Data Analysis Methods for DNA Microarrays: A Critical Review of Applications to Breast Cancer Research

Biochemistry 218/Bioinformatics 231

June 6, 2003

Caroline Reiss Smetana

I. Overview of Microarray Data Analysis

A. Introduction:

In a single assay, DNA microarrays allow researchers to analyze the mRNA expression levels of thousands genes, and consequently, detect global changes in the transcription of a genome. These chips, however, produce unprecedented amounts of data that scientists must analyze for both biological and statistical significance. Three of the most common methods for organizing the data are: (1) Hierarchical Clustering; (2) Self-Organizing Maps (SOMs); and (3) Support Vector Machine (SVMs). Hierarchical clustering efficiently organizes groups of genes into sets, but the other two methods, SOMs and SVMs, better identify changes in patterns of expression over a time period. After explaining a common protocol for generating data using a DNA microarray, this paper describes the three major data analysis techniques. It then critically analyzes which of these three methods is best in certain applications of breast cancer research, and finally, suggests a novel method to improve support vector machines when used in the gene expression pattern analysis of tissues treated with various chemotherapies.

Many microarray experiments study the alterations in expression patterns over a period of time following a stimulus to the cellular environment. A common method for obtaining microarray data is as follows: RNA from experimental cells is removed at, for example, the different sequential time points, and then reverse transcribed in the presence of the fluorescent dye Cy5. A reference sample, such as one taken at time 0, is also reverse transcribed but in the presence of Cy3, and subsequently mixed with the experimental samples containing the Cy5 dye. Following hybridization of these samples to the DNA microarray, the Cy5/Cy3 ratio of each spot is measured, with the ratio being expressed as a log odds ratio in the base 2. The color red denotes over-expression of a gene relative to the control state, green signifies under-expression, and black indicates no change in which case the log ratio is 0 as the Cy5/Cy3 ratio is 1 (equal absolute expression during experimental and control states).[1] These ratios are entered into an "expression matrix" where each row represents a different gene and each column is a single experiment (i.e. a different moment in time after administration of a compound). However, the generated matrix comparing gene expression and experiments lacks organization. Consequently, the three data analysis methods all aim to reorganize this matrix in order to improve the presentation of the information and emphasize specific patterns of gene expression. The first two methods, hierarchical clustering and SOMs lack the presence of a teacher signal and are thus unsupervised methods, whereas SVMs describes a supervised method.[2]

B. Data Analysis Techniques:

1. Hierarchical Clustering[3]

This technique clusters genes with similar expression vectors.[4] The first step builds a matrix comparing all genes to one another. [4] One option is to compute a matrix containing the distances between expression vectors. In terms of "expression space," each experiment is a new axis in space, with the $\log_2$(ratio) of that gene in that experiment serving as the geometric coordinate. In other words, following the addition of a compound, one extracts the mRNA at time 0mins, 5mins, and 10mins, and the x-coordinate is the $\log_2$(ratio) at time 0, the y-coordinate is the $\log_2$(ratio) at time 5mins, and the z-coordinate is the $\log_2$(ratio) at time 10mins. Each gene consequently has its own expression vector (point in space). Genes with similar vectors are close

together, whereas those with different patterns are far apart.[4] It is then possible to calculate the distance between points in order to determine their "mathematical similarity." A Euclidean distance is the most common metric distance: if x_i and y_i are the expression values for genes X and Y in experiment i and there is a total of n experiments: $d = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$.(4) Computing the Euclidean distance, however, is only one of many ways to measure the mathematical similarity between expression vectors. The paper by *Eisen et al.* instead computed a matrix of correlation coefficients (such as the dot product of two normalized vectors). For two genes, x and y and a total of n experiments, their similarity score is computed by the formula: $S(X,Y) = \frac{1}{N} \sum_{i=1}^{n} \left(\frac{x_i - x_{offset}}{\sigma_x} \right) \left(\frac{y_i - y_{offset}}{\sigma_y} \right)$ where the $\sigma_{x,y}$ are the standard deviations.[3] The authors suggest that the standard correlation coefficient better fits the biological notion of coexpression as this statistic describes similarity in "shape" but deemphasizes the idea of magnitude between the measurements.[3] After obtaining a matrix that represents the mathematical similarity between genes, hierarchical clustering then proceeds to build a dendrogram assembling all elements into a single tree.[3] An overview of this process is as follows: a program searches the distance matrix for the two most similar (or different) genes/clusters. The two selected clusters merge together, and the distances between the new cluster and all other clusters are recalculated. This process continues until all clusters are merged into one. Again, there are many possible methods to build this tree including: (A) *Single-linkage clustering* in which a program calculates the distance between two clusters by determining the minimum distance between two members of each cluster, in other words, the nearest neighbor method. (B) *Complete-linkage clustering* in which a program calculates the distance between two clusters determining the farthest distance between two members of each cluster. It tends to produce compact clusters of similar size. (C) *Average-linkage clustering* in which a program calculates distance by averaging values (UPGMA). The distances between all members in a cluster are first averaged, and then the distances between all clusters are examined. The clusters having the lowest distance scores are joined. (D). *Weighted pair-group average* is similar to average-linking but the size of each cluster is used to weight the distances, and is consequently best when there is an expectation for cluster sizes to be uneven.[4]

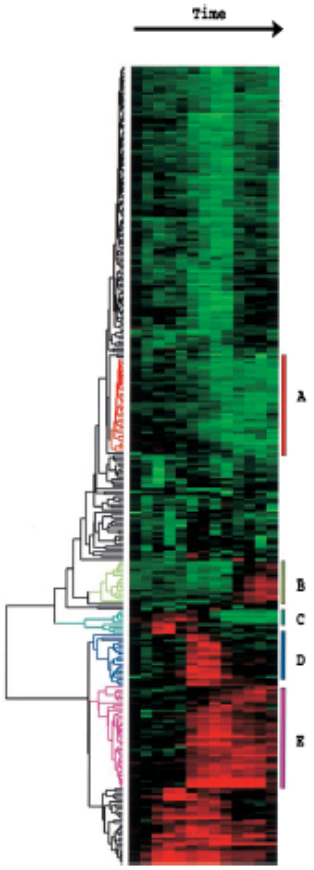

Figure 1: Example of Hierarchical Clustering taken

Eisen et al. use the last method listed. First an upper-diagonal similarity matrix containing the similarity scores for the pair of genes is computed. A computer program scans the matrix in order to identify the highest correlation coefficients. It then creates a node joining the

two genes and computes a gene expression profile for that node by averaging the values of the observations (weighing values by the number of genes each cluster already contains).[3] The matrix is constantly updated a total on *n*-1 times until a single element remains. The programs TREEVIEW AND CLUSTER implement this algorithm.

Two major problems with hierarchical clustering are: (1) as the elements within a cluster increase, the expression vector computed for that cluster may become unrepresentative of the elements it contains. Consequently, the patterns of expression themselves, loose their importance. (2) A mistake, or poor assignment early on, is irrevocable. [1,4] Further, using different clustering methods can yield dissimilar trees. Without additional biological information, however, any order can be considered "correct."[4] Thus, hierarchical clustering is perhaps best for applications of true hierarchical descent (i.e. evolution) and not for cases examining gene expression patterns in a mutated or experimental situation.

An improvement to hierarchical clustering is k means clustering which can be used when the researcher knows the total number of clusters the program should create.[4] The result is a series of k clusters internally similar but externally divergent. Basically, all expression profiles are initially randomly assigned to one of the k clusters. The program then computes an average expression vector for each cluster and between clusters. A recursive algorithm then methodically chooses expression vectors, and after calculating the intra and inter distances, moving the vector only if the selected cluster is more similar to the point than the original one. Finally, following the move, the average expression profile for each cluster is recalculated.[4] K means clustering allows biological knowledge to be integrated into the clustering method, but scientists must still judge if in fact these classes are significantly distinct.

2. Self-Organizing Maps[5]

SOMs, a neural network based clustering system, are better designed for explanatory data analysis as they allow one to partition the data based upon similar expression patterns.[4] Similarly to k means clustering, however, the user must be able to specify the number of desired clusters. First, the user chooses a geometric configuration for the nodes in two dimensions. These nodes are then arranged randomly in *n* dimensional space and recursively adjusted. The nodes next migrate to fit the data points. At each iteration, a data point P is randomly selected, and the node closest to that data point, Np, is moved most, while the other nodes are proportionally adjusted depending upon their distance to Np in the initial geometry. *Tamayo et al.* describe the algorithm (implemented by the program GENECLUSTER) as follows: a distance function $d(N_1,N_2)$ relates each node to one another, and the position of each node at iteration *i* is denoted by $f_i(N)$. The initial mapping $[f_0(N)]$ is random and at each iteration the node closest to the selected point P is identified, and all nodes then move by the algorithm $f_{i+1}(N) = f_i(N) + \alpha(d(N,N_p),i)(P - f_i(N))$. α, the learning rate, decreases with each

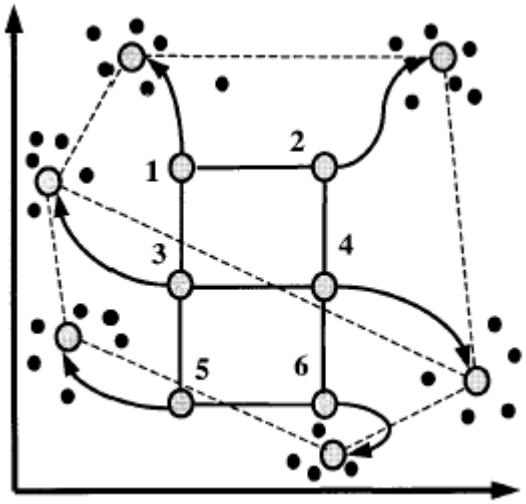

Figure 2: The Tamayo *et al.* description of the principles behind a Self Organizing Map. [5]

iteration, and is defined as $\Delta(x,i) = 0.02T/(T+100i)$.[5] In other words, the new position of each node depends upon its initial position, its distance from the node closest to the point, and its subsequent distance from the point.

3. Support Vector Machines[2]

SVMs are a binary classification method that discriminate one set of data points from another.[1] Using previous information obtained about gene expression, the SVM learns the expression features for a specific class and then classifies genes based upon their expression patterns as either included in class or excluded from it.[4] As the SVM learns to distinguish between class members and outliers, an optimal hyperplane is drawn to divide these points. In real world data, however, it is often difficult to clearly discriminate between positive and negative examples. SVMs solve this problem mapping the data points into a higher-dimensional space called the 'feature space' instead of into the 'input space' where one finds the training examples.[4] The feature space is so named because one calls each entry in the expression vector a feature.[18] Furthermore, algorithms determining the hyperplane in the feature space can be expressed exclusively in terms of vectors in the input space and dot products in the feature space. Consequently, the SVMs, by defining a 'kernel function' that assumes the role of the dot product in the feature space, can identify the hyperplane without ever having to actually represent the feature space.[2] As *Eisen et al.* also noted (see section on hierarchical clustering), the dot product of two normalized vectors is the simplest way to measure the similarity in expression vectors between two genes: $K(X,Y) = \vec{X} \bullet \vec{Y} = \sum_{i=1}^{n} X_i Y_i$.[2] (A 1 is added to this function for technical reasons, see reference). By raising the kernel function to a higher power, d, one obtains a hyperplane of higher degrees in the input space, and for any gene, there now exist d-fold interactions between RNA measurements in this kernel's feature space ($X_{i1}X_{i2}...X_{id}$): $\sum_{i=1}^{n} (X_i Y_i + 1)^d$. SVMs can consequently account for correlations within gene expression measurements.[2] *Brown et al.* explored kernel functions raised to the power d=1, 2, and 3. In the event that a SVM is unable to create an effective separating hyperplane, one usually also implements a 'soft margin' that allows some training examples to be misclassified. One technique for introducing the soft margin (controlling the trade-off between false positives and negatives) is to replace the kernel matrix, K(X,Y) with K + λD. λ, the diagonal factor, controls the training error and the risk of misclassification decreases with appropriate choices of λ. D, a diagonal matrix, contains entries that are either d^+ or d^- depending on their correspondence to positive and negative examples.[6] Although the higher level math is omitted from this paper, α_i when associated with the training point x^i, expresses the strength of the correlation with which the training point is rooted in the final decision function.[6] The above replacement allows one to control the value of α_i in a way that is proportional to the size of the class. Classes with smaller d are kept further away from the decision boundary.

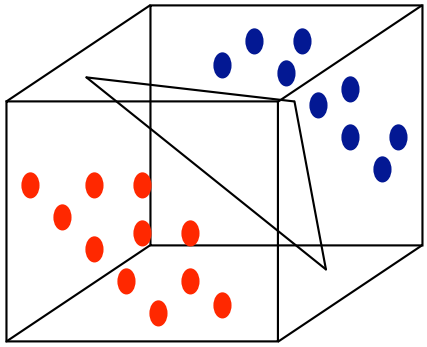

Figure 3: Support Vector Machine in which a hyperplane separates the positive data points (red) from the negative points (purple). Imagine reproduced from Mount, p. 522.[1]

Researchers must thus specify not only the kernel function, but also the penalty for violating the soft margin.

To obtain the feature selection criteria, studies by Furey *et al.* used the following methods: The goal is to locate which genes have expression vectors that the researcher can use to differentiate between two classes. Samples are labeled Y _ {+1, -1} (i.e. cancer and normal), and for each gene x_j, the mean μ_j^+ (μ_j^-) and standard deviation σ_j^+ (σ_j^-) are calculated. A score, $F(x_j)$, is then assigned to each gene. The highest scores are for genes having expression levels that differ most on average in the two classes, while preference also goes to genes having small score deviations in their respective classes: $F(x_j) = \left| \frac{\mu_j^+ - \mu_j^-}{\sigma_j^+ + \sigma_j^-} \right|$. Genes with the highest $F(x_j)$ scores are selected as top features. In sum, the entire SVM method can be summarized as follows: (1) choose kernel function; (2) adjust kernel function by changing the diagonal factor (soft margin); and (3) train the SVM to classify unknown samples using the genes selected in a "hold-one out procedure." In other words, train the SVM with all but one of the samples and then use the SVM to classify the held out sample. This process is repeated until all samples are held out once.[6]

Finally, the SVM software is available by downloading the GIST package from Columbia University.[18] The SVM program takes three files as inputs, train <filename> (the list of training examples), class <filename> (specified classifications for the training examples), and test <filename> (the to be classified data). It outputs two files, a weights file containing the weight of each training set example, and a prediction file with the predicted classification of the test set. Examples of these files are listed below and taken from: http://svm.sdsc.edu/svm-io.html [18]

train <filename>

corner	feature_1	feature_2	feature_3	feature_4
example_1	-0.9	-3.9	-3.1	0.7
example_2	2.1	1.1	0.3	-1.6
example_3	3.5	2.0	-0.3	3.1
example_4	-2.3	-0.4	-0.4	-0.1

class <filename>

corner	class
example_1	-1
example_2	1
example_3	1
example_4	-1

test <filename>

corner	feature_1	feature_2	feature_3	feature_4
example_11	0.3	0.3	-2.2	-0.1
example_12	-1.9	-1.8	0.5	2.6
example_13	-1.0	3.0	2.1	-0.1

output weights file

corner	class	weight	train_classification	train_discriminant
example_1	-1	-0	-1	-2.341
example_2	1	0.1321	1	0.9991
example_3	1	0	1	1.83
example_4	-1	-0	-1	-1.058
example_5	-1	-0.09971	-1	-1

output predicted classification

corner	classification	discriminant
example_11	-1	-0.03785
example_12	1	0.0522

```
example_13        -1            -0.08235
```

GIST uses the following programs to train the SVM and perform the classification of the test set: COMPUTE-WEIGHTS uses the iterative update procedure (first described by Jaakkola, Diekhans, and Haussler) to train a support vector machine. Inputs include: (1) train <filename>: the file of training examples in which the first column contains the example names and the subsequent columns contain the gene expression levels in each experiment. (2) class <filename>: the file which classifies the training set into the positive (+1) or negative category (-1). Again, the first column contains example names and subsequent columns contain the binary classification. The number of elements must be the same in each of the first two files. The output is the weights file containing five columns: (1) example name; (2) class; (3) learned weights for each example [non-zero weights are given to training examples considered "support vectors" (the point lies near to the inside/outside of the separating hyperplane determined by the SVM algorithm)]; (4) predicted (train) classification; and (5) corresponding discriminate value (determined by calculating proportional distance between example and hyperplane). CLASSIFY uses the trained SVM to classify an unlabeled set of vectors. Inputs are: (1) the train <filename> from above; (2) learned <filename> which is the output file from COMPUTE-WEIGHTS, and the kernel function parameters are read from the header of this file; and (3) test <filename> containing the test set to be classified. Output includes 1 file with three columns containing (1) test set names; (2) binary classification(1, -1); and (3) discriminants. KERNEL-PCA for the set of training examples, performs the kernel principal component analysis by computing kernel-based eigenvectors. It uses the train <filename> as an input, and returns a matrix in which columns correspond to eigenvectors as the output. PROJECT inputs are the same as CLASSIFY, and the output includes a matrix in which the test data has been projected onto the given set of eigenvectors

Additionally, there are a few auxiliary programs that help to manage the data: FSELECT uses a specified measure for feature quality to select features from a given data set. RDB-MATRIX manipulates rows and columns in a specified matrix. SCORE-SVM-RESULTS takes the outputs from COMPUTE-WEIGHTS and CLASSIFY and performs statistical analysis. It calculates the number of false positives, false negatives, true positives, and true negatives in both the training set and the experimental set, as well as an ROC score which is determined by calculating the area under the curve that plots true positives as a function of false positives for differing decision thresholds (perfect scores correspond to 1.0). FIT-SIGMOID converts the COMPUTE-WEIGHTS discriminant values into probabilities, and finally, GIST3HTML converts GIST output files into HTML format. [18]

GIST was written by William Stafford Noble from the Columbia University computer science department and by Paul Pavlidis of the Columbia Genome Center.

II. Microarray Applications in Breast Cancer Research:

A. Background Information:

To date, the molecular basis of breast tumorigenesis is poorly understood.[7] Extreme genetic heterogeneity exists in breast cancers, and no single genetic mutation induces all forms of this disease. Tumors consequently can have differing clinical outcomes and reactions to therapies. The advent of DNA microarrays allows much of the current research in this field to focus on "expression profiling" in which genes are analyzed according to similar expression

patterns. Microarrays permit genome wide comparisons of tumors having, for example, mutations in the BRCA1 and BRCA2 genes. It is thus now possible for scientists to distinguish characteristics specific to each form of breast cancer and develop better diagnostic, prognostic, and therapeutic techniques. Despite some overlap, two main avenues in breast cancer research exist: (1) Issues related to the specific functioning of the disease; and (2) general diagnostic and classification studies. The first uses arrays to better understand the functioning of oncogenesis. Such studies can uncover novel genes and regulatory pathways via comparison of expression profiles in the stages leading from normal to metastic conditions.[8] Scientists may focus on a specific mechanism by either over expressing a gene, such as BRCA1,[9] or under expressing one, such as p53.[10] The other major research area examines methods for tumor classification in order to better categorize patients and guide them toward the most appropriate treatments. The specific question being studied, however, often dictates the most appropriate data analysis method (HC, SOMs, or SVMs). The supervised methods are by nature predictive, whereas the unsupervised methods simply reduce the complexity of the data and allows the naked eye to observe common structures.[8]

The remainder of this paper examines which data analysis methods are best for studying survival outcomes of patients receiving various chemotherapies. Adjuvant anthracyclin-based chemotherapy, for example, has a 40% failure rate, and with the increasing availability of novel therapies, physicians wish to know from which treatments each patient will most benefit.[8] Although risks of distant metastases can diminish by one third when physicians treat patients with hormonal therapies or chemotherapies, approximately 70-80% would have survived anyway.[11] By using microarrays to discriminate between patients with positive and negative prognosis, scientist can not only refine the prognostic classification of breast cancer, but also better understand the mechanisms of these therapies thereby providing novel targets for future research. This information would also allow the possibility for patient-tailored therapy strategies.

Many outcome predictions studies use clustering methods to organize the microarray data. This technique, however, may be inappropriate given the goal of these studies. When a project aims to distinguish specimens such that expression profiles are similar within groups and different between groups, then clustering methods are useful. If, however, the samples already come from known prespecified groups, then the goal (as in the case of comparative chemotherapy studies) is usually to identify differently expressed genes or global differences between groups.[12] Korn *et al.*, using another form of statistical analysis (the step down permutation approach),[19] revisited data that was first analyzed using hierarchical programs (Perou *et al.*[13]*)* to discover 17 genes for which the original authors failed to account.[12] Analysis of these 17 genes helps provide biological incite into the responses of tumors to doxorubicin treatment. Thus, although clustering methods highlight possible relationships between genes, they give no absolute answers and can miss vital changes in expression patterns. Furey *et al.* have already begun to apply support vector machines to classify cancer tissues. The remainder of this paper examines the progress in the SVM-breast cancer field and suggests possible improvements to SVMs that analyze tissues between patients receiving different forms of chemotherapy.

B. Supervised Clustering Methods for Analysis of Changes in Gene Expression Patterns:

1. Current Research

van't Veer *et al.* used a combination of supervised and unsupervised methods in their study of gene expression profiles to predict clinical outcomes of breast cancer patients.[14] Clustering methods effectively distinguished between 'good prognosis' and 'poor prognosis' tumors. However, in order to analyze specific genes with differing gene expression profiles between the two groups, the authors utilized supervised classification methods instead. A correlation coefficient comparing the expression of each gene with disease outcome was calculated for 5,000 (of 25,000) genes. 231 were found to be statistically significant, and the program created a rank ordered list. To optimize the number genes in the "prognosis classifier," subsets of 5 genes were sequentially added from top to bottom of the rank list. The authors implemented the 'leave-one-out' method to determine the program's ability to classify samples correctly, and the optimal number of marker genes was 70. By classifying genes via this method, researchers obtained a better understanding of the biological mechanisms leading to rapid metastases. Many of the genes found in the poor prognosis classifier were involved in the cell cycle regulation, angiogenesis, and signal transduction (i.e. cyclin E2, MCM6, metalloproteinases, and the BEGF receptor FLT1).[14] The authors make note that none of their 70 marker genes have been previously emphasized in studies correlating changes in expression patterns with different disease outcomes. They explain this phenomena by stating how earlier clinical studies relied on observing single genes in isolation, and thus emphasize the need multi-gene approach based methods.

Although few studies have actually applied SVMs to breast cancer, Furey *et al.* have, applied SVMs to analyzing ovarian cancer tissues, and it is possible replicate their methods for other diseases. SVMs can not only separate expression vector between classes, but can also identify errors in previous classification data.

2. Suggestions

One of the major problems with a SVM used in the Furey *et al.* study was after classifying genes into the positive group, in some cases, the machine gave high rankings to biologically meaningless genes and failed to recognize known tumor genes. Consequently, to improve SVM functioning, additional methods for identifying and ranking important genes must be devised.[6] It thus seems that an additional stage is needed before the final output of data. A machine capable of not only dividing genes between classes, but also then able to go through the sub classes and make associations between genes within the positive class, would be valuable.

In their paper, Furey *et al.* give a figure illustrating the SVM classification margins for ovarian tissues. During the classification process, the SVM calculates a margin, or the distance of the specific gene to the decision boundary. The output, however, lacks organization, and it is this data that should be reorganized by the program before final display. To start, it might be helpful to implement a hierarchical clustering method at this point in the program. Although SOMs are well designed for identifying small numbers of important classes in a data set,[15] the user may not know in advance how many clusters to expect. Using clustering methods, one could thus identify genes with similar

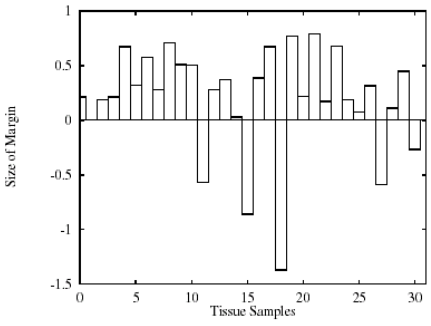

Figure 4: SVM classification margins for

expression patterns in the positive category. This method would also reorganize the data in terms of similarities within member of the class, as opposed to ranking the data by similarity to the training set.

One of the questions Golub *et al.* addressed in their study on cancer class discovery and prediction was: if a sample has been partitioned into a positive and negative category, how can one determine if in fact these clusters are correct if the "right answer" is not already known. The authors broached this question because they had used SOMs in order to classify cancer samples into either the AML or ALL family, but the scenario is almost identical to that of SVMs. After the SVM has classified a sample as positive or negative based upon gene expression patterns, how can one then determine if these samples are classified correctly? Golub *et al.* suggest that assuming the positive class reflects the "truth," then building a new class predictor using the members of that class should perform reasonably well. Following this logic, it is even more imperative that the data within the positive class of the SVM be well analyzed. Continuously analyzing and rebuilding a class predictor portfolio can only help to further the understanding of this disease.

To modify the GIST program, for example, a logical place to insert this improvement is in the SCORE-SVM-RESULTS auxiliary program. A function would need to sort through the CLASSIFY output matrix extracting the vectors with a positive classification and forming a new matrix. This new matrix, then, could be manipulated in order to find similarities among expression patterns within members of the same class.

III. Conclusion

A variety of data analysis methods for DNA microarray currently exist. Generally, the data gathered from microarrays prompts two categories of questions: (1) those about variables themselves, such as, which genes or clusters of genes are associated with a specific phenotype, biological mechanism, or outcome; and (2) those regarding biological samples, such as, what predictions can be made about a specific tissue (either for diagnostic and/or prognostic reasons).[16] Most statistical methods, including clustering, easily address the first question. Pattern classifier, such as SVMs and other machine learning systems, however, are much better for the second. It is vital that the scientist use the most appropriate computational analysis technique in order to extract the maximum amount of information from his sample. In studies examining different clinical outcomes of patients receiving a variety of chemotherapies, support vector machines provide the most informative results. These machines, however, still have flaws. It may be possible to improve these machines by using other data analysis techniques, such as clustering, on a specified subsets of SVM data.

IV. References

1. Mount DW. Bioinformatics: Sequence and Genome Analysis. Cold Spring Harbor Laboratory Press; New York: 2001.
2. Brown MP, Grundy WN, Lin D, Cristianini N, Sugnet CW, Furey TS, Ares M Jr, Haussler D. Knowledge-based analysis of microarray gene expression data by using support vector machines. Proc Natl Acad Sci U S A. 2000 Jan 4;97(1):262-7.
3. Eisen MB, Spellman PT, Brown PO, Botstein D. Cluster analysis and display of genome-wide expression patterns. Proc Natl Acad Sci U S A. 1998 Dec 8;95(25):14863-8.

4. Quackenbush J. Computational analysis of microarray data. Nat Rev Genet. 2001 Jun;2(6):418-27. Review.
5. Tamayo P, Slonim D, Mesirov J, Zhu Q, Kitareewan S, Dmitrovsky E, Lander ES, Golub TR. Interpreting patterns of gene expression with self-organizing maps: methods and application to hematopoietic differentiation. Proc Natl Acad Sci U S A. 1999 Mar 16;96(6):2907-12.
6. Furey TS, Cristianini N, Duffy N, Bednarski DW, Schummer M, Haussler D. Support vector machine classification and validation of cancer tissue samples using microarray expression data. Bioinformatics. 2000 May 19; 16(10):906-14.
7. Ma XJ, Salunga R, Tuggle JT, Gaudet J, Enright E, McQuary P, Payette T, Pistone M, Stecker K, Zhang BM, Zhou YX, Varnholt H, Smith B, Gadd M, Chatfield E, Kessler J, Baer TM, Erlander MG, Sgroi DC. Gene expression profiles of human breast cancer progression. Proc Natl Acad Sci U S A. 2003 May 13;100(10):5974-9.
8. Bertucci F, Viens P, Hingamp P, Nasser V, Houlgatte R, Birnbaum D. Breast cancer revisited using DNA array-based gene expression profiling. Int J Cancer. 2003 Feb 20;103(5):565-71. Review.
9. Welcsh PL, Lee MK, Gonzalez-Hernandez RM, Black DJ, Mahadevappa M, Swisher EM, Warrington JA, King MC. BRCA1 transcriptionally regulates genes involved in breast tumorigenesis. Proc Natl Acad Sci U S A. 2002 May 28;99(11):7560-5.
10. Zhao R, Gish K, Murphy M, Yin Y, Notterman D, Hoffman WH, Tom E, Mack DH, Levine AJ. Analysis of p53-regulated gene expression patterns using oligonucleotide arrays. Genes Dev. 2000 Apr 15;14(8):981-93.
11. Early Breast Cancer Trialists' Collaborative Group. Polychemotherapy for early breast cancer: an overview of randomized trials. Lancet 1998; 352:930-42.
12. Korn EL, McShane LM, Troendle JF, Rosenwald A, Simon R. Identifying pre-post chemotherapy differences in gene expression in breast tumours: a statistical method appropriate for this aim. Br J Cancer. 2002 Apr 8;86(7):1093-6.
13. Perou CM, Sorlie T, Eisen MB, van de Rijn M, Jeffrey SS, Rees CA, Pollack JR, Ross DT, Johnsen H, Aksien LA, Fluge O, Pergameschikov A, Williams C, Zhu SX, Lonning PE, Borresen-Dale AL, Brown PO, Botstein D. Molecular portraits of human breast tumors. Nature 2000 406:747-52.
14. van't Veer LJ, Dai H, van de Vijver MJ, He YD, Hart AAM, Mao M, Peterse HL, van der Kooy K, Marton MJ, Witteveen AT, Schreiber GJ, Kerkhoven RM, Roberts C, Linsley PS, Bernards R, Friend SH. Gene expression profiling predicts clinical outcome of breast cancer. Nature 2002 Jan 31;415:530-36.
15. Golub TR, Slonim DK, Tamayo P, Huard C, Gassenbeek M, Mesirov JP, Coller H, Loh ML, Downing JR, Caligiuri MA, Bloomfield CD, Lander ES. Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring. Science 1999 Oct 15; 286:531-37.
16. Olshen AB, Jain AN. Deriving quantitative conclusions from microarray expression data. Bioinformatics 2002 Jan 24; 18(7):961-70.
17. Schena M (Editor). DNA Microarrays: A Practical Approach. Oxford University Press; Oxford, 2002. *(Source of general information regarding DNA Microarrays)*
18. GIST Software Package: http://microarray.cpmc.columbia.edu/gist/ Columbia University, May 20, 2003.

19. Westfall P.H., Young SS (1993) *Resampling- Based Multiple Testing,* pp 72-4 New York: Wiley
20. Davies Erin. *A critical review of computational methods used to manage microarray data sets*. 10 March 2003; BioChem218 Final Project.