SUPPLEMENTARY INFORMATION

Multi-class Cancer Diagnosis Using Tumor Gene Expression Signatures

http://www-genome.wi.mit.edu/MPR/GCM.html

Sridhar Ramaswamy [*†], Pablo Tamayo [*], Ryan Rifkin [***], Sayan Mukherjee [***], Chen-Hsiang Yeang [*††], Michael Angelo [*], Christine Ladd [*], Michael Reich [*], Eva Latulippe [¶], Jill P. Mesirov [*], Tomaso Poggio [**], William Gerald [¶], Massimo Loda [†§], Eric S. Lander [*‖], Todd R. Golub [*‡‡‡]

[*] Whitehead Institute / Massachusetts Institute of Technology Center for Genome Research, Cambridge, MA 02138; [†] Departments of Adult and [‡] Pediatric Oncology, Dana-Farber Cancer Institute, Boston, MA 02115; [§] Department of Pathology, Brigham & Women's Hospital, Boston, MA 02115; [¶] Department of Pathology, Memorial Sloan-Kettering Cancer Center, New York, NY 10021; Departments of [‖] Biology, [**] McGovern Institute, CBCL, and Artificial Intelligence Laboratory, and [††] Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA 02139

This document provides supplementary information and details not included in the paper "Multi-class Cancer Diagnosis Using Tumor Gene Expression Signatures." Other sources of information can be found at our web site www.genome.wi.mit.edu/MPR, and also in Yeang et al. (2001) and Rifkin et al. (2002) (to appear).

TABLE OF CONTENTS

INTRODUCTION 2

MATERIALS & METHODS 2

- **PATIENT DATA** 3
- **MICROARRAY HYBRIDIZATION** 3

CLUSTERING 4

CLASS-SPECIFIC MARKER SELECTION 5

- **PERMUTATION TEST AND NEIGHBORHOOD ANALYSIS FOR MARKER GENES** 6
- **ADDITIONAL NOTES** 8

MULTI-CLASS SUPERVISED LEARNING 12

- **K-NN AND WEIGHTED VOTING** 15

SUPPORT VECTOR MACHINES 16
RECURSIVE FEATURE ELIMINATION 17
PROPORTIONAL CHANCE CRITERION 17
MULTI-CLASS PREDICTION RESULTS 17
SVM/OVA MULTI-CLASS PREDICTION 18
APPENDIX: SUPPORT VECTOR MACHINES 32
REFERENCES 36

Introduction

The accurate classification of human cancer based on anatomic site of origin is an important component of modern cancer treatment. It is estimated that upwards of 40,000 cancer cases per year in the U.S. are difficult to classify using standard clinical and histopathologic approaches. Molecular approaches to cancer classification have the potential to effectively address these difficulties. However, decades of research in molecular oncology have yielded few useful tumor-specific molecular markers. An important goal in cancer research, therefore, continues to be the identification of tumor specific genetic markers and the use of these markers for molecular cancer classification.

Oligonucleotide microarray-based gene expression profiling allows investigators to study the simultaneous expression of thousands of genes in biological systems. In principle, tumor gene expression profiles can serve as molecular fingerprints that allow for the accurate and objective classification of tumors. Previously, our group developed computational approaches (unsupervised and supervised learning) using gene expression data to accurately distinguish between two common blood cancer classes: acute lymphocytic and acute myelogenous leukemia (Golub et al 1999, Slonim et al 2000). The classification of primary solid tumors, by contrast, is a harder problem due to limitations with sample availability, identification, acquisition, integrity, and preparation. Moreover, a solid tumor is a heterogeneous cellular mix and gene expression profiles might reflect contributions from non-malignant components further confounding classification. In addition, there are intrinsic computational complexities in making multi-class, as opposed to binary class, distinctions.

We have asked whether it is possible to achieve a general, multi-class molecular-based cancer classification solely using tumor gene expression profiles. This document describes the biologic, algorithmic, and computational details of this method and provides a first look at the technical difficulties and computational challenges associated with this approach.

Materials & Methods

The gene expression datasets were obtained following a standard experimental protocol described schematically in Figure 1.

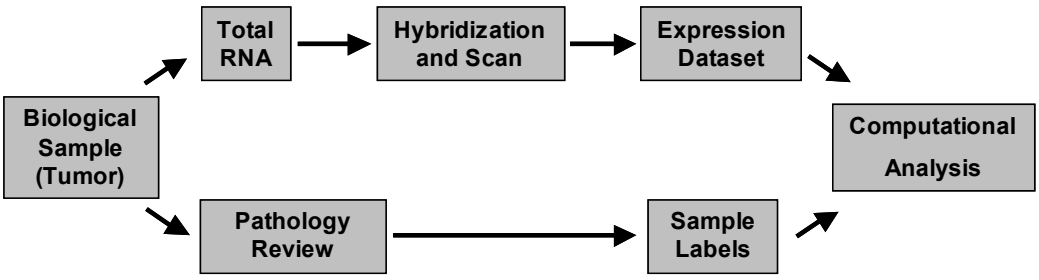


Figure 1: Experimental protocol and dataset creation.

Patient Data

Snap-frozen human tumor and normal tissue specimens, spanning 14 different tumor classes, were obtained from NCI / Cooperative Human Tissue Network, Massachusetts General Hospital Tumor Bank, Dana-Farber Cancer Institute, Brigham and Women's Hospital, and Children's Hospital (all in Boston, MA) and Memorial Sloan-Kettering Cancer Center (New York, NY). Tissue was collected and studied in an anonymous fashion under a discarded tissue protocol approved by the Dana-Farber Cancer Institute Institutional Review Board.

Initial diagnoses were made at university hospital referral centers using all available clinical and histopathologic information. Tissues underwent centralized clinical and pathology review at the Dana-Farber Cancer Institute and Brigham & Women's Hospital or Memorial Sloan-Kettering Cancer Center to confirm initial diagnosis of site of origin. All tumors were:

1. biopsy specimens from primary sites (except where noted)
2. obtained prior to any treatment
3. enriched in malignant cells (>50%) but otherwise unselected.

Normal tissue RNA (Biochain, Inc. (Hayward, CA)) was from snap-frozen autopsy specimens collected through the International Tissue Collection Network.

The file **SAMPLES.xls** includes relevant clinical data for these specimens.

Microarray hybridization

RNA from whole tumors was used to prepare "hybridization targets" with previously published methods (4). For a detailed protocol, see http://www.genome.wi.mit.edu/MPR. Briefly, snap frozen tumor specimens were homogenized (Polytron, Kinematica, Lucerne) directly in Trizol (Life Technologies, Gaithersberg, MD), followed by a standard RNA isolation according to the manufacturer's instructions. RNA integrity was assessed by non denaturing gel electrophoresis (1% agarose) and spectrophotometry. The amount of starting total RNA for each reaction was 10 ug. First strand cDNA synthesis was performed using a T7-linked oligo-dT primer, followed by second strand synthesis. An *in vitro* transcription reaction was done to generate cRNA containing biotinylated UTP and CTP, which was subsequently chemically fragmented at 95 C for 35 minutes. Fifteen micrograms of the fragmented, biotinylated cRNA was sequentially hybridized in MES buffer (2-[N-Morpholino]ethansulfonic acid) containing 0.5 mg/ml acetylated bovine serum albumin (Sigma, St. Louis, MO) to Affymetrix (Santa Clara, CA) Hu6800 and Hu35KsubA oligonucleotide microarrays at 45 C for 16 hours. Arrays were washed and stained with streptavidin-phycoerythrin (SAPE, Molecular Probes). Signal amplification was performed using a biotinylated anti-streptavidin antibody (Vector Laboratories, Burlingame, CA) at 3 μg/ml followed by a second staining with SAPE. Normal goat IgG (2 mg/ml) was used as a blocking agent. Scans were performed on Affymetrix scanners and expression values for each gene was calculated using Affymetrix GENECHIP software. Hu6800 and Hu35KsubA arrays contain a total of 16,063 probe sets representing 14,030 Genbank and 475 TIGR accession numbers. For

subsequent analysis, the output of each probe set (i.e. the “average difference” value calculated from matched and mismatched probe hybridization) was considered as a separate gene.

Of 314 tumor samples and 98 normal tissue samples processed, 218 tumors and 90 normal tissue samples passed quality control criteria and were used for subsequent data analysis. The remaining 104 samples either failed quality control measures of the amount and quality of RNA, as assessed by spectrophotometric measurement of optical density (O.D.) and agarose gel electrophoresis, or yielded poor quality scans. Scans were rejected if mean chip intensity exceeded 2 standard deviations from the average mean intensity for the entire scan set, if the proportion of “Present” calls was less than 10%, or if microarray artifacts were visible. This resulting dataset has approximately 5 million gene expression values.

Data (308 samples) was organized into four sets:

1. **GCM_Training.res** (Training Set; 144 primary tumor samples)
2. **GCM_Test.res** (Independent Test Set; 54 samples; 46 primary and 8 metastatic)
3. **GCM_PD.res** (Poorly differentiated adenocarcinomas; 20 samples)
4. **GCM_Total.res** (Training set + Test set + normals (90); 280 samples).

Associated **.cls** files are also provided to allow for supervised learning analysis of these datasets using our GeneCluster software package (available at http://www-genome.wi.mit.edu/MPR/software.html).

In each dataset, columns represent each gene profiled, rows represent samples, and the values are raw average difference value output from the Affymetrix software package.

Clustering

A threshold of 20 units was imposed before analysis of the dataset because at very low values the data is noisy and not reproducible. A ceiling of 16,000 units was also imposed due to saturation effects at very high measurement values. Gene expression values were subjected to a variation filter that excluded genes showing less than a 5-fold variation and an absolute variation of less than 500 across samples (comparing max/min and max-min with predefined values and excluding genes not obeying both conditions). The data were also normalized by standardizing each row (gene) to mean 0 and variance 1.

Self Organizing Maps analysis was performed using our GeneCluster clustering package available at http://www-genome.wi.mit.edu/MPR/software.html. The Self Organizing Map is a method for performing unsupervised learning (i.e. classifying data where the true class for the data samples is assumed to be unknown prior to model training). In general, unsupervised learning presents a more difficult problem than supervised learning methods but can be useful for discovering classes during exploratory data analysis. With the SOM, one randomly chooses the geometry of the grid (e.g., a 3 x 2 grid) and maps it into the k-dimensional feature space. Initially the features are randomly mapped to the grid but during training the mapping is iteratively adjusted to reflect the data structure. Multiple clustering runs using different SOM architectures with Dataset A (Training set; 144 samples, 14 tumor classes) failed to separate the samples along tissue of origin lines. A representative 5 x 5 SOM is shown below.

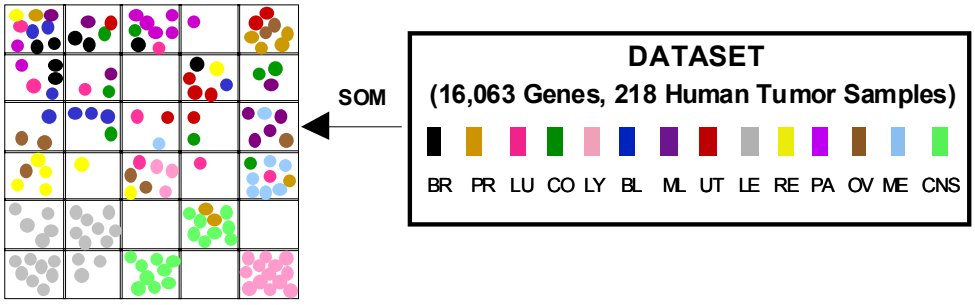


Figure 1: A 5 x 5 Self-Organizing Map of Dataset A

Hierarchical Clustering is another unsupervised learning method useful for dividing data into natural groups. Samples are clustered hierarchically by organizing the data into a tree structure based upon the degree of similarity between features (genes). We used the Cluster and TreeView software (available at http://rana.lbl.gov/EisenSoftware.htm) to perform average linkage clustering, which organizes all of the data elements into a single tree with the highest levels of the tree representing the discovered classes. As seen with Self-Organizing Maps, hierarchical clustering of Dataset A (Training set; 144 samples, 14 tumor classes) also failed to separate the samples along tissue of origin lines.

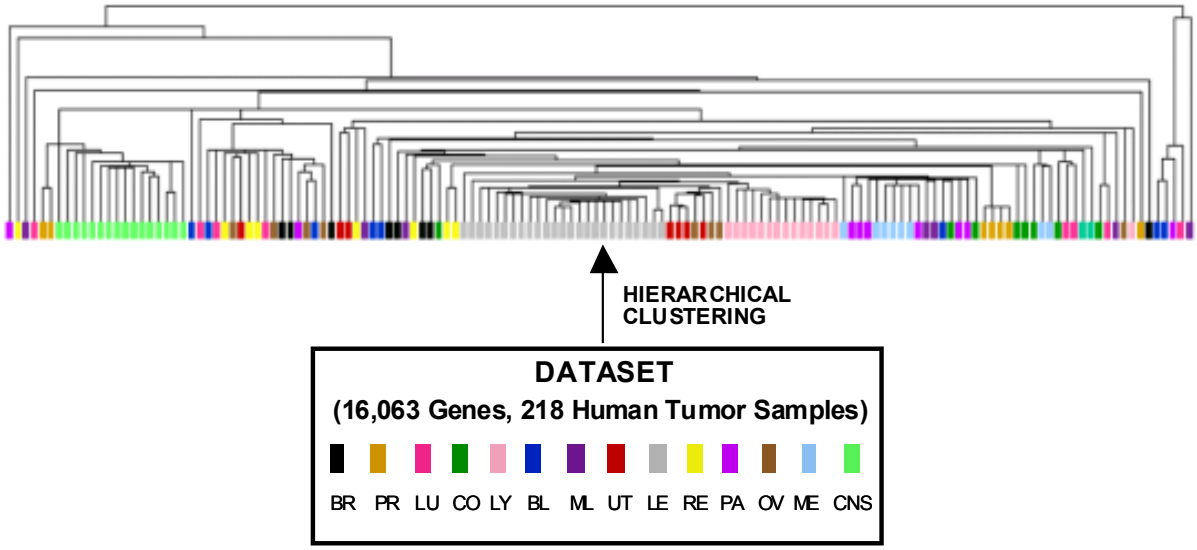


Figure 2: Hierarchical clustering of Dataset A

Class-specific Marker Selection

Genes correlated with one particular class versus all other classes (one-versus-all (OVA) markers) were identified by sorting all of the genes on the array according the signal-to-noise statistic (S2N) $(\mu_{\text{one tumor class}} - \mu_{\text{all other tumor classes}})/(\sigma_{\text{one tumor class}} + \sigma_{\text{all other tumor classes}})$ where μ and σ represent the mean and standard deviation of expression of each gene, respectively, for each “class.” Thus, markers represent genes that are differentially expressed by a single class that might be useful, individually or as groups, as molecular markers

for the differential diagnosis of cancer. These marker genes can also be used, in combination, to build the k-nearest neighbor and weighted voting classifiers (see below).

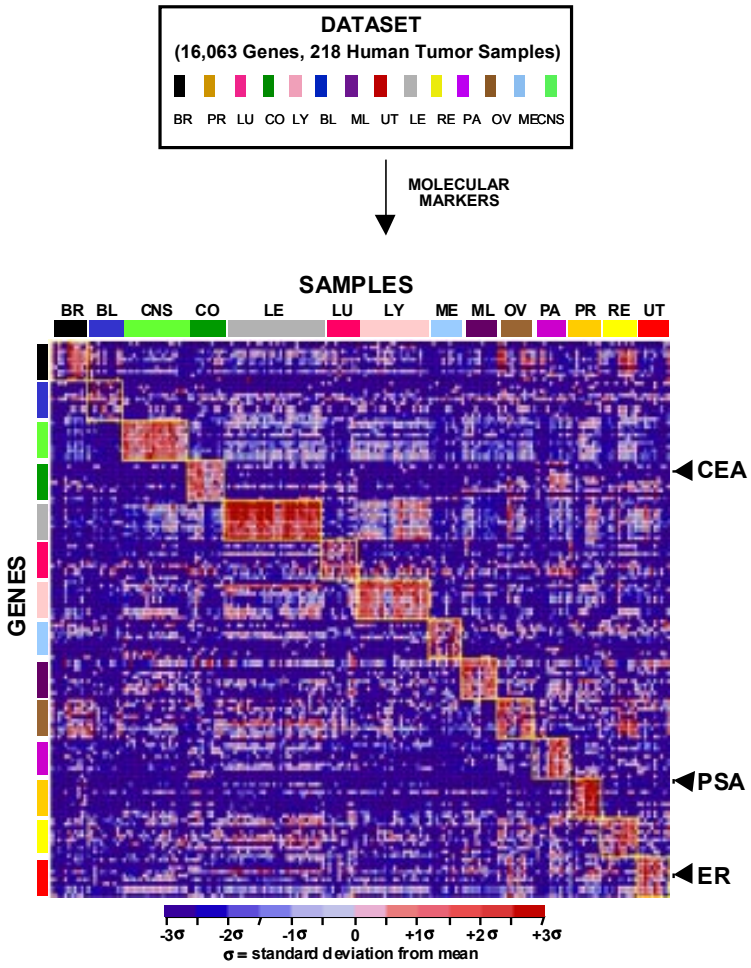


Figure 3: Colorgram of tumor class-specific markers.

Permutation test and neighborhood analysis for marker genes

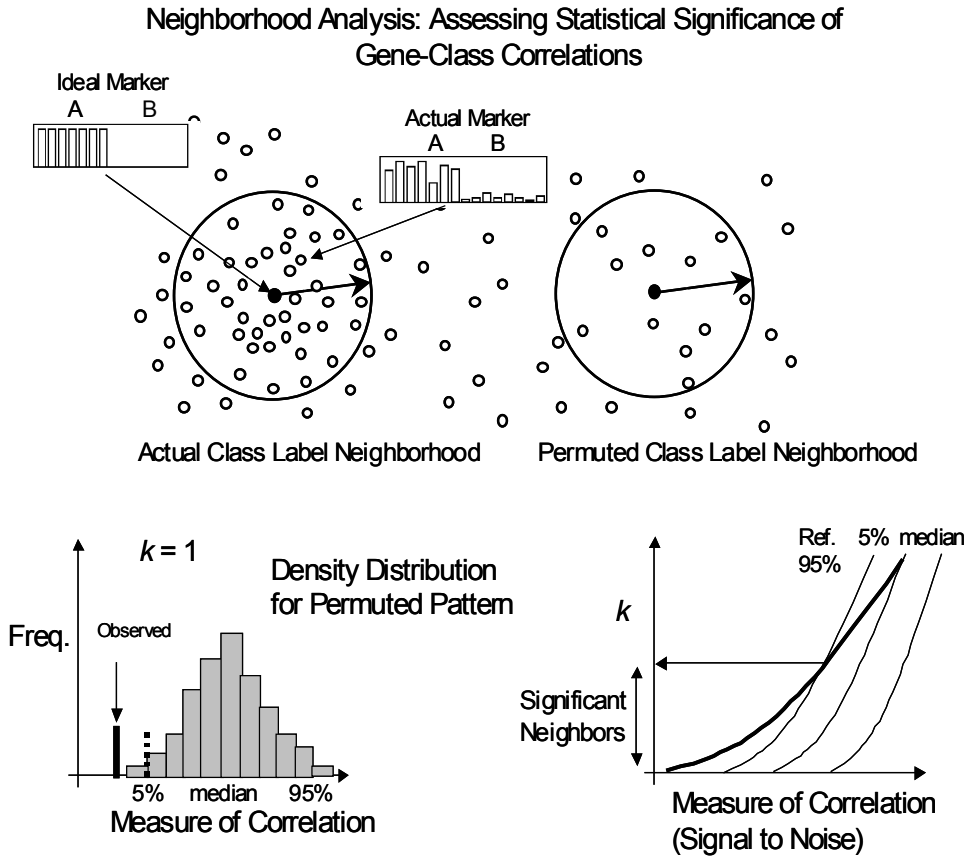
A permutation test was used to calculate which OVA marker genes were statistically significant. This procedure addresses the following question: what is the likelihood that individual marker genes, selected by signal-to-noise, represents a chance correlation? In permutation testing, the signal-to-noise scores for each marker gene is calculated and compared with the corresponding signal-to-noise scores after random permutation of the class labels. One thousand random permutations are used to build histograms for the top marker, the second best etc. Based on this histogram the 10%, 5% and 1% significance levels are determined. In detail the permutation test procedure is as follows:

1. Generate signal-to-noise scores (μ one tumor class - μ all other tumor classes)/(σ one tumor class + σ all other tumor classes) for all genes that pass a variation filter using the actual class labels and sort them from best to worst. The best match (k=1) is the gene “closer” or more correlated to the phenotype using the signal to noise as a distance function. In fact one can imagine the reciprocal of the signal to noise as a “distance” between the phenotype and each gene as shown in the figure (see next page).

2. Generate 1000 random permutations of the class labels. For each case of randomized class labels generate signal-to-noise scores and sort genes accordingly.

3. Build a histogram of signal to noise scores for each value of k. For example one for all the 500 top markers (k=1), another one for the 500 second best (k=2) etc. These histograms represent a reference statistic for the best match, second etc. and for a given value of k different genes contribute to it. Notice that the correlation structure of the data is preserved by this procedure. Then for each value of k one determines the 10%, 5% and 1% significance levels. See the bottom diagrams in the figure.

4. Compare the actual signal to noise scores with the different significance levels obtained for the histograms of permuted class labels for each value of k. This test helps to assess the statistical significance of gene markers in terms of target class-correlations.



In **OVA MARKERS.xls** and **TUMOR NORMAL MARKERS.xls**, the values for permutation tests of the top 1000 marker genes for each given class are reported in tables with this format:

Distinction	Distance	Perm 1%	Perm 5%	Perm 10%	Feature	Description
class 0	0.96694607	1.0144908	0.8333578	0.6280173	M93119_at	INSM1 Insulinoma-associated 1
class 0	0.9096911	0.8600172	0.7669801	0.5740431	M30448_s_at	Casein kinase II beta subunit
class 0	0.90010124	0.85051423	0.7251496	0.5494933	S82240_at	RhoE
class 1	0.832689	0.84354156	0.7071885	0.5292253	U44060_at	Homeodomain protein (Prox 1)
class 1	0.83225346	0.8009565	0.68034023	0.5169537	D80004_at	KIAA0182 gene

class 1	0.6520017	0.9831643	0.84544426	0.6230137	X86693_at	High endothelial venule
class 2	1.2436218	0.88150144	0.7559189	0.5795857	M93426_at	PTPRZ Protein tyrosine phosphatase, receptor-type, zeta polypeptide
class 2	1.2317128	0.86047184	0.70928395	0.5539352	U48705_rna1_s_at	Receptor tyrosine kinase DDR gene
class 2	1.2259983	0.8433512	0.68909335	0.5358038	X86809_at	Major astrocytic phosphoprotein PEA-15
class 3	1.214929	0.8281318	0.6849929	0.5217813	U45955_at	Neuronal membrane glycoprotein M6b mRNA, partial cds
class 3	1.2095517	0.79365546	0.6711517	0.510208	U53204_at	Plectin (PLEC1) mRNA
…….						

The distinction represents the class for which the markers are high (and low in the other classes). Distance is the signal to noise to the actual phenotype. Perm 1%, 5% and 10% refer to the significance cut-off values derived from histograms of random permutation signal to noise scores for each given gene. Feature is the gene accession number and Description the gene name and annotation.

Additional Notes

- This test helps to assess the statistical significance of gene markers in terms of class-gene correlations but if a group of genes fails to pass the test that by itself does not necessarily imply that they cannot be used to build an effective classifier.

- The choice of the signal to noise is somewhat ad hoc but not unreasonable as a choice of class distance. The reason the signal to noise ratio was chosen instead of a t-statistic or other class distance measures was mainly historical and empirical: it performed slightly better in previous studies of gene expression feature selection combined with a weighted voting classifier.

- We deal with the problem of multiple hypotheses by performing a permutation test and use quantiles of the empirical distributions of rank signal-to-noise values to assess significance. This is a distribution-free approach that preserves the correlation structure of genes.

- The advantages of performing a permutation test are multiple:

    1) It is a direct empirical to test the significance of the matching of a given phenotype to a particular set of genes (dataset).
    2) It doesn't assume a particular functional form for the distribution or correlation structure of genes.
    3) As the permutation test is done on the entire distribution of genes (as scored by signal to noise from the phenotype) the gene-to-gene correlation structure is preserved and therefore one doesn't need to explicitly compensate for multiple hypothesis testing (for example by Bonferroni, Sidak's or some other procedure that makes strong assumptions about the distribution, correlations or independence of genes).

- Another more geometrical and sometimes more intuitive way to look at this procedure is to consider the figure above as a hypothetical projection of normalized gene expression space where each dimension represents an experiment and each data point a gene. The entire dataset of filtered genes will be represented by a collection of data points distributed in that space. Each gene is represented by a point and the closer two points are the more correlated they are (i.e. across the set of experiments being considered). Now imagine projecting a point that corresponds to an ideal marker gene that perfectly represents the phenotype of interest. This is for example a marker gene that is high and constant in one of the classes and low and constant in the other. This gene will be a perfect classifier to distinguish the two classes. We are interested in finding marker genes that are if not equal at least similar to this ideal marker. This can be accomplished by computing a distance or correlation measure between the class labels (phenotype) and the genes. In this sense we are looking at the "neighborhood" of a phenotype in gene expression space trying to find "close" neighbors. A permutation test in

this context is equivalent to moving the ideal gene point randomly (as the labels are permuted) and studying the distribution of neighbors each time it lands to a new reference point in expression space. By building a histogram of distance distributions to these random locations one can assess how “typical” is the actual neighborhood of the actual phenotype. For example if only once in a thousand random tries we found a set of top 10 markers as correlated as in the actual neighborhood then we will consider those markers to be significant.

Figure 4: Results of permutation testing for tumor versus normal gene markers. Blue = S2N values for genes and Red = mean S2N values for genes after 1000 random permutations. Markers for which the permuted value exceeds the actual value are considered statistically significant. (**TUMOR NORMAL MARKERS.xls)**

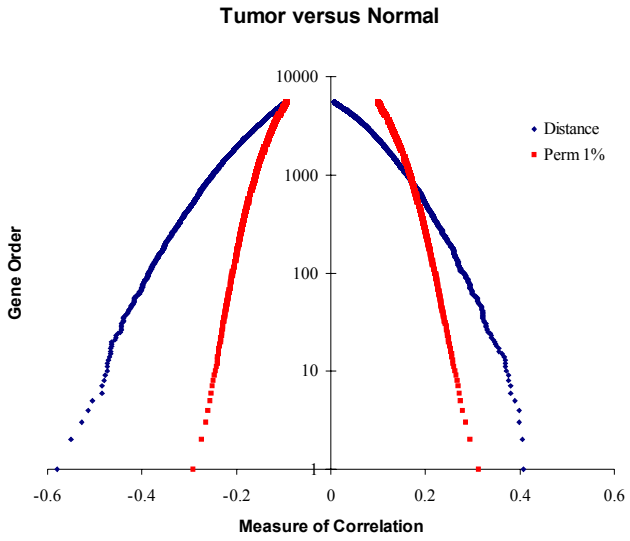
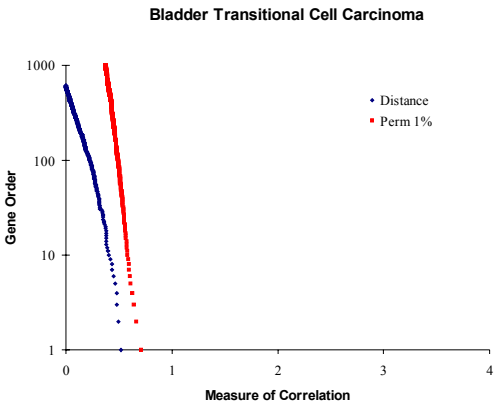
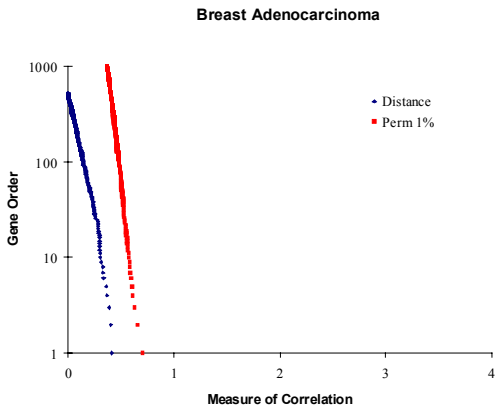
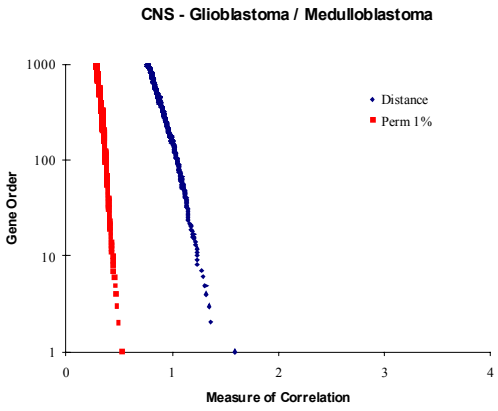
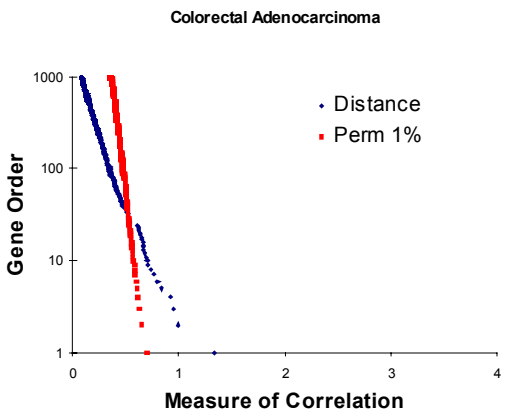


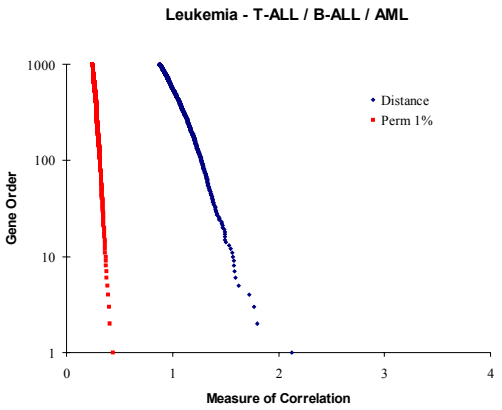
Figure 5: Results of permutation testing for each set of one versus all (OVA) tumor class-specific markers. Blue = S2N values for genes and Red = mean S2N values for genes after 1000 random permutations. Markers for which the permuted value exceeds the actual value are considered statistically significant. (**OVA MARKERS.xls**)

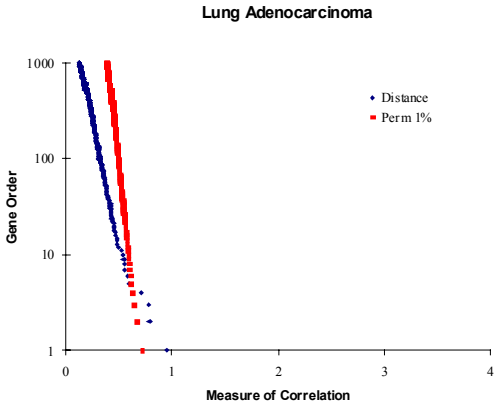

Bladder Transitional Cell Carcinoma
1000
100
10
1
Gene Order
0
1
2
3
4
Measure of Correlation
Distance
Perm 1%

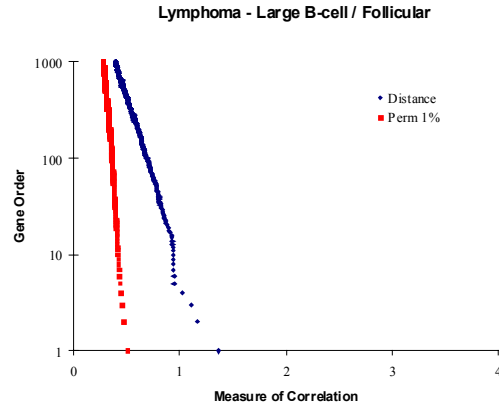

Breast Adenocarcinoma
1000
100
10
1
Gene Order
0
1
2
3
4
Measure of Correlation
Distance
Perm 1%

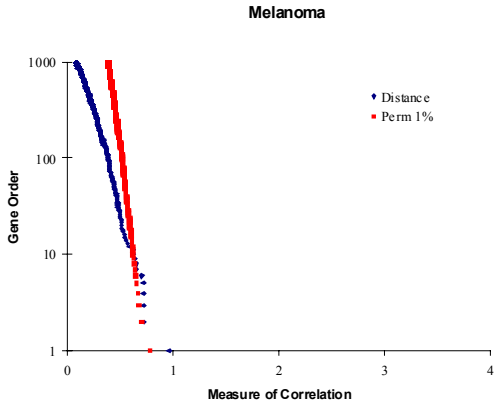

CNS - Glioblastoma / Medulloblastoma
1000
100
10
1
Gene Order
0
1
2
3
4
Measure of Correlation
Distance
Perm 1%

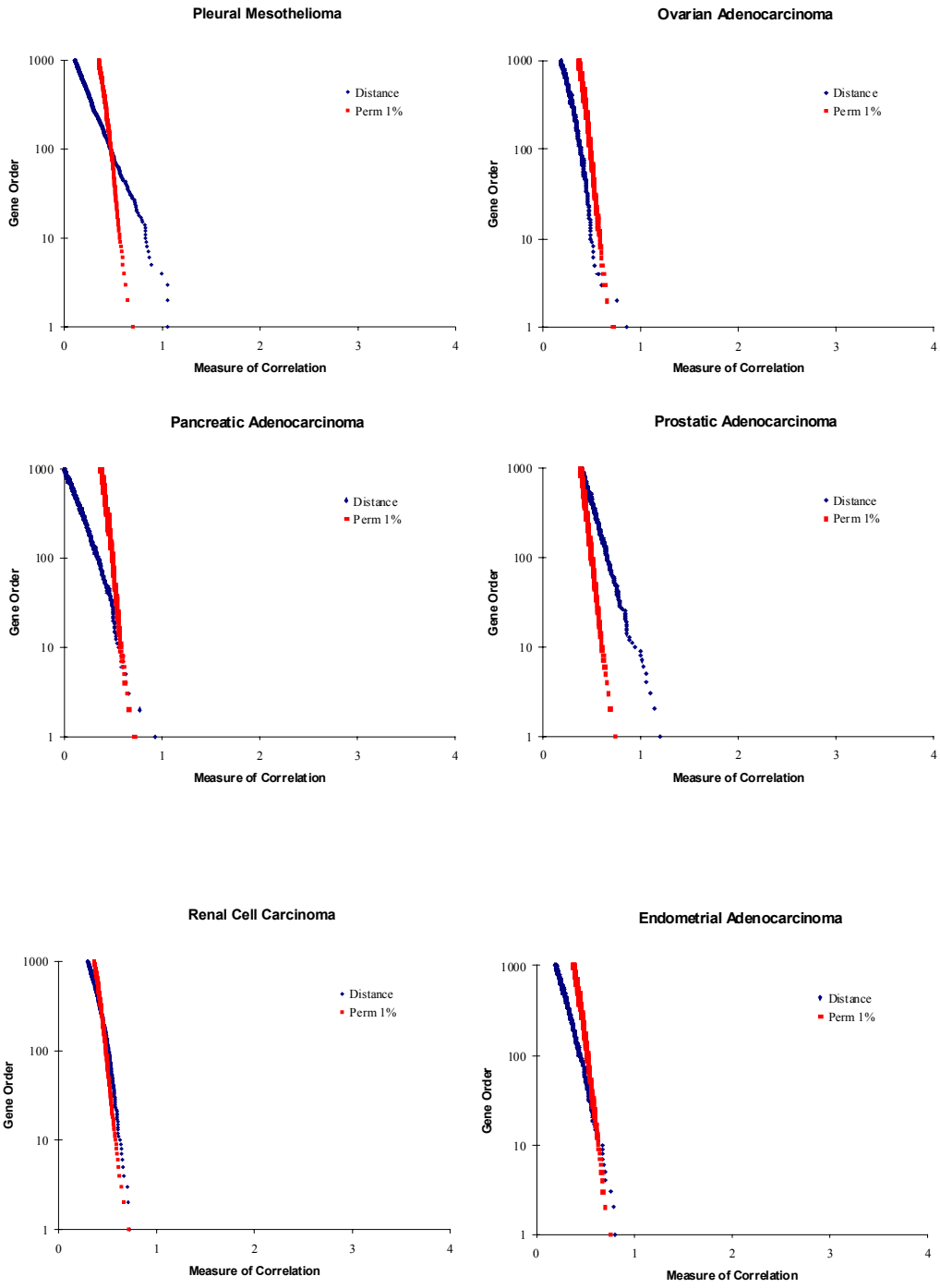

Colorectal Adenocarcinoma
1000
100
10
1
Gene Order
0
1
2
3
4
Measure of Correlation
Distance
Perm 1%


Leukemia - T-ALL / B-ALL / AML
Gene Order
1000
100
10
1
0
1
2
3
4
Measure of Correlation
Distance
Perm 1%


Lung Adenocarcinoma
Gene Order
1000
100
10
1
0
1
2
3
4
Measure of Correlation
Distance
Perm 1%


Lymphoma - Large B-cell / Follicular
Gene Order
1000
100
10
1
0
1
2
3
Measure of Correlation
Distance
Perm 1%


Melanoma
Gene Order
1000
100
10
1
0
1
2
3
4
Measure of Correlation
Distance
Perm 1%



Multi-Class Supervised Learning

Supervised learning in this case involves “training” a classifier to recognize distinctions among the 14 clinically-defined tumor classes in our dataset, based on gene expression patterns, and then

testing the accuracy of the classifier in a blinded fashion. The methodology for building a supervised classifier that we followed differed based on the algorithm used for prediction.

Multi-class classification in this context is especially challenging for several reasons including:

- the large dimensionality of our datasets
- the small but significant uncertainty in the original labelings
- the noise in the experimental and measurement processes
- the intrinsic biological variation from specimen to specimen
- the small number of examples

Multi-class prediction is also intrinsically harder than binary prediction because the classification algorithm has to learn to construct a greater number of separation boundaries or relations. In binary classification an algorithm can "carve out" the appropriate decision boundary for only one of the classes, the other class is simply the complement. In multi-class classification each class has to be explicitly defined. Errors can occur in the construction of any one of the many decision boundaries so the error rates on multi-class problems can be significantly greater than that of binary problems. For example, in contrast to a balanced binary problem where the accuracy of a random prediction is 50%, for K classes the accuracy of a random predictor is of the order of 1/K.

There are basically two types of multi-class classification algorithms. The first type deals directly with multiple values in the target field. For example Naïve Bayes, k-Nearest Neighbors, and classification trees are in this class. Intuitively, these methods can be interpreted as trying to construct a conditional density for each class, then classifying by selecting the class with maximum a posteriori probability. The second type decomposes the multi-class problem into a set of binary problems and then combines them to make a final multi-class prediction. This group contains support vector machines, boosting, and weighted voting algorithms, and, more generally, any binary classifier. In certain settings the later approach results in better performance than the multiple target approaches. Intuitively, with high dimensional input space and very few samples per class, it is very difficult to construct accurate densities, and our data has these characteristics.

The basic idea behind combining binary classifiers is to decompose the multi-class problem into a set of easier and more accessible binary problems. The main advantage in this divide-and-conquer strategy is that any binary classification algorithm can be used. Besides choosing a decomposition scheme and a base classifier, one also needs to devise a strategy for combining the binary classifiers and providing a final prediction. The problem of combining binary classifiers has been studied in the computer science literature (Hastie and Tibshirani 1998, Allwein et al 2000, Guruswami and Sahai 1999) from a theoretical and empirical perspective. However, the literature is inconclusive, and the best method for combining binary classifiers for any particular problem is open.

The decomposition problem in itself is quite old and can be considered as an example of the collective vote-ranking problem addressed by Condorcet and others at the time of the French Revolution. Condorcet was interested in solving the problem of how to deduce a collective ranking of candidates based on individual voters' preferences. He proposed a decomposition based on binary questions (Michaud 1987) and then introduced analytical rules to obtain a consistent collective ranking based on the individual's answers to these binary questions. It turns out that a consistent collective ranking is not guaranteed in all cases and this led to the situation known as Condorcet or Arrow's paradox (Arrow 1951).

Standard modern approaches to combining binary classifiers can be stated in terms of what is called "output coding" (Dietterich and Bakiri 1991). The basic idea behind output coding following: given *K* classifiers trained on various partitions of the classes a new example is mapped into an output vector. Each element in the output vector is the output from one of the *K* classifiers, and a "codebook" is then used to map from this vector to the class label (see Figure 3). For example,

given three classes the first classifier may be trained to partition classes one and two from three, the second classifier trained to partition classes two and three from one, and the third classifier trained to partition classes one and two from three.

Two common examples of output coding are the one-versus-all (OVA) and all-pairs (AP) approaches. In the OVA approach, given *K* classes, *K* independent classifiers are constructed where the *ith* classifier is trained to separate samples belonging to class *i* from all others. The codebook is a diagonal matrix and the final prediction is based on the classifier that produces the strongest confidence,

$$\boldsymbol{class} = \arg\max_{i=1..K} \boldsymbol{f}_i ,$$

where $\boldsymbol{f}_i$ is the signed confidence measure of the *i*th classifier. In the all-pairs approach *K(K-1)/2* classifiers are constructed with each classifier trained to discriminate between a class pair (*i* and *j*). This can be thought of as a *K* by *K* matrix, where the *i-j* th entry corresponds to a classifier that discriminates between classes *i* and *j*. The codebook in this case is used to simply sum the entries of each row and select the row for which this sum is maximum,

$$\boldsymbol{class} = \arg\max_{i=1..K} \left[\sum_{j=1}^{K} \boldsymbol{f}_{ij} \right],$$

where as before $\boldsymbol{f}_{ij}$ is the signed confidence measure for the *ij*th classifier.

An ideal code matrix should be able to correct the mistakes made by the component binary classifiers. Dietterich and Bakiri used error-correcting codes to build the output code matrix where the final prediction is made by assigning a sample to the codeword with the smallest Hamming distance with respect to the binary prediction result vector (Dietterich and Bakiri 1991). There are several other ways of constructing error-correcting codes including classifiers that learn arbitrary class splits and randomly generated matrices (Bose and Ray-Chaudhuri 1960, Allwein et al 2000, Guruswami and Sahai 1999).

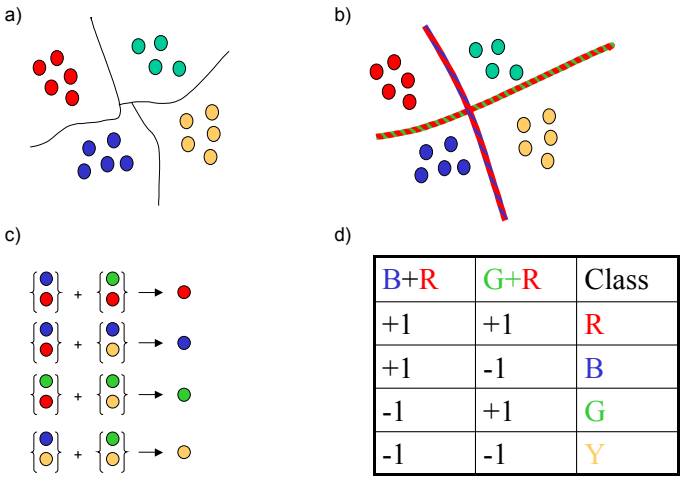


Figure 6. a) A four-class classification problem. b) Two binary classifiers are trained, the first discriminates the red and blue classes from the green and yellow and the second discriminates the red and green classes from the blue and yellow. c) The outputs of these two classifiers can

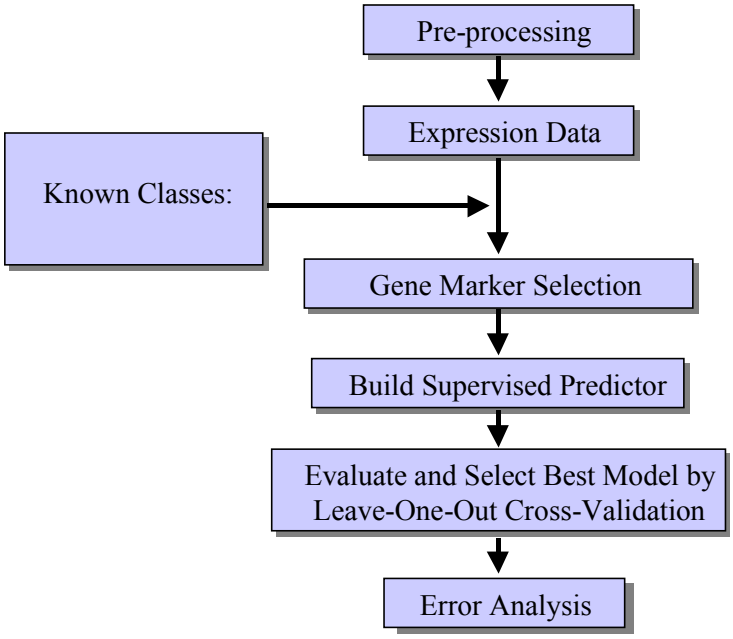
be combined to uniquely label a new example. d) The combination in (c) can be represented as a matrix.

Intuitively, there is a tradeoff between the OVA and AP approaches. The discrimination surfaces that need to be learned in the all-pairs approach are, in general, more natural and, theoretically, should be more accurate. However, with fewer training examples the empirical surface constructed may be less precise. The actual performance of each of these schemes, or others such as random codebooks, in combination with different classification algorithms is problem dependent.

k-NN and Weighted Voting

For Weighted Voting and k-NN algorithm based prediction, we:

- defined each target class based on histopathologic and clinical evaluation of tumor specimens;
- performed gene marker selection using signal-to-noise metric to identify class-specific marker sets useful for making each distinction;
- optimized a classifier on a Training Set using leave-one-out cross-validation (i.e. removing one sample, using the rest of the samples as a training set, predicting the class of the left out sample, and iteratively repeating this process for all the samples in the Training Set);
- evaluated the final prediction model on an Independent Test Set.



Algorithms

k-Nearest Neighbors (k-NN)

We developed a weighted implementation of the *k*-NN algorithm that predicts the class of a new sample by calculating the Euclidean distance (d) of this sample to the *k* "nearest neighbor" samples in "expression" space in the Training Set, and by selecting the predicted class to be that of the majority of the *k* samples. The method is defined in terms of Euclidean distances over standardized vectors so it is equivalent to using inner products: a . b / |a||b|. We first performed one class versus all other classes (OVA) marker gene selection and fed the *k*-NN algorithm with

class-specific OVA genes. Gene were selected by sorting each OVA marker according to the signal-to-noise metric (μ one tumor class - μ all other tumor classes)/(σ one tumor class + σ all other tumor classes). In our version of the algorithm the weight of each of the *k* neighbors was weighted according to 1/d. For our experiments, we set $k = 5$. The *k*-NN models were evaluated by 144-fold leave-one-out cross-validation whereby the training set of 143 samples was used to predict the class of a randomly withheld sample, and the cumulative error rate was recorded. Models with a variable total gene number (1-8400), selected according to their correlation with each tumor class distinction, were tested in this manner.

Weighted Voting

The weighted voting algorithm makes a weighted linear combination of relevant "marker" or "informative" genes obtained in the training set to provide a classification scheme for new samples. The selection of features (marker genes) is accomplished by computing the signal-to-noise statistic (S_x). The class predictor is uniquely defined by the initial set of samples and marker genes. In addition to computing S_x, the algorithm also finds the decision boundaries (half way) between the class means: b_x = (μ one tumor class - μ all other tumor classes)/(σ one tumor class + σ all other tumor classes)/2 for each gene. To predict the class of a test sample *y*, each gene *x* in the feature set casts a vote: $V_x = S_x (g_x^y - b_x)$ and the final vote for class a class is *sign* ($\mu_x V_x$). The strength or *confidence* in the prediction of the winning class *is* $(V_{win}-V_{lose})/(V_{win}+V_{lose})$ (i.e., the relative margin of victory for the vote).

Support Vector Machines

Support Vector Machines (SVMs) are powerful classification systems based on a variation of regularization techniques for regression (Vapnik 1998, Evgeniou et al 2000). SVMs provide state-of-the-art performance in many practical binary classification problems (Vapnik 1998, Evgeniou et al 2000). SVMs have also shown promise in a variety of biological classification tasks including some involving gene expression microarrays (Mukherjee et al 1999, Brown et al. 2000). For a detailed description of the algorithm see the appendix.

The algorithm is a particular instantiation of regularization for binary classification. Linear SVMs can be viewed as a regularized version of a much older machine-learning algorithm, the perceptron (Rosenblatt 1962, Minsky and Papert 1972). The goal of a perceptron is to find a *separating hyperplane* that separates positive from negative examples. In general, there may be many separating hyperplanes. In our problem, this separating hyperplane is the boundary that separates a given tumor class from the rest (OVA) or two different tumor classes (AP). The SVM chooses a separating hyperplane that has maximal *margin*, the distance from the hyperplane to the nearest point. Training an SVM requires solving a convex quadratic program with as many variables as training points.

The SVM experiments described in this paper were performed using a modified version of the SvmFu package (http://www.ai.mit.edu/projects/cbcl/). The advantages of SVM, when compared with other algorithms, are their sound theoretical foundations (Vapnik 1998, Evgeniou et al 2000), intrinsic control of machine capacity that combats over-fitting, capability to approximate complex classification functions, fast convergence, and good empirical performance in general. Standard SVMs assume the target values are binary and that the classification problem is intrinsically binary. We use the OVA methodology to combine binary SVM classifiers into a multiclass classifier. A separate SVM is trained for each class and the winning class is the one for with the largest margin, which can be thought of as a signed confidence measure.

In the experiments described in this paper there few data points in many dimensions. Therefore, we used a linear classifier in the SVM. Although we did allow the hyperplane to make misclassifications, in all cases involving the full 16,063 dimensions each OVA hyperplane fully separated the training data with no errors. In some of the experiments, involving explicit feature

selection with very few features, there were some training errors. Although this may indicate that we could select a very small number of features, and then use a kernel function to improve classification, preliminary experiments with this approach yielded no improvement over the linear case.

Recursive Feature Elimination

Many methods exist for performing feature selection. Similar results were observed with informal experiments using recursive feature elimination (RFE) (Guyon et al 2002), signal to noise ratio (Slonim et al 2000), and the radius-margin-ratio (Weston et al 2001). For the paper, we used RFE since it was the most straightforward to implement with the SVM. The method recursively removes features based upon the absolute magnitude of the hyperplane elements. Given microarray data with *n* genes per sample, the SVM outputs a hyperplane, w, which can be thought of as a vector with *n* components each corresponding to the expression of a particular gene. Assuming that the expression values of each gene have similar ranges, the absolute magnitude of each element in w determines its importance in classifying a sample, since,

$$f(x) = \sum_{i=1}^{n} w_i x_i + b ,$$

and the class label is $\mathrm{sign}[f(x)]$. The SVM is trained with all genes, the expression values of genes corresponding to $|w_i|$ in the bottom 10% are removed and the SVM is retrained with the smaller gene expression set.

Proportional chance criterion

In order to compute p-values for multi-class prediction, we used a "proportional chance criterion" to evaluate the probability that a random predictor will produce a confusion matrix with the same row and column counts as the gene expression predictor. For example, for a binary class (A vs. B) problem, if α is the prior probability of a sample being in class A and *p* is the true proportion of samples in class A then $C_p = p\,\alpha + (1-p)\,(1-\alpha)$ is the proportion of the overall sample that is expected to receive correct classification by chance alone. Then if C_{model} is the proportion of correct classifications achieved by the gene expression predictor one can estimate its significance by using a Z statistic of the form: $(C_{model} - C_p)/\mathrm{Sqrt}(C_p\,(1-C_p)/n)$, where *n* is the total sample count. For more details see chapter VII of Huberty's *Applied Discriminant Analysis*.

Multi-class Prediction Results

In a preliminary empirical study of multi-class methods and algorithms (Yeang et al 2001) we applied the OVA and AP approaches with three different algorithms: Weighted Voting, k-Nearest Neighbors and Support Vector Machines. The results, shown in Table 2, demonstrate that the OVA approach in combination with SVM gave us the most accurate method by a significant margin, and we describe this method in detail below. See Yeang et al 2001 for more details on using other algorithms.

Genes per Classifier	Weighted Voting	Weighted Voting	k-NN	k-NN	SVM	SVM
	OVA	All Pairs	OVA	All Pairs	OVA	All Pairs

30	60.0%	62.3%	65.3%	67.2%	70.8%	64.2%
92	59.3%	59.6%	68.0%	67.3%	72.2%	64.8%
281	57.8%	57.2%	65.7%	67.0%	73.4%	65.1%
1073	53.5%	52.4%	66.5%	64.8%	74.1%	64.9%
3276	43.4%	48.8%	66.3%	62.0%	74.7%	64.7%
6400	38.5%	45.6%	64.2%	58.4%	75.5%	64.6%
All	-	-	-	-	**78.0%**	64.7%

Table 1. Accuracy of different combinations of multi-class approaches and algorithms.

SVM/OVA Multi-class Prediction

The procedure for this approach is as follows:

- Define each target class based on histopathologic clinical evaluation (pathology review) of tumor specimens;
- Decompose the multi-class problem into a series of 14 binary OVA classification problems: one for each class.
- For each class optimize the binary classifiers on the training set using leave-one-out cross-validation i.e. remove one sample, train the binary classifier on the remaining samples, combine the individual binary classifiers to predict the class of the left out sample, and iteratively repeat this process for all the samples. A cumulative error rate is calculated.
- Evaluate the final prediction model on an independent test set.

This procedure is described pictorially in Figure 7 where the bar graphs on the lower right side show an example of actual SVM output predictions for a Breast adenocarcinoma sample.

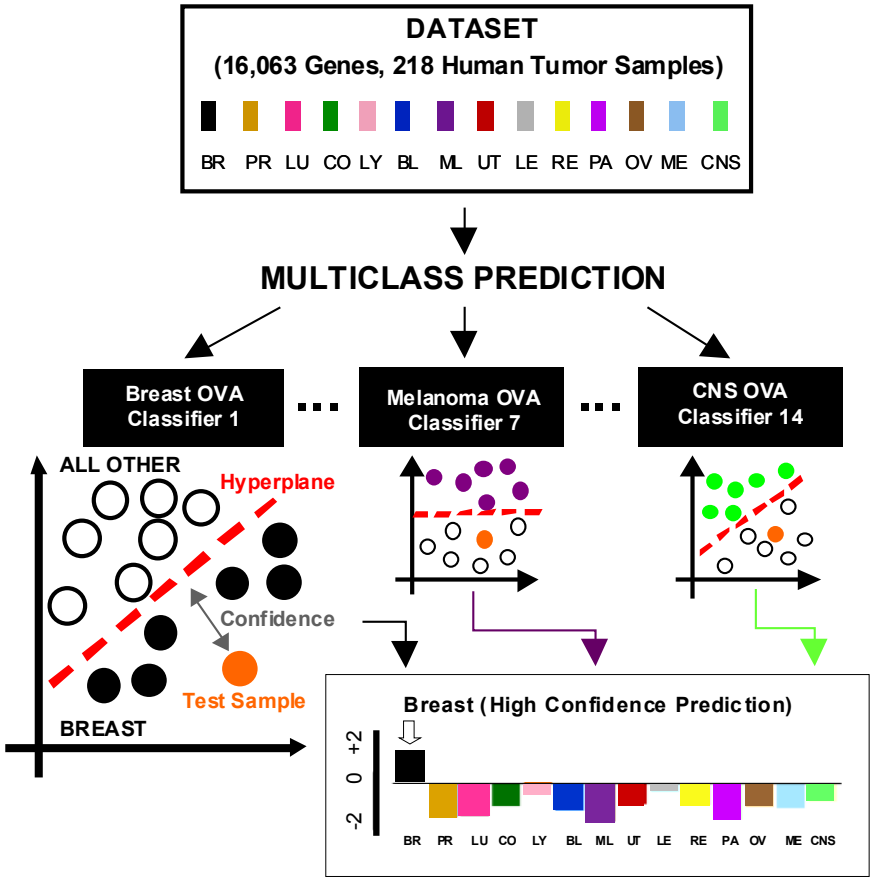


Figure 7. Multi-class methodology using a One vs. All (OVA) approach. The bars graphs on the lower right side show the results of a high and low confidence predictions.

The final prediction (winning class) of the OVA set of classifiers is the one corresponding to the largest confidence (margin),

$$\boldsymbol{class} = \arg\max_{i=1..K} \boldsymbol{f}_i ,$$

The confidence of the final call is the margin of the winning SVM. When the largest confidence is positive the final prediction is considered a "high confidence" call. If negative it is a "low confidence" call that can also be considered a candidate for a no-call because no single SVM "claims" the sample as belonging to its recognizable class. We analyze the error rates in terms of totals and also in terms of high and low confidence calls. In the example in the lower right hand side of Figure 7, an example of a high confidence call, the Breast classifier attains a large positive margin while the other classifiers all have negative margins.

Repeating this procedure we created a multi-class OVA-SVM model with all genes using the training dataset and then applied it to two test datasets (Independent Test Set and Poorly-differentiated adenocarcinomas). The results are summarized in Figure 8.

a

Dataset	Method	Samples	Accuracy	Confidence			
				High		Low	
				Fraction	Accuracy	Fraction	Accuracy
Training	CV	144	78%	80%	90%	20%	28%
Test	Train / Test	54	78%	78%	83%	22%	58%
PD	Train / Test	20	30%	50%	50%	50%	10%

b

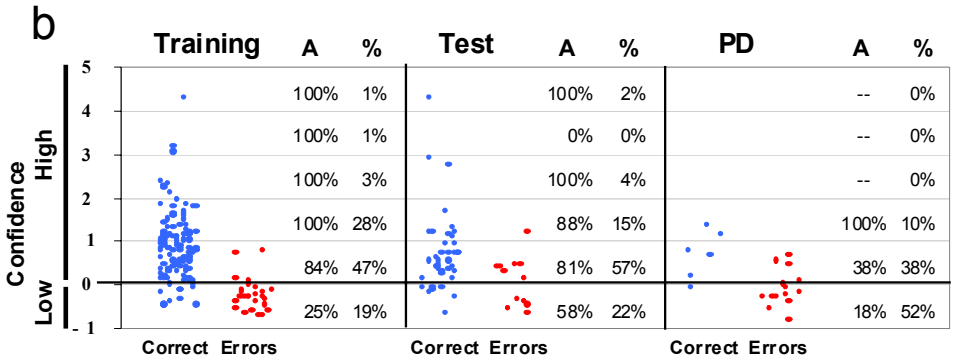


Figure 8: Accuracy results for the OVM / SVM classifier.

As can be seen in the table in cross validation the overall multi-class predictions were correct for 78% of the tumors. This accuracy is substantially higher than expected for random prediction (9% according to proportional chance criterion (see below)). More interestingly the majority of calls (80%) were high confidence, and for these the classifier achieved an accuracy of 90%. The remaining tumors (20%) had low confidence calls and lower accuracy (28%). The results for the test set are similar to the ones obtained in cross-validation: the overall prediction accuracy was 78% and the majority of these predictions (78%) were again high confidence with an accuracy of 83%. Low confidence calls were made on the remaining 22% of tumors with an accuracy of 58%. The actual confidences for each call and a bar graph of accuracy and fraction of calls versus confidence is shown in Figure 8(b). The confusion matrices for cross-validation (Train) and Independent Test Set (Test) are shown in Figure 9.

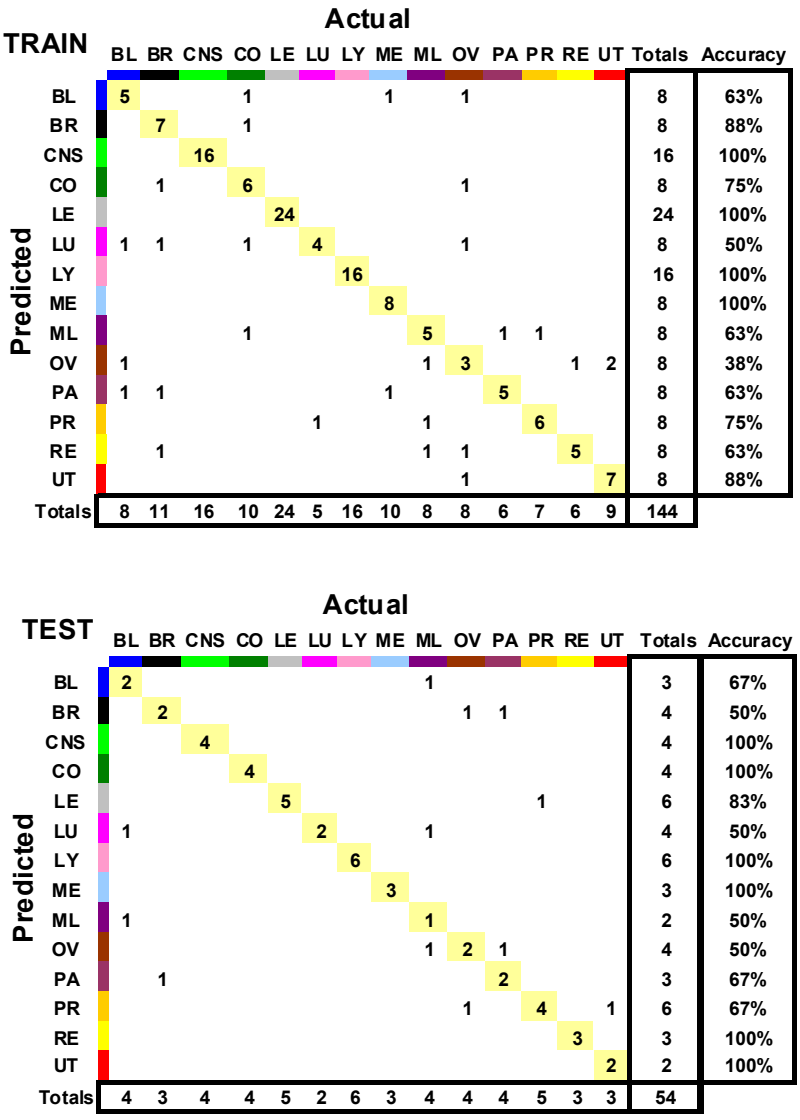


Figure 9. Confusion matrices for the OVA / SVM classifier.

An interesting observation concerning these results is that for 50% of the tumors that were incorrectly classified the correct answer corresponded to the second or third most confident (SVM) prediction. This is shown in Figure 10.

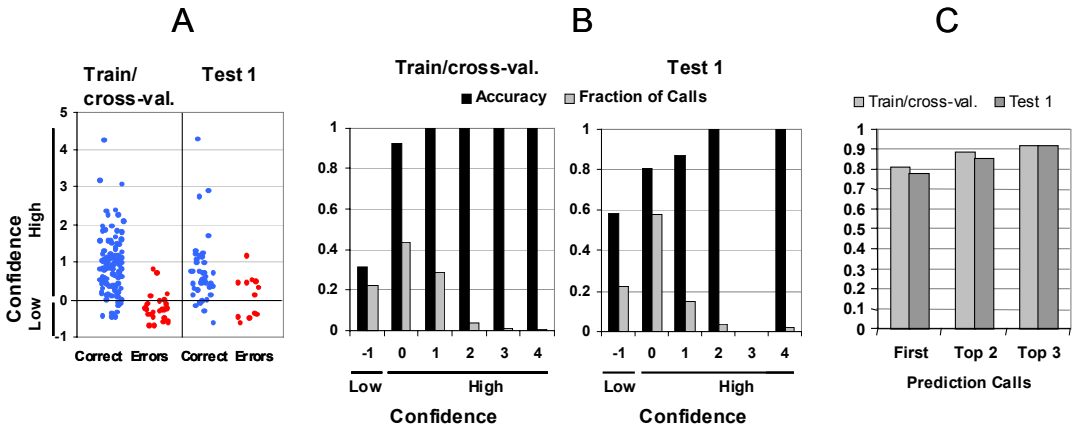
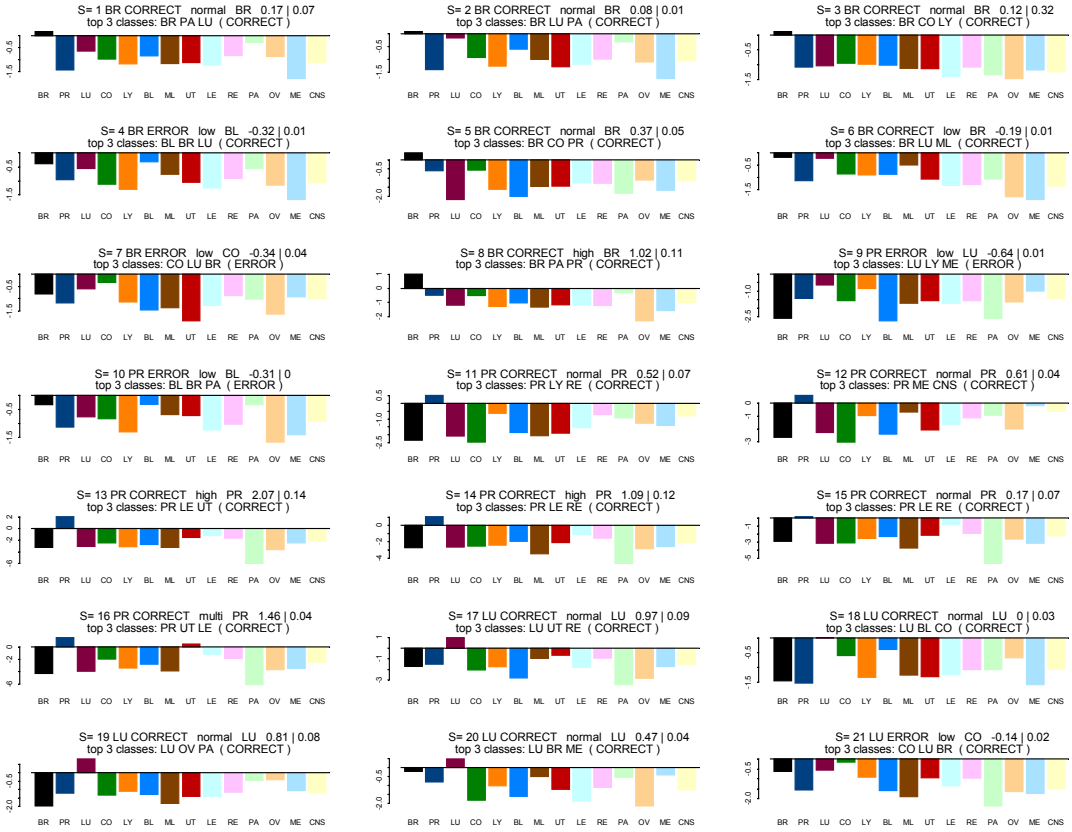
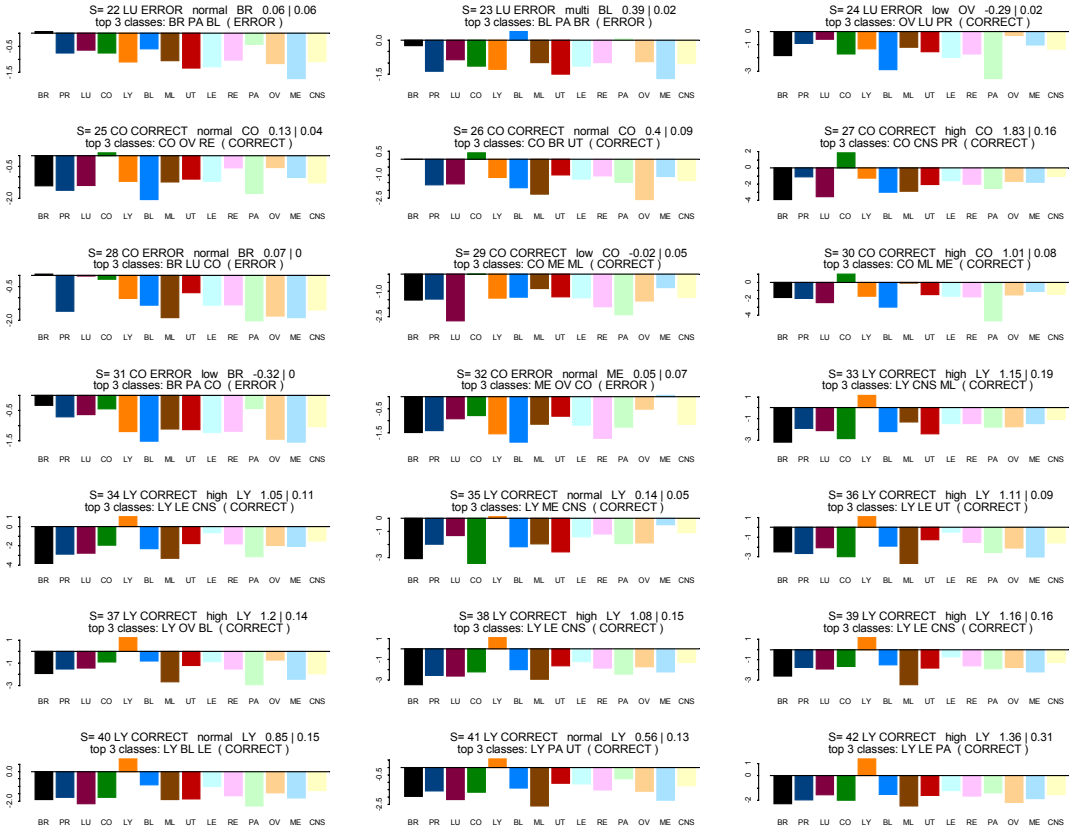


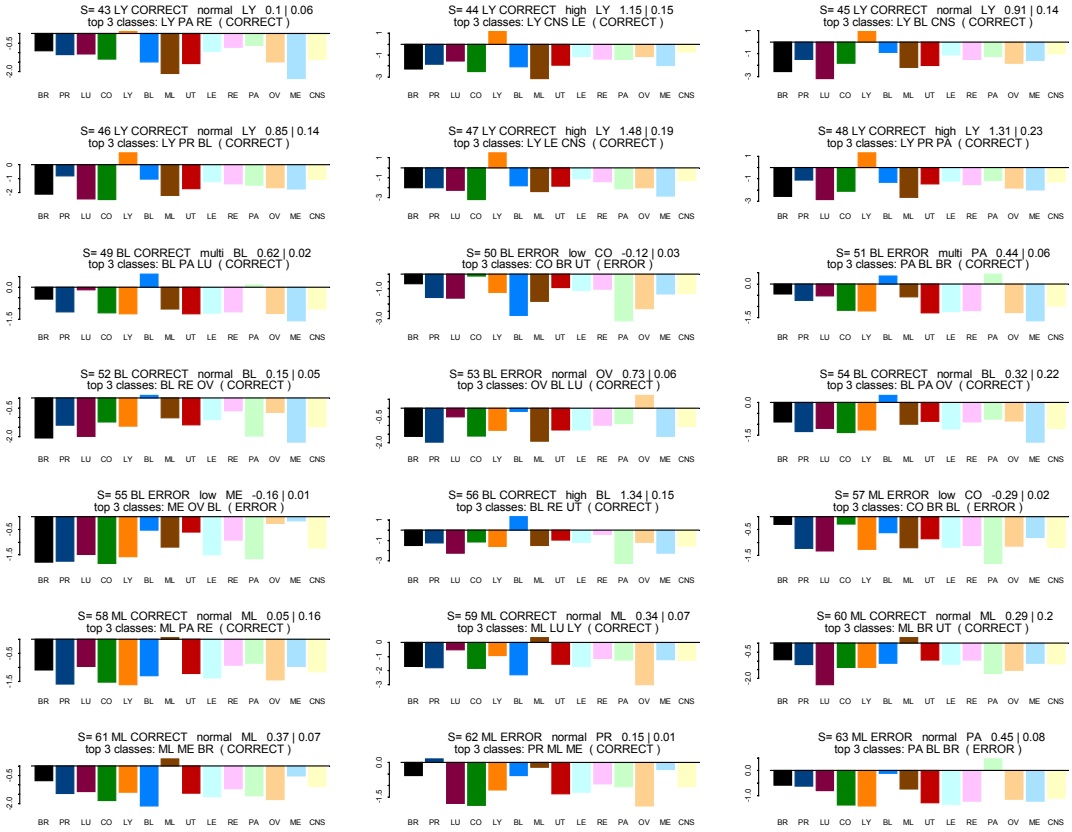
Figure 10. Individual sample errors and confidence calls (A), accuracy bar graphs (B) and performance considering second-and third most confident prediction (C) for the train and test datasets.

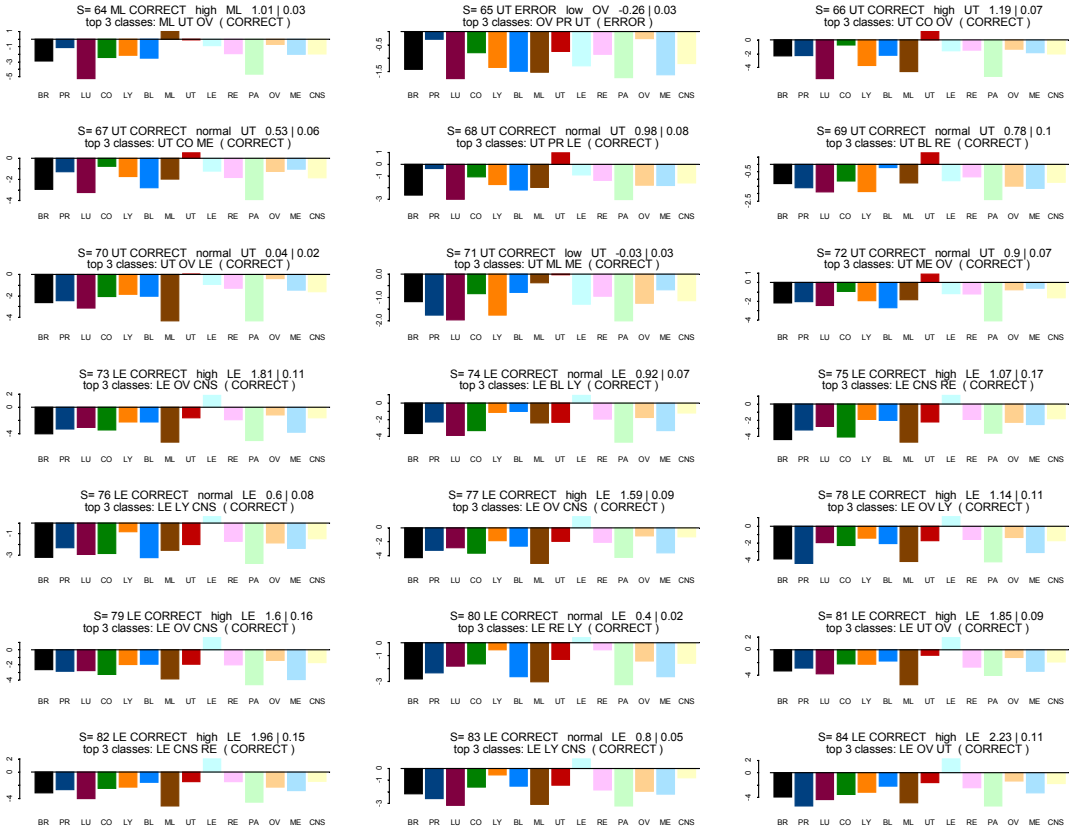
Below are histograms of prediction for the Training Set (cross-validation), Test Set, and Poorly Differentiated tumor set. The sample number (S), the actual class of the sample, Correct / Incorrect designation, High (high or normal) / Low (low) confidence calls, the prediction strength of the winning class, and the top 3 predicted classes are delineated for each sample. Multi refers to samples which triggered positive prediction strengths for more than one OVA predictor. Sample numbers refer to specimens as outlined in SAMPLES.xls.

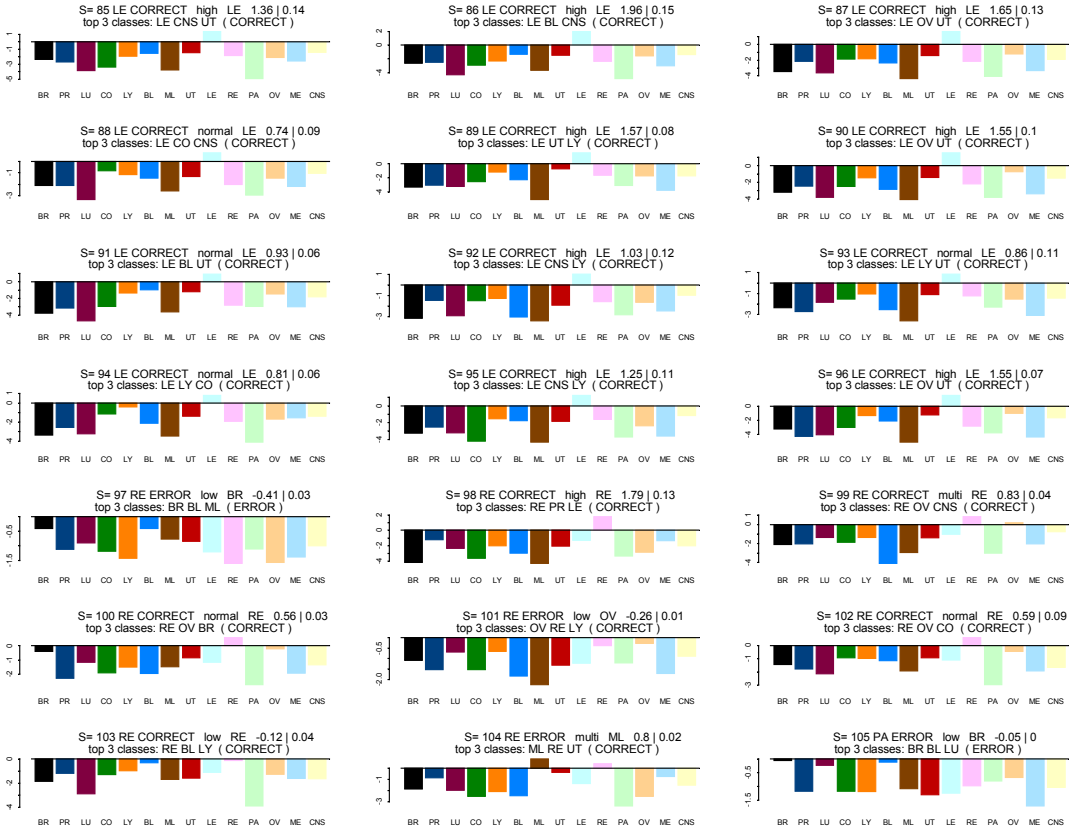
Training Set:

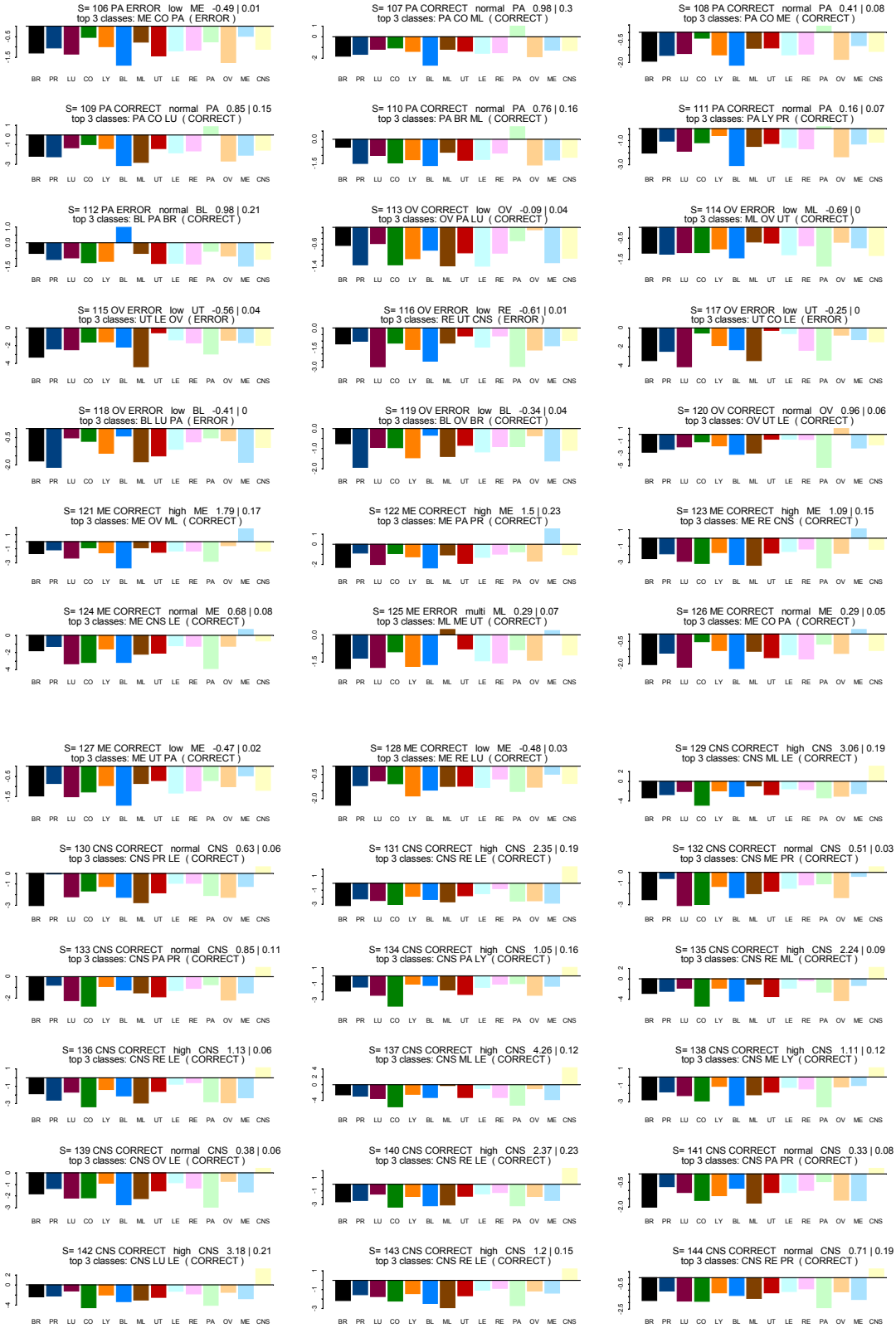

S= 1 BR CORRECT normal BR 0.17 | 0.07
top 3 classes: BR PA LU (CORRECT)
S= 2 BR CORRECT normal BR 0.08 | 0.01
top 3 classes: BR LU PA (CORRECT)
S= 3 BR CORRECT normal BR 0.12 | 0.32
top 3 classes: BR CO LY (CORRECT)
S= 4 BR ERROR low BL -0.32 | 0.01
top 3 classes: BL BR LU (CORRECT)
S= 5 BR CORRECT normal BR 0.37 | 0.05
top 3 classes: BR CO PR (CORRECT)
S= 6 BR CORRECT low BR -0.19 | 0.01
top 3 classes: BR LU ML (CORRECT)
S= 7 BR ERROR low CO -0.34 | 0.04
top 3 classes: CO LU BR (ERROR)
S= 8 BR CORRECT high BR 1.02 | 0.11
top 3 classes: BR PA PR (CORRECT)
S= 9 PR ERROR low LU -0.64 | 0.01
top 3 classes: LU LY ME (ERROR)
S= 10 PR ERROR low BL -0.31 | 0
top 3 classes: BL BR PA (ERROR)
S= 11 PR CORRECT normal PR 0.52 | 0.07
top 3 classes: PR LY RE (CORRECT)
S= 12 PR CORRECT normal PR 0.61 | 0.04
top 3 classes: PR ME CNS (CORRECT)
S= 13 PR CORRECT high PR 2.07 | 0.14
top 3 classes: PR LE UT (CORRECT)
S= 14 PR CORRECT high PR 1.09 | 0.12
top 3 classes: PR LE RE (CORRECT)
S= 15 PR CORRECT normal PR 0.17 | 0.07
top 3 classes: PR LE RE (CORRECT)
S= 16 PR CORRECT multi PR 1.46 | 0.04
top 3 classes: PR UT LE (CORRECT)
S= 17 LU CORRECT normal LU 0.97 | 0.09
top 3 classes: LU UT RE (CORRECT)
S= 18 LU CORRECT normal LU 0 | 0.03
top 3 classes: LU BL CO (CORRECT)
S= 19 LU CORRECT normal LU 0.81 | 0.08
top 3 classes: LU OV PA (CORRECT)
S= 20 LU CORRECT normal LU 0.47 | 0.04
top 3 classes: LU BR ME (CORRECT)
S= 21 LU ERROR low CO -0.14 | 0.02
top 3 classes: CO LU BR (CORRECT)
BR PR LU CO LY BL ML UT LE RE PA OV ME CNS


S= 22 LU ERROR normal BR 0.06 | 0.06
top 3 classes: BR PA BL (ERROR)
S= 23 LU ERROR multi BL 0.39 | 0.02
top 3 classes: BL PA BR (ERROR)
S= 24 LU ERROR low OV -0.29 | 0.02
top 3 classes: OV LU PR (CORRECT)
S= 25 CO CORRECT normal CO 0.13 | 0.04
top 3 classes: CO OV RE (CORRECT)
S= 26 CO CORRECT normal CO 0.4 | 0.09
top 3 classes: CO BR UT (CORRECT)
S= 27 CO CORRECT high CO 1.83 | 0.16
top 3 classes: CO CNS PR (CORRECT)
S= 28 CO ERROR normal BR 0.07 | 0
top 3 classes: BR LU CO (ERROR)
S= 29 CO CORRECT low CO -0.02 | 0.05
top 3 classes: CO ME ML (CORRECT)
S= 30 CO CORRECT high CO 1.01 | 0.08
top 3 classes: CO ML ME (CORRECT)
S= 31 CO ERROR low BR -0.32 | 0
top 3 classes: BR PA CO (ERROR)
S= 32 CO ERROR normal ME 0.05 | 0.07
top 3 classes: ME OV CO (ERROR)
S= 33 LY CORRECT high LY 1.15 | 0.19
top 3 classes: LY CNS ML (CORRECT)
S= 34 LY CORRECT high LY 1.05 | 0.11
top 3 classes: LY LE CNS (CORRECT)
S= 35 LY CORRECT normal LY 0.14 | 0.05
top 3 classes: LY ME CNS (CORRECT)
S= 36 LY CORRECT high LY 1.11 | 0.09
top 3 classes: LY LE UT (CORRECT)
S= 37 LY CORRECT high LY 1.2 | 0.14
top 3 classes: LY OV BL (CORRECT)
S= 38 LY CORRECT high LY 1.08 | 0.15
top 3 classes: LY LE CNS (CORRECT)
S= 39 LY CORRECT high LY 1.16 | 0.16
top 3 classes: LY LE CNS (CORRECT)
S= 40 LY CORRECT normal LY 0.85 | 0.15
top 3 classes: LY BL LE (CORRECT)
S= 41 LY CORRECT normal LY 0.56 | 0.13
top 3 classes: LY PA UT (CORRECT)
S= 42 LY CORRECT high LY 1.36 | 0.31
top 3 classes: LY LE PA (CORRECT)
BR PR LU CO LY BL ML UT LE RE PA OV ME CNS

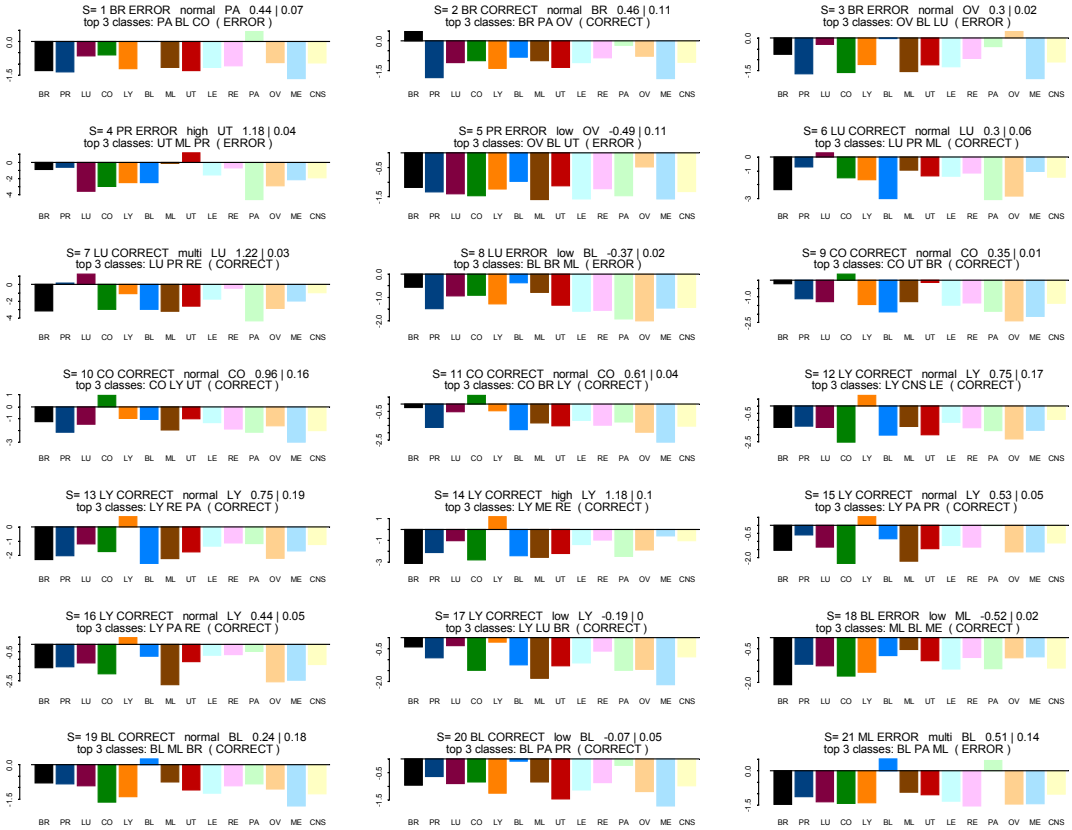

S= 43 LY CORRECT normal LY 0.1 | 0.06
top 3 classes: LY PA RE (CORRECT)
S= 44 LY CORRECT high LY 1.15 | 0.15
top 3 classes: LY CNS LE (CORRECT)
S= 45 LY CORRECT normal LY 0.91 | 0.14
top 3 classes: LY BL CNS (CORRECT)
S= 46 LY CORRECT normal LY 0.85 | 0.14
top 3 classes: LY PR BL (CORRECT)
S= 47 LY CORRECT high LY 1.48 | 0.19
top 3 classes: LY LE CNS (CORRECT)
S= 48 LY CORRECT high LY 1.31 | 0.23
top 3 classes: LY PR PA (CORRECT)
S= 49 BL CORRECT multi BL 0.62 | 0.02
top 3 classes: BL PA LU (CORRECT)
S= 50 BL ERROR low CO -0.12 | 0.03
top 3 classes: CO BR UT (ERROR)
S= 51 BL ERROR multi PA 0.44 | 0.06
top 3 classes: PA BL BR (CORRECT)
S= 52 BL CORRECT normal BL 0.15 | 0.05
top 3 classes: BL RE OV (CORRECT)
S= 53 BL ERROR normal OV 0.73 | 0.06
top 3 classes: OV BL LU (CORRECT)
S= 54 BL CORRECT normal BL 0.32 | 0.22
top 3 classes: BL PA OV (CORRECT)
S= 55 BL ERROR low ME -0.16 | 0.01
top 3 classes: ME OV BL (ERROR)
S= 56 BL CORRECT high BL 1.34 | 0.15
top 3 classes: BL RE UT (CORRECT)
S= 57 ML ERROR low CO -0.29 | 0.02
top 3 classes: CO BR BL (ERROR)
S= 58 ML CORRECT normal ML 0.05 | 0.16
top 3 classes: ML PA RE (CORRECT)
S= 59 ML CORRECT normal ML 0.34 | 0.07
top 3 classes: ML LU LY (CORRECT)
S= 60 ML CORRECT normal ML 0.29 | 0.2
top 3 classes: ML BR UT (CORRECT)
S= 61 ML CORRECT normal ML 0.37 | 0.07
top 3 classes: ML ME BR (CORRECT)
S= 62 ML ERROR normal PR 0.15 | 0.01
top 3 classes: PR ML ME (CORRECT)
S= 63 ML ERROR normal PA 0.45 | 0.08
top 3 classes: PA BL BR (ERROR)
BR PR LU CO LY BL ML UT LE RE PA OV ME CNS

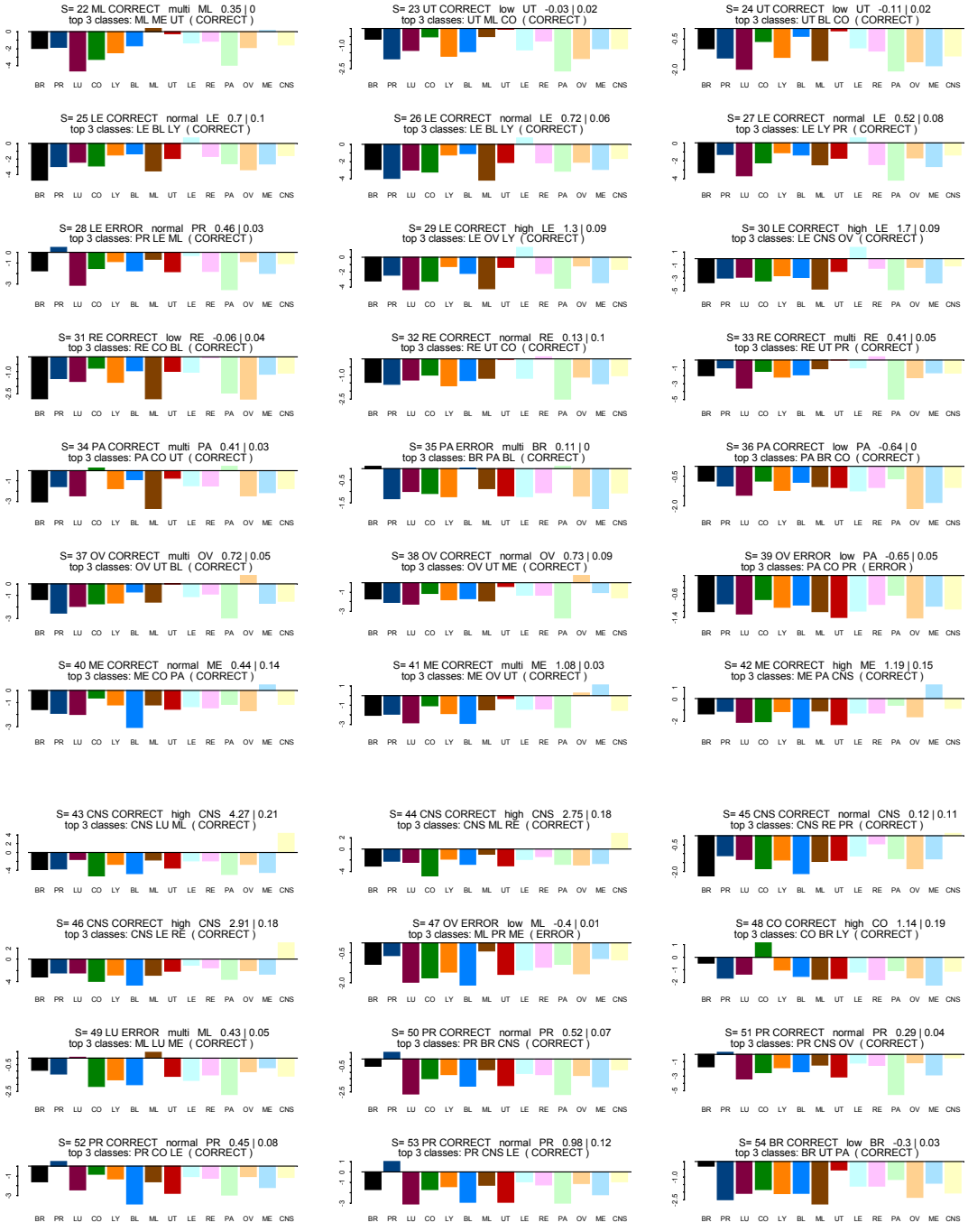

S= 64 ML CORRECT high ML 1.01 | 0.03
top 3 classes: ML UT OV (CORRECT)
S= 65 UT ERROR low OV -0.26 | 0.03
top 3 classes: OV PR UT (ERROR)
S= 66 UT CORRECT high UT 1.19 | 0.07
top 3 classes: UT CO OV (CORRECT)
S= 67 UT CORRECT normal UT 0.53 | 0.06
top 3 classes: UT CO ME (CORRECT)
S= 68 UT CORRECT normal UT 0.98 | 0.08
top 3 classes: UT PR LE (CORRECT)
S= 69 UT CORRECT normal UT 0.78 | 0.1
top 3 classes: UT BL RE (CORRECT)
S= 70 UT CORRECT normal UT 0.04 | 0.02
top 3 classes: UT OV LE (CORRECT)
S= 71 UT CORRECT low UT -0.03 | 0.03
top 3 classes: UT ML ME (CORRECT)
S= 72 UT CORRECT normal UT 0.9 | 0.07
top 3 classes: UT ME OV (CORRECT)
S= 73 LE CORRECT high LE 1.81 | 0.11
top 3 classes: LE OV CNS (CORRECT)
S= 74 LE CORRECT normal LE 0.92 | 0.07
top 3 classes: LE BL LY (CORRECT)
S= 75 LE CORRECT high LE 1.07 | 0.17
top 3 classes: LE CNS RE (CORRECT)
S= 76 LE CORRECT normal LE 0.6 | 0.08
top 3 classes: LE LY CNS (CORRECT)
S= 77 LE CORRECT high LE 1.59 | 0.09
top 3 classes: LE OV CNS (CORRECT)
S= 78 LE CORRECT high LE 1.14 | 0.11
top 3 classes: LE OV LY (CORRECT)
S= 79 LE CORRECT high LE 1.6 | 0.16
top 3 classes: LE OV CNS (CORRECT)
S= 80 LE CORRECT normal LE 0.4 | 0.02
top 3 classes: LE RE LY (CORRECT)
S= 81 LE CORRECT high LE 1.85 | 0.09
top 3 classes: LE UT OV (CORRECT)
S= 82 LE CORRECT high LE 1.96 | 0.15
top 3 classes: LE CNS RE (CORRECT)
S= 83 LE CORRECT normal LE 0.8 | 0.05
top 3 classes: LE LY CNS (CORRECT)
S= 84 LE CORRECT high LE 2.23 | 0.11
top 3 classes: LE OV UT (CORRECT)
BR PR LU CO LY BL ML UT LE RE PA OV ME CNS


S= 85 LE CORRECT high LE 1.36 | 0.14
top 3 classes: LE CNS UT (CORRECT)
S= 86 LE CORRECT high LE 1.96 | 0.15
top 3 classes: LE BL CNS (CORRECT)
S= 87 LE CORRECT high LE 1.65 | 0.13
top 3 classes: LE OV UT (CORRECT)
S= 88 LE CORRECT normal LE 0.74 | 0.09
top 3 classes: LE CO CNS (CORRECT)
S= 89 LE CORRECT high LE 1.57 | 0.08
top 3 classes: LE UT LY (CORRECT)
S= 90 LE CORRECT high LE 1.55 | 0.1
top 3 classes: LE OV UT (CORRECT)
S= 91 LE CORRECT normal LE 0.93 | 0.06
top 3 classes: LE BL UT (CORRECT)
S= 92 LE CORRECT high LE 1.03 | 0.12
top 3 classes: LE CNS LY (CORRECT)
S= 93 LE CORRECT normal LE 0.86 | 0.11
top 3 classes: LE LY UT (CORRECT)
S= 94 LE CORRECT normal LE 0.81 | 0.06
top 3 classes: LE LY CO (CORRECT)
S= 95 LE CORRECT high LE 1.25 | 0.11
top 3 classes: LE CNS LY (CORRECT)
S= 96 LE CORRECT high LE 1.55 | 0.07
top 3 classes: LE OV UT (CORRECT)
S= 97 RE ERROR low BR -0.41 | 0.03
top 3 classes: BR BL ML (ERROR)
S= 98 RE CORRECT high RE 1.79 | 0.13
top 3 classes: RE PR LE (CORRECT)
S= 99 RE CORRECT multi RE 0.83 | 0.04
top 3 classes: RE OV CNS (CORRECT)
S= 100 RE CORRECT normal RE 0.56 | 0.03
top 3 classes: RE OV BR (CORRECT)
S= 101 RE ERROR low OV -0.26 | 0.01
top 3 classes: OV RE LY (CORRECT)
S= 102 RE CORRECT normal RE 0.59 | 0.09
top 3 classes: RE OV CO (CORRECT)
S= 103 RE CORRECT low RE -0.12 | 0.04
top 3 classes: RE BL LY (CORRECT)
S= 104 RE ERROR multi ML 0.8 | 0.02
top 3 classes: ML RE UT (CORRECT)
S= 105 PA ERROR low BR -0.05 | 0
top 3 classes: BR BL LU (ERROR)
BR PR LU CO LY BL ML UT LE RE PA OV ME CNS

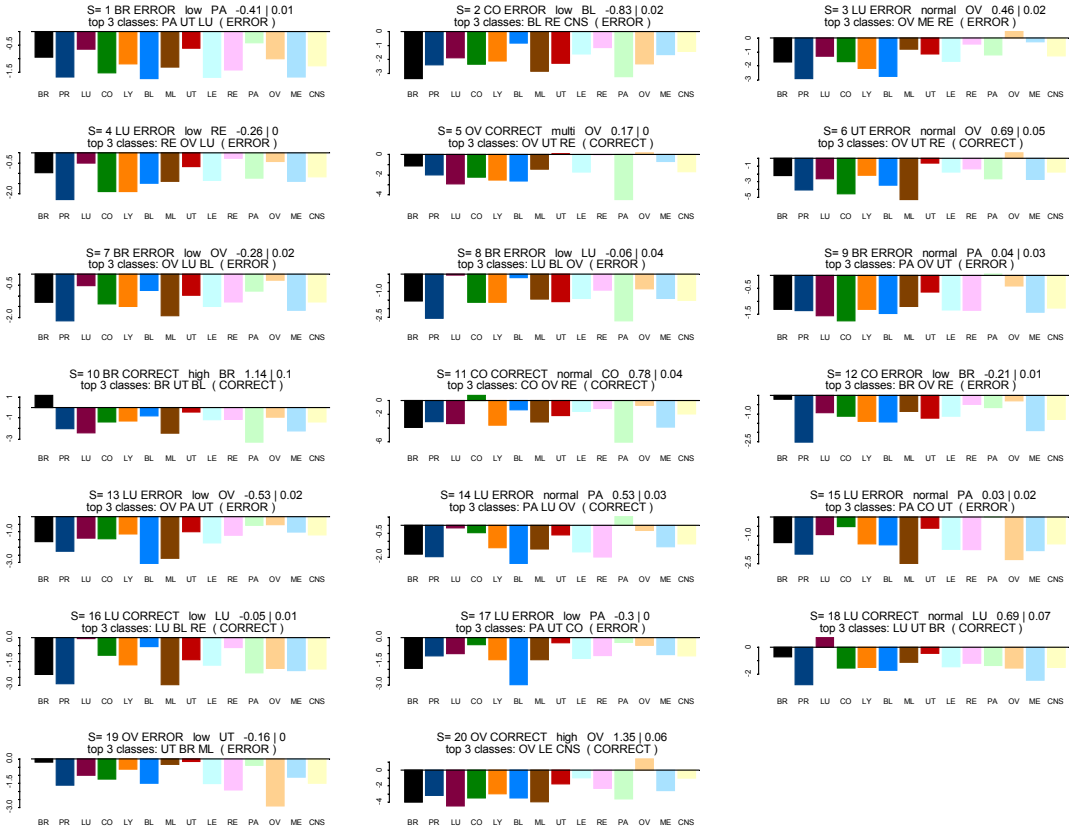

S= 106 PA ERROR low ME -0.49 | 0.01
top 3 classes: ME CO PA (ERROR)
S= 107 PA CORRECT normal PA 0.98 | 0.3
top 3 classes: PA CO ML (CORRECT)
S= 108 PA CORRECT normal PA 0.41 | 0.08
top 3 classes: PA CO ME (CORRECT)
S= 109 PA CORRECT normal PA 0.85 | 0.15
top 3 classes: PA CO LU (CORRECT)
S= 110 PA CORRECT normal PA 0.76 | 0.16
top 3 classes: PA BR ML (CORRECT)
S= 111 PA CORRECT normal PA 0.16 | 0.07
top 3 classes: PA LY PR (CORRECT)
S= 112 PA ERROR normal BL 0.98 | 0.21
top 3 classes: BL PA BR (CORRECT)
S= 113 OV CORRECT low OV -0.09 | 0.04
top 3 classes: OV PA LU (CORRECT)
S= 114 OV ERROR low ML -0.69 | 0
top 3 classes: ML OV UT (CORRECT)
S= 115 OV ERROR low UT -0.56 | 0.04
top 3 classes: UT LE OV (ERROR)
S= 116 OV ERROR low RE -0.61 | 0.01
top 3 classes: RE UT CNS (ERROR)
S= 117 OV ERROR low UT -0.25 | 0
top 3 classes: UT CO LE (ERROR)
S= 118 OV ERROR low BL -0.41 | 0
top 3 classes: BL LU PA (ERROR)
S= 119 OV ERROR low BL -0.34 | 0.04
top 3 classes: BL OV BR (CORRECT)
S= 120 OV CORRECT normal OV 0.96 | 0.06
top 3 classes: OV UT LE (CORRECT)
S= 121 ME CORRECT high ME 1.79 | 0.17
top 3 classes: ME OV ML (CORRECT)
S= 122 ME CORRECT high ME 1.5 | 0.23
top 3 classes: ME PA PR (CORRECT)
S= 123 ME CORRECT high ME 1.09 | 0.15
top 3 classes: ME RE CNS (CORRECT)
S= 124 ME CORRECT normal ME 0.68 | 0.08
top 3 classes: ME CNS LE (CORRECT)
S= 125 ME ERROR multi ML 0.29 | 0.07
top 3 classes: ML ME UT (CORRECT)
S= 126 ME CORRECT normal ME 0.29 | 0.05
top 3 classes: ME CO PA (CORRECT)
S= 127 ME CORRECT low ME -0.47 | 0.02
top 3 classes: ME UT PA (CORRECT)
S= 128 ME CORRECT low ME -0.48 | 0.03
top 3 classes: ME RE LU (CORRECT)
S= 129 CNS CORRECT high CNS 3.06 | 0.19
top 3 classes: CNS ML LE (CORRECT)
S= 130 CNS CORRECT normal CNS 0.63 | 0.06
top 3 classes: CNS PR LE (CORRECT)
S= 131 CNS CORRECT high CNS 2.35 | 0.19
top 3 classes: CNS RE LE (CORRECT)
S= 132 CNS CORRECT normal CNS 0.51 | 0.03
top 3 classes: CNS ME PR (CORRECT)
S= 133 CNS CORRECT normal CNS 0.85 | 0.11
top 3 classes: CNS PA PR (CORRECT)
S= 134 CNS CORRECT high CNS 1.05 | 0.16
top 3 classes: CNS PA LY (CORRECT)
S= 135 CNS CORRECT high CNS 2.24 | 0.09
top 3 classes: CNS RE ML (CORRECT)
S= 136 CNS CORRECT high CNS 1.13 | 0.06
top 3 classes: CNS RE LE (CORRECT)
S= 137 CNS CORRECT high CNS 4.26 | 0.12
top 3 classes: CNS ML LE (CORRECT)
S= 138 CNS CORRECT high CNS 1.11 | 0.12
top 3 classes: CNS ME LY (CORRECT)
S= 139 CNS CORRECT normal CNS 0.38 | 0.06
top 3 classes: CNS OV LE (CORRECT)
S= 140 CNS CORRECT high CNS 2.37 | 0.23
top 3 classes: CNS RE LE (CORRECT)
S= 141 CNS CORRECT normal CNS 0.33 | 0.08
top 3 classes: CNS PA PR (CORRECT)
S= 142 CNS CORRECT high CNS 3.18 | 0.21
top 3 classes: CNS LU LE (CORRECT)
S= 143 CNS CORRECT high CNS 1.2 | 0.15
top 3 classes: CNS RE LE (CORRECT)
S= 144 CNS CORRECT normal CNS 0.71 | 0.19
top 3 classes: CNS RE PR (CORRECT)
BR PR LU CO LY BL ML UT LE RE PA OV ME CNS

Test Set:


S= 1 BR ERROR normal PA 0.44 | 0.07
top 3 classes: PA BL CO (ERROR)
S= 2 BR CORRECT normal BR 0.46 | 0.11
top 3 classes: BR PA OV (CORRECT)
S= 3 BR ERROR normal OV 0.3 | 0.02
top 3 classes: OV BL LU (ERROR)
S= 4 PR ERROR high UT 1.18 | 0.04
top 3 classes: UT ML PR (ERROR)
S= 5 PR ERROR low OV -0.49 | 0.11
top 3 classes: OV BL UT (ERROR)
S= 6 LU CORRECT normal LU 0.3 | 0.06
top 3 classes: LU PR ML (CORRECT)
S= 7 LU CORRECT multi LU 1.22 | 0.03
top 3 classes: LU PR RE (CORRECT)
S= 8 LU ERROR low BL -0.37 | 0.02
top 3 classes: BL BR ML (ERROR)
S= 9 CO CORRECT normal CO 0.35 | 0.01
top 3 classes: CO UT BR (CORRECT)
S= 10 CO CORRECT normal CO 0.96 | 0.16
top 3 classes: CO LY UT (CORRECT)
S= 11 CO CORRECT normal CO 0.61 | 0.04
top 3 classes: CO BR LY (CORRECT)
S= 12 LY CORRECT normal LY 0.75 | 0.17
top 3 classes: LY CNS LE (CORRECT)
S= 13 LY CORRECT normal LY 0.75 | 0.19
top 3 classes: LY RE PA (CORRECT)
S= 14 LY CORRECT high LY 1.18 | 0.1
top 3 classes: LY ME RE (CORRECT)
S= 15 LY CORRECT normal LY 0.53 | 0.05
top 3 classes: LY PA PR (CORRECT)
S= 16 LY CORRECT normal LY 0.44 | 0.05
top 3 classes: LY PA RE (CORRECT)
S= 17 LY CORRECT low LY -0.19 | 0
top 3 classes: LY LU BR (CORRECT)
S= 18 BL ERROR low ML -0.52 | 0.02
top 3 classes: ML BL ME (CORRECT)
S= 19 BL CORRECT normal BL 0.24 | 0.18
top 3 classes: BL ML BR (CORRECT)
S= 20 BL CORRECT low BL -0.07 | 0.05
top 3 classes: BL PA PR (CORRECT)
S= 21 ML ERROR multi BL 0.51 | 0.14
top 3 classes: BL PA ML (ERROR)
BR PR LU CO LY BL ML UT LE RE PA OV ME CNS



Poorly Differentiated:



To confirm the stability and reproducibility of the prediction results for this collection of samples we repeated the train and test procedure for 100 random splits of a combined dataset. The results were similar to the reported case. Figure 11 shows the mean of the error rate for the different test-train splits as a function of the total number of genes. Due to the fact the different test-train splits were obtained by reshuffling the dataset the empirical variance measured is optimistic (Efron and Tibshirani, 1993).

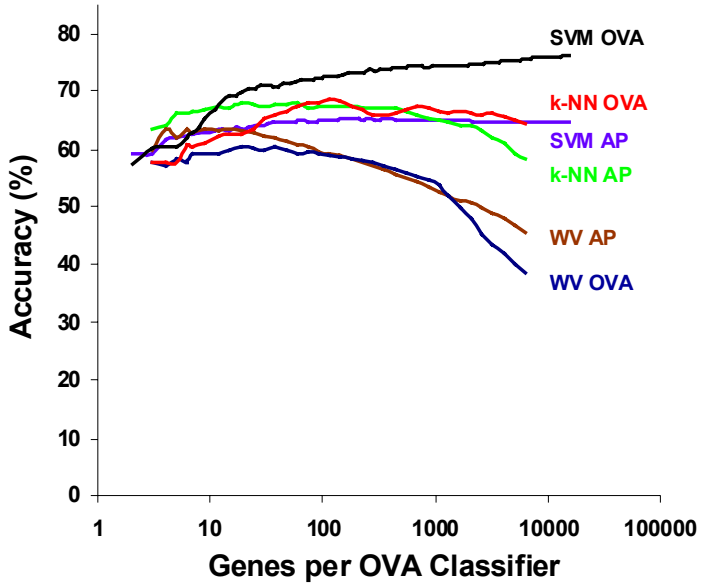


Figure 11. Mean classification accuracy and standard deviation plotted as a function of number of genes used by the classifier. The prediction accuracy decreases with decreasing number of genes.

We also analyzed the accuracy of the multi-class SVM predictor as a function of the number of genes. The algorithm inputs all of the 16,063 genes in the array and each of them is assigned a weight based on its relative contribution to each OVA classification. Practically all genes were assigned weakly positive and negative weights in each OVA classifier. We performed multiple runs with different numbers of genes selected using RFE. Results are also shown in Figure 11, where total accuracy decreases as the number of input genes decreases for each OVA distinction. Pairwise distinctions can easily be made between some tumor classes using fewer genes but multi-class distinctions among highly related tumor types are intrinsically more difficult. This behavior can also be the result of the existence of molecularly distinct but unknown sub-classes within known classes that effectively decrease the predictive power of the multi-class method. Despite the increasing accuracy with increased number of genes trend, significant but modest prediction accuracy can be achieved with a relatively small number of genes per classifier (e.g. about 70% with about 200 total genes).

Appendix: Support Vector Machines

The problem of learning a classification boundary given positive and negative examples is a particular case of the problem of approximating a multivariate function from sparse data. The problem of approximating a function from sparse data is ill-posed and regularization theory is a classical approach to solving it (Tikhonov and Arsenin 1977).

Standard regularization theory formulates the approximation problem as a variational problem of finding the function f that minimizes the functional

$$\min_{f \in H} \frac{1}{\ell} \sum_{i=1}^{\ell} V(y_i, f(x_i)) + \lambda \|f\|_K^2$$

where $V(\cdot,\cdot)$ is a loss function, $\|f\|_K^2$ is a norm in a Reproducing Kernel Hilbert Space defined by the positive function K (Aronszsajn 1950), ℓ is the number of training examples, and λ is the regularization parameter. Under rather general conditions the solution to the above functional has the form

$$f(x) = \sum_{i=1}^{\ell} \alpha_i K(x, x_i).$$

SVMs are a particular case of the above regularization framework (Evgeniou et al 2000).

The SVM runs described in this paper were performed using a modified version of SvmFu (http://www.ai.mit.edu/projects/cbcl/software-datasets/index.html). For a more comprehensive introduction to SVM see Evgeniou et al 2000 and Vapnik 1998.

For the SVM the regularization functional minimized is the following

$$\min_{f \in H} \frac{1}{\ell} \sum_{i=1}^{\ell} (1 - y_i f(x_i))_+ + \lambda \|f\|_K^2,$$

where the hinge loss function is used, $(a)_+$ is the $\min(a, 0)$. The solution again has the from $f(x) = \sum_{i=1}^{\ell} \alpha_i K(x, x_i)$ and the label output is simply $\text{sign}(f(x))$.

The SVM an also be developed using a geometric approach. A hyperplane is defined via its normal vector *w*. Given a hyperplane w and a point x, define x_0 to be the closest point to x on the hyperplane -- the closest point to x that satisfies $w \cdot x_0 = 0$ (see figure x). We then have the following two equations:

$w \cdot x = k$ for some k
$w \cdot x_0 = 0$.

Subtracting these two equations, we obtain,

$$w \cdot (x - x_0) = k.$$

Dividing by the norm of w, we have,

$$\frac{w}{\|w\|} \cdot (x - x_0) = \frac{k}{\|w\|}$$

Noting that $\frac{w}{\|w\|}$ is a unit vector, and the vector $x - x_0$ is parallel to w, we conclude that,

$$\|x - x_0\| = \frac{|k|}{\|w\|}.$$

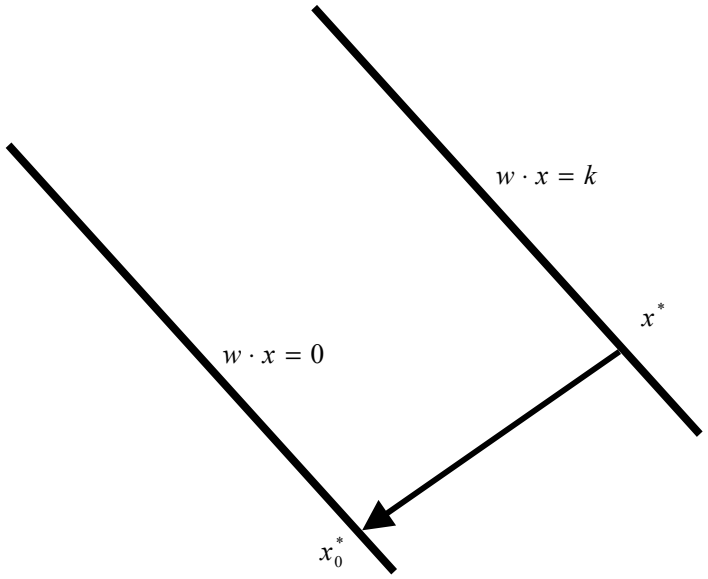


Figure 12. The distance between points x and x_0.

Our goal is to maximize the distance between the hyperplane and the closest point, with the constraint that the points from the two classes lie on separate sides of the hyperplane. We could try to solve the following optimization problem:

$$\max_{w} \min_{x_i} \frac{y_i (w \cdot x_i)}{\|w\|} \text{ subject to } y_i (w \cdot x_i) > 0 \text{ for all } x_i .$$

Note that $y_i (w \cdot x_i) = |k|$ in the above derivation. For technical reasons, the optimization problem stated above is not easy to solve. One difficulty is that if we find a solution w, then cw for any positive constant c is also a solution. In some sense, we are interested in the *direction* of the vector w, but not its length.

If we can find any solution w to the above problem, for example by scaling w, we can guarantee that $y_i (w \cdot x_i) \geq 1$ for all x_i. Therefore, we may equivalently solve the problem,

$$\max_{w} \min_{x_i} \frac{y_i (w \cdot x_i)}{\|w\|} \text{ subject to } y_i (w \cdot x_i) \geq 1 .$$

Note that the original problem has more solutions than this one, but since we are only interested in the direction of the optimal hyperplane, this would suffice. We now restrict the problem further: we are going to find a solution such that for any point *closest* to the hyperplane, the inequality constraint will be satisfied as an *equality*. Keeping this in mind, we can see that,

$$\min_{x_i} \frac{y_i (w \cdot x_i)}{\|w\|} = 1 .$$

So the problem becomes,

$\max \frac{1}{\|w\|}$ subject to $y_i(w \cdot x_i) \geq 1$.

For computational reasons, we transform this to the equivalent problem,

$\min \frac{1}{2}\|w\|^2$ subject to $y_i(w \cdot x_i) \geq 1$.

Note that so far, we have considered only hyperplanes that pass through the origin. In many applications, this restriction is unnecessary, and the standard separable SVM problem is written as,

$\min \frac{1}{2}\|w\|^2$ subject to $y_i(w \cdot x_i + b) \geq 1$,

where b is a free *threshold* parameter that translates the optimal hyperplane away from the origin.

In practice, datasets are often not linearly separable. To deal with this situation, we add *slack variables* that allow us to violate our original distance constraints. The problem becomes:

$\min \frac{1}{2}\|w\|^2 + C\sum_i \chi_i$ subject to $y_i(w \cdot x_i + b) \geq 1 - \chi_i$, $\chi_i \geq 0$ for all i.

This new program trades off the two goals of finding a hyperplane with large margin (minimizing $\|w\|$), and finding a hyperplane that separates the data well (minimizing the χ_i). The parameter C controls this tradeoff. It is no longer simple to interpret the final solution of the SVM problem geometrically; however, this formulation often works very well in practice. Even if the data at hand *can* be separated completely, it may be preferable to use a hyperplane that makes some errors, if this results in a much smaller ||w||.

There also exist SVMs which can find a *nonlinear* separating surface. The basic idea is to nonlinearly map your data to a *feature space* of high or possibly infinite dimension,

$x \rightarrow \phi(x)$.

A linear separating hyperplane in the feature space corresponds to a nonlinear surface in the original space. We can write the program as follows,

$\min \frac{1}{2}\|w\|^2 + C\sum_i \chi_i$ subject to $y_i(w \cdot \phi(x_i) + b) \geq 1 - \chi_i$, $\chi_i \geq 0$ for all i.

Note that as phrased above, *w* is a hyperplane in the feature space. In practice, we solve the Wolfe dual of the optimization problems presented. A nice consequence of this is that we avoid having to work with w and $\phi(x)$, the hyperplane and the feature vectors, explicitly. Instead, we only need a function $K(x, y)$ that acts as a dot product in feature space,

$K(x_i, x_j) = \phi(x_i) \cdot \phi(x_j)$.

For example, if we use as our kernel function a Gaussian kernel,

$$K(x_i, x_j) = \exp(-\|x_i - x_j\|^2).$$

This corresponds to mapping our original vectors x_i to a certain countably infinite dimensional feature space when x is in a bounded domain and an uncountably infinite dimensional feature space when the domain is not bounded.

References

1. Alizadeh, A.A., Eisen, M.B., Davis, R.E., Ma, C., Lossos, I.S., Rosenwald, A., Boldrick, J.C., Sabet, H., Tran, T., Yu, X., et al. (2000) *Nature* **403**, 503-511.
2. Allwein (2000). Reducing multiclass to binary: A unifying approach for margin classifiers. *Proc ICML 2000*.
3. Aronszajn (1950). Theory of Reproducing Kernels. *Trans Am Math Soc* **686**, 337-404.
4. Arrow (1951). *Social Choice and Individual Values*. John Wiley & Sons, New York, NY
5. Ben-Dor, A., Bruhn, L., Friedman, N., Nachman, I., Schummer, M., Yakhini, Z. (2000). Tissue classification with gene expression profiles*, J Comp Biol*, **7**, 559-584.
6. Bienz, M. & Clevers, H. (2000) *Cell* **103**, 311-320.
7. Bittner, M., Meltzer, P., Chen, Y., Jiang, Y., Seftor, E., Hendrix, M., Radmacher, M., Simon, R., Yakhini, Z., Ben-Dor, A., et al. (2000) *Nature* **406**, 536-540.
8. Bose, R.C. & Ray-Chaudhuri, D.K. (1960). On a class of error correcting binary group codes. *Information Control* **3**, 68-79.
9. Brown, M.P., Grundy, W.N., Lin, D., Christianini, N., Sugnet, C.W., Furey, T.S., Ares, M., & Haussler, D. (2000) *Proc Natl Acad Sci USA* **97**, 262-267.
10. Califano, A. (1999). Analysis of gene expression microarrays for phenotype classification. *Proceedings of the Eighth International Conference on Intelligent Systems for Molecular Biology, San Diego, California, August 19-23*, 75-85.
11. Chapelle, O., Vapnik, V., Bousquet, O., Mukherjee, S. (2002) *Machine Learning (Special Issue)*, (in press).
12. Connolly, J.L., Schnitt, S.J., Wang, H.H., Dvorak, A.M., & Dvorak, H.F. (1997) in *Cancer Medicine*, eds. Holland, J.F., Frei, E., Bast, R.C., Kufe, D.W., Morton, D.L., & Weichselbaum, R.R. Williams & Wilkins, Baltimore, MD, pp.533-555.
13. Dasarathy, V.B. (1991) in *NN Pattern Classification Techniques* (IEEE Computer Society Press, Los Alamitos, CA).
14. Dhanasekaran, S.M., Barrette T.R., Ghosh D., Shah R., Varambally S., Kurachi K., Pienta K.J., Rubin M.A., Chinnaiyan A.M. (2001) *Nature* **412,** 822-826.
15. Dietterich & Bakiri (1991). Error correcting output codes: A general method for improving multiclass inductive programs. *Proc AAAI*, 572-577.
16. Efron, B. & Tibshirani, R. (1993). *Introduction to the Bootstrap*. Chapman and Hall, New York, NY.
17. Eisen, M.B., Spellman, P.T., Brown, P.O., & Botstein, D. (1998) *Proc Natl Acad Sci USA* **95**,14863-14868.
18. Evgeniou, T., Pontil, M., Poggio, T. (2000) *Advances in Computational Mathematics*, **13**, 1-50.
19. Furey, T., Christianini, N., Duffy. N., Bednarski, D.W., Schummer, M., & Haussler, D. (2000) *Bioinformatics* **16**, 906-914.
20. Golub, T.R., Slonim, D.K., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J.P., Coller H., Loh, M.L., Downing, J.R., Caligiuri, M.A., Bloomfield, C.D., & Lander, E.S. (1999) *Science* **286**, 531-537.
21. Guruswami, V. & Sahai, A. (1999). Multi-class learning, boosting and error-correcting codes. *Proceedings of the Twelfth Annual Conference on Computational Learning Theory*, ACM Press 145-155.

22. Guyon, I., Weston, J., Barnhill, S., & Vapnik, V. (2002) *Machine Learning (Special Issue)*, (in press). http://homepages.nyu.edu/~jaw281/genesel. pdf
23. Hainsworth, J.D. & Greco, F.A. (1993) *N Engl J Med* **329**, 257-263.
24. Hair, J.F., Anderson, R.E., Tatham, R.L., & Black, W.C. (1998) in *Multivariate Data Analysis* (Prentice Hall, New Jersey).
25. Hastie, T. and Tibshirani, R. (1998). Classification by pairwise coupling. *Advances on Neural Processing Systems* 10, MIT Press.
26. Hastie, T., Tibshirani, R., Eisen, M.B., Alizadeh, A., Levy, R., Staudt, L., Chan, W.C., Botstein, D., Brown, P. (2000) *Genome Biol* **1**: RESEARCH003
27. Hedenfalk, I., Duggan, D., Chen, Y., Radmacher, M., Simon, R. Meltzer, P, Gusterson, B., Esteller, M., Kallioniemi, O.P., Wilfond, B., et al. (2001) *N Engl J Med* **344**,539-548.
28. Huberty, C.J. (1994) *Applied Discriminant Analysis*. John Wiley & Sons, New York, NY.
29. Khan, J., Wei, J.S., Ringner, M., Saal, L.H., Ladanyi, M., Westermann, F., Berthold, F., Schwab, M., Antonescu, C.R., Peterson, C., et al. (2001) *Nat Med* **7**, 673-679.
30. Lickert, H., Domon, C., Huls, G., Wehrle, C. Duluc, I., Clevers, H., Meyer, B.I., Freund, J.N., & Kemler, R. (2000) *Development* **127**, 3805-3813.
31. Michaud, P. (1987). Condorcet – A man of the avant-garde. *Applied Stochastic Models and Data Analysis*, **3**, 173-189.
32. Minsky, M. and Papert, S. (1972). *Perceptrons: An introduction to computational geometry*. MIT Press.
33. Mukherjee, S. (1999) Support vector machine classification of microarray data. CBCL Paper #182 Artificial Intelligence Lab. Memo #1676, MIT, http://www.ai.mit.edu/projects/cbcl/publications/ps/cancer.ps
34. Perou, C.M., Sorlie, T., Eisen, M.B., van de Rijn, M., Jeffrey, S.S., Rees, C.A., Pollack, J.R., Ross, D.T., Johnsen, H., Akslen, L.A., et al. (2000) *Nature* **406**,747-752.
35. Ramaswamy, S. & Golub, T.R. (2001) DNA microarrays in clinical oncology. *J Clin Onc* (in press)
36. Ramaswamy, S., Osteen, R.T., & Shulman, L.N. (2001) in *Clinical Oncology*, eds. Lenhard, R.E., Osteen, R.T., & Gansler, T. (American Cancer Society, Atlanta, GA), pp.711-719.
37. Rifkin, R. Mukherjee, S., Tamayo, P., Ramaswamy, S., Yeang, C.H., Angelo, M., Reich, M., Poggio, T., Golub, T.R., Lander, E.S., & Mesirov, J.P. (2002) An analytical method for multi-class molecular cancer classification. American Mathematical Society Conference Proceedings (in press)
38. Rosenblatt 1962. *Principles of Neurodynamics*. Spartan Books, New York, NY.
39. Scherf, U., Ross, D.T., Waltham, M., Smith, L.H., Lee, J.K., Tanabe, L., Kohn, K.W., Reinhold, W.C., Myers, T.G., Andrews, D.T., et al. (2000) *Nat Genet* **24**, 236-244.
40. Slonim, D.K. (2000) in *Proceedings of the Fourth Annual International Conference on Computational Molecular Biology (RECOMB)* (Universal Academy Press, Tokyo, Japan), pp. 263-272.
41. Staunton, J.E., Slonim, D.K., Coller, H.A., Tamayo, P., Angelo, M.J., Park, J., Scherf, U., Lee, J.K., Reinhold, W.O., Weinstein, J.N., et al. (2001) *Proc Natl Acad Sci U S A* **98**, 10787-10792.
42. Taipale, J. & Beachy, P.A. (2001) *Nature* **411**,349-354.
43. Tamayo, P., Slonim, D., Mesirov, J., Zhu, Q., Kitareewan, S., Dmitrovsky, E., Lander, E.S., & Golub, T.R. (1999) *Proc Natl Acad Sci USA* **96**, 2907-2912.
44. Tikhonov and Arsenin (1977). *Solutions of ill-posed problems*, W.H. Winston, Washington D.C.
45. Tomaszewski, J.E. & LiVolsi, V.A. (1999) *Cancer* **86**, 2198-2200.
46. Vapnik, V.N. (1998) in *Statistical Learning Theory* (John Wiley & Sons, New York).
47. Yeang, C.H., Ramaswamy, S., Tamayo, P., Mukherjee, S. Rifkin, R., Angelo, M., Reich, M., Mesirov, J.P., Lander, E.S., Golub, T.R. (2001) Molecular classification of multiple tumor types, *Bioinformatics* 17(s1), s316-s322.
48. Ziemer, L.T., Pennica, D., & Levine, A.J. (2001) *Mol Cell Biol* **21**, 562-574.