GroupAdaBoost for selecting important genes

T. Takenouchi,
Department of Fundamental Statistical Theory, The Institute of Statistical Mathematics, 4-6-7, Minami-Azabu, Minato-ku, Tokyo 106-8569, Japan.

M. Ushijima,
Genome Center, Japanese Foundation for Cancer Research, 1-37-1, Kami-Ikebukuro, Toshima-ku, Tokyo 170-8455, Japan.

S. Eguchi *
Department of Fundamental Statistical Theory, The Institute of Statistical Mathematics, 4-6-7, Minami-Azabu, Minato-ku, Tokyo 106-8569, Japan.
Phone: +81-3-3446-1501
Fax: +81-3-5421-8728

**Key words:** Microarray, classification, Boosting method.
ABSTRACT

This paper discusses the challenging problem for predicting the association of the disease status of an individual with observed gene expressions based on learning a training dataset of gene expressions. This difficult problem has been addressed as “$p \gg n$” problem with the number $p$ of tested genes and the number $n$ of individuals in a microarray experiment for observing gene expressions. We propose a new statistical method for prediction without any complicated pre-procedure by extending AdaBoost to GroupAdaBoost in the context of statistical learning theory.

Any usual method for predicting the genetic causes of diseases is apt to over-learn from any particular training dataset. The proposed method, called GroupAdaBoost, aims to regularize the learning step from this excess information. We observed that GroupAdaBoost gave a robust performance for the excess number of tested genes in situations where the usual methods have failed to yield any sensible result. In several real datasets, which are publicly available from web-pages, we compared the analysis of results from the proposed method and others. We considered simple reasoning of the superiority of GroupAdaBoost to conventional methods in a theoretical discussion and have provided a small scale simulation study.
1 INTRODUCTION

Microarray technology is a recently developed biotechnology that allows us to monitor several thousand gene expressions in a single experiment. This poses a serious problem in that this technology leads us beyond the usual assumptions of conventional statistical analysis. There are two main objectives with microarray analysis. One is discriminant analysis (supervised learning), which aims to predict the unknown class label of a new individual from a monitored gene expression profile. The other is the identification of those marker genes (variable selection) that characterize the different disease classes. For the discriminant analysis, there are many proposed methods such as Fisher discriminant analysis, support vector machine and machine learning methods as bagging or Boosting.

Boosting method constructs a classification machine by combining a lot of weak classification machines and its learning process is implemented by sequentially reweighting all the examples according to classification results. Ben-Dor et al. (2000) and Dudoit et al. (2002) compared the typical Boosting algorithm AdaBoost with other methods and reported that AdaBoost does not yield many impressive results. Dettling and Bühlmann (2003) applied LogitBoost (see Friedman et al. (2000)), which is a variation of AdaBoost, and obtained some results for publicly available datasets.

The serious problem with classification from gene expression profiles is that the sample size $n$ is generally too small relative to the number $p$ of monitored genes. Many gene expressions are considered to be non-differentially expressed across the sample and do not give any important information. This superabundance of information on gene expression makes it difficult to get any useful predictive results. Moreover, there are several solutions for prediction, which situation inhibits the building of medical knowledge from the analysis. That $p$ is extremely huge makes conventional classification algorithms in-executable: the training dataset often can be completely learned with the training error 0 even when any gene expressions do not have information. To avoid those problem, conventional methods apply a
variable selection method to expression data for the reduction of gene expression and then only those selected gene expressions are used to construct a classification machine for producing the diagnosis. However, this can be problematic because the set of all the genes involves a considerable number of non-informative genes and the pre-procedure often falls into a difficult situation in which there are no evident separations between important genes and unimportant genes. Once a gene expression has been truncated by the pre-procedure, the information is never utilized in the prediction procedure. AdaBoost can be applied to microarray data without any pre-selection but does not sufficiently catch important expressions. Because the usual Boosting method selects the best representative expression as a classification machine in each learning step, the algorithm tends to look over important genes having similar expressions over individuals. For example, in our data analysis, AdaBoost selects only 15 genes for classification of the well-known publicly dataset, Leukemia and this result by AdaBoost gives less information than that of Golub et al. (1999). However in their naive analysis, the set of 50 genes was suggested to have an association with leukemic diseases.

To overcome the above difficulty, we propose a new Boosting algorithm called GroupAdaBoost where the similarity between gene expressions is positively incorporated into AdaBoost. GroupAdaBoost deals with genes having similar performance as a group and enables us to jointly execute a selection of important genes and design of a classification machine without pre-selection. Consequently, GroupAdaBoost can select important genes that are highly influential for the classification of diseases.
2 GroupAdaBoost

2.1 Framework of Boosting

We focus on the binary classification problem. See McLachlan (1992) for extensive discussion for classification methods. Let us assume a set of training dataset \( D = \{ (x_i, y_i) : i = 1, \cdots, n \} \), where \( x \in \mathbb{R}^p \) is an input vector and \( y \in \{1, -1\} \) is a class label. In our context the input vector \( x \) has expression profiles of \( p \) monitored genes as the components, \( y \) represents a disease status of an individual having expression \( x \). Typically, the sample size \( n \) is about several tens, and there are several thousands or larger of monitored genes, \( p \).

The aim of the classification problem is to construct a classification machine \( F : x \rightarrow \mathbb{R} \) that minimizes a misclassification error \( \Pr(\text{sgn}(F(x)) \neq y) \).

In the context of Boosting, we intentionally use only weak classification machines, \( f(x) \in \{1, -1\} \). We employ a decision stump as a weak classification machine:

\[
f(x; j, b) = \text{sgn}(x_j - b),
\]

where \( j = 1, \cdots, p \) and \( b \) is a threshold value in a range of the \( j \)-th gene expression profile \( x_j \). This implies that the classification machine \( f(x; j, b) \) determines the label for an input \( x \) whether the quantity of the expression level of \( j \)-th gene, \( x_j \), is larger than the threshold \( b \) or not. The decision stump has a preferable property for analyzing gene expression data: the decision result is dependent on only ranking of the expression levels \( x_j \) and thus the stump is invariant for any kind of monotone transformation in preprocessing such as centralization or log transformation. Let \( \mathcal{F} \) be a set of weak classification machines.

\[
\mathcal{F} = \{ f(x; j, b) | j = 1, \cdots, p, b \in \mathbb{R} \}.
\]

The Boosting method aims to construct a strong classification machine \( \text{sgn}(F_T(x)) \) by linearly combining weak classification machines as

\[
F_T(x) = \sum_{t=1}^{T} \alpha_t f(x; j_t, b_t), \tag{1}
\]
where $j_t$ is the gene number related in the $t$-step and $b_t$ is the optimized threshold. The derivation of $(\alpha_t, j_t, b_t)$ will be given in a subsequence discussion. This discriminant function is viewed as a weighted majority vote of stumps associated with $T$ gene expressions and an absolute value of $F_T(x)$ represents a degree of confidence concerned with the classification of $x$. The prediction of label is decided by the sign of $F_T(x)$. A typical Boosting algorithm is AdaBoost, which is derived from a sequential minimization of the exponential loss function

$$L_{\text{exp}}(F) = \frac{1}{n} \sum_{i=1}^{n} \exp(-F(x_i)y_i).$$

We note that, if in the $i$-th example $F(x_i)$ has the same sign as $y_i$, it gives less influence for the exponential loss than that with distinct sign. Thus the minimization of $L_{\text{exp}}(F)$ with respect to $F$ qualitatively aims at matching the signs of $y_i$ and $F(x_i)$ over the training dataset. AdaBoost works efficiently for usual classification problems but is not appropriate for gene expression datasets. In the present dataset, there are too many feature variables comparing with $n$ and some of these have similar information. Because AdaBoost selects only one variable in one step, other variables with similar information are abandoned. To overcome this problem, we propose GroupAdaBoost, as a simple modification of AdaBoost in a learning step. We will demonstrate its performance using real datasets, and discuss the theoretical considerations.

### 2.2 GroupAdaBoost algorithm

In this section, we introduce the algorithm GroupAdaBoost. For this, we overview the learning step of algorithm of AdaBoost, which consists of the following three procedures and is proceeded by a sequentially updated weight distribution for examples. At first, the algorithm selects the best weak classification machine having minimum weighted error rate. Secondly, a coefficient for the selected classification machine is calculated according to the performance of the selected machine. Thirdly, the weight distribution is updated as putting a high weight into only misclassified examples. These three procedures are updated and finally provide the discriminant function (1). Let
us define GroupAdaBoost we propose. It follows almost the same procedures except for the first procedure. AdaBoost selects the best machine in terms of weighted error rates, whereas GroupAdaBoost selects a group of $G$ classification machines as follows.

GroupAdaBoost($G$)

1. Set the initial condition as $w_1(i) = \frac{1}{n}$ and $F_0(x) = 0$.

2. For $t = 1, \cdots, T$.

   (a) Select a weak machine for the $j$-th gene as

   $$f_t(x_j; b_j) = \arg\min_{f \in \mathcal{F}_j} \varepsilon_t(f),$$

   where $\mathcal{F}_j = \{f(x_j; b); b \in \mathbb{R}\}$ is a set of weak classification machines associated with $j$-th gene and $\varepsilon_t(f)$ is a weighted error rate,

   $$\varepsilon_t(f) = \sum_{i=1}^{n} w_t(i) I(f(x_i) \neq y_i).$$

   Note that the threshold value $b_j$ depends on the step number $t$. From a sequence of weak machines $\{f_t(x_1; b_1), \cdots, f_t(x_p; b_p)\}$, extract $G$ weak classification machines in the order of their weighted error rate, with the smallest first,

   $$\{f_t(x_{(1)}; b_{(1)}), \cdots, f_t(x_{(G)}; b_{(G)})\},$$

   where the subscript $(g)$ denotes the gene number of the $g$-th smaller weighted error rate. Thus, this is a group of the $G$ best-weighted error rates.

   (b) For the extracted weak classification machine, $f_t(x_{(g)}; b_{(g)})(g = 1, \cdots, G)$, calculate the coefficient

   $$\alpha_{t,(g)} = \frac{1}{2} \log \frac{1 - \varepsilon_t(f_t(\cdot; b_{(g)}))}{\varepsilon_t(f_t(\cdot; b_{(g)}))},$$

   and construct the $t$-th machine as

   $$\bar{f}_t(x) = \frac{1}{G} \sum_{g=1}^{G} \alpha_{t,(g)} f_t(x_{(g)}; b_{(g)}).$$
(c) Update a weight distribution as

\[ w_{t+1}(i) = \frac{w_t(i) \exp(-\tilde{f}_t(x_i)y_i)}{Z_t}, \]

where

\[ Z_t = \sum_{i=1}^{n} w_t(i) \exp(-\tilde{f}_t(x_i)y_i), \]

and update the discriminant function as

\[ F_t(x) = F_{t-1}(x) + \tilde{f}_t(x). \]

3. Output the function

\[ F_T(x) = \sum_{t=1}^{\tau} \tilde{f}_t(x). \]

If we set \( G = 1 \), then GroupAdaBoost reduces to the usual AdaBoost analysis. In the first procedure, GroupAdaBoost can be expected to catch a group of classification machines having similar properties or equivalent genes comparable with the best machine. The constant, \( G \), is typically determined to be a number required by medical information and the number within several tens is appropriate in our analysis. Alternatively, we can choose classification machines by other measures, for example, from a range of weighted error rates. The coefficient \( \alpha_{t,(g)} \) calculated in the procedure (b) is the same with AdaBoost and becomes higher as the weighted error rate gets lower. The discriminant function, \( \tilde{f}_t(x) \) of step \( t \) is constructed for the weighted majority vote. The sign of \( \tilde{f}_t(x) \) means a predicted label for the input \( x \), and the absolute value of \( \tilde{f}_t(x) \) represents a confidence of classification. In the procedure (c), the weight distribution is updated according to classification results and its confidences. A weight of an example with a high confidence is exponentially updated but its update is moderated if \( \tilde{f}_t(x) \) has a low confidence. By this update rule, GroupAdaBoost sequentially reinforces the discriminant function \( F_t(x) \). As a result, GroupAdaBoost jointly executes accurate classification and efficient correction of important gene expressions.

### 2.3 Loss function related with GroupAdaBoost

We now consider a relation between the GroupAdaBoost algorithm and the exponential loss function (2). Procedures of GroupAdaBoost are derived from
a approximate minimization of the exponential loss function while procedures of AdaBoost are derived from exactly sequential minimization. Let us assume that we obtain the discriminant function \( F_{t-1}(x) \) and consider an update from \( F_{t-1}(x) \) to \( F_t(x) \), where

\[
F_t(x) = F_{t-1}(x) + \frac{1}{G} \sum_{g=1}^{G} \alpha_{t,(g)} f_t(x_{(g)}; b_{(g)}).
\]

(3)

From the convexity of the exponential function, we obtain the following inequality:

\[
L_{\exp}(F_t) \leq \frac{1}{G} \sum_{g=1}^{G} L_{\exp}(F_{t-1} + \alpha_{t,(g)} f_t(x_{(g)}; b_{(g)})) \leq L_{\exp}(F_{t-1}).
\]

(4)

This shows that the loss function monotonically decreases by the update (3). A minimizer of the exponential function is equivalent to the Bayes rule which is the optimal discriminant function and minimizes the naive error rate. See Murata et al. (2004) for detailed discussion. GroupAdaBoost approximately minimizes the exponential loss function that is updated by a set of weak classification machines. Note that GroupAdaBoost does not directly minimize \( L_{\exp}(F) \).

### 2.4 Score of a gene

When the discriminant function \( F_T(x) \) is obtained, we define a score for \( x_j \) associated with the \( j \)-th gene as

\[
\frac{1}{T} \sum_{t=1}^{T} \sum_{g=1}^{G} I((g) = j) \alpha_{t,(g)}.
\]

(5)

Note that the number \((g)\) defined in step (a) of the algorithm in section 2.2 implicitly depends on \( t \). The score value implies the contribution of classification machines associated with the \( j \)-th gene expression per one step or total confidence of \( j \)-th gene expression. In section 3.5, we will discuss a selection of important genes based on the score.
2.5 Choice of the learning step number

GroupAdaBoost and any other Boosting method, including LogitBoost, are apt to over-fit the training dataset unless the algorithm is stopped at an appropriate step. See Takenouchi and Eguchi (2004). Thus, the number of learning steps \( T \) should be estimated. If we had sufficient examples, we could set aside a test dataset and use it to assess the performance of a method. Now the sample size \( n \) of gene expression dataset is typically small compared with \( p \), we employ a \( K \)-fold cross validation technique. If we set \( K = n \), this reduces to the leave one out cross validation employed in many data analysis. But the leave one out cross validation does not work well as an estimator of the generalization error, we use another value of \( K \), typically 10 as in Ambroise and McLachlan (2002).

First, we divide the training dataset into \( K \) roughly equal-sized sets \( D_1, \ldots, D_K \) in which each \( D_k \) is a subset of \( D \) and satisfies \( D = D_1 \cup \cdots \cup D_K \) and \( D_j \cap D_k = \emptyset \) for any different \( j, k \in \{1, \ldots, K\} \). Second, we run GroupAdaBoost on the dataset without \( D_k \) and construct the classification machines \( F_t^{(-k)}(x) \) \((t = 1, \ldots, T)\). We calculate the misclassification rate \( \epsilon(F_t^{(-k)}; D_k) \) on \( D_k \) and do this for \( k = 1, \ldots, K \). Finally, we compute the cross validation error at step \( t \) by averaging the \( K \) estimates of misclassification rate as

\[
\epsilon(t) = \frac{1}{K} \sum_{k=1}^{K} \epsilon(F_t^{(-k)}; D_k).
\]

Note that, if we set \( K = n \), the above method means the leave one out cross-validation. An optimal learning step is determined as a point that minimizes \( \epsilon(t) \). Figure 1 shows a \( k \)-th procedure of \( K \)-fold cross validation described above.

Insert Figure 1.

2.6 Experiment with a synthetic dataset

In this subsection, we investigate the performance of GroupAdaBoost with a synthetic dataset. In particular, we want to investigate the relationship
between the number of groups, \( G \), and a number of feature vectors giving important information. Assume that the feature vector \( \mathbf{x} \) is uniformly distributed on \([-1, 1]^p\) and the conditional probability of \( y \) is in the logistic model,

\[
p(y|\mathbf{x}) = \frac{1}{1 + \exp(-2yF^*(\mathbf{x}))},
\]

where \( F^*(\mathbf{x}) = F^1(\mathbf{x}) + F^0(\mathbf{x}) \) and

\[
F^1(\mathbf{x}) = \sum_{s=1}^{u} 10x_s,
\]

\[
F^0(\mathbf{x}) = \sum_{s=u+1}^{p} \frac{0.1}{p} x_s.
\]

Under the above setting, the Bayes rule of (6) is \( F^*(\mathbf{x}) \) and is mainly influenced by \( F^1(\mathbf{x}) \) because the rule is determined by the sign of \( F^*(\mathbf{x}) \). See McLachlan (1992) for detail discussion for Bayes rule. Now, let us consider the relationship between \( u \) and \( G \). We generated 20 sets of training datasets containing 50 examples and a test dataset containing 2000 examples for \( p = 50, 1000 \), \( u = 2, 10, 20, 50 \). The number of learning steps \( T \) is estimated by the 10-fold cross-validation with each training dataset, and so we can calculate the mean test error rates. Figure 2 shows the result of this case, \( p = 50 \) and Figure 3 shows the case of \( p = 1000 \). Differences in the cross-validation error rate between AdaBoost and GroupAdaBoost for fixed \( p \) and \( G \) are shown in each figure. The level 0 indicates AdaBoost; if lines are under the level 0, then GroupAdaBoost is superior to AdaBoost. If we appropriately choose \( G \), GroupAdaBoost is superior to AdaBoost. We could observe that the number \( G \) that minimizes the cross-validation error is near to the number \( u \) of influential feature variables.

Insert Figure 2.
Insert Figure 3.
3 RESULTS

We applied GroupAdaBoost to three publicly available real datasets. The test error rates were estimated from the 10-fold cross-validation. Note that the stopping parameter $T$ was also determined by the 10-fold cross-validation for a dataset without validation examples. Consequently, we performed two sequences of cross-validation: one was to estimate the generalization performance of GroupAdaBoost, and the other was to estimate the optimal stopping point $T$.

3.1 Leukemia

We explored the performance of our method on a leukemia dataset. This dataset contains gene expression data from patients suffering from acute lymphoblastic leukemia (ALL) or acute myeloid leukemia (AML). The dataset consisted of 72 observations and a feature vector $x$ containing 7129 variables. More information about this dataset can be found in Golub et al. (1999). The averaged training error and 10-fold cross-validation error rate against $G$ are shown in Figure 4.

Insert Figure 4.

3.2 Colon

This dataset contains 2000 gene expressions of 62 patients, with 40 tumor and 22 normal colon tissues, measured using Affymetrix gene chip technology. This dataset is available at http://microarray.princeton.edu/oncology/. Figure 5 shows the result.

Insert Figure 5.

3.3 Estrogen and Nodal

This dataset monitors 7129 genes in 49 breast tumor samples. The datasets are available at http://mgm.duke.edu/genome/dna_micro/work/ and were
obtained using Affymetrix gene chip technology. In these datasets, there are two differently labeled variables. One describes the status of the estrogen receptor (ER), in which 25 samples are positive (ER+) and 24 are negative (ER−). The other describes lymph node (LN) status, which is an indicator for the metastatic spread of the tumor and a very important risk factor for disease outcome. Here, 25 samples are positive (LN+) and the remaining 24 samples are negative (LN−). The results for this dataset are shown in Figure 6 and Figure 7 shows the results for the nodal dataset.

Insert Figure 6.
Insert Figure 7.

3.4 Discussion

For the leukemia, colon, and estrogen datasets, the mean training error is minimized near $G = 1$, which supports AdaBoost or GroupAdaBoost results in a small $G$, whereas the validation error rate was minimized at a relatively large $G$. We observed that AdaBoost apt to fall into over-learning; GroupAdaBoost escapes from the over-learning thanks to the improved step in the algorithm. Note that an extremely large $G$ also loads to over-learning and $G$ of several tens seems to be appropriate. GroupAdaBoost worked much better than AdaBoost in the sense of the estimated test error and this result is comparable to that of Dettling and Bühlmann (2003), using the intensive pre-selection of genes. For the leukemia dataset, GroupAdaBoost wrongly classified only one example through the validation process. Only for the nodal dataset, AdaBoost gave the best performance in terms of validation error rate but the averaged training error was minimized by GroupAdaBoost with a large value of $G$. By monitoring the learning process, we suspect that the estimate of the stopping parameter using the validation technique did not work well.
3.5 Selection of important genes

Another objective of this paper was to select important genes as the weak learners. To verify the importance of selected genes, we refer to the original papers. Here, we discuss only the leukemia and estrogen datasets because there was no detailed information for important genes for the colon and nodal datasets in the original papers.

Tables 1 and 2 show the list of the top 30 genes for the leukemia and estrogen datasets identified by GroupAdaBoost with the highest score. The score for the classification machine is defined in (5) and we averaged over 10 classification machines, obtained by 10-fold cross validation.

For the leukemia dataset, we refer to Golub et al. (1999) who showed 50 genes as informative genes which were most closely correlated with AML-ALL class distinction. Twelve genes in Table 1 are listed in Golub et al. (1999). In particular, the genes CD33 and MB-1 are known to be useful in distinguishing lymphoid from myeloid lineage cells, and so the genes likely to be associated with ALL-AML can be distinguished.

For the estrogen dataset, we refer to West et al. (2001) who showed the list of the 40 genes most highly correlated with ER status. Fifteen genes in Table 2 are also listed in Table 1 of West et al. (2001), and 8 genes are in the protein synthetic pathway of ER or are involved in ER itself. For example, pS2, LIV-1, and GATA3 have already been reported to have a relationship with ER status in several articles.

Therefore, we can confirm that many important genes are included in our lists through the use of GroupAdaBoost. Thus, important genes can be selected accurately as those effective for sample classification.

Insert Table 1.
Insert Table 2.
4 CONCLUSIONS

We have proposed the new algorithm GroupAdaBoost for analyzing microarray problems and have explored the performance of the algorithm with publicly available real datasets and synthetic datasets. Using the adaptive selection of tuning parameters, GroupAdaBoost was shown to overcome the sensitivity of the pre-selection of genes and to have a high generalization ability. Additionally, the grouping and selection of weak classification machines using GroupAdaBoost worked effectively for the identification of important genes.
REFERENCES


West, M., Blanchette, C., Dressman, H., Huang, E., Ishida, S., Spang, R.,
Zuzan, H., Olson, J., Marks, J. and Nevins, J. 2001. Predicting the clinical
status of human breast cancer by using gene expression profiles. *PNAS*,
98, 11462–11467.
Figure 1: The outline of $K$-fold cross validation.
Figure 2: The difference in the mean cross-validation error rate against $G$. The level 0 corresponds to AdaBoost. The dataset was generated by (6) and the dimension $p$ of feature vector was 50.
Figure 3: The difference in the mean cross-validation error rate against $G$. The level 0 corresponds to AdaBoost. The dataset was generated by (6) and the dimension $p$ of feature vector was 1000.
Figure 4: The mean of 10-fold cross-validation error rate against $G$ for the leukemia dataset. The point $G = 1$ indicates AdaBoost.
Figure 5: The mean of 10-fold cross-validation error rate against $G$ for the colon dataset. The point $G = 1$ indicates AdaBoost.
Figure 6: The mean 10-fold cross-validation error rate against $G$ for the estrogen dataset. The point $G = 1$ indicates AdaBoost.
Figure 7: The mean of 10-fold cross-validation error rate against $G$ for the nodal dataset. The point $G = 1$ indicates AdaBoost.
Table 1: Top 30 genes associated with disease from the leukemia dataset

<table>
<thead>
<tr>
<th>GenBank ID</th>
<th>LocusLink Symbol</th>
</tr>
</thead>
<tbody>
<tr>
<td>D88422</td>
<td>CSTA</td>
</tr>
<tr>
<td>J05243</td>
<td>SPTAN1</td>
</tr>
<tr>
<td>M11722</td>
<td>DNTTT</td>
</tr>
<tr>
<td>M23197*</td>
<td>CD33 (CD33)</td>
</tr>
<tr>
<td>M27891*</td>
<td>CST3 (Cystatin C)</td>
</tr>
<tr>
<td>M31166</td>
<td>TSG-14</td>
</tr>
<tr>
<td>M31303*</td>
<td>STMN1 (Op18)</td>
</tr>
<tr>
<td>M63379</td>
<td>TRPM-2</td>
</tr>
<tr>
<td>M68891</td>
<td>GATA2</td>
</tr>
<tr>
<td>M84526*</td>
<td>ADN (Adipsin)</td>
</tr>
<tr>
<td>M92287*</td>
<td>CCND3 (CCND3)</td>
</tr>
<tr>
<td>M96326*</td>
<td>AZU1 (Azurocidin)</td>
</tr>
<tr>
<td>U05259*</td>
<td>CD79A (MB-1)</td>
</tr>
<tr>
<td>U46499</td>
<td>MGST1</td>
</tr>
<tr>
<td>U88047</td>
<td>DRIL1</td>
</tr>
<tr>
<td>X62320</td>
<td>GRN</td>
</tr>
<tr>
<td>X62654</td>
<td>CD63</td>
</tr>
<tr>
<td>X90858</td>
<td>UPP1</td>
</tr>
<tr>
<td>X95735*</td>
<td>ZYX (Zyxin)</td>
</tr>
<tr>
<td>Y07604</td>
<td>NME4</td>
</tr>
<tr>
<td>J02783</td>
<td>P4HB</td>
</tr>
<tr>
<td>U22376*</td>
<td>MYB (C-myb)</td>
</tr>
<tr>
<td>L07807</td>
<td>DNM1</td>
</tr>
<tr>
<td>L09209</td>
<td>APLP2</td>
</tr>
<tr>
<td>M83652*</td>
<td>PFC (Properdin)</td>
</tr>
<tr>
<td>M83667</td>
<td>CEBPD</td>
</tr>
<tr>
<td>X85116*</td>
<td>STOM (Epb72)</td>
</tr>
<tr>
<td>U49020</td>
<td>MEF2A</td>
</tr>
<tr>
<td>L11672</td>
<td>ZNF91</td>
</tr>
<tr>
<td>M31523*</td>
<td>TCF3 (E2A)</td>
</tr>
</tbody>
</table>

*The “*” marked genes are also listed in 50 informative genes in Golub et al. (1999). Gene names or symbols used in Golub et al. (1999) are written in parentheses.
Table 2: Top 30 genes associated with disease from the estrogen dataset

<table>
<thead>
<tr>
<th>GenBank ID</th>
<th>LocusLink Symbol</th>
</tr>
</thead>
<tbody>
<tr>
<td>D38437*</td>
<td>PMS2L3 (PMS2L3)</td>
</tr>
<tr>
<td>J03778**</td>
<td>MAPT (MAPT)</td>
</tr>
<tr>
<td>J03827</td>
<td>NSEP1</td>
</tr>
<tr>
<td>L08044**</td>
<td>TFF3 (TFF3)</td>
</tr>
<tr>
<td>L12535</td>
<td>RSU-1</td>
</tr>
<tr>
<td>L17131</td>
<td>HMGA1</td>
</tr>
<tr>
<td>L26336</td>
<td>HSPA2</td>
</tr>
<tr>
<td>L40401</td>
<td>ZAP128</td>
</tr>
<tr>
<td>M29874*</td>
<td>CYP2B6 (CYP2B)</td>
</tr>
<tr>
<td>M33493</td>
<td>TPSB2</td>
</tr>
<tr>
<td>M99701</td>
<td>TCEAL1/TCEAL3</td>
</tr>
<tr>
<td>U05340</td>
<td>CDC20</td>
</tr>
<tr>
<td>U07919</td>
<td>ALDH1A3</td>
</tr>
<tr>
<td>U09770*</td>
<td>CRIP1 (CRIP1)</td>
</tr>
<tr>
<td>U22376**</td>
<td>MYB (MYB)</td>
</tr>
<tr>
<td>U41060**</td>
<td>SLC39A6 (LIV-1)</td>
</tr>
<tr>
<td>U42408</td>
<td>LAD1</td>
</tr>
<tr>
<td>U79293*</td>
<td>Clone 23948 (N/A)</td>
</tr>
<tr>
<td>X03635**</td>
<td>ESR1 (ESR1)</td>
</tr>
<tr>
<td>X13238*</td>
<td>COX6C (COX6C)</td>
</tr>
<tr>
<td>X17059*</td>
<td>NAT1 (NAT1)</td>
</tr>
<tr>
<td>X52003**</td>
<td>TFF1 (pS2)</td>
</tr>
<tr>
<td>X52947</td>
<td>GJA1</td>
</tr>
<tr>
<td>X55037**</td>
<td>GATA3 (GATA3)</td>
</tr>
<tr>
<td>X58072**</td>
<td>GATA3 (GATA3)</td>
</tr>
<tr>
<td>X76180</td>
<td>SCNN1A</td>
</tr>
<tr>
<td>X83425</td>
<td>LU</td>
</tr>
<tr>
<td>X87176</td>
<td>HSD17B4</td>
</tr>
<tr>
<td>X87212*</td>
<td>CTSC (CTSC)</td>
</tr>
<tr>
<td>Z48633</td>
<td>N/A</td>
</tr>
</tbody>
</table>

*The marked genes are in the 40 genes list which are correlated with ER status, and the "**" marked genes are in the protein synthetic pathway of ER or are involved in ER itself. Gene symbols used in West et al. (2001) are written in parentheses.