Analysis of gene expression profiles: class discovery and leaf ordering

Chris H.Q. Ding
NERSC Division, Lawrence Berkeley National Laboratory
University of California, Berkeley, CA 94720
chqding@lbl.gov
(Extended Abstract)

Abstract

We approach the class discovery and leaf ordering problems using spectral graph partitioning methodologies. For class discovery or clustering, we present a min-max cut hierarchical clustering method and show it produces subtypes quite close to human expert labeling on the lymphoma dataset with 6 classes. On optimal leaf ordering for displaying the gene expression data, we present a sequential ordering method that can be computed in $O(n^2)$ time which also preserves the cluster structure. We also show that the well known statistic methods such as F-statistic test and the principal component analysis are very useful in gene expression analysis.

1 Introduction

Clustering analysis of DNA microarray gene expression proflies is among the first steps in understanding the activities of genes during biological process and their responses to certain desease conditions. By grouping tissue samples into homogeneous groups that correlates highly to particular macroscopic phenotypes such as different cancer types or other clinical syndromes [17, 1], more systematic characterization can be developed and new subtypes discovered.

Severl clustering methods applied to gene expressions data has been studied, the hierarchical agglomerative methods [14], self-organized maps [24, 17], simulated annealing [15], graph partioning methods [5], [22] [25]. However, many of these methods are more focused on clustering genes; when clustering tissue samples, many of them applied only to small number of phenotypes, typically 2$\sim$3.

In this paper we present a min-max cut hierarchical clustering algorithm and applied it to several publically available gene expression datasets. The effectiveness of our algorithm is demonstrated on the lymphoma dataset [1] where our algorithm can correctly identify 6 phenotypes based on standard correlation alone. We also found out that several samples of DLBCL type have high correlation with T-cell lines type, differ from the original study [1]. (see section 6).

As the second main contribution of the paper, we present an optimal leaf node ordering algorithm. In both hierarchical agglomerative clustering and divisive clustering, the clusters and the members of clusters contained in the leaf nodes of the binary hierarchical tree are often displayed in linear order. Biological and clinic studies are often performed in the context of this leaf node linear ordering, making it significant part of the clustering analysis. In Eisen et al.[14], the leaf nodes are ordered based on the average expression levels and patches of visible structures. In self-organizing maps [24], clusters are organized as a 2D topological mesh which does not always match those of hierarchical clustering method. Alon et al [15] used similarity between nodes and their parent's siblings to order the leaf nodes. Most recently, an optimal ordering method based on similarity of adjacent nodes is proposed by Bar-Joseph, et al.[3].

Here we propose a new optimal ordering objective function that both maximizes the similarities on adjacent nodes, but also *minimizes* similarities on large distance pairs of nodes. We then present an efficient algorithm to compute an approximate optimal ordering based on this ordering objective. This algorithm can also compute an optimal or-

dering that preserves the clustering structure. We apply this algorithm on the lymphoma datasets to illustrate the usefulness of our approach (see section 7).

The min-max cut algorithm follows a min-max clustering principle — tissue samples are grouped into clusters such that the similarity between clusters are minimized while similarities within each clusters are maximized. It is a new development in spectral graph partition [11, 16, 20] that makes use of the eigenvectors of Laplace matrix of a graph. It is more effective in finding balanced clusters than earlier algorithms. The optimal leaf node ordering algorithm uses a spectral formulation that is closely related to the spectral graph partitioning. Due to widely available software for efficient computation of eigenvectors (LAPACK, ARPACK, etc.), these methods can be efficiently implemented on a variety of computer architectures.

In this work, we use F-statistic for gene selection and show it is effective method. We also use principal component analysis for preliminary understanding of the data (see Figures 3 and 1). These well established statistical methods are quite useful in gene expression profiles analysis.

2 Gene selection

Of the thousands of genes measured in a microarray experiment, many of them show little variations across the tissue samples. and therefore are not useful in distinguishing different phenotypes. Furthermore, many genes are highly correlated, exhibit a large degree of redundancy. Selection of those *informative* genes [17] which show large variance among the targeted phenotypes is an important part of clustering analysis. There exist several methods for gene selection, from the simple t-statistic like tests [15, 17] to more sophisticated ones, such as information gain and Markov blanket. In this paper, we emphasizes the multi-cluster nature of the problem and use the F-statistic test which is a generalization of t-statistic for two class. Given a gene expression across n tissue samples $\mathbf{g} = (g_1, g_2, \cdots, g_n)$, the F-statistic is defined as

$$F = \left[\sum_k n_k (\bar{g}_k - \bar{g})^2 / (K-1) \right] / \sigma^2, \qquad (1)$$

where $\bar{g}$ is the average expression across all samples, $\bar{g}_k$ is the average within class C_k, and σ^2 is the *pooled* variance:

$$\sigma^2 = \left[\sum_k (n_k - 1)\, \sigma_k^2 \right] / (n - K)$$

where n_k and σ_k are the size and variance of gene expression within class C_k. For $K = 2$,

$$F = t^2, \quad t = \sqrt{\frac{n_1 n_2}{n_1 + n_2}}\, \frac{\bar{g}_1 - \bar{g}_2}{\sigma}, \qquad (2)$$

F-statistic reduces to t-statistic. We pick genes with large F-values or t-values. (If gene expressions follow Gaussian distribution, F-value of genes follow $F(K-1, n-K)$ distribution and we can compute p-values and confidence levels.) F-statistic for gene selection is used a classification study [13].

3 Similarity metric

For automatic class discovery, the association or similarity between tissue samples are the main factors. We wish to group tissue samples into clusters such that similarities between clusters are minimized and similarities within each clusters are maximized. There are a number of ways to define the similarity. A popular method is to measure the Pearson correlation [14, 15] $c(i, j)$ between two tissue samples i, j, and define the similarity as

$$s_{ij} = \exp(c(i,j)/\langle c \rangle),$$

where $\langle c \rangle$ is an average correlation between nearest k neighbors. Another method is to measure the Euclidean distance $d(i, j)$ and define

$$s_{ij} = \exp(-d(i,j)/\langle d \rangle),$$

where $\langle d \rangle$ is some average distance between nearest k neighbors. These similarity metrics are generic and are used in wide areas of applications. There are, however, more detailed modeling of similarities or weights based on statistical properties of the underlying populations [22].

4 Hierarchical divisive clustering

Many current research on gene expression cluster analysis uses hierarchical *agglomerative* clustering

methods [14, 1] which builds clusters from bottom up, gradually merging clusters into bigger and bigger clusters [12].

Hierarchical *divisive* clustering follows a top-down approach. It first partitions the samples into two clusters, and then recursively partition each leaf clusters into more clusters. This approach naturally uses a graph partitioning method. The similarities between all pairs of samples are first computed and stored in a matrix $W = (w_{ij})$. W then defines a weight matrix, or the adjacency matrix of an undirected graph with each node as a tissue sample. (Here we focus on clustering tissue samples. One can equivalently consider clustering genes according their responses to all tissue samples or other experiment conditions). Clustering becomes partitioning the graph into subgraphs based on certain objective or cost criteria. Clustering gene expression data using graph partitioning approach has also been studied in [22, 25].

4.1 Min-max cut

We briefly introduce the min-max cut graph partition and clustering method very recently developed for internet newsgroup clustering [10]. Given a weighted graph G with weight matrix W, we wish to partition it into two subgraphs A, B using the above mentioned min-max clustering principle. The similarity or association between A, B is the sum of weights between the two clusters,

$$\mathrm{sim}(A, B) = s(A, B) = \sum_{i \in A, j \in B} w_{ij}, \qquad (3)$$

The similarity or association within a cluster is the sum of all edge weights within A or B:

$$\mathrm{sim}(A, A) = s(A, A), \quad \mathrm{sim}(B, B) = s(B, B). \quad (4)$$

The clustering principle requires minimizing $s(A, B)$ while maximizing $s(A, A)$ and $s(B, B)$ independently at the same time. These requirements are simultaneously satisfied by the objective function,

$$J_{\mathrm{Mcut}} = \frac{s(A, B)}{s(A, A)} + \frac{s(A, B)}{s(B, B)}. \qquad (5)$$

J_{Mcut} is called min-max cut (Mcut) objective[10].

The solution of partition problem can be represented by an indicator vector $\mathbf{q}$, where the element of $\mathbf{q}$ on node i is

$$q_i = \begin{cases} a & \text{if } \ i \in A \\ -b & \text{if } \ i \in B \end{cases} \qquad (6)$$

where a and b ($0 < a, b < 1$) are two constants. Finding the optimal partition is NP-complete. An effective solution is the following. First, one can show that

$$\min_{\mathbf{q}} J_{\mathrm{Mcut}}(A, B) \Rightarrow \min_{\mathbf{q}} \frac{\mathbf{q}^T (D - W)\mathbf{q}}{\mathbf{q}^T D \mathbf{q}}, \qquad (7)$$

subject to $\mathbf{q}^T W \mathbf{e} = \mathbf{q}^T D \mathbf{e} = 0$, where $D = (d_i)$ is a diagonal matrix and $d_i = \sum_j w_{ij}$ is the degree of node i and $\mathbf{e} = (1, \cdots, 1)^T$. Second, we relax q_i from discrete indicators a and -b to real number in $(-1, 1)$. The solution of $\mathbf{q}$ for minimizing the Rayleigh quotient of Eq.(7) is given by

$$(D - W)\mathbf{q} = \zeta D \mathbf{q}. \qquad (8)$$

The solution to this generalized eigenvalue problem is the second eigenvector $\mathbf{q}_2$, called the Fiedler vector. Third, we sort the Fiedler vector $\mathbf{q}$ to establish a linear search index order. Fourth, using the linear index order, given any cutpoint i_{cut}, we partition the graph into two subgraphs (clusters):

$$A = \{q_i \mid i \le i_{cut}\}, \quad B = \{q_i \mid i > i_{cut}\}.$$

A contains all nodes left of the cutpoint i_{cut} and B contains all nodes on the right. We search for the cutpoint i_{cut} such that $J_{\mathrm{Mcut}}(A, B)$ is minimized. The corresponding A and B are the final clusters of the Mcut algorithm.

This Mcut algorithm is the latest development along the line of spectral graph partitioning that is based on the properties of eigenvectors of the Laplacian matrix $L = D - W$ [11, 16, 20]. Besides the min-max cut objective function, the *ratio cut* objective, $J_{\mathrm{Rcut}} = s(A, B)/|A| + s(A, B)/|B|$ is proposed earlier [6, 18] to balance the *sizes* of the partitions. The *normalized cut* objective, $J_{\mathrm{Ncut}} = s(A, B)/s(A, G) + s(A, B)/s(B, G)$, proposed in [23] attempts to balance the *volumes* of the partitions ($s(A, G)$ is the volume of subgraph A [7]). In contrast, J_{Mcut} balances within-cluster similarity. Both theoretical analysis and experiments on

internet newsgroup data sets indicate [10] J_{Mcut} gives more balanced clusters while J_{Rcut} and J_{Ncut} sometimes cut a small subgraph away from a large graph resulting in unbalanced clusters. Note that although $J_{\text{Rcut}}, J_{\text{Ncut}}$ and J_{Mcut} objective functions are first *proposed* based on appropriate intuitions, they can also be obtained automatically as the eigenvalues of the Fiedler vector in perturbation analysis [9]. This further justifies using the Fiedler vector for finding the (approximate) optimal partitions based on these objective functions.

4.2 Recursive Clustering

Once a cluster is partitioned into two clusters, we can further partition each of them using the same method. This process is repeated several times and a binary partition tree is established where the each node contains a cluster during the process.

A stopping criteria is necessary to stop the divisive process. The Mcut objective provides such a criteria. For a cluster G_k on the leaf node, we compute the Fiedler vector $\mathbf{q}$, find the optimal cut, and obtained $J_m = \min J_{\text{Mcut}}(q)$ value. If J_m is large, the overlap between two resulting subclusters is large in comparison to the within-subcluster similarity, hence cluster G_k should not be further partitioned. We set

$$J_{\text{stop}} = 1.0,$$

as the threshold for J_m in our experiments.

The complete clustering algorithm is:

1. For the current leaf node G_k, solve Eq.(5) for the second lowest eigenvector $\mathbf{q}$.
2. Sort $\mathbf{q}$. Find the cutpoint i_{cut} with $\min(J_{\text{Mcut}})$.
3. If $\min(J_{\text{Mcut}}) < J_{\text{stop}}$, cut G_k into two children clusters A_k, B_k. A_k and B_k become new active leaf nodes on the binary tree. If $\min(J_{\text{Mcut}}) > J_{\text{stop}}$, G_k becomes a dead-end leaf node.
4. Examine all active leaf nodes until none of them can be further partitioned.

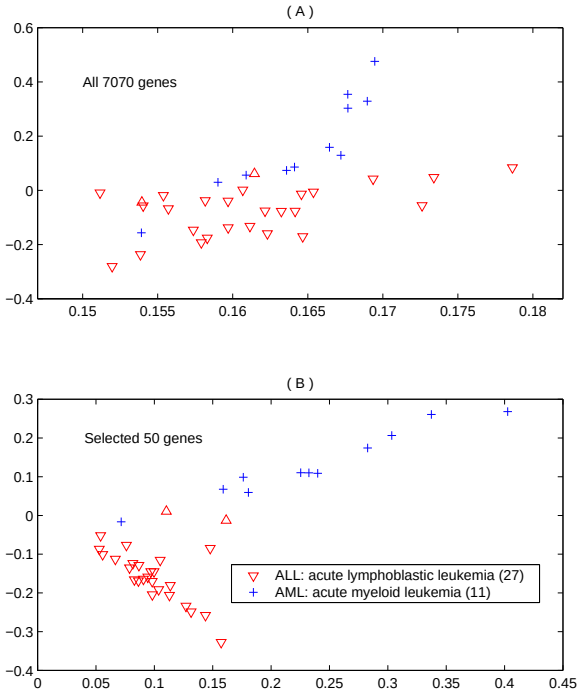

Figure 1: Leukemia dataset as shown in 2D space of the first two principal components. (A) All genes are used, i.e., we used PCA to reduce the data from the original 7070 dimensions to 2 dimensions. (B) Only 50 selected genes are used in PCA.

5 Analysis of Leukemia subtypes

The leukemia dataset of Golub et al [17] is well studied. Here we study the training dataset: 7070 gene expressions of 38 tumor tissue samples. The goal here is to see if we can automatically detect the two phenotypes of the cancer: acute lymphoblastic leukemia (ALL) and acute myeloid leukemia (AML). To gain insight, we perform the principal component analysis (PCA) and show results on the first two principal components in Fig.1a. One can see the structure of two phenotypes. The two classes overlap substantially when all 7070 genes are used. We used the t-test statistic criteria to select 50 genes (shown in Fig.2). Using 50 selected genes, the two classes separate clearly (Fig.1b).

We perform the cluster algorithm on the dataset using Pearson correlation. The cluster result using all 7070 genes is shown in the two-way contingency table (Table 1). We use the simple Q-accuracy [21, 8] (sum of the diagonal elements divided by

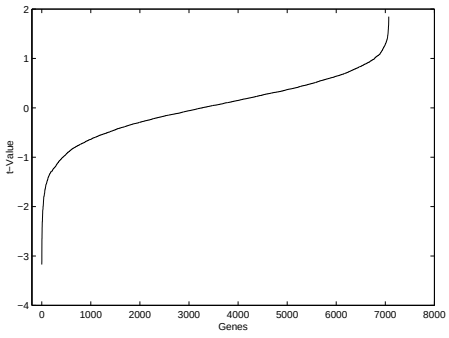

Figure 2: The t-values (Eq.2) for all 7070 genes in the Leukemia dataset of Golub et al [17]. Gene indices are reordered according to t-values. We select 25 genes with largest positive t-values and 25 genes with largest negative t-values.

the total number of samples).

	C_1	C_2
AML	10	1
ALL	9	18

Table 1: A contingence table summarizes the discovered clusters C_1, C_2 using all 7070 genes. The accuracy is $Q = (10+18)/(10+1+9+18) = 0.737$.

Using the 50 selected genes, the clustering results (contingency table T) and the accuracy are

$$T = \begin{bmatrix} 11 & 0 \\ 2 & 25 \end{bmatrix}, \quad Q = 0.947.$$

Only two samples of the ALL class (the two points with the $\triangle$ symbol in Fig.1b instead of the ∇ symbol for the rest of ALL samples) are incorrectly clustered into the AML class. Clearly, these two samples are on the boundary between the clusters. We note that if we use the Euclidean distance option to define the similarity metric, these two samples will be correctly clustered, while one sample from AML class (the point nearest to ALL samples in Figure 1b) is mis-clustered into ALL class. The accuracy will be $Q = 0.974$. Thus our clustering algorithm performs well, and from the PCA analysis we understand the origin of clustering errors.

This dataset is studied in [25]. The CLIFF algorithm begins with 360 genes to perform iterative feature selection and clustering, to graduately reduce the number of genes. We perform the clustering using 360 genes selected by the t-statistic and the results are identical to that using 50 genes, although in 2D PCA space, the two classes mix more than the case of 50 genes (not shown). This indicate the effectiveness of the t-statistic in gene selection.

6 Analysis of Lymphoma classes

This dataset contains 4029 gene expression of 96 tissue samples from Alizadeh et al.[1]. Using biological and clinic expertise, Alizadeh et al classify the tissue samples into 9 classes as shown in Figure 3. Because of the large number of classes and also highly uneven number of samples in each classes $(46, 2, 2, 10, 6, 6, 9, 4, 11)$, it is a reletively difficult problem. We use F-statistic to select 200 genes for this study as shown in Figure 4. We also ignore 8 tissue samples belonging to classes C2, C3, and C8 because (i) the number of samples in these classes are too small. (ii) as discussed in [1], C2 (germinal center B), C3 (lymph node/tonsil) are very close to C1 (DLBCL) — in fact, they are clustered together in [1]. Therefore, we focus on 6 largest classes of 88 samples. Using PCA, we first examine the samples in the first two principal components as in Fig.3. The structure of 6 classes are visible in Fig.3. This motivate us to further study the automatic class discovery using the clustering algorithm.

We perform the clustering algorithm this dataset. The partition tree is shown in Figure 5. The clustering results (contingency table) and accuracy are listed below:

$$\begin{bmatrix} 39 & \cdot & 1 & \cdot & \cdot & 6 \\ \cdot & 10 & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & 9 & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & 11 & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & 6 & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & 6 \end{bmatrix}$$

$$Q = 0.921.$$

These results are quite reasonable for this relatively difficult problem with such a large number of classes and varied sizes of each class.

We independently verified these results by checking the sample-sample correlations. Samples OCI-Ly3, OCI-Ly10, DLCL-0042, DLCL-0017 of C1 class

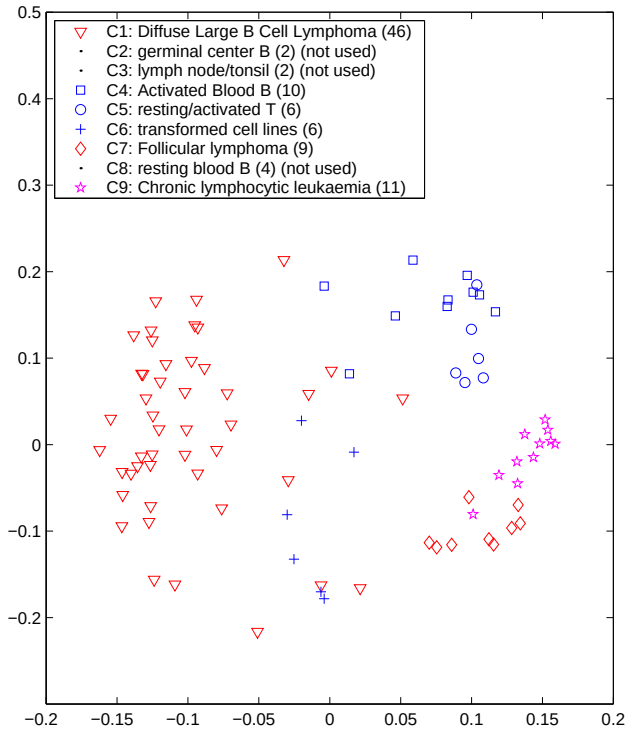

Figure 3: The lymphoma dataset of [1] in PCA space, using 200 genes selected based on F-statistic (class labels are according to Fig. 1 of [1]).

have high correlations with samples in C6 class and have low correlations with the rest of samples in C1. These samples should belong to C6 if the sample-sample correlation is the only factor in determining the class information. These results do not change if we use 100 genes, and therefore reflect the inherent structure of the gene expression data. Further studies are necessary to understand why they differ from the expectations of human expertise.

One of the main results of [1] is using the gene expression profiles to further detect two subtypes of DLBCL which are previous unknown and are more subtle to detect. Indeed, using our algorithm, we can further split DLBCL samples into the Germinal center B-like DLBCL and the Activated B-like DLBCL, although the J_{Mcut} value are larger, indicating these two subtypes mix more than other phenotypes (see Figure 5).

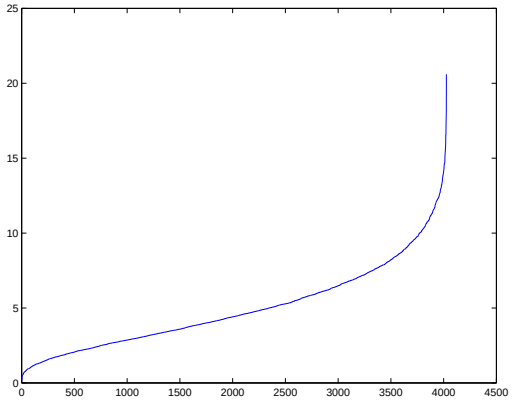

Figure 4: The F-value (Eq.1) for the 4029 genes in the lymphoma dataset of Alizadeh et al. [1]. We select 200 genes with largest F-values.

7 Ordering tissue samples

Once the cluster structures are discovered (and also before that), very often we need to order the genes or tissue samples in a linear order such that adjacent tissue samples are *similar* and samples far away along this sequential order are *different*. This is quite useful both for displaying results and for further inspection and study[14, 15, 3]. Here we present a new ordering objective function and an efficient algorithm to compute an approximate optimal solution. This optimal leaf ordering also preserve cluster structure, i.e., all nodes within a cluster should be adjacent to one another.

7.1 Leaf ordering objective function

In [3], the objective of leaf node ordering (defined by index permutation $\pi = (\pi_1, \cdots, \pi_n)$) is to insure that adjacent nodes are similar. This is achieved by maximizing the sum of similarity between adjacent nodes:

$$\max_{\pi} J_{d=1}(\pi), \quad J_{d=1}(\pi) = \sum_i s_{\pi_i, \pi_{i+1}} \qquad (9)$$

However, this objective ignore the similarity between larger distance nodes. To see this point, we list the different *distances* of a 5-node graph below:

$$\begin{bmatrix} 0 & 1 & 2 & 3 & 4 \\ & 0 & 1 & 2 & 3 \\ & & 0 & 1 & 2 \\ & & & 0 & 1 \\ & & & & 0 \end{bmatrix}$$

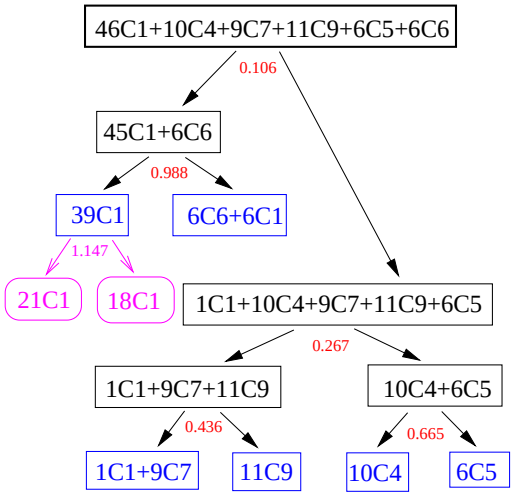

Figure 5: The binary partition tree outlines the divisive clustering process. Each node contains a current cluster, whose content, the number of samples in each class are given, e.g., 46C1 means 46 samples of class C1. The six leaf nodes contain the final 6 clusters discovered by the Mcut algorithm. The $\min(J_{\text{Mcut}})$ value for each divisive partitioning is also shown. We also attempted to further partition the DLBCL cluster (39C1) into two subtypes, 21C1 and 18C1. The $\min(J_{\text{Mcut}})$ is 1.147, slightly bigger than $J_{stop} = 1$. We verified that the 21C1 cluster corresponds to GC B-like DLBCL and the 18C1 cluster corresponds to Activated B-like DLBCL (see Figure 3 in [1]).

There are four $d = 1$ pairs, and their similarity is contained in $J_{d=1}$. There are also three $d = 2$ pairs. We believe the similarities on these pairs should be *smaller* than on those $d = 1$ pairs, if the final leaf order is meaningful. Further more, there are two $d = 3$ pairs and the similarities on these pairs should be smaller than both $d = 1$ and $d = 2$ pairs. This consideration goes all the way to $d = n - 1$ pair. All these considerations (except for $d = 1$ pairs) are not taken into account in the $J_{d=1}$ objective.

For this reason, we believe a more appropriate, distance-sensitive, objective function in leaf node ordering is the following

$$\min_{\pi} J_d(\pi), \quad J_d(\pi) = \sum_{\ell=1}^{n-1} \ell^2 J_{d=\ell}(\pi) \tag{10}$$

where

$$J_{d=\ell}(\pi) = \sum_i s_{\pi_i, \pi_{i+\ell}} \tag{11}$$

Here we penalize large distance similarities with larger weights to ensure that the *larger* the distance between a pair of nodes is , the *less* similar this two nodes are. We may rewrite the distance-sensitive objective function J_d as

$$J_d(\pi) = 4s + \left[\sum_{\ell=3}^{n-1} (\ell^2 - 4) J_{d=\ell}(\pi)\right] - 3J_{d=1}(\pi).$$

where $s = \sum_{ij} s_{ij}$ is the total weight of the graph which is a constant. Therefore, minimizing J_d is equivalent to simultaneously maximizing adjacent similarities $J_{d=1}$ and minimizing large distance similarities.

7.2 Approximation algorithm

An exact algorithm to find the optimal solution is NP-complete. However, an efficient $O(n^2)$ algorithm exist to compute an approximate solution to minimizing J_d. First, we note that J_d can be written as

$$J_d(\pi) = \frac{1}{2} \sum_{\pi_i, \pi_j} s_{\pi_i, \pi_j} (i - j)^2.$$

In the summation, replacing π_i by i and i by π_i^{-1} (π^{-1} is the inverse permutation), $J_d(\pi)$ remains identical. With shiting and rescaling, we have

$$J_d(\pi) = \frac{n^2}{8} \sum_{i,j} s_{i,j} \left(\frac{\pi_i^{-1} - \frac{n+1}{2}}{n/2} - \frac{\pi_j^{-1} - \frac{n+1}{2}}{n/2} \right)^2$$

For simplicity, we define

$$p_i \equiv \frac{\pi_i^{-1} - \frac{n+1}{2}}{n/2} \in \{\frac{1-n}{n}, \frac{3-n}{n}, \cdots, \frac{n-1}{n}\} \tag{12}$$

Note that

$$\sum_{ij} s_{ij}(p_i - p_j)^2 = \sum_{ij} s_{ij}(p_i^2 + p_j^2 - 2p_i p_j)$$

$$= \sum_{ij} 2p_i(d_{ii}\delta_{ij} - s_{ij})p_j = 2\mathbf{p}^T(D - S)\mathbf{p} \tag{13}$$

where $\delta_{ij} = 1$ if $i = j$; $\delta_{ij} = 0$ otherwise. Therefore, the inverse index permutation π^{-1} is obtained

by minimizing $\mathbf{p}^T(D-S)\mathbf{p}$ for p_i taking those discrete values of Eq.(12) in $(-1,1)$.

So far everything is exact. The critical approximation step here is that we *relax* p_i from these discrete values to continuous values in $(-1,1)$. With this, $\mathbf{p}^T(D-S)\mathbf{p}$ can be minimized by solving an eigenvalue problem.

Since $s_{ij} \geq 0$, from Eq.(13), $\mathbf{p}^T(D-S)\mathbf{p} \geq 0$ and trivial solutions such as $\mathbf{p}_0 = 0$ or $\mathbf{p}_0 = \mathbf{e}$ will minimize $\mathbf{p}^T(D-S)\mathbf{p}$: $\mathbf{p}_0^T(D-S)\mathbf{p}_0 = 0$. Therefore we need to impose a constraint on the normalization of $\mathbf{p}$ and other constraint so that $\mathbf{p} \neq \mathbf{e}$. These two constraints are

$$\mathbf{p}^T D\mathbf{p} = \text{const}, \ \mathbf{p}^T D\mathbf{e} = 0. \tag{14}$$

These constraints can be simultaneously satisfied with the scaled ordering objective

$$\tilde{J}_D = \frac{\mathbf{p}^T(D-S)\mathbf{p}}{\mathbf{p}^T D\mathbf{p}}. \tag{15}$$

The above approximation by relaxation of discrete permutation indicators for computing the optimal solution was first proposed in different forms in [19]and [4]. One can see it has a close connection to spectral graph partition (Eq.7).

Clearly, the solution to the minimization problem is the eigenvector of $(D-S)\mathbf{p} = \lambda D\mathbf{p}$. The lowest eigenvector is trivial $\mathbf{p}_1 = \mathbf{e}$ with $\lambda_1 = 0$, which should be discarded. The correct solution is the second lowest eigenvector $\mathbf{p}_2$, which automatically satisfies the constraints of Eq.(14). Once $\mathbf{p}_2$ is computed, π^{-1} can be inferred from Eq.(12). A more efficient way is to sort $\mathbf{p}_2$ to increasing order, The induced ordering gives the desired index permutation π.

To measure the quality of leaf ordering, we define the large-distance similarity ratio

$$r_d = J_d(\pi)/J_d(\text{random}),$$

and the adjacent pair similarity ratio

$$r_1 = J_{d=1}(\pi)/J_{d=1}(\text{random}),$$

where

$$J_d(\text{random}) = \langle s_{ij}\rangle \sum_{ij} |i-j|^2, \langle s_{ij}\rangle = \sum_{ij} s_{ij}/n^2.$$

and $J_{d=1}(\text{random}) = \langle s_{ij}\rangle(n-1)$, where $n-1$ accounts for the number of adjacent pairs on the ordering. r_d includes all pairs of distances, $n-1$ adjacent pairs and $(n-1)(n-2)/2$ pairs with $d > 1$, and r_d is dominated by large distance pairs (thus the name large-distance similarity ratio). If the dataset is randomly permuted, we expect $r_d \simeq 1$ and $r_1 \simeq 1$, which can be easily verified. As the leaf ordering is improved, the large-distance similarity ratio r_d will decrease while the adjacent pair similarity ratio r_1 will increase. For the 88 tissue samples in the lymphoma dataset, we obtained

$$r_d = 0.18, \quad r_1 = 3.39.$$

Thus the large-distance similarities are reduced about a factor of $1/0.18 = 5.6$ from random ordering and the adjacent pair similarities increase by 239%. The results of optimal leaf ordering on the lymphoma dataset is shown in Figure 6. Note that we can also reorder genes by first computing gene-gene similarity using Pearson correlation (see §3) and then ordering them using the same method. This is done in Figure 6.

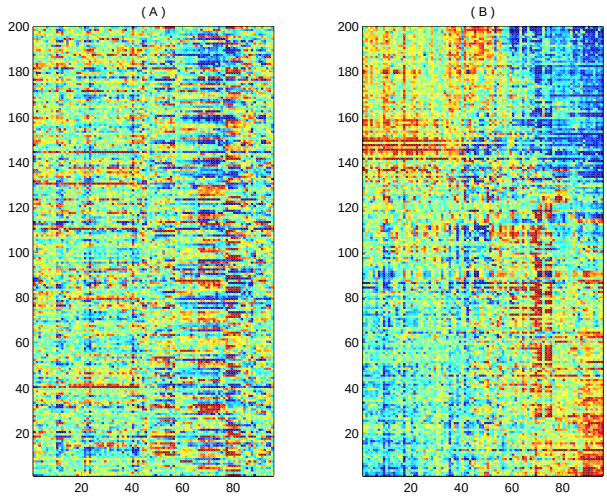

Figure 6: Optimal leaf node ordering of the lymphoma dataset: 88 tissue samples with 200 genes. (A) Data are displayed as the original order from Alizadeh et al [1]. (B) Both tissue samples and genes are ordered according to J_d objective computed from Eq.(15).

7.3 Preserving cluster structure

In the above distance-sensitive ordering heuristic, our main goals are (i) to maximize the similarities on adjacent pairs of nodes and (ii) to minimize similarities on large distance pairs. However, these considerations do not take into account the cluster structure — it sometimes occurs that nodes of a cluster are not consecutively ordered, and nodes from another cluster could mix in between.

Here we propose a method to take this into account in leaf ordering. Our approach is not to modify the J_d objective. Instead, we modify the similarity matrix with the following considerations: (1) preserve local ordering within each cluster, while (2) enforce nodes within a cluster to stay together. The first consideration implies that similarity between nodes of a cluster, relatively, should remain unchanged. The second consideration suggest we reduce the similarities between different clusters (or equivalently, increase within-cluster similarities uniformly). The following re-weighting achieves both goals:

$$\bar{s}_{ij} = s_{ij}(1 + \alpha\delta_{c_i,c_j}) \qquad (16)$$

where c_i is the cluster id of node i. $\alpha > 0$ is a parameter that adjusts how much we increase the within-cluster similarity. If $\alpha \gg 1$, clusters will become well separated. Thus $\alpha \simeq 1$ is good choice.

In Fig.7, we show the effects on modifying the leaf order that preserves the cluster structure for the 88 sample dataset. We set $\alpha = 1$ in Eq.(16). The cluster structure is 6-class structure discovered in Figure 5. One can see that the cluster structure is preserved in the leaf ordering.

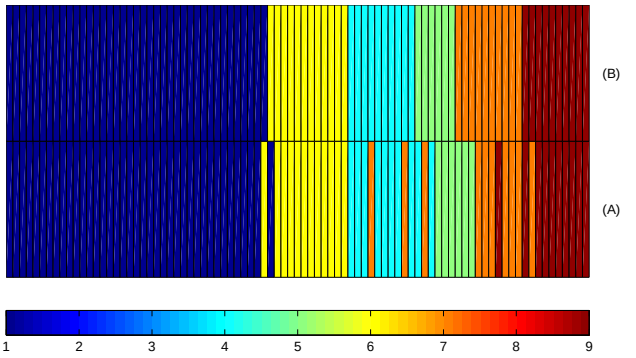

Figure 7: Leaf node ordering that preserves cluster structure. The cluster ids (C1, C4, C5, C6, C7, C9) for each sample are color coded. (A) Original ordering computed from Eq.(15). There are several samples from different clusters mix together. (B) The modified ordering computed with the similarity matrix modified according to Eq.(16). The cluster structure is preserved.

8 Discussions

The main contributions of this work are two fold: (1) we introduce the min-max cut hierarchical divisive clustering algorithm and show it produces good cluster results on the gene expression dataset with large number of classes. (2) we introduce a fast and effective leaf nodes ordering method for tissue samples and genes that maximize similarities on adjacent nodes and minimize similarities on large distance pairs of nodes. A simple modification of the method leads to leaf ordering that also preserves cluster structure.

This work also demonstrate that the well-known statistic methods such as F-statistic test and PCA are quite useful in gene expression analysis. The F-statistic is effective in gene selection. PCA is effective in gaining initial knowledge of the cluster structure of the dataset. PCA has been used in [2] for different goals, and is recently criticized [26] for not being effective in cluster analysis.

Our clustering results on lymphoma dataset also reveals some difference in class labeling of several tissue samples. This needs to be more carefully studied. More details and analysis results on the lymphoma dataset will be collected in a website (www.nersc.gov/∼cding/lymphoma).

Acknowledgements. This work is supported by U.S. Department of Energy (Office of Science, Office of Advanced Scientific Research/MISC Division and Office of Laboratory Policy and Infrastructure/LDRD) under contract DE-AC03-76SF00098.

References

[1] A.A. Alizadeh, M.B. Eisen, et al. Distinct types of diffuse large B-cell lymphoma identified by gene expression profiling. *Nature*, 403:503–511, 2000.

[2] O. Alter, P. O. Brown, and D. Botstein. Singular value decomposition for genome-wide expression data processing and modeling. *Proc. Nat'l Acad Sci USA*, 97:10101–10106, 2000.

[3] Z. Bar-Joseph, D.K. Gifford, and T.S. Jaakkola. Fast optimal leaf ordering for hierarchical clustering. *Bioinformatics*, 17:S22–S29, 2001.

[4] S. T. Barnard, A. Pothen, and H. D. Simon. A spectral algorithm for envelope reduction of sparse matrices. *Proc. Supercomputing '93, IEEE*, pages 493–502, 1993.

[5] A. Ben-Dor, R. Shamir, and Z. Yakhini. Clustering gene expression patterns. *J. Computational Biology*, 6:281–297, 1999.

[6] C.-K. Cheng and Y.A. Wei. An improved two-way partitioning algorithm with stable performance. *IEEE. Trans. on Computed Aided Desgin*, 10:1502–1511, 1991.

[7] F.R.K. Chung. *Spectral Graph Theory.* Amer. Math. Society, 1997.

[8] C. Ding and I. Dubchak. Multi-class protein fold recognition using support vector machines and neural networks. *Bioinformatics*, 17:349–358, 2001.

[9] C. Ding, X. He, and H. Zha. A spectral method to separate disconnected and nearly-disconnected web graph components. In *Proc. ACM Int'l Conf Knowledge Disc. Data Mining (KDD)*, pages 275–280, 2001.

[10] C. Ding, X. He, H. Zha, M. Gu, and H. Simon. A min-max cut algorithm for graph partitioning and data clustering. *Proc. IEEE Int'l Conf. Data Mining*, pages 107–114, 2001.

[11] W.E. Donath and A. J. Hoffman. Lower bounds for partitioning of graphs. *IBM J. Res. Develop.*, 17:420–425, 1973.

[12] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification, 2nd ed.* Wiley, 2000.

[13] S. Dudoit, J. Fridyand, and T.P. Speed. Comparison of discrimination methods for the classification of tumor using gene expression data. *Univ. of California, Dept of Statistiscs, Tech Report 576*, 2000.

[14] M.B. Eisen, P.T. Spellman, P.O. Brown, and D. Botstein. Cluster analysis and display of genome-wide expression patterns. *Proc. Nat'l Acad Sci USA*, 95:14863–14868, 1998.

[15] U. Alon et al. Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. *Proc. Nat'l Acad Sci USA*, 96:6745–6750, 1999.

[16] M. Fiedler. Algebraic connectivity of graphs. *Czech. Math. J.*, 23:298–305, 1973.

[17] T.R. Golub, D.K. Slonim, P. Tamayo, et al. Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science*, 286:531–537, 1999.

[18] L. Hagen and A.B. Kahng. New spectral methods for ratio cut partitioning and clustering. *IEEE. Trans. on Computed Aided Desgin*, 11:1074–1085, 1992.

[19] K. M. Hall. R-dimensional quadratic placement algorithm. *Management Science*, 17:219–229, 1971.

[20] A. Pothen, H. D. Simon, and K. P. Liou. Partitioning sparse matrices with egenvectors of graph. *SIAM Journal of Matrix Anal. Appl.*, 11:430–452, 1990.

[21] B. Rost and C. Sander. Prediction of protein secondary structure at better than 70% accuracy. *J. Mol. Bio.*, 232:584–599, 1993.

[22] R. Sharan and R. Shamir. CLICK: A clustering algorithm with applications to gene expression analysis. *Proc. ISMB 2000*, pages 307–316, 2000.

[23] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE. Trans. on Pattern Analysis and Machine Intelligence*, 22:888–905, 2000.

[24] P. Tamayo, D. Slonim, J. Mesirov, Q. Zhu, S. Kitareewan, E. Dmitrovsky, E.S. Lander, and T.R. Golub. Interpreting patterns of gene expression with self-organizing maps: Methods and application to hematopoietic differentiation. *Proc. Nat'l Acad Sci USA*, 96:2907–2912, 1999.

[25] E.P. Xing and R.M. Karp. CLIFF: clustering of high-dimensional microarray data via iterative feature filtering using normalized cuts. *Bioinformatics*, 17:S306–S315, 2001.

[26] K. Y. Yeung and W. L. Ruzzo. Principal component analysis for clustering gene expression data. *Bioinformatics*, 17:763–774, 2001.