A learned comparative expression measure for Affymetrix GeneChip DNA microarrays

Will Sheffler
Dept. of Genome Sciences
University of Washington
wsheffle@u.washington.edu

Eli Upfal
Dept. of Computer Science
Brown University
eli@cs.brown.edu

John Sedivy
Dept. of Molecular and Cell
Biology and Biochemistry
Brown University
john_sedivy@brown.edu

William Stafford Noble
Dept. of Genome Sciences
Dept. of Computer Science and Engineering
University of Washington
noble@gs.washington.edu

Abstract

Perhaps the most common question that a microarray study can ask is, "Between two given biological conditions, which genes exhibit changed expression levels?" Existing methods for answering this question either generate a comparative measure based upon a static model, or take an indirect approach, first estimating absolute expression levels and then comparing the estimated levels to one another.

We present a method for detecting changes in gene expression between two samples based on data from Affymetrix GeneChips. Using a library of over 200,000 known cases of differential expression, we create a learned comparative expression measure (LCEM) based on classification of probe-level data patterns as changed or unchanged. LCEM uses perfect match probe data only; mismatch probe values did not prove to be useful in this context. LCEM is particularly powerful in the case of small microarry studies, in which a regression-based method such as RMA cannot generalize, and in detecting small expression changes. At the levels of selectivity that are typical in microarray analysis, the LCEM shows a lower false discovery rate than either MAS5 or RMA trained from a single chip. When many chips are available to RMA, LCEM performs better on two out of the three data sets, and nearly as well on the third. Performance of the MAS5 log ratio statistic was notably bad on all datasets.

Key words: microarrays, gene expression, support vector machine

1 Introduction

The most common use for DNA microarrays is to find genes with changed expression between two samples. For example, one might look for genes that are expressed more highly in cancerous versus healthy lung tissue, or for genes with different expression in neurons grown in a petri dish versus a 3D matrix.

Typically, the comparison of expression levels is carried out in two steps: the absolute expression level is estimated, and these levels are then compared using differences or ratios. Many methods exist for estimating expression levels [5, 14, 16]. Using such a method, differentially expressed genes can then be identified by comparing the estimated expression levels in two samples. This is, however, an indirect approach. By analogy, to test if one item is heavier than another we could weigh each individually, but might prefer to place them on opposite sides of a balance and see which way it tips. Directly comparative methods, such as the MAS5 log ratio and change p-value, are usually based on static models such as rank test, t-tests, and ANOVA [3, 15, 18, 4]. In some cases these comparative methods are better at discriminating genes with changed expression from genes with unchanged expression. Dynamic expression models are generally better for quantitating the level of change.

The learned comparative expression measure (LCEM) is, to our knowledge, the first application of machine learning to probe-level analysis of high-density oligonucleotide arrays. The LCEM algorithm builds an implicit non-parametric model based upon a training set of perfect-match probe set pairs. Each probe set pair is derived from the same

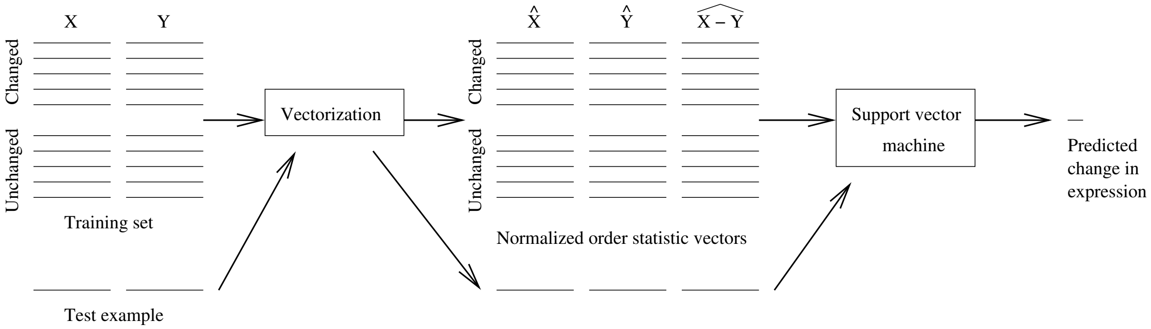


Figure 1. Schematic of the LCEM algorithm: The training set consists of vectors of pefect-match measurements (represented as horizontal lines) from two chips (X and Y) per gene. The vectorization procedure converts the raw expression measurements into 48 dimensional vectors of normalized order statistics, as described in the text. An SVM then learns to discriminate between the differentially expressed and non-differentially expressed genes.

gene measured under two different conditions. These training set pairs must be labeled *a priori* as exhibiting changed or unchanged expression. Using this training data, plus a test group of probe set pairs for which the expression status is unknown, the algorithm proceeds in two stages, summarized in Figure 1. First, each probe set pair in the training set is converted to a vector representation using sorted perfect match probe values. Second, a support vector machine (SVM) classifier [22, 7] learns to discriminate between the vectors corresponding to changed and unchanged gene expression. The resulting SVM can be used to predict the expression change associated with probe set pairs in the test set.

A significant contribution of our work is a method for extracting examples of small but certain gene expression changes from mixtures of samples (described in Section 3.1). We apply this method to the Gene Logic dilution data set (`http://www.genelogic.com/media/studies/dilution.cfm`). Previous comparative studies of differential expression measures [6, 15, 12] rely only on data derived from spike-in style data sets, which have a number of limitations. Spike-In studies typically contain examples of only two fold and larger expression changes and use only 10 to 42 spiked-in transcripts, limiting the diversity of data patterns represented. The examples we draw from the dilution data set are of much smaller fold changes, many in the range between 1.1 and 2 fold, and are drawn from 2,942 different human genes. In contrast, most fold changes in the spike-in data sets are 10-100 times higher. These borderline examples provide valuable training data and account for the high accuracy of the LCEM. With over 220,000 positive and 1.2 million negative examples of differential expression, this data provides a realistic and rigorous test set for evaluating the performance of differential expression measures.

We perform comparative experiments with three diverse data sets: the GeneLogic dilution study mentioned above, and spike-in studies performed by GeneLogic and Affymetrix. We compare the LCEM to two Affymetrix static models, the log ratio (LR) and the change p-value (Pval), as well as to RMA trained from a single chip and trained from all available data. Our results indicate that, at levels of selectivity that are typical in microarray analysis, the LCEM shows a lower false discovery rate than either MAS5 or RMA trained from a single chip. Even when many chips are made available to RMA and only two chips to LCEM, LCEM performs better than RMA on two out of the three data sets and nearly as well on the third. LCEM performs particularly well on the examples drawn from the dilution study, indicating a strong ability to detect small expression changes. LCEM's good performance across these three data sets shows that the method generalizes across different chip architectures with different numbers of probes per gene. LCEM shows good quantitative as well as discriminative performance. On the dilution examples the correlation between LCEM and RMA is 0.976 on genes for which there is a change in expression, showing that LCEM quantitates changes about as well as RMA. This is not because LCEM and RMA always agree: the correlation is -0.028 for cases with no change in gene expression.

Below, we describe the LCEM method in detail, followed by a description of our experimental design, results and discussion.

2 Algorithm

The LCEM algorithm, depicted in Figure 1, consists of two steps: a vectorization step, in which labeled probe set pairs are converted to an order-statistic based vector representation, and support vector machine training and classification step. This section describes these two steps in detail.

The goal of the vectorization step is to produce a data representation that generalizes across different probe sets and chip architectures. Before vectorization, the data set is background corrected and normalized according to the method of [13] using the R [11] package Bioconductor [9]. Each data vector consists of three parts. Say we are comparing chip A to chip B. Data points 1 thru 16 correspond to perfect-match probe values on chip A, data points 17 thru 32 correspond to perfect-match probe values on chip B, and data points 33 thru 48 are ratios between corresponding perfect-match probe values with values from chip A divided by values from chip B. All data point are log-transformed to stabilize variance, and data in each of the three parts is sorted in increasing order. In cases where more or fewer than 16 perfect-match probe values are available, the sorted probe values are linearly interpolated after sorting to produce 16 datapoints. We chose to include both absolute and ratio data so that the learning machine would be able to distinguish between cases with lower and higher overall expression and treat them correspondingly types.

The second step of LCEM involves training an SVM to discriminate between "changed" and "unchanged" genes. We use the SVM implemented in the R package e1071 [8], with a radial basis kernel function $K(X,Y) = exp(\gamma||X - Y||/2)$, $\gamma = 0.001$ and a soft margin ($C = 10$). In order to assign an expression value to each test set example, each probe set pair in the test set is first vectorized, as described above. The SVM then assigns to the vector a discriminant value that is proportional to the vector's distance from the separating hyperplane.

As constructed, the LCEM only measures upward changes. In order to measure downward changes, we reverse the data vectors and run LCEM again. A combined comparative expression measure can then be obtained by taking the max of the forward and reverse LCEM and multiplying by the sign of the difference. Although it is theoretically possible a data vector could be classified as both changed-up and changed-down using this scheme, Figure 2 shows that in practice no data vectors generated such contradictions. It is possible that both the forward and reversed LCEM could be slightly negative, indicating no change in either direction.

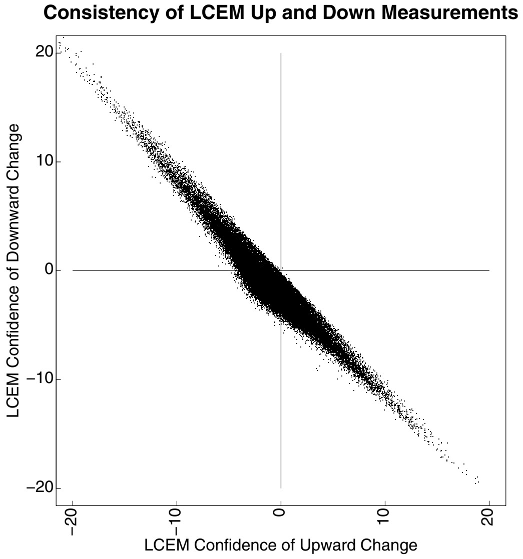


Figure 2. Consistency of LCEM: This figure shows forward and reversed LCEM measures which correspond to measurements of upward and downward changes. The upward and downward measurements are consistent with each other, as

3 Methods

LCEM is a learned measure of comparative expression. Because a learned method can be only as good as the data on which it is trained, we required a large set of high quality example microarray data. Data for both known cases of differential expression, and also known cases of unchanged expression were needed. Furthermore, for a discriminatory learning machine such as a SVM to work well, examples close to the boundary between changed and unchanged must be found.

Section 3.1 describes in detail a method to generate such close-call examples from an existing microarray dataset. Section 3.2 describes our method of comparison to existing micorarray expression measures.

3.1 Data sets

We used a subset of the GeneLogic dilution data set (`www.genelogic.com`) to train the LCEM, and we tested the generalization of the method on the rest of the dilution data set, as well as on two independent spike-in data sets (Table 1). The GeneLogic dilution data set provides a rich source of small but reliable fold changes which are not available in the spike-in studies (see online supple-

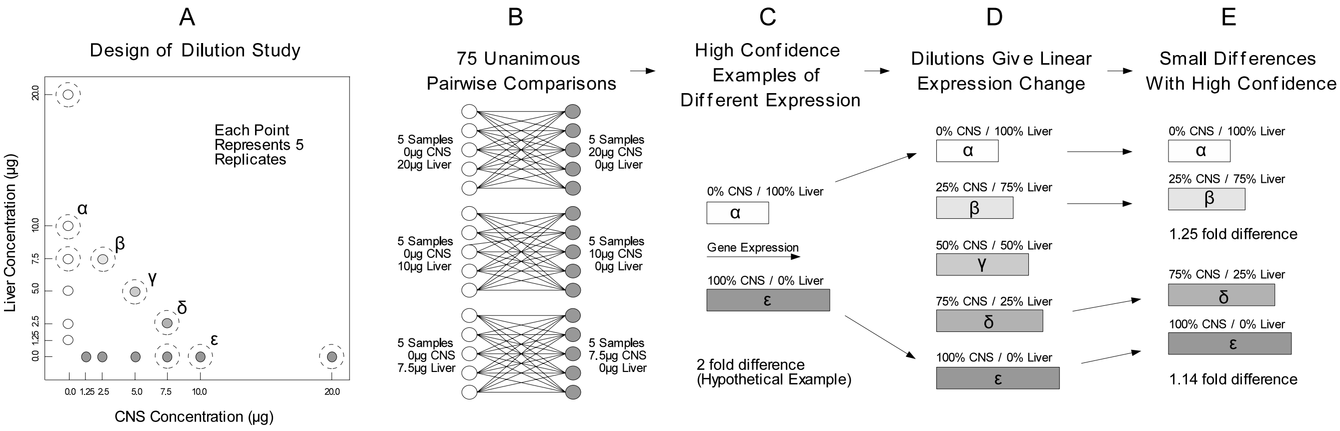

A
Design of Dilution Study
Each Point Represents 5 Replicates
Liver Concentration (µg)
CNS Concentration (µg)
0.0
1.25
2.5
5.0
7.5
10.0
20.0
α
β
γ
δ
ε
B
75 Unanimous Pairwise Comparisons
5 Samples 0µg CNS 20µg Liver
5 Samples 20µg CNS 0µg Liver
5 Samples 0µg CNS 10µg Liver
5 Samples 10µg CNS 0µg Liver
5 Samples 0µg CNS 7.5µg Liver
5 Samples 7.5µg CNS 0µg Liver
C
High Confidence Examples of Different Expression
0% CNS / 100% Liver
α
Gene Expression
100% CNS / 0% Liver
ε
2 fold difference (Hypothetical Example)
D
Dilutions Give Linear Expression Change
0% CNS / 100% Liver
α
25% CNS / 75% Liver
β
50% CNS / 50% Liver
γ
75% CNS / 25% Liver
δ
100% CNS / 0% Liver
ε
E
Small Differences With High Confidence
0% CNS / 100% Liver
α
25% CNS / 75% Liver
β
1.25 fold difference
75% CNS / 25% Liver
δ
100% CNS / 0% Liver
ε
1.14 fold difference

Figure 3. Creation of the dilution data set: (A) This plot summarizes the design of the Gene Logic dilution study. Each point represents five technical replicates with the given liver and CNS sample concentrations. Circled points are used in our study. (B) Construction of training examples starts with 75 pairwise comparisons of 100% liver to 100% CNS samples. These comparisons use the six circled samples along the x and y axes. Genes for which all 75 comparisons agree are taken as changed. (C-E) Relatively large expression differences, detected in (B), between the pure samples α and ϵ imply correspondingly smaller differences when mixed samples β, γ, and δ are considered. A hypothetical example of a gene with a two fold difference between samples α and ϵ, shown in (C), must imply the much smaller differences shown in (E) because of the relationship between the dilution mixtures (D). Thus a two-fold change, which is potentially easy to detect, can be used to find examples of 1.25 and 1.14 fold changes that would have been impossible to detect directly.

Table 1. Characteristics of the three data sets: The table lists the total number of positive and negative examples in each data set.

Data set	Use	Positive	Negatives
GeneLogic dilution	train	5,000	5,000
GeneLogic dilution	test	215,650	1,257,500
GeneLogic spike-in	test	70	832,590
Affymetrix latin square	test	4104	2,176,400

ment). Thus, the dilution data set is of paramount importance to our work.

The dilution data set was created with two distinct biological samples mixed at various dilution levels and hybridized in five technical replicates to a total of 75 HG_U95Av2 GeneChips. A graphical summary of the fifteen dilutions is shown in Figure 3(A), with the nine samples used in our study circled. For the purposes of our study, the details of the liver and CNS samples are unimportant. We care only that they are significantly different biologically so that there are many differences in gene expression between them. Using these two very different samples, we find examples of small expression changes by first selecting genes which have changed expression between the liver and CNS samples. Second, we draw examples of the genes' probe patterns from mixtures of the two samples. Expression differences in the mixtures go in the same direction as the differences between the pure samples but must have reduced magnitude, providing more subtle examples.

In the first step toward generating the dilution examples, genes with different expression levels are identified via 75 pairwise comparisons between pure liver and CNS samples. The 75 comparisons come from three sets of five versus five: one set at 20 μg total sample, one set at 10 μg total sample, and one at 7.5 μg total sample, as illustrated in Figure 3(B). The Affymetrix Microarray Suite 5.0 change p-value statistic was used to perform the comparisons. Genes for which all 75 comparisons showed a change Pval above 0.9 (decreased expression) or below 0.1 (increased expression) were selected as differentially expressed. Though 0.1 and 0.9 are not very high confidence levels for individual comparisons, our requirement for 75-way unanimous agreement makes selecting an unchanged gene this way very unlikely. Interpreting p-values literally and assuming samples are independent, the chance of selecting an unchanged gene is 1 in 10^{75}. Of the 12,650 total genes on the HG_U95Av2 chip, 2,942 were selected as changed in this way.

Once a set of genes is determined to have changed expression between the liver and CNS samples with very high confidence, we can infer that there must be smaller expression differences between mixtures of these samples. Let samples α, β, γ, δ, and ϵ have CNS/liver mixtures of 0%/100%, 25%/75%, 50%/50%, 75%/25%, 100%/0% respectively, as shown in Figure 3(A). If we know that there exists a difference in expression for a particular gene between samples α and ϵ, then we can be equally sure that samples α and β, δ and ϵ, or any other combination also have different expression for that gene. However, the expression differences between mixtures will be lower than corresponding expression differences between pure samples. Figure 3(C-E) illustrates this principle, showing how a gene which has a 2 fold expression difference between α and ϵ would have a 1.25 fold difference between α and β and a 1.14 fold difference between δ and ϵ. Using this technique, we can use a set of large, high confidence expression changes to create a library of much smaller expression changes of which we can be equally confident.

This data set of subtle expression changes, along with no-change examples, is used to train the LCEM. Examples of unchanged expression patterns were obtained within the five replicates of each dilution level. Each replicate measures exactly the same sample, so data for a pair of replicates must represent unchanged expression. A total of 5,000 of these unchanged examples were chosen uniformly at random. These, together with the 5,000 randomly chosen changed examples, form the training set for LCEM. The remaining 215,650 changed and 1,257,500 unchanged examples were used for testing.

In addition to testing on the dilution examples, we tested the quality of the LCEM on two independent spike-in data sets, the Affymetrix HG_U133A "latin square" data set (`www.affymetrix.com/support/technical/sample_data/datasets.affx`), and the GeneLogic spike-in data set (`www.genelogic.com/media/studies`). These data sets were generated by adding, or "spiking in," a set of genetic transcripts at various known concentrations to a uniform background containing some typical sample of genomic transcripts. The background is added in the same concentration to all chips, while the concentrations of the spike-in transcripts are varied in a known way. The Affymetrix spike-in data set contains 42 HG_U133A chips and has 42 spike-in transcripts. The GeneLogic spike-in data set contains 94 HG_U95A chips and 12 spike-in transcripts. Of the two spike-in data sets, we draw examples mainly from the newer and more comprehensive Affymetrix data set; only 12 of the Gene Logic spike-in chips are used. Extracting changed and unchanged examples from the spike-in data sets is straightforward: unchanged examples are taken from the background, and changed examples are taken from the spike-in genes. We can be certain these examples are truthful ones, provided that the samples were prepared properly.

3.2 Comparison methods

We compare the LCEM to two state-of-the-art microarray expression analysis methods: the regression based Robust Multi-chip Average (RMA) [12] and Affymetrix Microarray Suite 5.0, (MAS5) [2]. Most analysis of Affymetrix chips is done with one of these two methods. A third popular method, dChip [16, 17] was not considered in our study. Our model problem is small, two to six chip microarray studies, and dChip performs best on large datasets where probe effects can be effectively modeled.

The MAS 5.0 software produces two comparative expression statistics: a change p-value (Pval) based on nonparametric signed rank tests, and a signal log ratio (LR), which is a robust average ratio incorporating both perfect-match values and differences between perfect-match and mismatch values. The Pval and LR computations are convoluted and some steps are unpublished [2, 3]. Briefly, the Pval is based on multiple Wilcoxon signed rank tests between probe sets using several types of normalization and mismatch correction. The Pval is a summary of the resulting set of rank test p-values and is reported as a multiple of 10^{-6}, which can limit resolution at high levels of selectivity. The LR is a one-step tukey biweight mean log of mismatch corrected probe values. The mismatch correction is done by subtracting either mismatch probe values or an average of mismatch probe values if the mismatch is higher than the perfect match. The LR is reported as a multiple of 0.1. In our experiments, the Pval and LR statistics were computed with MAS 5.0 using default parameter settings [1] with normalization on all probe pairs. Normalization settings have no bearing on the Pval, for which a different set of normalization routines are used. Though Affymetrix is phasing out MAS5 in favor of their newer Gene Chip Operating Software (GCOS), the statistical algorithms used to generate expression measurements remain the same in GCOS [3].

RMA expression measurements were computed using the Bioconductor library [9] for the R language [11]. RMA performs a robust regression across background corrected and normalized [5] perfect match data for all available chips. For probe i of gene k on chip j, the following model is fit: $log_2(PM_{ij}^{(k)}) = \alpha_i^{(k)} + \beta_j^{(k)} + \epsilon_{ij}^{(k)}$, where $\alpha_i^{(k)}$ is a probe effect, $\beta_j^{(k)}$ is the log_2 expression value for gene k, and $\epsilon_{ij}^{(k)}$ is the minimized error term. Mismatch probe values are ignored. This model computes absolute expression measurements which we then use to compute differential expression by taking differences between RMA statistics on pairs of chips. The authors of RMA present many comparative tests done this way [6] RMA performs best when run on many chips, while LCEM and MAS5 are based on pairs of chips. We thus test RMA in two different ways in order to provide a fair comparison. First, we run RMA on all chips available in the data set in aggregate: 42, 25, and 12 chips for the Affymetrix spike-in, dilution and GeneLogic spike-in data sets, respectively. Second, we normalize and background correct all of the data and then run the RMA probe summarization on individual chips. Because the datasets we consider contain many more chips than a typical microarray experiment, a fair assessment of RMA's performance lies somewhere in between its performance on individual chips and its performance on all chips in a data set.

The performance of a given method is evaluated using a receiver operating characteristic (ROC) curve [10]. For each of our data sets, the true expression status ("changed" or "unchanged") is known for each probe set pair. An expression analysis method produces as output a ranking of genes, from genes that exhibit extremely large changes in expression down to genes that show no change at all. Setting a decision threshold at any location in this ranked list produces a list of true positives, false positives, true negatives and false negatives. The ROC curve is generated by varying the decision threshold along the ranked list and plotting the percent of true positives as a function of percent of false positives. The ROC score is the area under this curve. A perfect method will rank all of the positives above the negatives and receive a score of 1; a random ranking will receive a score close to 0.5.

In practice, most users of microarray technology are not interested in the performance of a given method beyond a relatively small number of false positives. Therefore, we also plot ROC$_P$ curves, which are simply ROC curves up to a fraction of false positives P. We set the threshold P equal to the fraction of false positives attained using MAS5 as the manufacturer recommends, using both a p-value threshold of 0.003 and a log ratio threshold of 1. Although 0.003 is a fairly low p-value, for the large sets of genes tested in a microarray experiment, this is not a particularly strict threshold. The fraction false positives obtained this way are 0.00172, 0.00124, and 0.000566 for the dilution, latin square, and spike-in examples respectively.

4 Results

Our results show that the LCEM provides high quality expression change measurements across three different data sets (Figures 4 and 5). In Figure 4, LCEM shows overall ROC scores similar or better than RMA and MAS5 except for RMA$_{42}$ on the latin square data set. More detail can be seen in the ROC plots in Figure 5. LCEM performs particularly well in comparison to the other methods on the dilution examples, showing uniformly more true positives than any other method at any threshold. RMA$_{42}$ and MAS Pval are marginally better overall on the latin square set, but LCEM performs better than MAS Pval at typical thresholds. LCEM is marginally best overall for the spike-in set, and is clearly best at typical thresholds. One striking result is the poor performance of the MAS5 LR, especially at the typical thresholds shown on the bottom of Figure 5.

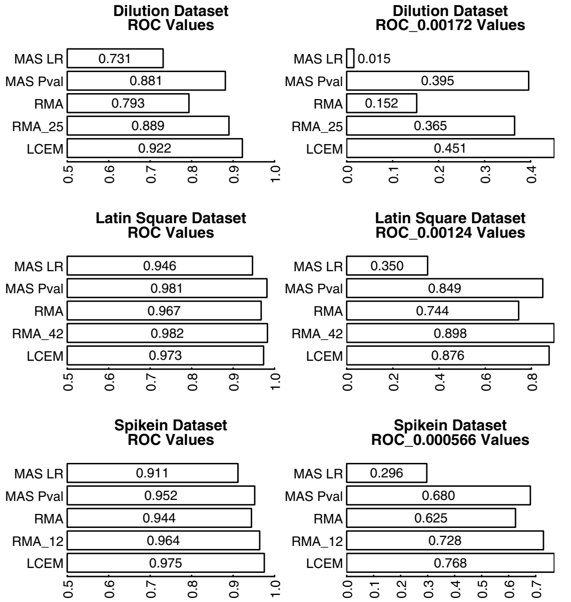
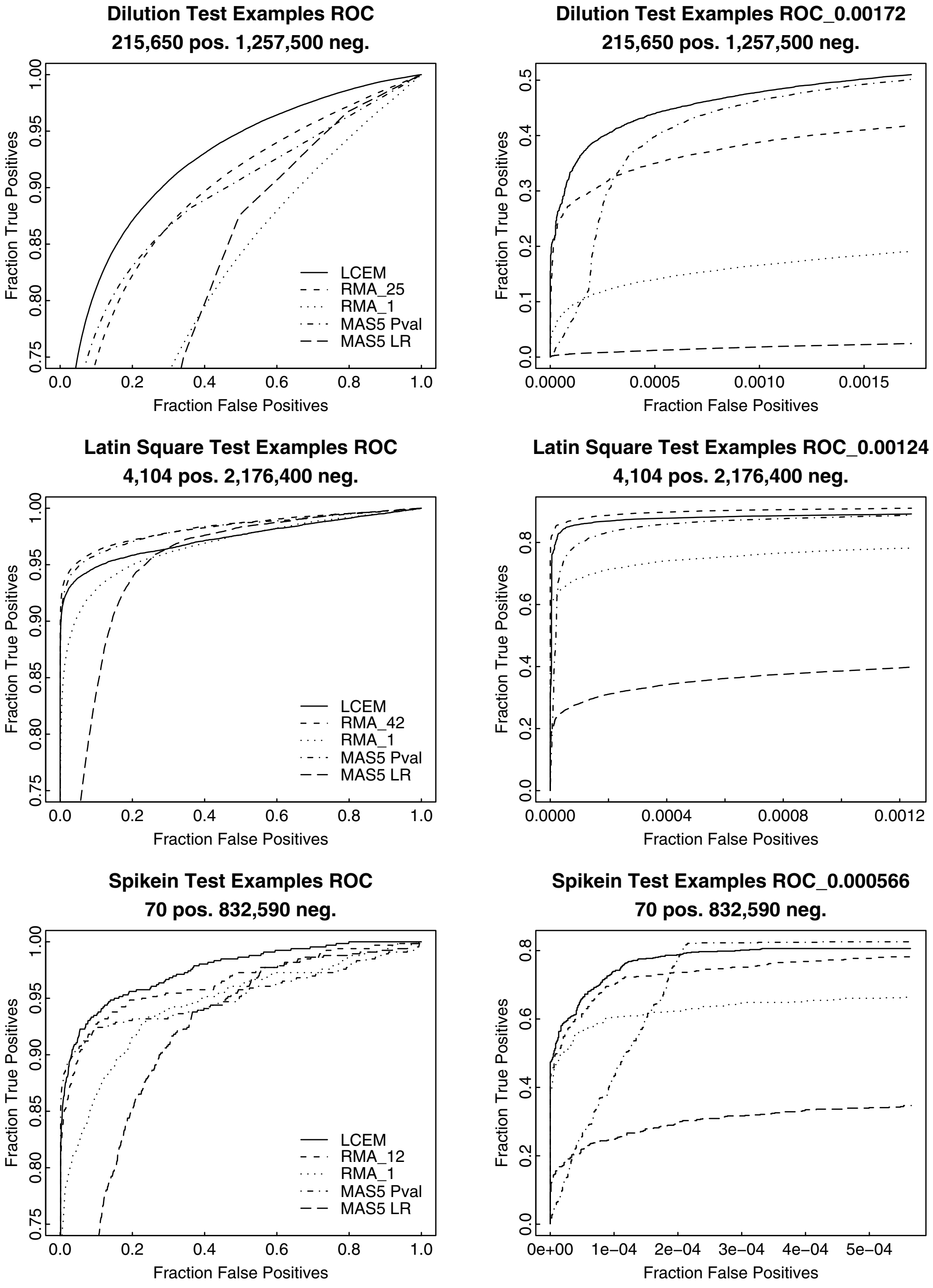


Figure 4. ROC Values for MAS, RMA and LCEM on three different data sets: Each figure plots the areas under the ROC curves in Figure 5. Values are shown for the learned comparative expression measure (LCEM), Microarray Analysis Suite 5.0 log ratio (MAS LR) and p-value statistics (MAS Pval), the Robust Multi-chip Average run on all N chip in the data set (RMA_N) and on individual chips (RMA).

For comparing expression levels, the dilution examples are much more challenging than the latin square and spike-in examples, because the fold changes in the dilution examples are small, typically 10%–20%. Accordingly, the average ROC score on the dilution set of 0.844 is significantly less than the ROC scores of 0.970 and 0.949 for the latin square and spike-in sets (Figure 4). Similarly, ROC scores over typical thresholds average 0.276 for the dilution set, 0.744 for the latin square set, and 0.620 for the spike-in set. On these challenging examples of small fold changes LCEM clearly outperforms the other expression measures, expecially MAS LR, which performs little better than a ran-



Dilution Test Examples ROC
215,650 pos. 1,257,500 neg.

Dilution Test Examples ROC_0.00172
215,650 pos. 1,257,500 neg.

Latin Square Test Examples ROC
4,104 pos. 2,176,400 neg.

Latin Square Test Examples ROC_0.00124
4,104 pos. 2,176,400 neg.

Spikein Test Examples ROC
70 pos. 832,590 neg.

Spikein Test Examples ROC_0.000566
70 pos. 832,590 neg.

Figure 5. Comparison of MAS5, RMA and LCEM on three different data sets: Each curve plots the fraction of true positives as a function of false positives for varying classification thresholds. Curves labeled "ROC" include all false positives, whereas curves marked "ROC*" include only false positives up to a threshold determined by a MAS5 p-value of 0.003 and MAS5 log ratio of 1. Thus the ROC* curves show performance at typical threshold settings. In each plot, the five series correspond to the learned comparative expression measure (LCEM), Microarray Analysis Suite 5.0 log ratio (MAS LR) and p-value statistics (MAS Pval), the Robust Multi-chip Average run on all N chips in the data set (RMA_N) and on individual chips (RMA). The data set from which each curve was generated is listed above each plot.

Table 2. True positive / false positive trade-offs for three data sets: Each row in the table lists the number of false positives which must be suffered in order to find a given number of true positives using five different comparative expression measures across three different data sets. These numbers are based on a hypothetical data set that contains 100 changed genes in a background of 10,000 unchanged genes and has expression patterns like those in the dilution, latin square and spike-in data sets, respectively. The five measures used are LCEM, the MAS5 p-value (Pval) and log ratio (LR), and RMA on all N chips (RMA_N) and on individual chips (RMA).

	Dilution data set false positives					Latin square data set false positives					Spike-in data set false positives				
TP	LCEM	Pval	LR	RMA_{25}	RMA_1	LCEM	Pval	LR	RMA_{42}	RMA_1	LCEM	Pval	LR	RMA_{12}	RMA_1
10	0	2	344	1	2	0	0	0	0	0	0	0	0	0	0
20	1	3	770	1	20	0	0	0	0	0	0	0	0	0	0
30	1	3	1123	2	123	0	1	2	0	0	0	0	0	0	0
40	3	5	1545	12	371	1	1	13	0	0	0	0	0	0	0
50	14	16	1984	59	797	1	1	64	0	0	0	0	67	0	0
60	70	85	2479	211	1478	1	1	196	0	0	0	0	306	0	0
70	259	400	3137	623	2502	1	1	428	0	1	0	0	755	0	15
80	870	1481	4183	1622	4089	1	1	812	0	15	0	0	1515	0	264
90	2809	5135	6439	4125	6580	15	16	1428	4	428	304	374	2735	514	1744

dom classifier at higher selectivities.

On the latin square examples, RMA_{42} performs better than all other methods. This is not surprising, because RMA_{42} has access to all 42 chips in the data set, giving a large advantage over the other methods. This advantage is more pronounced on the latin square examples because the sample concentrations of all unchanged example genes are uniform across all 42 chips. Genes for which concentrations are the same on all 42 chips are particularly easy for the regression-based method to determine as uniform. LCEM may have performed less well on this data set because our training set of dilution examples contains mainly examples of small and medium changes of 1.1–4 fold, whereas the smallest fold changes in the latin square set are 2 fold and the mean fold change over 200 fold. Furthermore, the chips used in the latin square study typically contain 11 probes per gene, whereas our training data contained 16–20 probes per gene.

In the latin square data set results shown in Figures 5 and Figure 4, data from a set of 56 genes were removed. These genes (listed in the online supplement) were part of the background, and thus were among the examples of unchanged expression. MAS5, RMA and LCEM all judged some examples of these genes as highly changed, to the point where overall results on the dataset were affected. Because these outliers were observed independently with two different software packages (MAS 5.0 and R/Bioconductor) it seems likely they are a feature of the dataset, possibly due to nonspecific hybridization. Of the methods considered here, performance of RMA was effected most and LCEM least (see online supplement), providing some indication that LCEM is robust to unusual data patterns.

To provide a more intuitive interpretation of these results as they could apply to practical microarray analysis, a table of true positive / false positive tradeoffs is presented in Table 2. Values in the table are scaled to represent an experiment in which there are 100 genes with changed expression between two samples and 10,000 genes with unchanged expression between the samples. The number of false positives which must be suffered in order to find 10 through 90 true positives is shown. True positive / false positive tradeoffs on the dilution examples clearly show the strength of LCEM. In every case LCEM generates fewer false positives than any other method, with the single exception of a tie with RMA_{25} for 1 false positive in 20 true positives. For high selectivity cases of 10-30 true positives, RMA_{25} is second best while MAS5 Pval is second best for 40 or more true positives. Overall, MAS LR and RMA_1 show very poor performance, in most cases generating more than 10 times as many false positives as LCEM and in some cases 100 or even 1000 times more false positives. False positive rates on the latin square examples are very low using any measure other than MAS5 LR. RMA_{42} is uniformly best overall on this data, which is unsurprising, as noted above. For 60 or fewer false positives RMA_1 performs well, generating no false positives for some cases where LCEM and MAS5 Pval generate 1 false positive. For 80 and above true positives, LCEM and MAS5 Pval are very close and are both much better than RMA_1. On the spike-in examples, all methods except LR generate very few false positives. LCEM is best overall by at least 23%. These true positive / false positive tradeoffs show that RMA performs well at higher levels of selectivity, MAS Pval performs well at lower levels of selectivity, LCEM performs well at all levels of selectivity, and MAS LR performs universally badly.

One striking feature of the MAS5 Pval results is the linearly sloped region at the left of the ROC curves for the dilution and spike-in examples (Figure 5). The reason for this shape is that many samples, including some false positives, are given a Pval of 0. ROCs are computed such that ties are broken randomly, yielding a roughly straight line for regions of equal measure. In the spike-in examples, about 80% of the 70 positive examples have a Pval of 0. In the dilution examples, the slope of the region with Pval 0 is lower then the following portion of the ROC, indicating that a MAS5 Pval of 0 is less likely to indicate a true change than a slightly higher Pval.

Figure 6 shows how LCEM values correlate with MAS LR and RMA. Data shown is from a real world micorarray study used in a screen of genes regulated by cMyc [20] in a rat model. The portion of this data set used consists of six Affymetrix RGU_34A chips. Three biological replicates of two samples were performed, and the data shown are from all nine possible pairwise comparisons between two groups of three chips. The striped pattern in the Figure 6(A) is due to rounding performed by MAS5. The most striking feature in this plot is the cross pattern. This pattern indicates the presence of many genes which the LR measures as highly changed but which LCEM measures as unchanged. Because the MAS LR generates 10–100 times more false positives than LCEM, most of these points are probably false positives. The MAS LR has a clear bias towards false positives at low expression levels (see online supplement). This effect is presumably due to mismatch correction, in which mismatch signals, after processing to avoid negative expression, are subtracted from perfect match signals. If gene expression is low, then perfect match and mismatch signal will be highly random with similar mean. The corrected values will thus be fairly uniformly distributed close to zero. The MAS LR is a log ratio of two such quantities, which will be highly variable even if the original signals are the same.

Figure 6(B) plots LCEM versus RMA. The plot shows that the metrics generally agree with one another, with most points occuring on a diagonal line. Qualitatively, RMA appears to emphasize small differences near zero. When the two methods strongly disagree, LCEM tends to be more conservative. The correlation between RMA and LCEM on the dilution examples is 0.976 for changed examples and -0.0284 on unchanged examples. Thus, when genes have changed expression RMA and LCEM agree very well on how changed they are, but on unchanged examples LCEM and RMA show little overall agreement or disagreement.

In order to gain insight into what information LCEM was using as compared to RMA, we studied case examples for which the two methods disagreed. Four types of data patterns were examined: changed examples which RMA predicts correctly and LCEM incorrectly, changed examples which LCEM classifies correctly and RMA incorrectly, unchanged examples which LCEM classifies correctly and RMA incorrectly, and unchanged examples which RMA predicts correctly and LCEM incoretcly. See the online

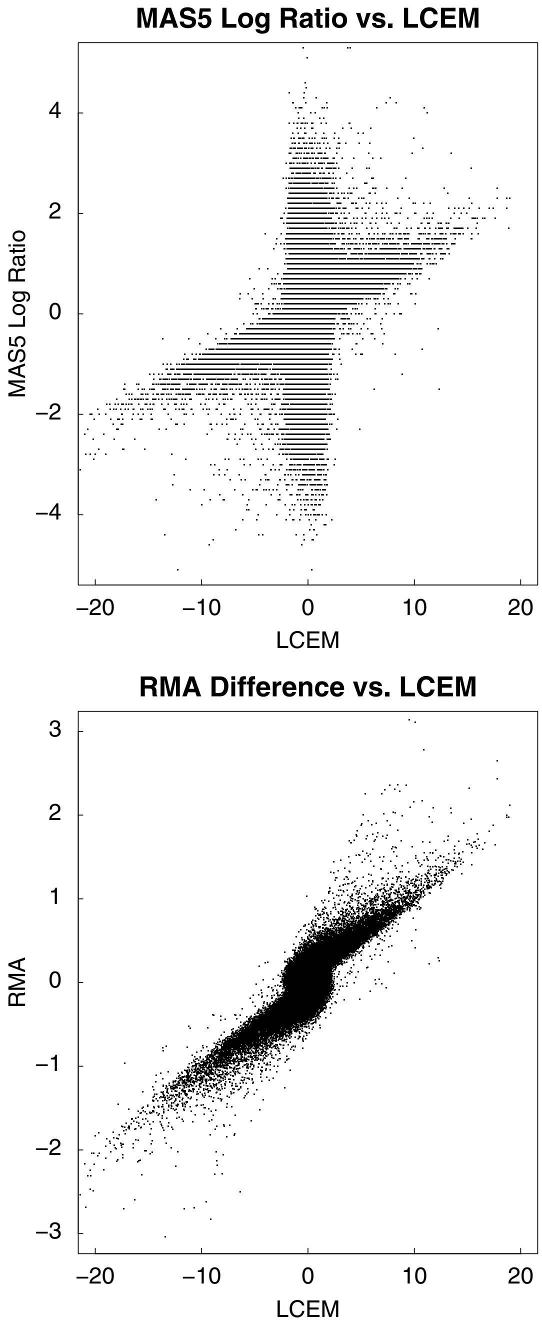


Figure 6. Scatterplot of LCEM vs. RMA and MAS5 Statistics: Each figure plots LCEM values versus the corresponding MAS5 log ratio, MAS5 p-value and RMA values. Data shown is from an independent data set [20]. Many genes for which LCEM is low show strong indications of change under the MAS5 log ratio.

supplement for examples from these four groups. We make two main observations about these data patterns. First, LCEM makes more use of information from the highest and lowest 25% of probe values than RMA. In both true and false cases where LCEM predicted changes and RMA did not, probe values in the top of each set showed significant differences, while the middle 50% of probe values were similar. Second, LCEM measurements were more conservative on genes with lower absolute expression. In many cases where RMA predicted changed expression, correctly or incorrectly, and LCEM did not, half or more probe values were very close to background. There is no bias against low expressors in LCEM, as LCEM and RMA show similar differential expression vs. average expression (see online supplement).

5 Discussion

The Learned Comparative Expression Measure is a powerful approach for evaluating gene expression changes in an Affymetrix GeneChip experiment. From a user's perspective, LCEM's primary benefits, as inferred from the experiments reported here, are four-fold. First, the method works on any Affymetrix data set without requiring re-training. Second, the method works well with data from a small number of GeneChips. Third, LCEM provides excellent ability to discriminate between genes with changed vs. unchanged expression at both higher and lower levels of selectivity. Fourth, LCEM blends discriminative ability and accurate quantitation of expression changes in a single statistic.

LCEM performs comparisons between two GeneChips. We choose to focus on two-chip comparisons for a number of reasons. First, to allow direct comparison with MAS5, which operates only on pairs of chips. Second, the majority of comparative microarray experiments involve a small number of chips due to cost. Third, two-chip comparison can easily be scaled up to any number of chips by performing all pairwise comparisons between experiment and control groups. And fourth, replicates in small comparative experiments are usually biological rather than technical. Biological replicates come from different biological samples; thus, equal expression levels between replicates cannot be assumed. In this case, the utility of a multi-chip probe level model is questionable.

Although machine learning techniques, and in particular SVMs, have been used extensively to analyze microarray data, to our knowledge LCEM is the first method to successfully apply machine learning techniques to probe-level microarray analysis. Any learning approach to this problem benefits from the large amount of available microarray data. Indeed, the success of the LCEM depended in part upon the availability of the GeneLogic dilution data set. The examples extracted from this data set were constructed from real biological samples, emphasizing small fold changes over large ones, and provide over 220,000 examples of changed gene expression. In comparison, spike-in data sets are constructed with a limited number of artificial samples, and can provide comparatively few examples of changed expression. Without the dilution examples, a machine learning approach to this problem would likely not have been successful.

An obvious extension to the LCEM would be to include in each vector a representation of the mismatch probe data. We tried several variants of this idea: order statistics of mismatch probe values, mismatch values ordered by the value of their corresponding perfect match probes, order statistics of perfect match minus mismatch values, and order statistics of perfect match minus mismatch values truncated to 0. Both log values and log ratios between chips were considered. In all cases where mismatch-derived data was presented to the learning algorithm, the resulting performance was equal to or worse than when only perfect match data was considered. This effect was observed whether mismatch derived data was presented instead of or in addition to perfect match data. We were thus unable to fruitfully make use of the mismatch probe data.

An alternative approach to the LCEM would be to train using regression, rather than a classifier. The regression approach is appealing, because one of our goals is to quantitate expression changes. However, we chose SVM classification over regression for two reasons. First, we had difficulty quantitating the degree of change in training examples. Our data extraction method allowed us to identify challenging examples of changed genes with high confidence, but estimates of fold change could not be obtained without heavy reliance on uncertain expression measures. Second, we found that the regression approach performed badly at discriminating between changed and unchanged genes. Given the high correlation between LCEM and RMA on genes with expression changes, it seems that a regression based approach is not required to provide good quantitation of expression changes.

Finally, one could imagine using alternative classification algorithms or alternative SVM kernel functions in training the LCEM. Although, by performing some careful algorithm and model selection, it might be possible to improve the performance of the LCEM, we do not expect such an improvement to be large. In practice, the SVM produces state-of-the-art classification performance across a wide variety of problem domains [19].

This paper represents a proof-of-concept for the LCEM. Our next step is to improve its usability by mapping the SVM outputs to probabilities using a sigmoid curve fit [21]. This is a straightforward, widely used method that would yield, for each gene, a probability of changed expression, allowing the users to better set thresholds. In future work, we also plan to make a user-friendly version of the software available for download.

References

[1] Affymetrix. Fine tuning your data analysis. 2001.

[2] Affymetrix. Microarray suite user's guide, version 5.0. 2001.

[3] Affymetrix. Statistical algorithms reference guide. 2001.

[4] L. Barrera, C. Benner, Y. Tao, E. Winzeler, and Y. Zhou. Leveraging two-way probe-level block design for identifying differential gene expression with high-density oligonucleotide arrays. *BMC Bioinformatics*, 5(42):unknown, 2004.

[5] B. M. Bolstad, R. A. Irizarry, M. Astrand, and T. P. Speed. A comparison of normalization methods for high density oligonucleotide array data based on bias and variance. *Bioinformatics*, 19(2):185–193, 2003.

[6] L. Cope, R. Irizarry, H. Jaffee, Z. Wu, and T. Speed. A benchmark for affymetrix genechip expression measures. *Bioinformatics*, 20(3):323–331, 2003.

[7] N. Cristianini and J. Shawe-Taylor. *An Introduction to Support Vector Machines*. Cambridge UP, Cambridge, UK, 2000.

[8] E. Dimitriadou, K. Hornik, F. Leisch, D. Meyer, and A. Weingessel. e1071: Misc functions of the dept of statistics (e1071), tu wien. Technical report, TU Wien, unknown.

[9] R. C. Gentleman, V. J. Carey, D. M. Bates, B. Bolstad, M. Dettling, S. Dudoit, B. Ellis, L. Gautier, Y. Ge, J. Gentry, K. Hornik, T. Hothorn, W. Huber, S. Iacus, R. Irizarry, F. L. C. Li, M. Maechler, A. J. Rossini, G. Sawitzki, C. Smith, G. Smyth, L. Tierney, J. Y. H. Yang, and J. Zhang. Bioconductor: Open software development for computational biology and bioinformatics. *Genome Biology*, 5:R80, 2004.

[10] J. A. Hanley and B. J. McNeil. The meaning and use of the area under a receiver operating characteristic (ROC) curve. *Radiology*, 143:29–36, 1982.

[11] R. Ihaka and R. Gentleman. R: A language for data analysis and graphics. *Journal of Computational and Graphical Statistics*, 5(3):299–314, 1996.

[12] R. A. Irizarry, B. M. Bolstad, F. Collin, L. M. Cope, B. Hobbs, and T. P. Speed. Summaries of affymetrix genechip probe level data. *Nucleic Acids Research*, 31(4):e15, 2003.

[13] R. A. Irizarry, B. Hobbs, F. Collin, Y. D. Beazer-Barclay, K. J. Antonellis, U. Scherf, and T. P. Speed. Exploration, normalization, and summaries of high density oligonucleotide array probe level data. *Biostatistics*, 4(2):249–264, 2003.

[14] E. N. Lazaridis, D. Sinibaldi, G. Bloom, S. Mane, and R. Jove. A simple method to improve probe set estimates from oligonucleotide arrays. *Math Biosci*, 176(1):53–58, 2002.

[15] W. Lemon, S. Liyanarachchi, and M. You. A high performance test of differential gene expression for oligonucleotide arrays. *Genome Biology*, 4(10):R67, 2003.

[16] Li and W. Wong. Model-based analysis of oligonucleotide arrays: Expression index computation and outlier detection. *Proceedings of the National Academy of Science*, 98:31–36, 2001.

[17] Li and W. Wong. Model-based analysis of oligonucleotide arrays: model validation, design issues and standard error application. *Genome Biology*, 2:1–11, 2001.

[18] M. A. Newton, A. Noueiry, D. Sarkar, and P. Ahlquist. Detecting differential gene expression with a semiparametric hierarchical mixture method. *Biostatistics*, 5(2):155–176, 2004.

[19] W. S. Noble. *Kernel methods in computational biology*, chapter Support vector machine applications in computational biology, pages 71–92. MIT Press, Cambridge, MA, 2004.

[20] B. O'Connell, A. F. Cheung, C. P. Simkevich, W. Tam, X. Ren, M. K. Mateyak, and J. M. Sedivy. A large scale genetic analysis of c-Myc-regulated gene expression patterns. *J Biol Chem.*, 278(14):12563–73, 2003.

[21] J. C. Platt. Probabilities for support vector machines. In A. Smola, P. Bartlett, B. Schölkopf, and D. Schuurmans, editors, *Advances in Large Margin Classifiers*, pages 61–74. MIT Press, 1999.

[22] V. N. Vapnik. *Statistical Learning Theory*. Adaptive and learning systems for signal processing, communications, and control. Wiley, New York, 1998.