVO and DAGSVM to perform as good as WW, CS, and OVR (for example, see [Hsu2002]).

According to Tables 3 and 4 and results of application of the binary SVM implementation SVMLight [Joachims1999] to *DLBCL* and *Prostate_Tumor* datasets (not shown here), we conclude that our implementations of MC-SVM algorithms perform the same classifications as binary SVMs and, hence, handle binary diagnostic problems appropriately as expected.

We tried to explain classification performance of the best MC-SVM algorithms OVR, WW, and CS by fitting inverse power curves motivated by the ideas described in [Cortes1993]. We found that in high-dimensional spaces of microarray gene expression data, the number of samples divided by the product of the number of variables times the number of categories explains observed classification accuracies in the datasets. When we reduced dimensionality by gene selection, or employed RCI performance metric, or used other classification algorithms, this behavior disappeared. More details can be found in Appendix E, section 3 [Statnikov2005]. It is important to note that curve fitting procedure used in this study is very simplistic since it does not incorporate predictors describing degree of biological difficulty and assumes that datasets and learning tasks used in this study are representative.

Table 5: Total time of classification experiments without gene selection for all 11 datasets and two experimental designs.

Method		Time in hours	
		Design I	Design II
MC-SVM	OVR	19.28	772.43
	OVO	9.86	388.11
	DAGSVM	9.93	390.97
	WW	7.95	290.77
	CS	7.88	289.01
non-SVM	KNN	**3.40**	**109.60**
	NN	195.68	N/A
	PNN	186.19	N/A

Finally, we also analyzed execution time for all learning algorithms applied without gene selection (Table 5). The fastest MC-SVM methods CS and WW took 7.95 and 7.88 hours for Design I and 289.01 and 290.77 hours for Design II, respectively. The slowest MC-SVM technique OVR completed within 19.28 hours for Design I and 772.43 hours for Design II. This technique is slowest among MC-SVM algorithms since it constructs several classifiers repeatedly employing all samples from the training dataset. The fastest overall algorithm KNN took 3.40 hours for Design I and 109.60 hours for Design II, while the slowest overall algorithms NN and PNN took 195.68 hours and 186.19 hours, respectively, for Design I. All experiments were executed in the Matlab R13 environment on eight Intel Xeon 2.4GHz dual-CPU workstations connected in a cluster.

Classification with gene selection

The summary of application of the four gene selection methods BW, S2N-OVR, S2N-OVO, and KW to the four most "challenging" datasets *9_Tumors*, *14_Tumors*, *Brain_Tumor1*, and *Brain_Tumor2* using accuracy and RCI as a performance metric is presented in Figures 7 and 8, respectively. It should be noted that a more rigorous way to do gene selection with validation of number of genes and gene selection method is very expensive computationally (that is why it was not pursued here as explained in the methods chapter).

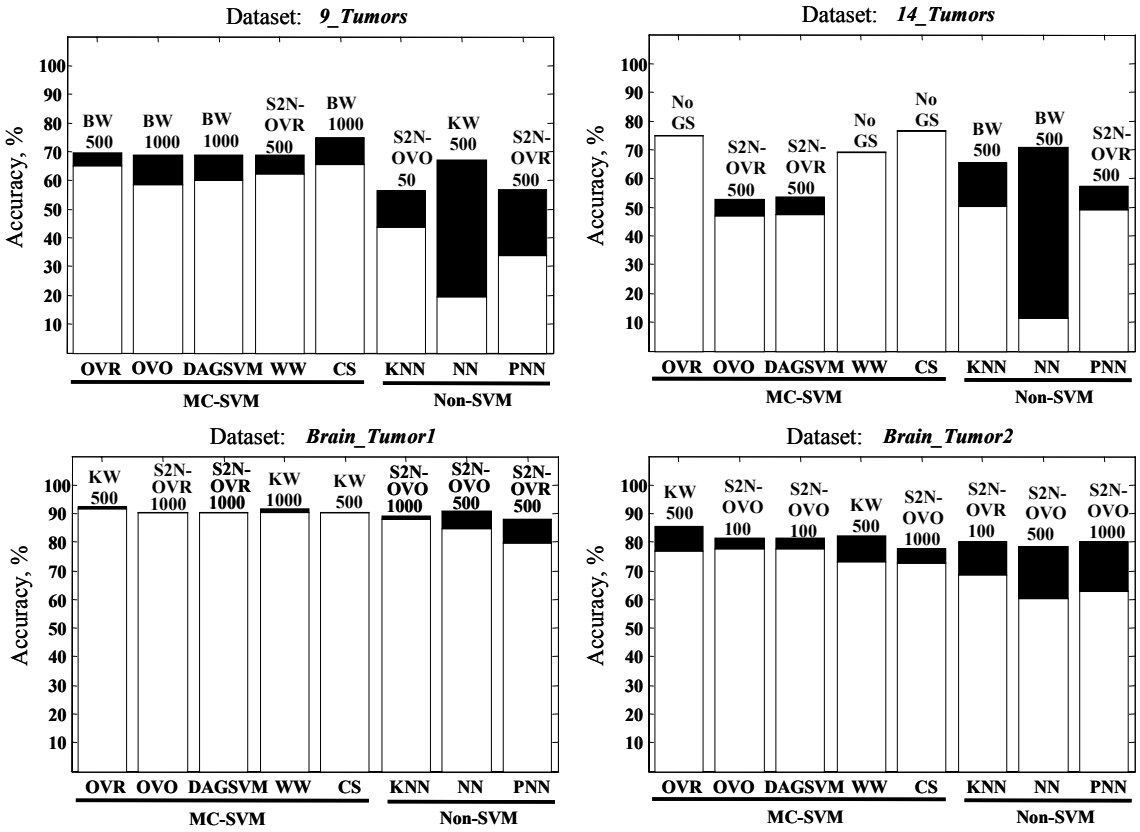

Figure 7: Performance results (accuracies) of classification experiments with gene selection obtained using a nested stratified 10-fold cross-validation design (Design I) for 4 datasets: 9_Tumors, 14_Tumors, Brain_Tumor1, Brain_Tumor2. The white bars correspond to classification results without gene selection. The black bars correspond to results with gene selection. The text above each bar indicates the optimal combination of gene selection method and number of genes for a specific classifier. The abbreviation "No GS" stands for "No gene selection".

The results show that gene selection significantly improves classification performance of non-SVM learners. In particular, for some datasets, accuracy is improved by up to 14.97%, 59.78%, 22.67% and RCI is improved by up to 19.52%, 69.95%, 34.98% for KNN, NN, and PNN, respectively. Gene selection also improves accuracy of MC-SVMs up to 9.53% and, hence, improves accuracy of the overall best classifier. Although KNN, NN, and PNN perform closer to MC-SVMs, three MC-SVM algorithms, OVR, WW, and CS, still outperform non-SVM methods in most of cases. We also found that these three MC-SVM methods are not statistically significant from each other and NN at the 0.05 level (Appendix E, section 2 [Statnikov2005]). The remaining algorithms, MC-SVMs OVO and DAGSVM, KNN and PNN, have statistically significant poorer performance. Finally, neither of the four gene selection methods performs significantly better than the other ones.

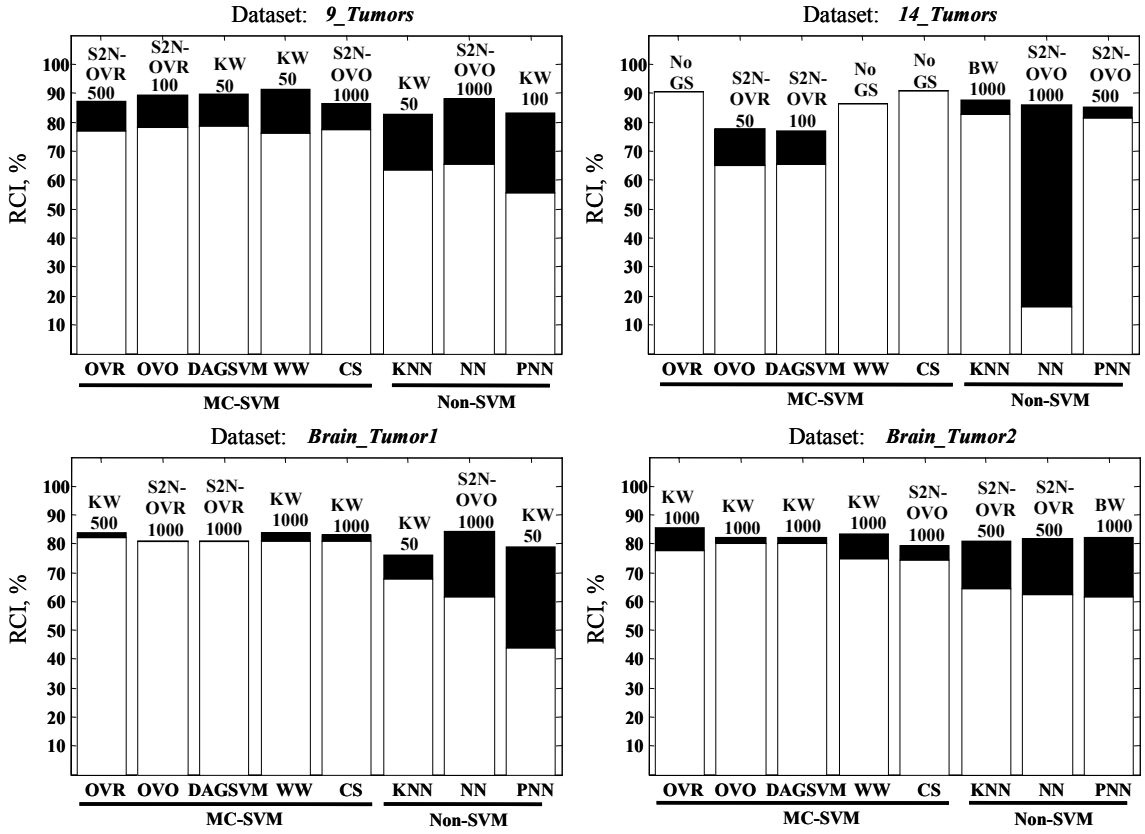

Figure 8: Performance results (RCI) of classification experiments with gene selection obtained using a nested stratified 10-fold cross-validation design (Design I) for 4 datasets: 9_Tumors, 14_Tumors, Brain_Tumor1, Brain_Tumor2. The white bars correspond to classification results without gene selection. The black bars correspond to results with gene selection. The text above each bar indicates the optimal combination of gene selection method and number of genes for a specific classifier. The abbreviation "No GS" stands for "No gene selection".

As we have empirically found, the non-SVM methods KNN, PNN, and NN benefit significantly more than MC-SVMs from gene selection. A number of observations can explain this behavior: In high-dimensional spaces, KNN has high variance of the prediction since all training points are located close to the edge of the sample [Hastie2001]. Furthermore, many irrelevant variables in the data dominate distances between samples which presents a significant problem for prediction [Mitchell1997]. PNN encounter problems similar to KNN, in particular because they rely on Parzen windows for density estimation which generally require exponential sample to the data dimensionality [Duda2001]. Backpropagation Neural Networks are sensitive to high dimensionality for at least two reasons: First notice, that the larger the number of variables, the larger is the number of weights in this type of neural network. Because of this, (I) there may be more local minima in the error landscape and it is thus more probable for backpropagation to get "trapped" in one of them, and (II) the model space becomes exponentially larger with the addition of

each weight, and so it becomes harder to identify a model that generalizes. In comparison, the family of SVMs allows for effective optimization search procedure by utilizing convex formulation with a single optimum justified by Statistical Learning Theory [Vapnik1998]. Furthermore, SVMs seem relatively insensitive to the curse of dimensionality, possibly due to the specific regularization mechanism they employ. In particular, this is reflected by the following: (I) many established generalization bounds do not depend on the data dimensionality [Herbrich2002], and (II) even linear SVMs assign zero weights to irrelevant variables [Hardin2004]. On the other hand, the SVM algorithm may assign non-zero weights to weakly relevant variables [Hardin2004] which explains why effective variable selection can still improve SVM classification.

Ensemble classification

For the case when no gene selection was performed, ensembles do not outperform the best non-ensemble methods with the exception of the Decision Trees ensemble classifier for *Brain_Tumor2* dataset, which improves classification accuracy by 1.67%. Other ensembles often achieve similar performance to the best non-ensemble methods (Appendix E, section 4 [Statnikov2005]).

Next, we considered three datasets *9_Tumors*, *Brain_Tumor1*, and *Brain_Tumor2* where we previously observed improvement of classification performance by gene selection. For each dataset we selected a subset of genes yielding the best classification performance (over all gene selection methods, subsets of genes, and learning algorithms) and constructed combined classifiers. According to results, ensembles perform worse than the best non-ensemble models (Appendix E, section 4 [Statnikov2005]).

We believe that in our study, ensemble classifiers did not improve final classification performance for the following two reasons: First, samples misclassified by non-SVM algorithms are almost always a strict superset of samples misclassified by MC-SVM algorithms. Second, SVM algorithms are fairly stable in a sense that small changes in the training data do not result in large changes in the predictive model's behavior [Kutin2002], and according to [Dudoit2002] stable algorithms do not usually tend to benefit from the ensemble classification.

Comparison with previously published results

Most of the results from this study are not exactly comparable with the analyses provided in the original studies due to differences in the setup of dataset/learning task, experimental design, gene selection, classifiers, etc. that vary from study to study. However, the reported results in the literature confirm that MC-SVMs as applied here perform equally as well, or even better, compared to previously published models on the same datasets (Appendix E, section 5 [Statnikov2005]).

CHAPTER V

DISCUSSION AND LIMITATIONS OF ALGORITHMIC EVALUATION

One of the limitations of the present study is that we use accuracy and RCI as our performance measures. These metrics do not incorporate information about confidence of the predictions as well as different misclassification costs of diagnostic categories. On the other hand, accuracy was used in published studies and it is easy to interpret and simplifies statistical comparison, while RCI is insensitive to prior class probabilities and accounts for the difficulty of the learning problem. There are currently no mature performance metrics applicable for multiclass domains and suitable for our classifiers with both confidence information and consideration of misclassification costs. Initial attempts were introduced by [Lee2003], [Mossman1999], and [Ferri2003], however much needs to be done before we obtain a workable metric for experiments such as those presented here.

As we mentioned, the choice of KNN, NN, and PNN classifiers as the baseline techniques was grounded on prior successful applications to gene expression based cancer diagnosis (e.g., [Khan2001], [Ramaswamy2001], [Pomeroy2002], [Nutt2003], [Signh2002], and [Berrar2003]). We have also experimented with other non-SVM classifiers, such as Decision Trees (DT) [Murthy1997] and Weighed Voting (WV) classifiers applied both in OVR and OVO fashion ([Golub1999], [Ramaswamy2001], and [Yeang2001]). We found that both with and without gene selection, DT perform significantly worse than MC-SVMs, worse than KNN, and similarly or worse than NN and PNN. Likewise, WV classifiers are significantly outperformed by MC-SVMs, KNN, NN, and PNN. More details about these additional experiments with DT and WV classifiers can be found in Appendix E, section 6 [Statnikov2005].

A particularly interesting direction for future research is to improve our existing gene selection procedures with selection of the "optimal" number of genes by cross-validation[2]. Furthermore, we are interested in applying various multivariate Markov blanket and local neighborhood algorithms which have been previously successfully applied to cancer gene expression and several other domains and do guarantee efficient identification of a set of relevant attributes under fairly broad assumptions ([Aliferis2003c], [Tsamardinos2003]).

[2] The functionality to cross-validate number of genes is already implemented in the software system.

To the best of our knowledge, currently there exists only one work aimed at evaluation of MC-SVM algorithms [Hsu2002]. That study is outside of the realm of biomedicine since [Hsu2002] considered such classification tasks as wine recognition, letter recognition, shuttle control, etc. with the number of variables ranging from 4 to 180 and sample sizes greater than 500 in the majority of tasks, which is not typical for microarray cancer gene expression datasets. However, it is worthwhile to mention the major conclusions of that evaluation. The authors empirically found the following: (1) using a Gaussian radial basis kernel, all MC-SVM methods perform similarly; (2) DAGSVM and OVO have the fastest training time; and (3) for problems with large sample size, WW and CS yield fewer support vectors compared to OVR, OVO, and DAGSVM. The work by Hsu is complementary to ours and is not overlapping due to significant differences in the problem domain and dataset characteristics. For example, in our experiments, MC-SVM methods OVO and DAGSVM achieved inferior classification performance compared to other MC-SVM algorithms.

CHAPTER VI

CONCLUSIONS OF ALGORITHMIC EVALUATION

We conducted the most comprehensive systematic evaluation to date of multicategory diagnosis algorithms applied to the majority of multicategory cancer-related gene expression human datasets publicly available. Based on results of this evaluation, the following conclusions can be drawn:

- For multicategory classification of cancer from microarray gene expression data, Support Vector Machines (SVMs) are the best performing family among the tested algorithms outperforming K-Nearest Neighbors, Backpropagation Neural Networks, Probabilistic Neural Networks, Decision Trees, and Weighted Voting classifiers to a statistically significant degree;
- Among multicategory Support Vector Machines, the best performing techniques are: one-versus-rest, the method by Weston and Watkins, and the method by Crammer and Singer;
- The diagnostic performance can be moderately improved for SVMs and significantly improved for the non-SVM methods by gene selection;
- Ensemble classification does not improve performance of the best non-ensemble diagnostic models;
- The obtained results favorably compare with the primary literature on the same datasets.

We believe that practitioners and software developers should take note of these results when considering construction of decision support systems in this domain, or when selecting algorithms for inclusion in related analysis software.

CHAPTER VII

SYSTEM FUNCTIONALITY AND DEVELOPMENT

Based on results of conclusions of algorithmic evaluation, we have developed a system, *GEMS* (Gene Expression Model Selection). The system provides to the user an implementation of all and only the best performing learning algorithms in this domain. Given a microarray dataset on input, the system can automatically perform one the following tasks:

I. Generate a classification model optimizing the parameters of classification and gene selection algorithms as well as the choice of the classifier and gene selection methods using *cross-validation for model selection* (Figure 5);

II. Estimate classification performance of the optimized model by *nested cross-validation* (Figure 6);

III. Perform tasks I and II, i.e. generate a classification model and estimate its performance;

IV. Apply an existing model to a new set of patients.

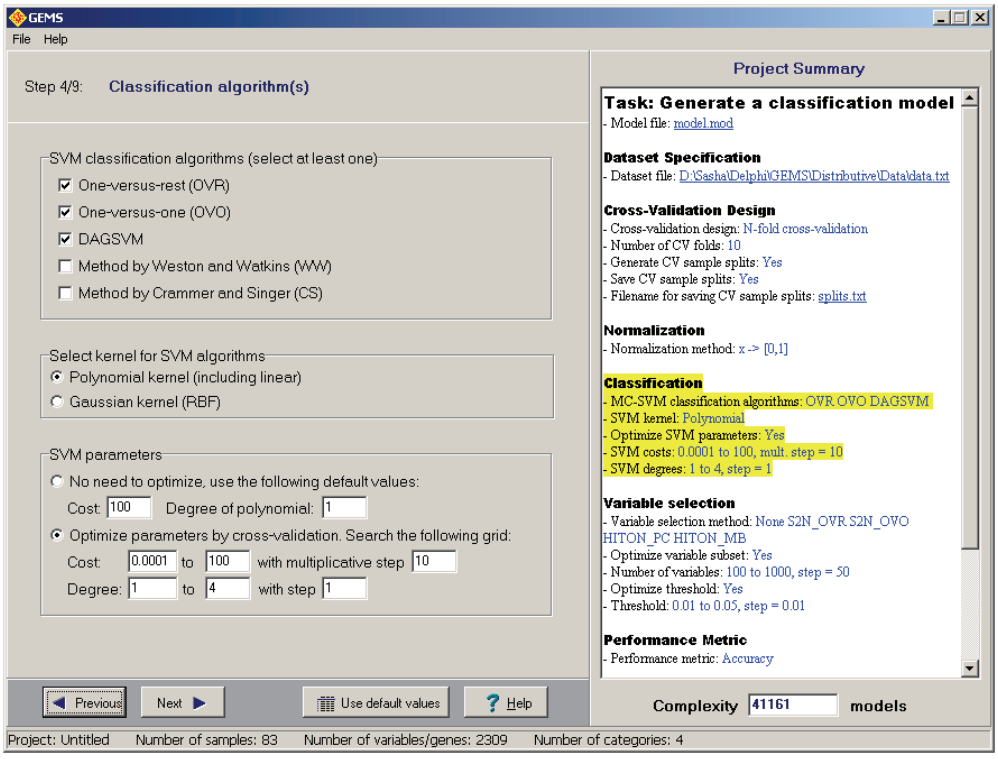

Figure 9: An example screen-shot of *GEMS*. The left part of the screen contains options for the current analysis step (classification algorithm). The summary of the entire project is shown in the right part of the screen.

Table 6: List of algorithms currently implemented in the system *GEMS*.

Classification algorithms	**Normalization techniques**	**Computational experimental design**
· Multi-Class SVM: One-Versus-Rest	· For every gene x → [a, b]	· Leave-one-out cross-validation for performance estimation (outer loop) and N-fold cross-validation for model selection (inner loop)
· Multi-Class SVM: One-Versus-One	· **For every gene x → [x - Mean(x)]/Std(x)**	
· Multi-Class SVM: DAGSVM	· **For every gene x → x/Std(x)**	· N-fold cross-validation for performance estimation (outer loop) and (N-1)-fold cross-validation for model selection (inner loop)
· Multi-Class SVM by Weston & Watkins	· **For every gene x → x/Mean(x)**	
· Multi-Class SVM by Crammer & Singer	· **For every gene x → x/Median(x)**	· Leave-one-out cross-validation for model selection
Gene selection methods	· **For every gene x → x/‖x‖**	· N-fold cross-validation for model selection
· Signal-to-noise ratio in one-versus-rest fashion	· **For every gene x → x - Mean(x)**	**Performance metrics**
· Signal-to-noise ratio in one-versus-one fashion	· **For every gene x → x - Median(x)**	· Accuracy
· Kruskal-Wallis nonparametric one-way ANOVA	· **For every gene x → |x|**	· Relative classifier information (entropy-based performance metric)
· Ratio of genes between-categories to within-category sum of squares	· **For every gene x → x + |x|**	
· **HITON_PC**		· **Area under ROC curve (AUC)**
· **HITON_MB**	· **For every gene x → Log(x)**	

In order to execute the tasks mentioned above, the user may select the type of the experimental design (*N*-fold cross-validation or leave-one-out cross-validation), the algorithm(s) to be used for classification, gene selection, and normalization, and the ranges of parameters over which optimization should take place. Table 6 summarizes all implemented algorithms. As the system evolved and based on discussions with our biomedical colleagues, we added new functionality to the system, namely, several simple gene expression normalization methods, area under ROC curve performance metric (for binary diagnostic problems), and two state of the art local causal discovery algorithms ([Aliferis2003c] and [Tsamardinos2003]) shown with boldface in Table 6. To guide the user's choices according to the available computational power and time, the system outputs the number of models to be generated while the user is selecting analysis options. *GEMS* provides an intuitive wizard-like user interface abstracting the microarray data analysis process and not requiring users to be experts in data analysis. Each step in the interface contains a form with options for a specific stage of analysis (Figure 9):

- overall task selection
- dataset specification
- cross-validation design
- normalization
- classification
- gene selection
- performance estimation
- logging
- report generation
- execution of analysis

Since the system can perform one out of four tasks outlined above, each task corresponds to a different sequence of steps. The overall software architecture of *GEMS* is shown in Figure 10. The system implements a client-server architecture consisting of a computational engine and an interface client. The computational engine is separated from the client and consists of intercommunicating functional units corresponding to different aspects of analysis. Upon completion of analysis, a detailed report is generated in HTML format with links to system input and output files as well as links to NCBI website with

information on selected genes. The *GEMS* graphics user interface is implemented using Borland Delphi 6.0 and the computational engine is programmed in Mathworks Matlab 6.5.1 and Microsoft Visual C++ 6.0.

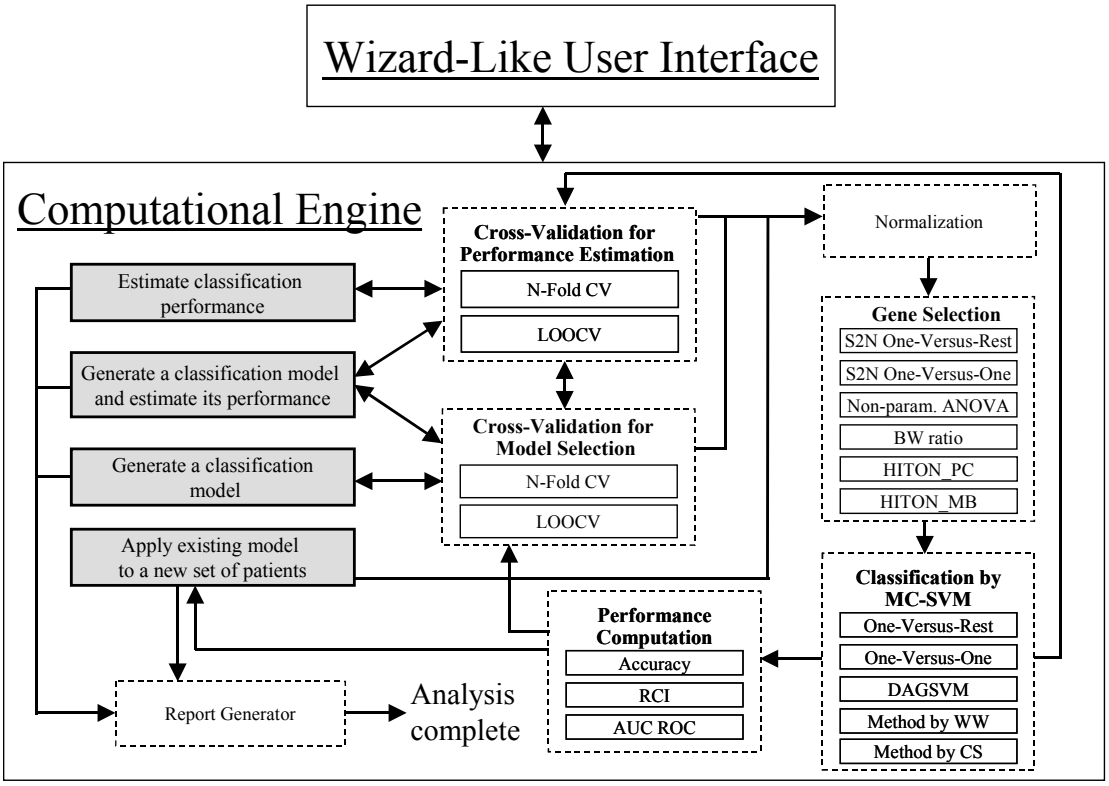

Figure 10: Software architecture of *GEMS*.

CHAPTER VIII

PRELIMINARY EVALUATION OF THE SYSTEM

In order to debug the interface and the algorithmic engine of the system, we repeated most of experiments performed in the algorithmic evaluation using *GEMS*, and we found that published results completely matched the system outputs. Next, we performed two studies to evaluate the system. First, we applied *GEMS* to several microarray datasets, not included in our previous study, and compared the resulting system performance with the published models. Second, we performed a cross-dataset evaluation of the system. This involved using the system to build a classifier from a gene-expression dataset, estimating its cross-validation performance on the same dataset, and then applying that classifier to a different dataset (using the same genes and diagnostic target) produced by an independent research group.

Application of GEMS to new datasets

We selected five human cancer microarray gene expression datasets to test our system: *6_Tumors* [Shedden2003], *Leukemia3* [Yeoh2002], *Lung_Cancer2* [Beer2002], *DLBCL2* [Savage2003], and *Lung_Cancer3*[3] [Gordon2003] (see Table 7 for description of datasets and diagnostic tasks). All five datasets were produced using Affymetrix oligonucleotide technology and processed as in our algorithmic evaluation. None of these datasets was included in our previous studies.

The results of application of *GEMS* to these datasets are presented in Table 8[4]. The analyses completed within 10-30 minutes per dataset and yielded performance results comparable or better than ones obtained by human analysts and previously published in literature.

[3] This dataset contains the same adenocarcinoma samples as in previously analysed *Lung_Cancer* data [Bhattacharjee2001]. However, *Lung_Cancer3* dataset contains additional mesothelioma samples and is now used to solve a different diagnostic problem (adenocarcinoma vs. mesothelioma) compared to diagnosis developed using *Lung_Cancer* data (adenocarcinoma vs. squamous vs. small-cell lung cancer vs. pulmonary carcinoids vs. normal tissues).

[4] Notice that for *DLCBL2* dataset Savage *et al.* reported 88.7% accuracy in their paper [Savage2003], however in the supplement the authors clarified that their classification procedure might be biased, since they optimized their classifier based on testing sets. When experiments were repeated using nested cross-validation, the authors obtained 83.9% accuracy (see supplement to [Savage2003]).

Table 7: Cancer-related human gene expression datasets used for preliminary evaluation of *GEMS* system. The column "Max. prior" indicates the prior probability of the dominant diagnostic category.

Dataset name	Diagnostic Task	Number of				Max. prior
		Samples	Variables (genes)	Categories	Variables / Samples	
6_Tumors	6 various human tumor types	353	7069	6	20	32.0%
Leukemia3	6 types of leukemia	248	12135	6	49	31.9%
Lung_Cancer2	Lung cancer and normal tissues	96	7129	2	74	89.6%
Lung_Cancer3	Mesothelioma and adenocarcinoma	181	12533	2	69	82.9%
DLBCL2	Diffuse large b-cell lymphomas (DLBCL) and mediastinal large B-cell lymphomas (MLBCL)	210	32404	2	154	83.8%

Table 8: Results of application of *GEMS* to five microarray datasets not employed for algorithmic evaluation.

Dataset name	GEMS classification accuracy	Published classification accuracy
6_Tumors	97.2%	96.0%
Leukemia3	98.4%	98.4%
Lung_Cancer2	100.0%	100.0%
Lung_Cancer3	99.4%	99.3%
DLBCL2	87.1%	83.9%

Cross-Dataset evaluation of the system

Many researchers believe that even though small cross-validation error is an important finding, it still requires further validation on an independent data [Simon2003]. There are two reasons for doing this: (1) cross-validation performance estimates in very small samples may have large variance, and (2) the dataset may not be representative of the general population. Therefore, we used *GEMS* to conduct two analyses with construction and evaluation of the classifier on one dataset and consecutive independent validation on another data. The results are summarized in Table 9, and the text below describes our experiments and findings in detail.

Table 9: Results of cross-dataset experiments: first, we used a dataset to build a diagnostic model and to estimate its future performance by cross-validation, and then we applied this model and computed its performance on a different dataset. More details on datasets used for these experiments are provided in Tables 2 and 7.

Dataset used for construction of a classification model		Performance estimate of the model* (AUC, %)	Dataset used for independent validation of the classification model		Performance on the independent dataset (AUC, %)
Name	**Distribution of samples**		**Name**	**Distribution of samples**	
Lung_Cancer	186 tumors 17 normals	100.00%	*Lung_Cancer2*	86 tumors 10 normals	100.00%
Leukemia2	24 ALL 28 AML	100.00%	*Leukemia1*	47 ALL 25 AML	99.15%

* This performance estimate was obtained by nested cross-validation on the dataset used for construction of the model.

First, we used the *Lung_Cancer* [Bhattachajee2001] and *Lung_Cancer2* [Beer2002] datasets with the diagnostic task to differentiate between cancerous and normal tissues. The *Lung_Cancer* dataset contains 186 tumor and 17 normal samples, and *Lung_Cancer2* dataset contains 86 tumor and 10 normal samples. The datasets were produced using different microarray technologies: *Lung_Cancer* dataset was obtained using Affymetrix Human Genome U95A chips with 12,600 oligonucleotide probes, while *Lung_Cancer2* dataset was obtained using Affymetrix HuGeneFL chips with 7,129 oligonucleotide probes. The mapping of 6,623 probes from HuGeneFL to 7,094 probes from Human Genome U95A was derived using Affymetrix array comparison spreadsheets [Jiang2004]. Next, we used *GEMS* to generate a classification model and estimate its performance in a nested cross-validation fashion using *Lung_Cancer* dataset. We decided to use area under ROC curve (AUC) as a performance metric since both datasets are not balanced in terms of distribution of cancerous and normal samples. *GEMS* created a classification model and estimated its cross-validation performance to be 100% AUC. When this model was applied to *Lung_Cancer2* data, the actual performance was again 100% AUC. We emphasize that, *Lung Cancer2* was never seen by the model neither during training, nor during the performance estimation phase.

Similarly, we used *Leukemia1* [Golub1999] and *Leukemia2* [Armstrong2002] datasets with the goal to build a classifier to predict whether a patient has acute lymphoblastic leukemia (ALL) or acute myelogenous leukemia (AML). The *Leukemia1* dataset contains 47 ALL and 25 AML samples, and *Leukemia2* dataset contains 24 ALL and 28 AML samples. Again, the datasets were produced using different microarray technologies: *Leukemia2* dataset was obtained using Affymetrix Human Genome U95A chips, while *Leukemia1* dataset was obtained using Affymetrix HuGeneFL chips. We used a similar approach as described above to map probes between datasets. Next, we fed *Leukemia2* dataset to *GEMS* to create a classification model and estimate its performance in a nested cross-validation fashion (AUC = 100%). When this model was applied to *Leukemia1* data, the final classification performance was 99.15% AUC.

In summary, the performance of the models as estimated by the system on one dataset is approximately equal to its performance on the independent dataset. This provides further confidence on the use of nested-cross-validation design both for generating the models and for estimating their performance.

CHAPTER IX

LIMITATIONS AND FUTURE RESEARCH

Although *GEMS* is a highly robust system for cancer diagnosis and discovery, it can be improved in several ways. Since many biomedical researchers and practitioners are interested in causal discovery, we are planning to extend and perform an evaluation of the computational causal discovery algorithms implemented in the system. The system evaluation presented in this paper was laboratory based with authors functioning as users of the system. In the future we plan to conduct a fielded evaluation of the system, ideally, with various types of users from different institutions and organizations. We also believe that gene selection capabilities of the system can be extended by SVM-based gene selection, such as the RFE algorithm [Guyon2002], and additional Markov-blanket based techniques ([Aliferis2003c] and [Tsamardinos2003]). The current version of *GEMS* communicates with SVM classifiers by a file input/output interface. A dynamic linked library or similar interface can provide significant speed-up of *GEMS* by eliminating necessity to write and read multi-megabyte microarray data files. Finally, the output report produced by the system provides minimal links to existing knowledge about genes. In particular, it will be useful to link the report on selected genes to GO terms and known pathways and interactions.

CHAPTER X

CONCLUSION

In this work we described *GEMS* (Gene Expression Model Selector), a system for automated development and evaluation of cancer diagnostic models and biomarker discovery from microarray gene expression data. Unlike past efforts, this system is informed by a comparative evaluation of many classification and related algorithms (e.g., cross-validation, gene selection, etc) applicable for this task and domain. In a preliminary evaluation of the system with 5 cancer gene expression datasets not employed for the algorithmic comparison, *GEMS* completed the analysis of each dataset within 10-30 minutes and the output model performed as well as or better than previously published models obtained by human analysts. Also, we used this system to perform cross-dataset analysis of cancer diagnostic models using two pairs of different datasets corresponding to two different diagnostic tasks. We found that the diagnostic models obtained by *GEMS* in one dataset generalize well in data from a different laboratory and that nested cross-validation performance estimates well approximate the error obtained by the independent validation. The system is available for download from http://www.gems-system.org free of charge for non-commercial use.

REFERENCES

[Aliferis2003a] Aliferis C.F., I. Tsamardinos, P. Massion, A. Statnikov, N. Fananapazir, D. Hardin. "Machine Learning Models For Classification Of Lung Cancer and Selection of Genomic Markers Using Array Gene Expression Data", In Proceedings of the 16th International Florida Artificial Intelligence Research Society (FLAIRS) Conference, 2003.

[Aliferis2003b] Aliferis C.F., I. Tsamardinos, P. Massion, A. Statnikov, D. Hardin. "Why Classification Models Using Array Gene Expression Data Perform So Well: A Preliminary Investigation Of Explanatory Factors", In Proceedings of the 2003 International Conference on Mathematics and Engineering Techniques in Medicine and Biological Sciences (METMBS), 2003.

[Aliferis2003c] Aliferis C.F., I. Tsamardinos, A. Statnikov. "HITON, A Novel Markov Blanket Algorithm for Optimal Variable Selection", In Proceedings of the 2003 American Medical Informatics Association (AMIA) Annual Symposium, 2003.

[Armstrong2002] Armstrong S., et al. "MLL translocations specify a distinct gene expression profile that distinguishes a unique leukemia", Nature Genetics, volume 30, January 2002.

[Balmain2003] Balmain A, Gray J, Ponder B. The genetics and genomics of cancer. Nat Genet. 2003 Mar;33 Suppl:238-44.

[Beer2002] Beer DG, Kardia SL, Huang CC, Giordano TJ, Levin AM, Misek DE, Lin L, Chen G, Gharib TG, Thomas DG, Lizyness ML, Kuick R, Hayasaka S, Taylor JM, Iannettoni MD, Orringer MB, Hanash S. Gene-expression profiles predict survival of patients with lung adenocarcinoma. Nat Med. 2002 Aug;8(8):816-24.

[Berrar2003] Berrar D. "Multiclass Cancer Classification Using Gene Expression Profiling And Probabilistic Neural Networks", In Proceedings of the Pacific Symposium on Biocomputing (PSB), 2003.

[Bhattacherjee2001] Bhattacherjee, A., et al. "Classification of human lung carcinomas by mRNA expression profiling reveals distinct adenocarcinoma subclasses", Proc Natl Acad Sci U S A , 2001 Nov 20;98(24):13790-5.

[Causton2003] Causton HC, Quackenbush J, Brazma A. Microarray Gene Expression Data Analysis: A Beginner's Guide. Blackwell Publishing, 2003.

[Chang2003] Chang, Chih-Chung and Lin, Chih-Jen. "*LIBSVM: a library for Support Vector Machines",* http://www.csie.ntu.edu.tw/~cjlin/libsvm, 2003.

[Cortes1993] Cortes, C., Lawrence D. Jackel, Sara A. Solla, Vladimir Vapnik, John S. Denker: "Learning Curves: Asymptotic Values and Rate of Convergence", Advances in Neural Information Processing Systems (NIPS) 1993: 327-334.

[Crammer2000] Crammer, K. and Y. Singer. "On the Learnability and Design of Output Codes for Multiclass Problems", Proceedings of the Thirteen Annual Conference on Computational Learning Theory (COLT), 2000.

[Demuth2001] Demuth, H. and M. Beale. "Neural network toolbox user's guide. Matlab user's guide", 2001: The MathWorks Inc.

[Dietterich1998] Dietterich, T. G. "Approximate statistical tests for comparing supervised classification learning algorithms", Neural Computation, 10 (7), 1998.

[Duda2001] Duda, R.O, Hart, P.E, and Stork, D.G. "Pattern Classification", second edition, John Wiley, New York, 2001.

[Dudoit2002] Dudoit S., J. Fridlyand, and T. P. Speed. "Comparison of discrimination methods for the classification of tumors using gene expression data", Journal of the American Statistical Association, Vol. 97, No. 457, p. 77-87, 2002.

[Dudoit2003] Dudoit S and van der Laan MJ. Asymptotics of Cross-Validated Risk Estimation in Model Selection and Performance Assessment. U.C. Berkeley Division of Biostatistics Working Paper Series. Working Paper 126, February 5, 2003.

[Ferri2003] Ferri C., J. Hernández-Orallo, and M.A. Salido. "Volume under the ROC Surface for Multi-Class Problems", In Proc. of 14th European Conference on Machine Learning, ECML'03, LNAI Springer Verlag, Vol 2837, pages 108-120, 2003.

[Freund1995] Freund, Y. "Boosting a weak learning algorithm by majority", Information and Computation, 121(2):256-285, 1995.

[Friedman1996] Friedman, J. "Another approach to polychotomous classification", Technical report, Stanford Univeristy, 1996.

[Furey2000] Furey, T.S., et al.. "Support vector machine classification and validation of cancer tissue samples using microarray expression data", Bioinformatics 2000 16: 906-914.

[Gollub2003] Gollub J, Ball CA, Binkley G, Demeter J, Finkelstein DB, Hebert JM, Hernandez-Boussard T, Jin H, Kaloper M, Matese JC, Schroeder M, Brown PO, Botstein D, Sherlock G. The Stanford Microarray Database: data access and quality assessment tools. Nucleic Acids Res 2003 Jan 1;31(1):94-6.

[Golub1999] Golub, T., et al. "Molecular Classification of Cancer: Class Discovery and Class Prediction by Gene Expression Monitoring", Science, Vol 286, 15 October 1999.

[Gordon2003] Gordon GJ, Jensen RV, Hsiao LL, Gullans SR, Blumenstock JE, Richards WG, Jaklitsch MT, Sugarbaker DJ, Bueno R. Using gene expression ratios to predict outcome among patients with mesothelioma. J Natl Cancer Inst. 2003 Apr 16;95(8):598-605.

[Good2000] Good, P. I. "Permutation Tests: A Practical Guide to Resampling Methods for Testing Hypotheses", 2nd ed. New York: Springer-Verlag, 2000.

[Goodman1996] Goodman P. and Frank Harrell, "NevProp Manual with Introduction to Artificial Neural Networks Theory" http://brain.cs.unr.edu/publications/NevPropManual.pdf, 2004.

[Guyon2002] Guyon, I., et al. "Gene selection for cancer classification using support vector machines", Machine Learning, 2002, 46: 389-422.

[Hardin2004] Hardin D., I. Tsamardinos, C.F. Aliferis, "A Theoretical Characterization of Linear SVM-Based Feature Selection", In the Twenty-First International Conference on Machine Learning (ICML), 2004.

[Hastie2001] Hastie T., Robert Tibshirani and Jerome Friedman, "Elements of Statistical Learning: Data Mining, Inference and Prediction" Springer-Verlag, New York, 2001.

[Herbrich2002] Herbrich, R. "Learning Kernel Classifiers: Theory and Algorithms". MIT Press. 2002.

[Ho1994] Ho TK, Hull JJ, Srihari SN. "Decision combination in multiple classifier systems", IEEE Trans Pattern Analysis and Machine Intelligence 1994; 16(1):66-76.

[Hsu2002] Hsu, Chih-Wei and Chih-Jen Lin. "A Comparison of Methods for Multi-class Support Vector Machines", IEEE Transactions in Neural Networks 13(2) 415-425, 2002.

[Joachims1999] Joachims, T. "Making Large-Scale SVM Learning Practical", Advances in Kernel Methods - Support Vector Learning, B. Schölkopf, C. Burges, and A. Smola (ed.), MIT Press, 1999.

[Jiang2004] Jiang H, Deng Y, Chen HS, Tao L, Sha Q, Chen J, Tsai CJ, Zhang S. Joint analysis of two microarray gene-expression data sets to select lung adenocarcinoma marker genes. BMC Bioinformatics. 2004 Jun 24;5(1):81.

[Jones1997] Jones, B. "Matlab Statistics Toolbox", The MathWorks, Inc. Natick, MA, USA, 1997.

[Khan2001] Khan, J., et al. "Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks", Nature Medicine, volume 7, Number 6, June 2001.

[Kohavi1995] Kohavi, R. "A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection", In Proceedings of the International Joint Conferences on Artificial Intelligence (IJCAI), 1995.

[Kressel1999] Kressel, U. "Pairwise classification and support vector machines", In Advances in Kernel Methods: Support Vector Learning (Chapter 15), MIT Press, 1999.

[Kutin2002], Samuel Kutin, Partha Niyogi: Almost-everywhere Algorithmic Stability and Generalization Error. UAI 2002: 275-282.

[Lee2003] Lee, Yoonkyung and Cheol-Koo Lee. "Classification of multiple cancer types by multicategory support vector machines using gene expression data", Bioinformatics 2003 19: 1132-1139.

[Lin1999] Lin, Chih-Jen and Jorge J. Moré. "Newton's Method for Large Bound-Constrained Optimization Problems", SIAM Journal on Optimization, Volume 9, Number 4, pp. 1100-1127, 1999.

[Lu2002] Lu, J., et al. "Classical statistical approaches to molecular classification of cancer from gene expression profiling", in Methods of Microarray Data Analysis: Papers from CAMDA'00, eds. S.M. Lin, K.F. Johnson, pp. 97-107, Kluwer Academic Publishers, Boston/Dordrecht/London, 2002.

[Mitchell1997] Mitchell, T.M., "Machine Learning", McGraw-Hill, 1997.

[Mossman1999] Mossman, D. "Three-way ROCs", Medical Decision Making, 1999, 19: 78-89.

[Mukherjee2003] Mukherjee S. "Classifying Microarray Data Using Support Vector Machines", Understanding And Using Microarray Analysis Techniques: A Practical Guide. Boston: Kluwer Academic Publishers; 2003.

[Murthy1997] Murthy, S., "Automatic construction of decision trees from data: A multi-disciplinary survey", Data Mining and Knowledge Discovery, 1997.

[Ntzani2003] Ntzani EE and Ioannidis JP. "Predictive ability of DNA microarrays for cancer outcomes and correlates: and empirical assessment", Lancet. 2003 Nov 1;362(9394):1439-44.

[Nutt2003] Nutt, C., et al. "Gene expression-based classification of malignant gliomas correlates better with survival than histological classification", Cancer Res. 2003 Apr 1;63(7):1602-7.

[Parmigiani2003] Parmigiani G, Garrett ES, Irizarry R, Zeger SL. (eds) The analysis of gene expression data: methods and software, New York: Springer, 2003.

[Platt1999] Platt, J. "Fast Training of Support Vector Machines using Sequential Minimal Optimization", Advances in Kernel Methods - Support Vector Learning, B. Schölkopf, C. Burges, and A. Smola, (ed.), MIT Press,1999.

[Platt2000] Platt, J., N. Cristianini, and J. Shawe-Taylor. "Large margin dags for multiclass classification", Advances in Neural Information Processing Systems 12, pages 547-553. MIT Press, 2000.

[Pomeroy2002] Pomeroy, L., et al. "Prediction of central nervous system embryonal tumour outcome based on gene expression", Nature, vol 415, 24 January 2002.

[Ramaswamy2001] Ramaswamy, S., et al. "Multiclass cancer diagnosis using tumor gene expression signatures", Proc Natl Acad Sci U S A Dec 11, 2001.

[Rhodes2004a] Rhodes DR, Yu J, Shanker K, Deshpande N, Varambally R, Ghosh D, Barrette T, Pandey A, Chinnaiyan AM. ONCOMINE: A Cancer Microarray Database and Data-Mining Platform. Neoplasia. 2004 Jan-Feb;6(1):1-6.

[Rhodes2004b] Rhodes DR, Yu J, Shanker K, Deshpande N, Varambally R, Ghosh D, Barrette T, Pandey A, Chinnaiyan AM. Large-Scale Meta-Analysis of Cancer Microarray Data Identifies Common Transcriptional Profiles of Neoplastic Transformation and Progression. Proc Natl Acad Sci U S A. 2004 Jun 22;101(25):9309-14.

[Romualdi2003] Romualdi C., Campanaro S., Campagna D., Celegato B., Cannata N., Toppo S., Valle G. and Lanfranchi G. (2003) Pattern recognition in gene expression profiling using DNA array: a comparative study of different statistical methods applied to cancer classification. Hum. Mol. Gen. 12:823-36.

[Savage2003] Savage KJ, Monti S, Kutok JL, Cattoretti G, Neuberg D, De Leval L, Kurtin P, Dal Cin P, Ladd C, Feuerhake F, Aguiar RC, Li S, Salles G, Berger F, Jing W, Pinkus GS, Habermann T, Dalla-Favera R, Harris NL, Aster JC, Golub TR, Shipp MA. The molecular signature of mediastinal large B-cell lymphoma differs from that of other diffuse large B-cell lymphomas and shares features with classical Hodgkin lymphoma. Blood. 2003 Dec 1;102(12):3871-9.

[Scheffer1999] Scheffer T. Error Estimation and Model Selection. Ph.D. thesis, Technischen Universität Berlin, School of Computer Science, 1999.

[Simon2003] Simon R, Radmacher MD, Dobbin K, McShane LM. Pitfalls in the use of DNA microarray data for diagnostic and prognostic classification. J Natl Cancer Inst. 2003 Jan 1;95(1):14-8.

[Sindwani2001] Sindwani V., et al. "Information Theoretic Feature Crediting in Multiclass Support Vector Machines", First SIAM International Conference on Data Mining, 2001.

[Singh2002] Singh, D., et al. "Gene expression correlates of clinical prostate cancer behavior", Cancer Cell: March 2002, Vol. 1.

[Sharkey1996] Sharkey AJC (ed). "Special Issue: Combining Artificial Neural Networks: Ensemble Approaches", Connection Science 1996; 8(3 & 4).

[Shedden2003] Shedden KA, Taylor JM, Giordano TJ, Kuick R, Misek DE, Rennert G, Schwartz DR, Gruber SB, Logsdon C, Simeone D, Kardia SL, Greenson JK, Cho KR, Beer DG, Fearon ER, Hanash S. Accurate molecular classification of human cancers based on gene expression using a simple classifier with a pathological tree-based framework. Am J Pathol. 2003 Nov;163(5):1985-95.

[Shipp2002] Shipp, M., et al. “Diffuse large B-cell lymphoma outcome prediction by gene expression profiling and supervised machine learning”, Nature Medicine, Volume 8, Number 1, January 2002.

[Specht1990] Specht , DF., "Probabilistic neural network," Neural Networks, vol. 3, pp. 109-118, 1990.

[Statnikov2005] Statnikov A, Aliferis C, Tsamardinos I, Hardin D, and Levy S. Online supplement. http://www.gems-system.org, 2004.

[Staunton2001] Staunton, J., et al. “Chemosensitivity prediction by transcriptional profiling”, Proc Natl Acad Sci U S A, September 11, 2001, vol. 98, no. 19, 10787-10792.

[Su2001] Su, A.I., et al. “Molecular classification of human carcinomas by use of gene expression signatures”, Cancer Res. 2001 Oct 15;61(20):7388-93.

[Tsamardinos2003] Tsamardinos, I., C.F. Aliferis, A. Statnikov. “Time and Sample Efficient Discovery of Markov Blankets and Direct Causal Relations”, 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2003.

[UPMC2004] University of Pittsburgh, Benedum Oncology Informatics Center. UPITT Cancer Gene Expression Data Set Link Database. http://bioinformatics.upmc.edu/Help/UPITTGED.html (accessed 10/2004)

[Valentini2003] Valentini G., M. Muselli, and F. Ruffino. “Bagged Ensembles of SVMs for Gene Expression Data Analysis”, IJCNN2003, The IEEE-INNS-ENNS International Joint Conference on Neural Networks, Portland, USA, 2003.

[Vapnik1998] Vapnik, V. “Statistical Learning Theory”, Wiley-Interscience, 1998.

[Weiss1991] Weiss S.M. and C.A. Kulikowski. “Computer systems that learn”, Morgan Kaufmann, 1991.

[Weston1999] Weston, J. and C. Watkins. “Support Vector Machines for Multi-Class Pattern Recognition”, Proceedings of the Seventh European Symposium On Artificial Neural Networks, 1999.

[Wouters2003] Wouters L, et al. "Graphical exploration of gene expression data: a comparative study of three multivariate methods", Biometrics. 2003 Dec;59(4):1131-9.

[Yeang2001] Yeang, C., et al. “Molecular Classification of Multiple Tumor Types”, In: Proceedings of Ninth International Conference on Intelligent Systems in Molecular Biology, Copenhagen, Denmark (July 21-25, 2001), S316-S322, 2001.

[Yeo2001] Yeo G., Poggio, T. “Mutliclass Classification of SRBCT Tumors”, Technical Report AI Memo 2001-018 CBCL Memo 206. MIT.

[Yeoh2002] Yeoh EJ, Ross ME, Shurtleff SA, Williams WK, Patel D, Mahfouz R, Behm FG, Raimondi SC, Relling MV, Patel A, Cheng C, Campana D, Wilkins D, Zhou X, Li J, Liu H, Pui CH, Evans WE, Naeve C, Wong L, Downing JR. Classification, subtype discovery, and prediction of outcome in pediatric acute lymphoblastic leukemia by gene expression profiling. Cancer Cell. 2002 Mar;1(2):133-43.