BIOINFORMATICS

Vol. 19 no. 1 2003
Pages 45–52

# Evolutionary algorithms for finding optimal gene sets in microarray prediction

*J. M. Deutsch*

*University of California, Santa Cruz, USA*

Received on August 8, 2001; revised on December 5, 2001; July 10, 2002; accepted on July 12, 2002

**ABSTRACT**

**Motivation:** Microarray data has been shown recently to be efficacious in distinguishing closely related cell types that often appear in different forms of cancer, but is not yet practical clinically. However, the data might be used to construct a minimal set of marker genes that could then be used clinically by making antibody assays to diagnose a specific type of cancer. Here a replication algorithm is used for this purpose. It evolves an ensemble of predictors, all using different combinations of genes to generate a set of optimal predictors.

**Results:** We apply this method to the leukemia data of the Whitehead/MIT group that attempts to differentially diagnose two kinds of leukemia, and also to data of Khan *et al.* to distinguish four different kinds of childhood cancers. In the latter case we were able to reduce the number of genes needed from 96 to less than 15, while at the same time being able to classify all of their test data perfectly. We also apply this method to two other cases, Diffuse large B-cell lymphoma data (Shipp *et al.*, 2002), and data of Ramaswamy *et al.* on multiclass diagnosis of 14 common tumor types.

**Availability:** http://stravinsky.ucsc.edu/josh/gesses/

**Contact:** josh@physics.ucsc.edu

## INTRODUCTION

cDNA and oligonucleotide microarrays have been used with great success to distinguish cell types from each other, and hence has promising applications to cancer diagnosis. While the histopathology of two cells may appear very similar, their clinical behavior, such as their response to drugs can be drastically different. The use of microarrays has been shown in many cases to provide clear differential diagnosis rivaling or surpassing other methods and leads to a clustering of data into different forms of a disease (DeRisi *et al.*, 1996; Alon *et al.*, 1999; Perou *et al.*, 1999; Zhu *et al.*, 1998; Wang *et al.*, 1999; Schummer *et al.*, 1999; Zhang *et al.*, 1997; Alizadeh *et al.*, 2000; Golub *et al.*, 1999; Khan *et al.*, 2001).

Many approaches have been used to classify microarray data. These include the use of artificial neural networks (Khan *et al.*, 2001; Furey *et al.*, 2000), logistic regression (Li and Yang, 2002), support vector machines (Brown *et al.*, 2000; Furey *et al.*, 2000), coupled two-way clustering (Getz *et al.*, 2000), weighted votes—neighborhood analysis (Golub *et al.*, 1999) and feature selection techniques (Xing *et al.*, 2001). For much of the data all these techniques appear to give similar results and their performance improves as the amount and quality of data increases.

To classify samples using microarray data, it is necessary to decide which genes should be included in a predictor. Including too few genes will not discriminate in a detailed enough manner to classify test data correctly. Having too many genes is not optimal either, as many of the genes are largely irrelevant to the diagnosis and mostly have the effect of adding noise, decreasing the 'information criterion' (Li and Yang, 2002; Akaike, 1974; Burnham, 1998; Schwarz, 1976). This is particularly severe with a noisy data set and few subjects. Therefore an effort is made to choose an optimal set of genes for which to start the training of a predictor. This is done in a variety of different ways, such as a kind of neighborhood analysis (Golub *et al.*, 1999), principle component analysis (Khan *et al.*, 2001), or gene shaving (Hastie *et al.*, 2000). A predictor can then be developed from this carefully chosen subset of genes.

Recent work (Li and Yang, 2002) addressed the problem of gene selection for a leukemia data set (Golub *et al.*, 1999). They initially ranked genes as had been done in the first analysis of Golub *et al.* and used the top ranked genes. They varied the number they included and found no clear indication of any optimum number aside from the conclusion that the number should be much smaller than the 50 that had been originally used.

Here we develop gene selection further by making it an integral part of the prediction algorithm itself. Instead of using all of the highest ranked genes, we find an effective method to greatly reduce this number. This can be done because gene expression tends to be highly correlated, making many of the initially chosen genes redundant or even deleterious because of the problem of added noise.

The method introduced here is named GESSES (genetic evolution of sub-sets of expressed sequences). It makes

use of a kind of evolutionary algorithm known as a replication algorithm that has been extensively used in quantum simulations (Ceperley and Kalos, 1979) and protein folding (Garel and Orland, 1990). It finds a set of highly relevant genes by considering a whole ensemble of predictors, and evolving this population by addition or deletion of genes until optimal performance has been achieved.

In the case of small round blue cell tumors, GESSES reduces the number of genes from 96 down to below 15 while still predicting the test data perfectly. Some of the perfect predictors have only ten genes.

It is hoped that GESSES will have applications in the clinical diagnosis of cancer (He and Friend, 2001). At the moment, microarray experiments are too costly and time consuming to be used clinically. However, if a subset of marker genes could be found whose expression levels could then be obtained using relatively inexpensive antibody assays, this might become a practical method. Therefore for this purpose it is important to use as few genes as possible and still obtain an accurate diagnosis of the disease.

With the same algorithms applied to leukemia data of Golub *et al.*, we find conclusions in accord with Li and Yang (2002) that there is no clear indication of an optimum number of genes to use in a predictor.

GESSES was applied to several additional data sets, two data sets related to Diffuse large B-cell lymphoma (DLBCL) (Shipp *et al.*, 2002) and work (Ramaswamy *et al.*, 2001) on the diagnosis of 14 different classes of tumors using microarrays. In these cases, GESSES was able to reduce the number of genes needed to make a prediction of a given error rate.

This paper is organized as follows. We discuss the algorithm used by first providing an overview of its basic features and then in detail by first defining the terminology and concepts used. Then we discuss the predictor used, the scoring function, the kind of evolutionary algorithms used and the annealing schedules. We then apply this to the SRBCT, leukemia data, two DLBCL data sets, and multiclass tumor data. Finally, we make some concluding remarks.

## THE ALGORITHM

### Overview

The algorithm can be divided up into several parts. First we are interested in the evolution of an *ensemble*, or population, of predictors. What distinguishes one predictor from another is the subset of genes it utilizes in making a prediction.

The most successful predictors will be the ones making fewest mistakes on test data. To determine which predictors are most successful, we utilize a scoring function which gives higher scores when more data points are correctly classified, that is the smallest classification error. Because we can only use a fixed amount of training data when evolving the predictors, we use leave-one-out cross validation (LOOCV) to calculate the score for a certain predictor. We obtain better predictors by adding an additional term to the scoring function to give higher marks to predictors that do a good job of grouping the data into well separated clusters, each cluster corresponding to the same type of cancer.

We would like a method that searches through a large number of different subsets of genes to come up with a population of the highest scoring predictors. This is often referred to as a *wrapper* method (Langley, 1994; Kohavi and John, 1997).

Most genes have little or no predictive value. The more of them that are included as possible choices, the more noise is added to the predictions which leads to a degradation in the performance of the prediction ensemble. Therefore we apply a *filter* (Xing *et al.*, 2001) method to construct an initial gene pool containing the most likely candidate genes. We use a simple method of ranking genes similar to previous work (Ben-Dor *et al.*, 2000).

We employ several methods for evolving our population of predictors. We produce offspring by random mutations and deletions of genes, with the number of replications of a particular predictor depending on how the mutations effect the scoring function. The notion of temperature is employed to control the degree to which less favorable mutations are kept in future generations. The higher the temperature, the more unfavorable predictors are kept. We slowly cool the system so that eventually we weed out all but the most fit predictors. This is a kind of simulated annealing. In addition we employ deterministic methods of evolution that try many combinations, only keeping the ones that score highest.

### Terminology

We have samples of microarray training data $\mathcal{D}_t \equiv \{D_1, D_2, \ldots\}$ with each sample consisting of $N$ genes. The complete set of genes $\mathcal{G}_t$ is the collection of genes 1 through $N$ and we will consider subsets of $\mathcal{G}_t$, for example the subset $\alpha_1, \alpha_2, \ldots, \alpha_m$. (e.g. genes 2, 5 and 9), which we denote $G_\alpha$.

The set of possible types is denoted $\mathcal{T}$. Each sample $D$ has a classification of type $T$, in this case the type of cancer, which can take one of $|\mathcal{T}|$ values.

### Predictor

We define a predictor $\mathcal{P}$ as a function that takes a data sample $D$ and outputs a type $T$, in this case the type of cancer that is associated with that data. That is $\mathcal{P}(D) \rightarrow T$.

In this work we will use a $k$-nearest neighbor search (Duda and Hart, 1973) to construct the predictor. In the results reported below, we use $k = 1$, that is, the set of samples that forms the training data $\mathcal{D}_t$ are compared with the target sample $D$ by finding the usual Euclidean distance between $D$ and each vector in the training set. The sample in the training set closest to $D$ gives the classification $T$ of $D$. The distance depends on what subspace of genes $G$ is used hence the predictor depends both on the training data and $G$.

We will use variants of this basic predictor when constructing a scoring function that we discuss below. For this we will not only need the closest point, but the values of the distances to all sample points.

### Scoring function

The scoring function has two parts and is closely related to LOOCV. We iteratively single out one data point and consider this to be pseudo test data. If this point is predicted correctly, we add 1 and also add a term that maximizes the separation between different classes as follows.

We consider the distances grouped by the classification type of the target points. We consider the shortest distance of each type which we call $d_1, d_2, \ldots, d_{|\mathcal{T}|}$. Of these we take the two shortest, $d_i$ and $d_j$ and add $C|d_i^2 - d_j^2|$ where $C$ is a constant chosen so that the value of these added terms is $\ll 1$.

After looping through the entire data set this way we obtain the total score.

The scoring function depends on the predictor, which in turn is determined by the training data and the subspace of genes $G$. We will denote this latter dependence as $\mathcal{S}_G$

### Initial Gene Pool

Often it is necessary to narrow down the genes that are considered from the many thousand that are measured on the microarray down to of order $10^2$ that are most relevant. There are many ways of doing this, one common method being principle component analysis. For the purposes here we choose instead a different method that is highly effective and similar to one previously used (Ben-Dor *et al.*, 2000).

We consider how genes distinguish two types $t_i, t_j$ from each other. For each gene $g$ we consider its expression levels in the training samples. We rank all the training samples in terms of the expression level of $g$. We are looking for genes that for high levels give type $t_i$ and for low levels give type $t_j$ (or vice-versa). When ranked this way, they sometimes will perfectly separate, that is the first part of the list is one type, and the last part is the other. These genes are ranked the highest. Most of the time however, a gene will not separate so clearly and there will be overlapping regions. Those with more overlaps of different types are ranked lower. In this way we have a ranking of the genes that are best able to distinguish $t_i$ from $t_j$, and we pick the top $M$ genes.

We then consider all distinct combinations of $t_i$ and $t_j$ and pick the best $M$ genes from each combination. Genes may overlap, narrowing the initial pool. This is our initial set of genes $\mathcal{G}_i$ that we will consider. A slight variant in this algorithm is necessary if the data set contains too few examples of a given class. In that case one compares type $t_i$ with all other classes, instead of only $t_j$. Otherwise a large number of irrelevant genes can rank highly by chance.

### Evolution Algorithms

Starting off with an ensemble of different gene subspaces we want to determine rules to evolve it to a new one that gives a better set of predictors. To do this, we have to have a measure of how well a predictor classifies samples into separate types. We do this by means of the scoring function described above. The evolutionary methods described below show how to utilize the scores to determine which predictors are kept and which are eliminated.

### Statistical Replication

In analogy with statistical mechanics, we can think of the scoring function as (negative) energy and invent a dynamics that evolves them towards the highest scoring (lowest energy) states. We do this at finite temperature to allow the system to accept predictors that occasionally may be less fit than their predecessors to get rid of local minima in predictor space and to allow for a diverse population of predictors.

Suppose the system has evolved to an ensemble of $n$ gene subspaces $\mathcal{E} \equiv \{G_1, G_2, \ldots, G_n\}$, we will now employ a variant of a replication algorithm used in other contexts (Ceperley and Kalos, 1979) to replicate and modify each of the $G_i$'s.

1. For each $G \in \mathcal{E}$ we produce a new subspace as follows.
   (a) A set of genes $G$ has genes $\{g_1, g_2, \ldots, g_m\}$. We randomly mutate genes according to three possibilities:
      i. Add an extra gene: We choose a randomly chosen gene $g_r$ from the initial set $\mathcal{G}_i$, and add it to $G$, producing a new set $G'$ of genes $\{g_1, g_2, \ldots, g_m, g_r\}$. If $g_r \in G$, $G' = G$.
      ii. Delete a gene: We randomly delete a gene from $G$ producing a new set with $m - 1$ total genes.
      iii. Keep $G$ the same.

(b) We compute the difference in the scoring functions $\delta\mathcal{S} = \mathcal{S}_{G'} - \mathcal{S}_G$.

(c) We compute the weight for $G'$, $w = \exp(\beta\delta\mathcal{S})$, where $\beta$ is the inverse 'temperature'.

2. Let $Z$ denote the sum of these weights. We normalize the weights by multiplying them by $n/Z$.

3. We replicate all subspaces according to their weights. With a weight $w$, the subspace is replicated $[w]$ and an additional time with probability $w - [w]$. Here $[w]$ denotes the largest integer $< w$.

Mutations as described in 1(a)ii and 1(a)iii can be optionally added, as the algorithm works well with just 1(a)i.

In summary, every subspace in the system is mutated and replicated in accordance with how much fitter it was than its predecessor. By carefully normalizing the system, the number of subspaces in the ensemble stays close to $n$. Note that we can also do more than one potential mutation in step 1. We will generalize this to allow $n_m$ potential mutations.

### Annealing

As the system evolves, the scoring function gives similar answers for all members of the ensemble. In order to improve convergence, it is useful to make the temperature a function of the spread in scores (or energy). A variety of schedules for the temperature were tested. The one that worked best lowered the temperature in accord with the fluctuation in the score (or energy) from predictor to predictor within the ensemble, $\sigma_E$. The temperature scale was adaptively chosen to be proportional to $\sigma_E$. This quickly changes energy scale when all training examples are correctly classified, but cools down slowly enough so as not to get trapped in local minima.

This schedule is also useful because it allows us to define a simple termination condition when all moves (additions and deletions) are allowed. In this case the condition is that the ensemble is unchanged for ten consecutive iterations. In practice the system terminates fairly rapidly because eventually the temperature decreases to essentially zero, leaving only a small number of systems left in the ensemble.

### Deterministic Evolution

As an alternative to the statistical replication method above, we also employed a method that is computationally more expensive but that often performs better. The statistical method does not explore all possible combinations of genes at each stage of growth. This can miss optimal gene combinations. We get around this by a deterministic exploration of the optimum gene combinations at every step. A single step goes as follows:

1. Construct all distinct unions of the $G$'s in the ensemble $\mathcal{E}$ with individual genes $g_i$ in the initial gene pool $\mathcal{G}_i$, i.e. $g_1, g_2, \ldots, g_m, g_i$.

2. Sort all of these combinations by their score, keeping the top $n_{top}$ of them.

To save computer time we tried various values for $n_{top}$. It was found that $n_{top} = n$, (the number of $G$'s in the ensemble) performed quite well. Another variant was to construct only half the unions and keep the top $n$, for computational efficiency.

## RESULTS

### SRBCT Data

Small round blue cell tumors (SRBCT) of childhood are hard to classify by current clinical techniques. They appear similar under a light microscope and several techniques are normally needed to obtain an accurate diagnosis. The paper (Khan *et al.*, 2001) used microarrays to study their classification using a single layer neural network. This work differed from previous studies in that they were attempting to distinguish between four different cancer types instead of the more usual two. They used 63 samples for training and tested with 20. By using a clever method combining principle component analysis and sensitivity of their neural network to a gene, they were able to reduce the number genes needed to 96 yet still classify all different forms of cancer in test data perfectly.

Here we use the same data set to reduce the number of genes needed and still classify the test data perfectly.

Starting with their data set of 2308 genes, we constructed the initial pool of genes by considering how well a gene discriminates type $i$ cancer from type $j$, as described above. Since there are four possible types, we have six combinations of $i$ and $j$. For each of these we take the top ten genes best able to discriminate for each $i$, $j$ pair. This gives a total of 50 genes, because it turns out that ten of these overlap between groups.

We then evolve these gene subspaces according to the statistical replication method with all mutational moves referred to in the section on statistical replication, 1(a)ii and 1(a)iii. We started with the same initial pool of genes as above. The results are shown in Figure 1. The average number of dimensions rises to a maximum of about 16 after 32 iterations, while the number of wrong classifications decreases from about nine down to about 0.5. By iteration 26 all members of the ensemble classify the *training data* perfectly. At this point, as was expected, the temperature falls very rapidly, so that the scoring function only probes its small second piece. Now the temperature drops and most classifiers are predicting perfectly. Eventually the systems predict the test data perfectly and we continue to cool it until

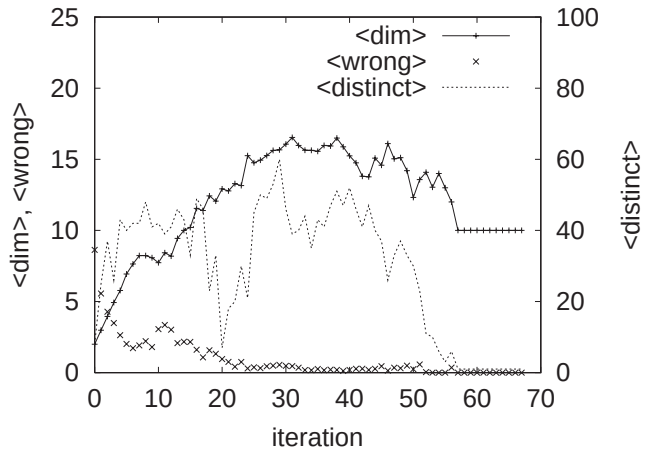


**Fig. 1.** The statistical algorithm using all mutational moves. The average number of dimensions (solid line with + symbols), the average number that the predictor got wrong (× symbols), and the number of distinct systems in the ensemble (dashed lines) as a function of the number of iterations. The number of genes used here was 50 ($n_m = 20$).

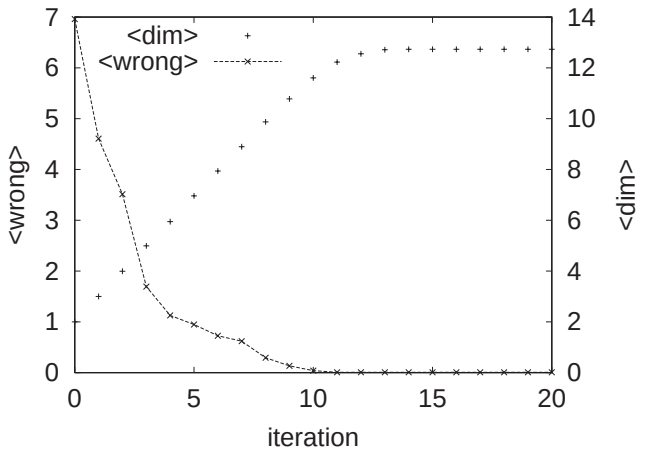


**Fig. 2.** The average number of genes and average number of mistakes made as a function of iteration in a predictor generated by the deterministic algorithm for an initial pool of 75 genes SRBCT data (Khan *et al.*, 2001). The parameters are described in the text.

the termination condition is met described above (see the section annealing). For the purposes of these figures, the number of distinct systems is the number of parents that the ensemble has in common.

We next use the deterministic evolution method described above starting with an initial pool of 90 genes of which 15 overlapped, giving a total of 75 initial genes. Evolving these with $n_{top} = 150$ gives the results shown in Figure 2. The + point in Figure 2 shows the average number of genes in a predictor as a function of the number of generations. Of the top 100 predictors, all predicted the test data perfectly. The average number of genes in a predictor was 12.7.

With this same initial pool of 75 we also ran the statistical algorithm allowing for all mutational moves, as was done for Figure 1. The features of this run are similar to those for the run with 50 initial genes, Figure 1. After iteration 28 the test data is predicted perfectly and the temperature rapidly decreases. At iteration 65 all system predict the test data perfectly. The temperature is further decreased until a steady state solution is reached, in this case where the number of distinct systems is three.

The implementation of GESSES is quite efficient and the above results took of order a few minutes to complete on a modest 450 Mhz Celeron machine, using of order 5 Mbytes RAM.

The genes found by these methods are mostly a subset of those found previously (Khan *et al.*, 2001). For example with 75 initial genes as described above (Figure 2), the union of all predictor genes found in the top 100 predictors gave a total of 24 genes. These were a subset of the 96 Khan *et al.* genes. However with other runs this is not always the case. For example with a run using statistical replication with only addition of genes and the same initial pool as Figure 1, we find that out of a total of 25 different genes that comprise all the possible genes used by the 50 predictors, four are different than those found by Khan *et al.* Of those four, one of them appears only one time, and two of them occur quite frequently in the predictors. One of these additional genes, neurofibromin 2 appears in all predictors, and the other thioredoxin appears in 37 of the 50 predictors. The third, homeobox B7 appears six times. Neurofibromin has been associated with tumorigenesis (Reed and Gutmann, 2001). It is believed that thioredoxin may play a role in cancer and Thioredoxin-1 is often associated with aggressive tumor growth (Powis and Montfort, 2001). In a study on multiple carcinigenesis of mouse skin (Chang *et al.*, 1998), Homeobox B7 appears to be expressed at a much lower level than in normal mouse skin. Because this gene only appears in 16% of predictors, this may not be a significant correlation.

## Leukemia Data

Microarray data (Golub *et al.*, 1999) was obtained from patients having two types of leukemia, acute lymphoblastic leukemia (ALL), and acute myeloid leukemia (AML). The data here was taken from bone marrow samples and the samples were of different cell types, for example B or T cells and different patient genders. Each sample was analyzed using an Affymetrix microarrays containing expression levels of 7129 genes. The data was divided into 38 training data points and 34 test points.

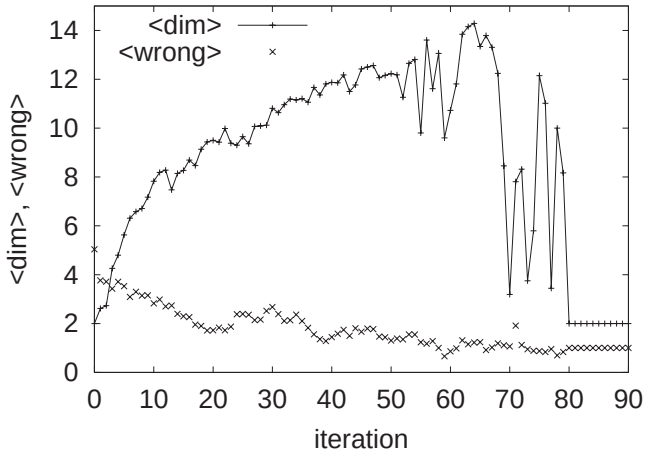


**Fig. 3.** The statistical algorithm using all mutational moves, which includes deletions with an initial pool of 50 genes for the leukemia data (Golub *et al.*, 1999). The average number of dimensions (top curve, solid line with + symbols), the average number that the predictor got wrong (bottom curve, × symbols), as a function of the number of iterations.

Various different replications algorithms were tried with this data: statistical replication algorithm without deletions, and with deletions, and deterministic evolution with different initial pool sizes. The predictors vary in accuracy; there are predictors that make no mistakes and some that make several. There appears to be no way of distinguishing between them short of using the test data. The lack of convergence to near perfect predictors is in agreement with other work on this data set (Furey *et al.*, 2000; Li and Yang, 2002; Golub *et al.*, 1999).

Results from statistical replication with all mutational moves is shown in Figure 3. It shows the results from starting from an initial pool of 50 distinct genes. The average number of dimensions in a predictor rises to more than 14 by iteration 63 and then declines, by iteration 80, to a dimension of only 2. During this evolution, the average number of mistakes made on the test data remains fairly constant at 1. Unlike the SRBCT data, there is no convergence to almost perfect prediction, and the individual predictors have a wide range of different dimensions all giving similar predictive ability. Note that although this is the case for the test data, the method predicts perfectly the training data, through LOOCV. For example for a pool of 50 genes after iteration 20 the test data prediction is perfect, with an average dimension of about 9.

Varying parameters such as the initial number of genes, $n_{top}$, and the method of scoring does not lead to a statistically significant improvement in the average number of mistakes made. Also, as the above cases illustrate, the optimum number of genes in a predictors varies between 3 to 25 depending on parameters. This is consistent with recent work on this data where also no clear cutoff in the number of genes needed for an optimal predictor was also found (Li and Yang, 2002).

### Diffuse large B-cell lymphoma

Recently microarrays in conjunction with supervised learning algorithms were used to study the important problem of Diffuse large B-cell lymphoma (DLBCL), the most common lymphoid malignancy in adults (Shipp *et al.*, 2002). Using 6817 genes from tumor specimens, the authors studied two problems. First, they studied whether their microarray data could be used to distinguish DLBCL from a related B-cell lymphoma, follicular lymphoma (FL). Then they studied if the success or failure of chemotherapy could be predicted from gene expression data of patients.

### DLBCL versus FL

Biopsies from patients before treatment were obtained from 58 patients diagnosed with DLBCL and 19 with FL. Shipp *et al.* used LOOCV to select their prediction algorithm. They found that a 30 gene predictor could correctly classify 71 of 77 tumors (91%).

GESSES was used to analyze the same data using statistical replication with the extra two mutational moves. With different random numbers and different numbers of starting top genes, 77 and 130, GESSES always predicted of 77 out of 77 (100%) of the data correctly. The final predictors ranged in number of genes, from four to 12. The four gene predictor shared three genes in common with those found previously (Shipp *et al.*, 2002).

It should be noted that LOOCV is expected to do better than it would on independent test data (Xing *et al.*, 2001). However the original work (Shipp *et al.*, 2002) did not provide any extra test data, but with 77 subjects it appeared plausible that an independent test could be carried out by splitting the data into two groups, one for test and one for training, to get a more conservative estimate of the predictive value. The data was split into 65 training and a 12 test samples. Half of the test data was DLBCL and the other FL. The predictor converged to one with two wrong and ten correctly classified. The number of final dimensions in the predictor was six. This gives a significance of $P < 1.2 \times 10^{-3}$ compared with random prediction. These numbers are expected to improve with larger data sets.

### DLBCL outcome analysis

Shipp *et al.* went on further to analyze the outcome of chemotherapy. The outcome of 58 patients was divided into two sets, 32 who were 'cured' and 26 who were 'fatal/refractory'. Using a similar analysis to the DLBCL

versus FL work, they used LOOCV to select their best predictor. The best predictor they found had 13 genes, and on LOOCV they found that it could predict 44 out of 58 (76%) correctly.

The same data was analyzed using GESSES utilizing the same parameters as were used above. Starting with the top 130 genes, it was able to find an ensemble of predictors where all outcomes were correctly classified by LOOCV. The number of genes for these 86 predictors ranged from a minimum of 22 to a maximum of 31. Out of these predictors, there were a total of 53 separate genes. Three of these genes are identical to ones in the 13 gene predictor of Shipp *et al.*

### Multiclass diagnosis of Common Tumors

Recent work (Ramaswamy *et al.*, 2001) used microarray data to attempt to distinguish 14 different kinds of tumors. They collected 214 tumor samples spanning these types and analyzed them with an array of different learning algorithms using the expression levels of 16 063 genes. Six different variants of SVM and k nearest neighbor algorithm were tried with different numbers of genes. LOOCV on the 144 subjects used for training data had their best predictor being able to distinguish the correct class on 78% of the samples. When this was tried on the 54 test samples, they also found that it worked on 78% of the test samples. This predictor was an SVM using all 16,063 genes. A slight complication with the analysis is that eight metastatic samples of different kinds were included in the test data. They found that six out of eight of these were identified correctly. The authors suggest that this indicates that many cancers retain their tissue of origin identity throughout their metastatic development.

A variety of separate runs were done on this data using GESSES with statistical replication and the two extra mutational moves. Excluding metastatic samples, the results typically range from 63% correct to 83%. With, metastatic genes included, the results range from 57% (worst with an initial pool of 182) to 80% (best with an initial pool of 273). For example, at the end of a run ($\beta = 681$), while there are many separate predictors (154), 5 predictors classified 12 out of 46 samples incorrectly, 16 predictors made 11 errors, 132 made ten errors, and one predictor made nine errors.

The number of genes used in these predictors ranged from about 40 to 70. The results from LOOCV are considerably higher, typically 92%. As mentioned above, this lower error rate is expected with any method where cross validation is used to select or optimize parameters in a model (Xing *et al.*, 2001). The degree of this bias clearly depends on the details of the algorithm employed. It is expected that this bias is higher with all mutational moves present rather than just gene addition.

## DISCUSSION

From the fact that a large number of different gene combinations perform similarly, and that the data is still quite noisy, one cannot expect to find the unique combination of genes that is optimum for cancer diagnosis. However from a practical point of view, the lack of a unique solution does not present a problem. Any one of the the predictors found for the SRBCT data would be a good starting point for the development of a clinical test based, for example, on antibody assays (He and Friend, 2001). In addition GESSES does not attempt to find a comprehensive set of relevant genes; there could very well be other relevant genes that are not employed in the final predictors.

In the case of SRBCT data (Khan *et al.*, 2001), this method was able to find predictors using fewer than 15 genes that were able to reliably classify test data into one of four groups. Some of the genes found were different than the 96 used originally to do this classification and may be of biological significant. The optimum number of genes to use in a predictor is approximately $12 \pm 2$.

GESSES was also applied successfully to several additional data sets. Leukemia data (Golub *et al.*, 1999), two data sets on Diffuse large B-cell lymphoma (Shipp *et al.*, 2002) and one on the classification of 14 different classes of tumors (Ramaswamy *et al.*, 2001).

For the multiclass tumor data, both GESSES and the initial work of Ramaswamy *et al.* fail to achieve 100% success even with the less conservative measure of LOOCV. The original work gives several reasons for why their experiment is particularly challenging. Among them are the possibility of mis-labeling, the noise in the data, and the small number of examples for each class coupled with the intrinsic biological variation from specimen to specimen. The same remarks are relevant to most of the current microarray data currently available.

It is hoped that using GESSES could help lead to practical uses of microarray data in cancer diagnosis, for example using antibody assays (He and Friend, 2001) from the handful of genes found in this work.

## ACKNOWLEDGMENTS

The author thanks Francoise Chanut, James Conklin, and David Draper for useful discussions.

## REFERENCES

Akaike,H. (1974) A new look at the statistical model identification. *IEEE Trans. Automat. Control*, **19**, 716–723.

Alizadeh,A., Eisen,M., Davis,R., Ma,C., Lossos,I., Rosenwald,A., Boldrick,J., Sabet,H., Tran,T., Yu,X. *et al.* (2000) Distinct types of diffuse large B-cell lymphoma identified by gene expression profiling. *Nature*, **403**, 503–511.

Alon,U., Barkai,N., Notterman,D., Gish,K., Ybarra,S., Mack,D. and Levine,A. (1999) Broad patterns of gene expression revealed by

clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. *Proc. Natl Acad. Sci. USA*, **96**, 6745–6750.

Ben-Dor,A., Friedman,N. and Yakhini,Z. (2000) Scoring genes for relevance. Technical Report AGL-2000-13 Agilent Laboratories.

Brown,M., Grundy,W., Lin,D., Cristianini,N., Sugnet,C., Furey,T. and Jr,M.A. (2000) Knowledge-based analysis of microarray gene expression data by using support vector machines. *Proc. Natl Acad. Sci. USA*, **97**, 262–267.

Ceperley,D. and Kalos,M. (1979) *Monte-Carlo Methods in Statistical Mechanics*. Springer, Berlin.

Chang,P., Kozono,T., Chida,K., Kuroki,T. and Huh,N. (1998) Differential expression of hox genes in multistage carcinogenesis of mouse skin. *Biochem. Biophys. Res. Commun.*, **248**, 749–752.

DeRisi,J., Penland,L., Brown,P., Bittner,M., Meltzer,P., Chen,M.R.Y. and Su,Y. (1996) Use of a cdna microarray to analyse gene expression patterns in human cancer. *Nature Genet.*, **14**, 457–460.

Duda,R. and Hart,P. (1973) *Pattern Classification and Scene Analysis*. New York, Wiley.

Furey,T., Cristianini,N., Duffy,N., Bednarski,D., Schummer,M. and Haussler,D. (2000) Support vector machine classification and validation of cancer tissue samples using microarray expression data. *Bioinformatics*, **16**, 906–914.

Garel,T. and Orland,H. (1990) Guided replication of random chain: a new Monte Carlo method. *J. Phys. A*, **23**, L621–L626.

Getz,G., Levine,E. and Domany,E. (2000) Coupled two-way clustering analysis of gene microarray data. *Proc. Natl Acad. Sci. USA*, **97**, 12079–12084.

Golub,T., Slonim,D., Tamayo,P., Huard,C., Gaasenbeek,M., Mesirov,J., Coller,H., Loh,M., Downing,J., Caligiuri,M. *et al.* (1999) Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science*, **286**, 531–537.

Powis,G. and Montfort,W. (2001) Properties and biological activities of thioredoxins. *Annu. Rev. Pharmacol. Toxicol.*, **41**, 261–295.

Hastie,T., Tibshirani,R., Eisen,M., Brown,P., Ross,D., Scherf,U., Weinstein,J., Alizadeh,A. and Staudt,L. (2000) Shaving: a new class of clustering methods for expression arrays. Technical report Stanford University.

He,Y.D. and Friend,S.H. (2001) Microarrays–the 21st century divining rod? *Nat. Med.*, **7**, 658–659.

Khan,J., Wei,J., Ringner,M., Saal,L., Ladanyi,M., Westermann,F., Berthold,F., Schwab,M., Antonescu,C.R., Peterson,C. and Meltzer,P. (2001) Classification and diagnostic prediction of cancers using expression profiling and artificial neural networks. *Nat. Med.*, **7**, 673–679.

Kohavi,R. and John,G. (1997) Wrapper for feature subset selection. *Artificial Intelligence*, **97**, 273–324.

KP Burnham,D.A. (1998) *Model Selection and Inference*. Springer.

Langley,P. (1994) Selection of relevant features in machine learning. In *Proceedings of the AAAI Fall Symposium on Relevance*. AAAI press.

Li,W. and Yang,Y. (2002) How many genes are needed for a discriminant microarray data analysis? In *Methods of Microarray Data Analysis*. Kluwer Academic, pp. 137–150.

Perou,C., Jeffrey,S., van de Rijn,M., Rees,C., Eisen,M., Ross,D., Pergamenschikov,A., Williams,C., Zhu,S., Lee,J. *et al.* (1999) Distinctive gene expression patterns in human mammary epithelial cells and breast cancers. *Proc. Natl Acad. Sci. USA*, **96**, 9212–9217.

Ramaswamy,S., Tamayo,P., Rifkin,R., Mukherjee,S., Yeang,C., Angelo,M., Ladd,C., Reich,M., Latulippe,E., Mesirov,J., Poggio,T. *et al.* (2001) Multiclass cancer diagnosis using tumor gene expression signatures. *Proc. Natl Acad. Sci. USA*, **98**, 15149–15154.

Reed,N. and Gutmann,D. (2001) Tumorigenesis in neurofibromatosis: new insights and potential therapies. *Trends Mol. Med.*, **7**, 157–162.

Schummer,M., Ng,W., Bumgarner,R., Nelson,P., Schummer,B., Bednarski,D., Hassell,L., Baldwin,R., Karlan,B. and Hood,L. (1999) Comparative hybridization of an array of 21 500 ovarian cdnas for the discovery of genes overexpressed in ovarian carcinomas. *Gene*, **238**, 375–385.

Schwarz,G. (1976) Estimating the dimension of a model. *Ann. Statist.*, **6**, 461–464.

Shipp,M., Ross,K., Tamayo,P., Weng,A., Kutok,J., Aguiar,R., Gaasenbeeck,M., Angelo,M., Reich,M., Pinkus,G. *et al.* (2002) Diffuse large B-cell lymphoma outcome prediction by gene-expression profiling and supervised machine learning. *Nat. Med.*, **8**, 68–74.

Wang,K., Gan,L., Jeffery,E., Gayle,M., Gown,A., Skelly,M., Nelson,P., Ng,W., M,M.S., Hood,L. and Mulligan,J. (1999) Monitoring gene expression profile changes in ovarian carcinomas using cDNA microarray. *Gene*, **229**, 101–108.

Xing,E., Jordan,M. and Karp,R. (2001) Feature selection for high-dimensional genomic microarray data. In *Proceedings of the Eighteenth International Conference on Machine Learning, (ICML2001)*. Morgan Kaufmann.

Zhang,L., Zhou,W., an S.E.d Kern,V.V., Hruban,R., Hamilton,S., Vogelstein,B. and Kinzler,K. (1997) Gene expression profiles in normal and cancer cells. *Science*, **276**, 1268–1272.

Zhu,H., Cong,J., Mamtora,G., Gingeras,T. and Shenk,T. (1998) Cellular gene expression altered by human cytomegalovirus: global monitoring with oligonucleotide arrays. *Proc. Natl Acad. Sci. USA*, **95**, 14470–14475.