

Virtual Gene: a Gene Selection Algorithm for Sample Classification on Microarray Datasets

Xian Xu and Aidong Zhang

{xianxu|azhang}@cse.buffalo.edu

State University of New York at Buffalo, Buffalo, NY 14260, USA

Abstract. Gene Selection is one class of most used data analysis algorithms on microarray dataset. The goal of gene selection algorithms is to filter out a small set of informative genes that best explains experimental variations. Traditional gene selection algorithms are mostly single-gene based. Some discriminative scores are calculated and sorted for each gene. Top ranked genes are then selected as informative genes for further study. Such algorithms ignore completely correlations between genes, although such correlations is widely known. Genes interact with each other through various pathways and regulative networks. In this paper, we propose to use, instead of ignoring, such correlations for gene selection. Experiments performed on three public available datasets show promising results.

1 Introduction

Microarray experiments enable biologists to monitor expression levels of thousands of genes or ESTs simultaneously [1, 5, 12]. Short sequences of genes or ESTs tagged with fluorescent materials are printed on a glass surface. The slice is then exposed to sample solution for hybridization (base-pairing). mRNA molecules are expected to hybridize with short sequences matching part of their complement sequences. After hybridization the slice is scanned and goes through various data processing steps including image processing, quality control and normalization [4]. The resulting dataset is a two dimensional array with thousands of rows (genes) and tens of columns (experiments). Element at i^{th} row and j^{th} column in such an array is the expression level measure for gene i in experiment j . When tissue samples used in the experiments are labeled (e.g., sample is cancer tissue or normal tissue), sample classification can be performed on such dataset. New samples are classified based on their gene expression profiles.

Such dataset poses special challenge for pattern recognition algorithms. The main obstacle is the limited number of samples due to practical and financial concerns. This results in the situation where the number of features (or genes) well outnumbers the number of observations. The term “curse of dimensionality” and “peaking phenomenon” are coined in the machine learning and pattern recognition community, referring to the phenomenon that inclusion of excessive features may actually degrade the performance of a classifier if the number of training examples used to build the classifier is relatively small compared to the number of features [9]. Typical treatment is to reduce the dimensionality of feature space before classification using feature extraction and feature selection. Feature extraction algorithms create new features based on transformation and/or combination of original features while feature selection algorithms aim

to select a subset of original features. Techniques like PCA and SVD have been used to create salient features [6, 7] for sample classification on microarray datasets. Feature selection, or in our case, gene selection generates a small set of informative genes, which not only leads to better classifiers, but also enables further biological investigation.

In order to find the optimal subset of features that maximizes some feature selection criterion function (we assume the higher value the criterion function, the better the feature subset), straightforward implementation would require evaluation of the criterion function for each feature subset, which is a classic NP hard problem. Various heuristics and greedy algorithms have been proposed to find sub-optimal solutions. Assuming independence between features, one attempt is to combine small feature subsets with high individual scores. This heuristic is widely used for gene selection. A class of gene selection algorithms calculates discriminative scores for individual genes and combines top ranked genes as selected gene set. We refer to this class of algorithms single gene based algorithms. Various discriminative scores have been proposed, including statistical tests (t-test, F-test) [3], non-parametric tests like TNoM [2], mutual information [15, 16], S2N ratio (signal to noise ratio) [5], extreme value distribution [11] and SAM [13] etc. Although simple, this class of algorithms is widely used in microarray data analysis and proven to be effective and efficient.

However, the assumption of independence between genes over simplifies the complex relationship between genes. Genes are well known to interact with each other through gene regulative networks. As a matter of fact, the common assumption of cluster analysis on microarray dataset [10] is that co-regulated genes have similar expression profiles. Bø [3] proposed to calculate discriminant scores for a pair of genes instead of each individual gene. Several of recent researches on feature selection especially gene selection [8, 14, 16] took into consideration the correlation between genes explicitly by limiting redundancy in resulting gene set. Heuristically, selected genes need to first have high discriminative scores individually and secondly not correlate much with genes that have already been selected. Generic feature selection algorithms like SFFS (sequential forward floating selection), SBFS (sequential backward floating selection), etc. have also been used for selecting informative genes from microarray datasets.

In this paper, we propose a totally different approach. Instead of trying to get rid of correlation in the selected gene set, we examine whether such correlation itself is a good predictor of sample class labels. Our algorithm is a supervised feature extraction algorithm based on new feature “virtual gene”. “Virtual genes” are linear combination of real genes on a microarray dataset. Top ranked “virtual genes” are used for further analysis, e.g., sample classification. Our experiments with three public available datasets suggest that correlations between genes are indeed very good predictors of sample class labels. Unlike typical feature extraction algorithms, the “virtual gene” bears biological meaning: the weighted summation or difference of expression levels of several genes.

The rest of this paper is organized as follows. We present our “virtual gene” algorithm in Sec. 2. Both a synthetic and a real example from Alon dataset [1] are given. In Sec. 3, extensive experimental results are reported using three public available datasets. We give our conclusion and future work of this paper in Sec. 4.

2 Virtual Gene: A Gene Selection Algorithm

2.1 Gene Selection For Microarray Experiments

In this section we formalize the problem of gene selection for microarray datasets. Symbols used in this section will be used throughout this paper.

Let \mathcal{G} be the set of all genes that are used in one study, \mathcal{S} be the set of all experiments performed, \mathcal{L} be the set of sample class labels of interest. We assume $\mathcal{G}, \mathcal{S}, \mathcal{L}$ are fixed for any given study. Let $n = |\mathcal{G}|$ be the total number of genes, $m = |\mathcal{S}|$ be the total number of experiments and $l = |\mathcal{L}|$ be the total number of class labels. Gene expression dataset used in our study can be defined as $\mathcal{E} = (\mathcal{G}, \mathcal{S}, \mathcal{L}, E)$, where \mathcal{L} is a list of sample class labels such that for $s \in \mathcal{S}$, $L(s) \in \mathcal{L}$ is the class label for sample s ; expression matrix E is an $n \times m$ matrix of real numbers. $E(g, s)$, where $g \in \mathcal{G}, s \in \mathcal{S}$, is the expression level of gene g in experiment s . For simplicity of presentation, we use a subscripting scheme to refer to elements in \mathcal{E} . Let $\mathcal{E}(G, S) = (G, S, L', E')$ where $G \subseteq \mathcal{G}$ and $S \subseteq \mathcal{S}$. L' is a sublist of \mathcal{L} containing class labels for samples S , E' is the subarray of E containing values of expression levels for genes G and experiments S . We also write $E' = E(G, S)$. We further use $L(S)$ to denote a list of class labels for the set of experiments S . Given training expression data $\mathcal{E}_{train} = (\mathcal{G}, \mathcal{S}_{train}, \mathcal{L}_{train}, E_{train})$, the problem of sample classification is to build a classifier that predict L_{new} for new experiment result $\mathcal{E}_{new} = (\mathcal{G}, \mathcal{S}_{new}, \mathcal{L}_{missing}, E_{new})$. $\mathcal{L}_{missing}$ indicates that the class labels of samples \mathcal{S}_{new} have not been decided yet. The problem of gene selection is to select a subset of genes $G' \subset \mathcal{G}$ based on \mathcal{E}_{train} so that classifiers built from $\mathcal{E}_{train}(G', \mathcal{S}_{train})$ predict L_{new} more accurately than classifiers built from \mathcal{E}_{train} . We use n' as the number of features being selected, or $n' = |G'|$.

2.2 An Example

Consider the following two examples as shown in Fig. 1. In each figure, the expression levels of two genes are monitored across several samples. Samples are labeled either cancerous or normal. In both cases, the expression levels of the selected genes vary randomly across the sample classes. However, their correlation is a good predictor of class labels. *Virtual gene* expression level is obtained using the Def. 2. In the case of Alon [1] dataset, the expression levels of H09719 are generally higher than that of L07648 in cancer tissues. In normal tissues, on the contrary, L07648 expresses consistently higher except in one sample. Such correlations could be good predictors of sample class labels. However, all feature selection algorithms listed in the previous section can not find and use such correlations. Single gene based algorithms will ignore both genes since neither of them is a good predictor of sample class labels in its own right. Correlation based algorithms will actually remove such correlations, should any of the genes have been selected.

2.3 Virtual Gene Algorithm

Definition 1. *Virtual Gene* is a triplet $VG = (G_v, W, b)$ where $G_v \subseteq \mathcal{G}$ is a set of constituent genes, $|G_v| = n_v$, W is a matrix of size $n_v \times 1$, b is a numeric value. The expression levels of a *virtual gene* is determined using Definition 2.

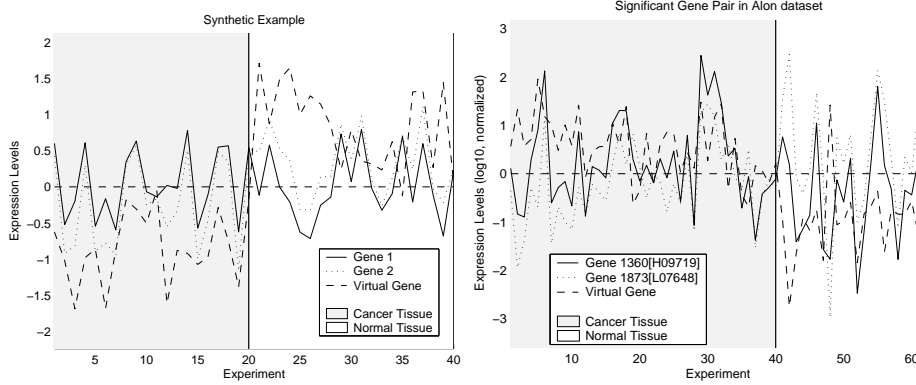


Fig. 1. Examples of gene pair being better predictor of class labels than single gene.

Definition 2. (Virtual Gene Expression) Given a *virtual gene* $VG = (G_v, W, b)$ and gene expression matrix E , where $|G_v| = n_v$, E is an $n_v \times m_v$ expression matrix, the *virtual gene expression* VE of a virtual gene VG is a linear combination of expression matrix E . $VE(VG, E) = W' \times E + b$, where W' is the transpose of W .

A *virtual gene* is a triplet $VG = (G, W, b)$ defined as Def. 1. Parameters W and b are chosen using FLD (fisher linear discriminant) to maximize linear separability between sample classes as listed in Algorithm 1. Discriminative power of a *virtual gene expression* with respect to sample classes can be measured using normal single gene based scores. We use t-score in this paper for this purpose. *Pairwise virtual gene* is a special case of *virtual gene* where the number of genes involved is limited to two. In this case, only the correlations between a pair of genes are considered. By limiting *virtual gene* to gene pairs, computation can be carried out efficiently. According to our experiments, it performs well on three public available datasets.

Algorithm 1 *gen_vg* : Calculating Virtual Gene From Training Data

Require: $\mathcal{E} = (G, S, L, E)$ as gene expression data.

Ensure: $VG = (G, W, b)$ as a virtual gene.

- 1: $(W, b) \leftarrow fld(E, L)$, (W, b) is the model returned by *fld* algorithm.
 - 2: **return** (G, W, b)
-

Definition 3. Pairwise virtual gene and its expression are special cases for *virtual gene* and its expression, where the number of genes involved is limited to two.

Exhaustive examination of all *pairwise virtual gene* requires $O(n^2)$ computation where n is the number of genes. For a large number of genes, exhaustive search of all gene pairs becomes inefficient. Such exhaustive search also invites unwanted noise

since not all gene pairs bare biological meaning. For example, for genes that are expressed in different locations in a cell, in different biological processes, without biological interactions, their relative abundance may not be biologically significant. Ideally, only gene pairs with some biological interaction shall be examined. We approximate this using a gene clustering approach. Each gene cluster corresponds roughly to some biological pathways. By limiting search among the gene pairs from the same gene cluster, we not only focus ourselves on these gene pairs that are more likely to interact biologically, but also make our gene selection algorithm much faster.

Algorithm 2 *pairwise_vg* : Pairwise Virtual Gene Selection

Require: $\mathcal{E} = (G, S, L, E)$; k as the number of genes to be selected; α ; β

Ensure: VGS: as set of virtual genes $VG = (G, W, b)$

- 1: Initialize VGS to be an empty set. Initialize *pair_score* to be a sparse $n \times n$ array.
- 2: Cluster genes based on their expression levels in E . Result stores in *Clusters*.
- 3: **for** each gene cluster $G' \in Clusters$ **do**
- 4: **for all** gene $g1 \in G'$ **do**
- 5: **for all** gene $g2 \in G'$ and $g2 \neq g1$ **do**
- 6: $vg \leftarrow gen_vg(\mathcal{E}((g1, g2), S))$
- 7: $ve \leftarrow VE(vg, E((g1, g2), S))$
- 8: $pair_score[g1, g2] \leftarrow t\text{-score}(ve, L)$
- 9: **end for**
- 10: **end for**
- 11: **end for**
- 12: **for** $i = 1$ to k **do**
- 13: $(g1, g2) \leftarrow \underset{(g1, g2)}{\operatorname{argmax}}(pair_score[g1, g2])$
- 14: $vg \leftarrow gen_vg(\mathcal{E}((g1, g2), S))$
- 15: add vg to VGS
- 16: multiply *pair_score* that involves $g1$ or $g2$ by α .
- 17: multiply *pair_score* that involves genes in same cluster of $g1$ or $g2$ by β .
- 18: $pair_score[g1, g2] \leftarrow$ minimum value
- 19: **end for**
- 20: **return** VGS

Algorithm 2 details *pairwise virtual gene selection* algorithm. Genes are first clustered based on their expression levels. For each pair of genes in the same cluster, virtual gene expression is calculated according to Def. 2. A single gene discriminative score with respect to the sample class labels is then derived from the virtual gene expression. All within-cluster pairwise virtual gene expression scores are calculated and stored for the next stage of analysis. The best scored virtual gene is then selected and pairwise scores are modified by two parameters. Pairwise scores of virtual genes that share constituent genes with the selected virtual gene are degraded by a constant α ranging $[0, 1]$. This dampens the effect of a single dominant salient gene. In the extreme case where α is set to 0, once a virtual gene is selected all virtual genes sharing constituent genes will not be further considered. The second parameter affecting the virtual gene selection is β , which controls how likely virtual genes in the same gene cluster are selected.

Different gene clusters correspond to different regulative processes in a cell. Choosing genes from different gene clusters broadens the spectrum of the selected gene set. β also ranges $[0, 1]$. In the extreme situation where $\beta = 0$, only one virtual gene will be selected for each gene cluster. After modifying pairwise scores, the algorithm begins next loop to find the highest scored virtual gene. This process repeats until k virtual genes have been selected. For performance comparison of *pairwise virtual gene* algorithm and single gene based algorithms, each *pairwise virtual gene* counts for two genes. For example, the performance of selecting 50 genes using single gene based algorithms would be compared to performance of selecting top 25 *pairwise virtual genes*.

2.4 Complexity of Pairwise Virtual Gene Algorithm

Pairwise virtual gene selection algorithm runs in three stages: (1) cluster genes based on expression profile (lines 1-2), (2) calculate discriminative scores for *pairwise virtual gene* (lines 3-11), and (3) select virtual genes with best discriminative scores (lines 12-20). We assume gene cluster number to be θ and n, m, k, α, β as discussed above.

In the first stage of analysis, k-means algorithm runs in $O(\theta n)$. In the second stage, the actual number of gene pairs examined is $O(\frac{n^2}{\theta})$, assuming gene clusters obtained in the previous stage are of roughly the same size. For each gene pair, the calculation of the virtual gene and its discriminative score require $O(m^2)$. Time complexity of the second stage is $O(\frac{m^2 n^2}{\theta})$. Stage three requires $O(k(\frac{n^2}{\theta} + m^2 + n + \frac{n}{\theta}))$ time. Putting them together, we have time complexity of $O(\theta n + \frac{m^2 n^2}{\theta} + k(m^2 + \frac{n^2}{\theta}))$. The most time consuming part in the previous expression is the term $O(\frac{m^2 n^2}{\theta})$. In our experiments, we choose $\theta \sim \Theta(n)$. Considering the fact that $k < n$, the time complexity of Algorithm 2 becomes $O(n^2 + nm^2)$. The $O(n^2)$ term is for k-means clustering, which runs rather quickly. If no clustering is performed in stage 1 (or $\theta = 1$, one gene cluster), the time complexity becomes $O(n^2 m^2 + kn^2)$. The savings in computation time is obvious.

Majority of space complexity for *virtual gene selection* algorithm comes from stage 2 in the algorithm where pairwise discriminative scores are recorded. The space needed for that is $O(\frac{n^2}{\theta})$ using sparse array. Under typical situation if we choose $\theta \sim \Theta(n)$, space complexity of Algorithm 2 becomes $O(n)$, although with a large constant.

3 Experiments

In order to assess the performance of our virtual gene feature selection algorithm, we use three benchmark datasets: colon cancer [1], leukemia [5] and multi-class cancer [12], all of which provide human gene expression levels using oligonucleotide microarrays. Samples are labeled with two class labels in the first two datasets. The multi-class dataset is labeled with multiple cancer types (and normal), of which we only use the cancer/normal distinction. Data are preprocessed using log transformation and normalization. Genes are filtered by fold change criteria. Performance of the feature selection method is measured by the classification accuracy of three different classifiers: KNN (k nearest neighbor), DLD (diagonal linear discriminant) and SVM (support vector machine, with linear kernel). Classification accuracy is estimated using cross validation

process. We test four FSS algorithms, single gene t-score, single gene S2N score [5], pairwise t-score [3] and our virtual gene. For our algorithm, the number of gene clusters varies from 128 to 400 and α and β range from 0.8 to 1.

Results are summarized in the following tables. Table 1 and Table 2 show FSS performance when 20 or 50 genes (10 or 25 virtual genes) are selected for sample classification. Virtual gene algorithm performs comparatively well in all cases. In some cases, the virtual gene algorithm outperforms other algorithms by a large margin, indicating correlations of gene expression levels being good predictors of sample class labels.

Table 1. Performance(%) of FSS methods using top 20 genes (10 virtual genes)

<i>FSS</i> <i>method</i>	<i>Colon Cancer</i>			<i>Leukemia</i>		<i>Multi-class</i>	
	<i>KNN</i>	<i>DLD</i>	<i>SVM</i>	<i>KNN</i>	<i>DLD</i>	<i>KNN</i>	<i>DLD</i>
t-score	81.58	80.10	81.35	94.97	92.75	78.92	78.34
S2N score	81.26	79.94	81.29	95.19	93.33	78.56	78.26
pairwise t-score	79.42	75.77	80.42	95.17	92.56	77.71	77.85
pairwise virtual gene	82.26	83.71	80.16	95.03	94.22	76.81	77.71

Table 2. Performance(%) of FSS methods using top 50 genes (25 virtual genes)

<i>FSS</i> <i>method</i>	<i>Colon Cancer</i>			<i>Leukemia</i>		<i>Multi-class</i>	
	<i>KNN</i>	<i>DLD</i>	<i>SVM</i>	<i>KNN</i>	<i>DLD</i>	<i>KNN</i>	<i>DLD</i>
t-score	81.52	78.10	81.81	95.47	94.14	81.81	78.85
S2N score	81.84	77.39	82.06	95.78	94.31	81.73	78.84
pairwise t-score	81.84	76.48	82.48	95.69	94.00	81.35	79.26
pairwise virtual gene	84.35	85.74	82.10	96.06	95.31	80.41	79.07

4 Conclusion and Future Work

Gene selection is crucial both for building a good sample classifier and for selecting smaller gene set for further biological investigation. Feature extraction algorithms (PCA, SVD, etc.), single gene based discriminative scores (t-score, S2N, TNoM, information gain, etc.) and correlation based algorithms have been proposed for this purpose. In this paper, we proposed a totally different approach. Instead of trying to minimize correlations within the selected gene set, we examined whether such correlations are good predictors of sample class labels. *Virtual gene* is a linear combination of a set of real genes. Our experiments confirm our assumption that the correlations between genes are indeed good predictors of sample class labels, better in many cases than single gene based discriminative scores. There are biological explanation for this: genes interact with each other. The relative abundance of genes is a better predictor than the absolute values. Using gene clustering algorithm to limit gene pair selection seems promising.

Our experiments show that by calculating pairwise scores for only a very small portion (0.5%) of all possible gene pairs, decent classification performance can still be achieved. This in turn shows most useful pairwise correlations are contained within gene clusters.

Our algorithm still has space for improvement. First but not least, we are interested in combining single gene based scores and virtual gene. In contrast to correlation based gene selection approaches, we can select top genes with high individual scores and top correlations between genes. We also want to examine larger virtual genes, virtual genes that combine more than two genes. Gene clustering is only a crude way of grouping co-regulated genes. We are currently working on using gene ontology as a way to group genes. Our algorithm is quite open, several other algorithms (e.g., cluster analysis and discriminative power of single gene) can be plugged into our algorithm without much modification. We leave this as future work as well.

References

1. U. Alon, N. Barkai, et al. Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissue probed by oligonucleotide arrays. *Proc. Natl. Acad. Sci. U.S.A.*, 96(12):6745–50, 1999.
2. A. Ben-Dor, L. Bruhn, N. Friedman, I. Nachman, M. Schummer, and Z. Yakhini. Tissue classification with gene expression profiles. volume 7, pages 559–83, 2000.
3. T.H. Bø and I. Jonassen. New feature subset selection procedures for classification of expression profiles. *Genome Biology*, 3(4):research0017.1–0017.11, 2002.
4. G. V. Bobashev, S. Das, and A. Das. Experimental design for gene microarray experiments and differential expression analysis. *Methods of Microarray Data Analysis II*, 2001.
5. T. R. Golub et al. Molecular classifications of cancer: Class discovery and class prediction by gene expression monitoring. *Science*, 286(5439):531–7, 1999.
6. T. Hastie, R. Tibshirani, et al. 'gene shaving' as a method for identifying distinct sets of genes with similar expression patterns. *Genome Biology*, 1(2), 2000.
7. j. Khan, J.S. Wei, et al. Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks. *Nature Medicine*, 7(6):673–9, 2001.
8. J. Jaeger, R. Sengupta, and W. L. Ruzzo. Improved gene selection for classification of microarrays. In *Proc. PSB*, 2003.
9. A .K. Jian, R. P. W. Duin, and J. Mao. Statistical pattern recognition: A review. *IEEE Transactions on pattern analysis and machine intelligence*, 22(1):4–37, 2000.
10. D. Jiang and A. Zhang. Cluster analysis for gene expression data: A survey. *IEEE Transactions on Knowledge and Data Engineering*, 16(11):1370–1386, 2004.
11. W. Li and I. Grosse. Gene selection criterion for discriminant microarray data analysis based on extreme value distributions. In *Proc. RECOMB*, 2003.
12. S. Ramaswamy, P. Tamayo, et al. Multiclass cancer diagnosis using tumor gene expression signatures. *PNAS*, 98(26):15149–15154, 2001.
13. Virginia Goss Tusher, Robert Tibshirani, and Gilbert Chu. Significance analysis of microarrays applied to the ionizing radiation response. *PNAS*, 98(9):5116–5121, April 2001.
14. Y. Wu and A. Zhang. Feature selection for classifying high-dimensional numerical data. In *CVPR 2004*, volume 2, pages 251–258, 2004.
15. E. P. Xing, M. I. Jordan, and R. M. Karp. Feature selection for high-dimensional genomic microarray data. In *Proc. ICML 2001*, pages 601–608, 2001.
16. L. Yu and H. Liu. Redundancy based feature selection for microarray data. In *Proc. of SIGKDD*, 2004.