Identifying Rare Cell Populations in Comparative Flow Cytometry

Ariful Azad[1], Johannes Langguth[2], Youhan Fang[1], Alan Qi[1], and Alex Pothen[1]

[1] Dept. Computer Science, Purdue University, West Lafayette, IN 47907, USA
[2] Department of Informatics, University of Bergen, Bergen, Norway
{aazad,yfang,alanqi,apothen}@cs.purdue.edu, johannes.langguth@ii.uib.no

Abstract. Multi-channel, high throughput experimental methodologies for flow cytometry are transforming clinical immunology and hematology, and require the development of algorithms to analyze the high-dimensional, large-scale data. We describe the development of two combinatorial algorithms to identify rare cell populations in data from mice with acute promyelocytic leukemia. The flow cytometry data is clustered, and then samples from the leukemic, pre-leukemic, and Wild Type mice are compared to identify clusters belonging to the diseased state. We describe three metrics on the clustered data that help in identifying rare populations. We formulate a generalized edge cover approach in a bipartite graph model to directly compare clusters in two samples to identify clusters belonging to one but not the other sample. For detecting rare populations common to many diseased samples but not to the Wild Type, we describe a clique-based branch and bound algorithm. We provide statistical justification of the significance of the rare populations.

Keywords: flow cytometry, edge cover, clique, mixture modeling, KL divergence, acute promyelocytic leukemia (APL).

1 Introduction

We describe two algorithms to identify rare cell populations characteristic of diseases such as leukemia by analyzing flow cytometric data obtained from diseased and healthy samples. The recent development of high-throughput, multi-channel flow cytometry creates high-dimensional and large-scale data that requires the concomitant development of algorithms for comparative analyses of data from diseased and healthy samples, and from diseased samples at various stages of disease. Specifically, we need algorithms that can match cell populations among diseased samples, and differentiate between cell populations that belong to diseased and healthy states. Such studies could distinguish cancer cells from healthy cells, and identify cancer stem cells that are responsible for generating new cancerous cells, which could lead to therapies targeting such cells.

In flow cytometry, fluorescently labeled antibodies are bound to antigens on the cell, and on excitation with a laser as cells flow in a fluid stream, the fluorochrome emits light of a specific wavelength, thus identifying the cell

V. Moulton and M. Singh (Eds.): WABI 2010, LNBI 6293, pp. 162–175, 2010.

populations that express the antigen. Flow cytometry is routinely used in the diagnosis of diseases and has many applications in clinical practice and research. Initially flow cytometry permitted the investigation of only one fluorophore, but recent advances allow close to twenty parallel channels to be monitored [5,12]. Various techniques have been developed in the past [7,10] to analyze this high dimensional data. A recent survey of data analysis methods in flow cytometry is provided in [2].

Early work on analyzing this high dimensional data has relied on projecting the data to lower dimensions and manual gating, which is labor intensive and influenced by analyst bias. Hence the development of efficient and accurate algorithms for analyzing the large-scale, high dimensional data is a critical need.

Performing comparative analysis of samples at the cell level is computationally expensive, and hence a more practical approach is to cluster cells in each sample first, and then perform the comparative analyses across the samples. Various techniques have been proposed to cluster flow cytometry data and form groups of cells [3,4,7], but there has been little work on the post-processing of the clustered data to identify common and distinct cell populations among diseased and healthy states.

A recent approach for downstream analysis of clustered data, flow analysis with automated multivariate estimation (FLAME), was proposed by Pyne et al. [10]. The fluorescense intensity matrix with rows corresponding to cells and columns corresponding to antibodies is first clustered into cell populations using the skew t distribution. The clusters across all samples are then pooled and a set of *global metaclusters* are obtained from them using an approach called Partitioning across Medoids. Each sample is then compared with the set of global metaclusters using an integer programming formulation of a weighted b-matching in a bipartite graph with additional constraints.

Our work is closest to the FLAME approach, while differing from it in significant ways. First, we use a non-parametric infinite mixture model in clustering phase, whereas FLAME used the skew t mixture model. Second, we compare clusters in two or more samples directly without creating metaclusters from the clusters in all samples. We propose a generalized edge cover formulation in a bipartite graph as a model for discovering outlying clusters using pairwise comparisons of samples. Third, we propose a weighted clique approach to compare multiple samples to identify outlying clusters and classify them further into distinctive and common outliers.

2 Problem Formulation

2.1 Description of Data

We analyze two different flow cytometry datasets on mouse bone marrow cells from Brigham and Women's Hospital in Boston [15]. In this work, an oncogene PML-RARα, was expressed in mice leading to acute promyelocytic leukemia (APL) in a course of weeks. Each dataset consists of flow cytometry data of cells from three leukemic mice (P^i), one pre-leukemic mouse (H) that has the oncogene expressed but has not developed APL yet, and a Wild Type (WT) sample that does not have the oncogene expressed. Each sample consists of multidimensional (6- or 7-dimensional) flow cytometry data of cells from a single mouse with each dimension representing a specific characteristic of the cell. A sample is represented as a matrix of size $N \times d$, where N is the number of cells, and d is the dimension of data. The data is shown in Table 1. We normalize each column of the matrix by converting it to a vector with mean equal to zero and standard deviation equal to one, and then apply a clustering method to be described in Sec. 3.1.

2.2 Model of the Data

Let each dataset consist of samples from N patients, labeled P^1 $P^2, \ldots, P^N$, and one WT. The i-th patient P^i has n_{P^i} clusters $P^i = \{u_1^i, u_2^i, \ldots, u_{n_{Pi}}^i\}$, where u_j^i is the j-th cluster in the i-th patient data. Similarly, WT has n_{WT} clusters $WT = \{w_1, w_2, \ldots, w_{n_{WT}}\}$. If multiple WT samples are available they can be combined beforehand to construct a unique WT model.

We use the Kullback-Leibler divergence as the measure of distance between two clusters. The KL-divergence [6], also known as the relative entropy, between two probability density functions $p(x)$ and $q(x)$ is:

$$KL(p\|q) = -\int p(x) ln \left\{ \frac{q(x)}{p(x)} \right\} dx. \tag{1}$$

For two d-dimensional Gaussian distributions N_0 and N_1 with means μ_0, μ_1 and covariance matrices Σ_0, Σ_1, respectively, the KL divergence has a closed-form expression:

$$\begin{aligned} KL(N_0\|N_1) = \frac{1}{2} \Bigg[\ln \left(\frac{\det \Sigma_1}{\det \Sigma_0} \right) + tr(\Sigma_1^{-1} \Sigma_0) \\ + (\mu_1 - \mu_0)^T \Sigma_1^{-1} (\mu_1 - \mu_0) - \frac{d}{2} \Bigg]. \end{aligned} \tag{2}$$

We make the distance measure symmetric by setting the average of $KL(p\|q)$ and $KL(q\|p)$ as the distance $d(p, q)$ between two clusters p and q. A few additional terms are needed to discuss our objective function.

2.3 Basic Definitions

Definition 1. Cohesion Index(CI): Given a set of clusters from N patients, $S = \{u_1, u_2 ... u_N\}$ such that $u_i \in P^i$, and $d(u_i, u_j)$ is the distance between clusters u_i and u_j, the Cohesion Index of the set S is the average distance between pairs of clusters (u_i, u_j) in the set S:

$$CI(S) = \frac{2}{N(N-1)} \sum_{\substack{u_i, u_j \in S \\ i<j}} d(u_i, u_j). \tag{3}$$

A small value of CI means that the clusters in S are similar, while large values indicate that they are dissimilar. The Cohesion Index CI for set S consisting of clusters represented by the large filled circles (within the circles denoting the P^i vertices) in Fig. 1 is the sum of the weights of the edges joining these clusters divided by ten.

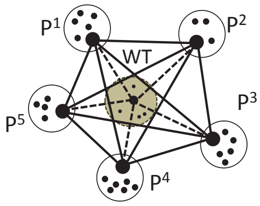

Fig. 1. Graph model of data with 5 patients and 1 Wild Type. The data from each individual has been clustered; vertices in the graph are the clusters, and the edge weights are derived from the Kullback-Leibler divergences between clusters.

Definition 2. Divergence Index(DI): Given a set of clusters from N patients, $S = \{u_1, u_2...u_N\}$ such that $u_i \in P^i$, and $d(w, u_i)$ is the KL-divergence between clusters $w \in WT$ and $u_i \in S$, the Divergence Index (DI) is the minimum sum of distances between each pair (w, u_i) in the set S:

$$DI(S) = \frac{1}{N} \min_{w \in WT} \left\{ \sum_{u_i \in S} d(w, u_i) \right\} . \tag{4}$$

A large value of DI means clusters in S are dissimilar from any WT cluster, while a small value of DI means the clusters in S are similar to some cluster in WT. In Fig. 1, the central grey circle represents the WT sample, the large filled circle within it corresponds to a cluster in WT with the least sum of distances from a set S of patient clusters denoted by the filled circles, and the average length of the edges between the WT and patient clusters yields DI for the set S.

To identify groups of similar outliers we look for sets of clusters S with low values of $CI(S)$ and high values of $DI(S)$. However, maximizing $(DI(S)-CI(S))$ does not suffice to guarantee both a low value of CI and a high value of DI. This observation leads to the next definition.

Definition 3. Coherence Confidence (CC): Given a set of clusters $S = \{u_1, u_2...u_N\}$ such that $u_i \in P^i$, the Coherence Confidence (CC) is the product of the normalized difference between $DI(S)$ and $CI(S)$ and a damping factor:

$$CC(S) = \frac{DI(S) - CI(S)}{DI(S) + CI(S)} \left[1 - a^{-(CI(S)+DI(S))} \right] , \tag{5}$$

where a is a constant greater than one. The damping factor prevents the ratio from becoming unstable for small values of the sum $CI(S) + DI(S)$. If this sum is small, then the factor is small enough to keep the value of $CC(S)$ low. As the sum increases, the factor increases to its maximum value of one, and does not significantly influence the CC value. The range of values of CC is $[-1, 1]$. The constant a in the damping factor is chosen so that it should not grow too quickly to its maximum value. We tried various values for the constant and $a \sim 1.2$ worked reasonably well with the data here. We will identify groups S consisting of similar outlying clusters from sets with large positive values of $CC(S)$.

2.4 Objectives

We now state the objectives of our analysis.

1. **Pairwise Outliers:** Identify dissimilar clusters in a diseased sample by pairwise comparison with WT. These are pairwise outliers which contain both distinctive and common outliers described below.
2. **Distinctive Outliers:** Identify the clusters in a diseased sample that are dissimilar to any WT clusters as well as clusters from other diseased samples. These are *distinctive outliers* that fail to form groups with low values of CI.
3. **Common Outliers:** Identify group of similar outliers, i.e., groups with members similar to each other in diseased samples but dissimilar to any WT cluster. These *common outliers* have high values for CC.

3 Methods

3.1 Clustering

We denote the flow cytometry data from a sample as $X = [x_1^T, x_2^T, \ldots, x_N^T]$, where x_i^T corresponds to the data from the i-th cell. We assume the data are generated from a hierarchical Bayesian model. First, the observation x_i is sampled from a likelihood function $f(\theta_i)$ where θ_i is the likelihood parameter for the i-th observation. Second, the parameter θ_i follows a distribution G, which is sampled from a Dirichlet process $DP(\alpha, G_0)$ with a concentration parameter α and a base distribution G_0. Note that the use of a Dirichlet prior will make many $\{\theta_i\}$'s share the same value, naturally inducing clustering of data. The model is known as the *Dirichlet Process Mixture* (DPM) model [1,9] and can be summarized as follows:

$$x_i|\theta_i \sim F(\theta_i), \quad \theta_i|G \sim G, \quad G \sim DP(\alpha, G_0), \tag{6}$$

where $X \sim S$ means that X follows the distribution S. Since G is a distribution, $G \sim DP(\alpha, G_0)$ suggests that the Dirichlet Process $DP(\alpha, G_0)$ is a distribution over distributions.

We used a publicly available Matlab implementation of DPM clustering by Teh [14], which is based on a Chinese Restaurant Process representation of the DPM model and uses simple Gibbs sampling. The computational cost per iteration is $O(Nd^2k)$, where N is the number of rows in a data sample matrix

X, d is the number of columns of X, and k is the number of clusters in the current iteration (k changes with iterations). The value of N is large (see Table 1 in Sec. 4), which makes each iteration computationally expensive even for a small number of clusters. Moreover, the quality of clustering improves with the number of iterations. In our experiments, a hundred iterations work reasonably well for the data as the clustering changes relatively little after that.

Although the DPM inference is expensive for large samples, we prefer the nonparametric model, DPMs, over classical parametric cluster methods, e.g, K-means. The reason is that the computational cost of selecting the number of clusters for a parametric model is prohibitively expensive for flow cytometry data analysis. DPMs circumvents the model selection problem by automatically determining the number of clusters for each sample, making it well suited as a clustering tool before the application of our outlier detection algorithms.

The DPM model is an infinite model in the sense that it contains a mixture of countably infinite components. For example, if $F(\cdot)$ is a Gaussian distribution, the DPM model can be viewed as a mixture of infinite Gaussians [11]. Given a finite number N data points, however, we compute the posterior distribution of the DPM model using Bayes theorem; the expected number of components in the posterior distribution is always finite and, often, much smaller than the number of data points.

3.2 Pairwise Comparison: Generalized Edge Cover

One method we used to identify outliers in the clustered data is pairwise comparison between samples. We model a pair of samples, say A_1 and A_2, using a complete bipartite graph with each cluster represented by a vertex, and edges joining pairs of clusters in different samples. Formally $G = (V_1, V_2, E)$ is a complete bipartite graph, where V_1 contains all clusters from A_1, V_2 contains all the clusters from A_2, and the edge weight function is $c : E \to \mathbb{R}$ where c_{ij} is the weight of edge $\{u_i, u_j\}$, with $u_i \in V_1$ and $u_j \in V_2$. The weight of an edge is the average KL divergence of its endpoint clusters. In this bipartite graph we seek to identify clusters that are common to samples A_1 and A_2, and also those that belong to only one sample.

Since low edge weight implies high similarity among clusters we could find a minimum-weight matching among all maximum cardinality matchings in the graph G and declare unmatched vertices to be outliers. However, this attempt at a model for outlier detection has a significant drawback. Since the number of clusters in the two samples is generally not the same, some clusters will remain unmatched even if they are highly similar to another cluster, and should not be identified as outliers. We address this issue by formulating the problem as a minimum-weight edge cover on a complete bipartite graph. An *edge cover* is a subset of edges such that each vertex in the graph has *at least* one edge incident on it, whereas a *matching* is a subset of edges such that each vertex in the graph has *at most* one edge incident on it. However, even an edge cover fails to accurately model the problem since clusters that represent outliers should not be covered in an edge cover. Hence we find a *generalized edge cover* that permits some vertices not to be covered, at the cost of a penalty, by adding a weight λ for each uncovered vertex to the weight of an edge cover. Thus λ acts as a cut-off value for long edges that would not be included in a generalized edge cover. This leads to a generalized edge cover formulation of the problem, where the cover EC leaves some uncovered vertices $V_{uc} \subseteq V_1 \cup V_2$, while minimizing the objective function:

$$min \left(\sum_{(v_i,v_j) \in EC} c_{ij} + \lambda * |V_{uc}| \right). \tag{7}$$

A generalized edge cover in G can be computed from an edge cover in a transformed graph G'. Let the graph G' be obtained from G by introducing two new distinguished vertices $v_1 \in V_1$ and $v_2 \in V_2$, and adding an edge $\{v_1, v_2\}$ with $c(\{v_1, v_2\}) = 0$, and edges $\{v_1, u_2\}$ for each $u_2 \in V_2$, $\{v_2, u_1\}$ for each $u_1 \in V_1$, with $c(\{u_1, v_2\}) = c(\{u_2, v_1\}) = \lambda$. If a minimum-weight edge cover includes added edges with weight λ, for each such edge, we leave the original vertex in G incident on this edge uncovered in a generalized edge cover of the original graph, thus paying a price of λ for the vertex, without changing the weight or structure of the remaining edge cover.

A minimum-weight edge cover in a graph can be computed in polynomial time by making a copy of the graph and connecting each vertex to its twin in the copy by an edge with weight equal to twice the minimum weight among original edges incident on it. A minimum-weight perfect matching in this graph can be used to compute a minimum-weight edge cover in the original graph [13].

Following the above discussion, our pairwise comparison algorithm for outlier detection can be formulated in the following stages:

1. **Pre-processing:** Add distinguished vertices $v_1 \in V_1$ and $v_2 \in V_2$, and an edge $\{v_1, v_2\}$ with $c(\{v_1, v_2\}) = 0$. Given a cut-off value λ, add edges $\{v_1, u_2\}$ for each $u_2 \in V_2$, and $\{v_2, u_1\}$ for each $u_1 \in V_1$, all with a weight of λ. Let $G' = (V', E')$ be the resulting graph.
2. **Duplicate Graph:** Let $\tilde{G}' = (\tilde{V}', \tilde{E}')$ be a disjoint copy of G'. Let $\bar{G}$ be the the graph formed by taking the union of G' and $\tilde{G}'$ and adding an edge $\{v, \tilde{v}\}$ connecting every $v \in V'$ with its twin $\tilde{v} \in \tilde{V}'$. Let $c(\{v, \tilde{v}\}) = 2\mu(v)$ for each $v \in V'$, where $\mu(v)$ is the minimum weight of the edges of G' incident on v.
3. **Matching:** Compute a minimum-weight perfect matching M in $\bar{G}$.
4. **Edgecover:** Obtain a minimum-weight edge cover EC' of G' by replacing every edge $\{v, \tilde{v}\} \in M$ by an edge of weight $\mu(v)$ in G' incident on v.
5. **Post-processing:** Remove all edges $\{v_1, o\}$, $\{v_2, o\}$ from EC', where o denotes an original vertex in $V_1 \cup V_2$; add each vertex o to the set of outliers O. Remove the distinguished vertices v_1 and v_2 from EC'. The resulting edge cover EC^* together with the set of uncovered vertices O is a solution to the generalized edge cover problem in G.

Lemma: The Algorithm described above computes an optimal generalized edge cover in G.

Proof: The correctness of the algorithm for computing the edge cover EC' in the graph G' was shown in [13]. We obtain a generalized edge cover in G by deleting, the vertices v_1 and v_2, the edges incident on these vertices in EC', and all vertices adjacent to these vertices in EC'. Let O be the set of the deleted vertices adjacent to v_1 and v_2, which will be identified as outliers. Let EC'' be the edges remaining from the edge cover EC' in the modified graph $G'' = G \setminus O$.

We claim that EC'' together with O is an optimal solution for the generalized edge cover problem in G. Assume that there is an optimal solution in G consisting of a set of outliers O and an edge cover EC^* in $G \setminus O$. Clearly $c(EC'') = c(EC^*)$, for otherwise one of the solutions could be improved upon, thereby contradicting their minimality in G'' or $G \setminus O$ respectively. It remains to prove that there is no solution in G with a different outlier set and smaller cost. Let $EC^{*'}$ together with O' be such a solution for G having a smaller cost $c' < c(EC')$. Then $EC^{*'}$ together with an edge $\{o, v_1\}$ or $\{o, v_2\}$ for every $o \in O'$ and the edge $\{v_1, v_2\}$ is an edge cover in G' with cost $c' < c(EC')$, contradicting the optimality of EC'. □

Thus we obtain a generalized edge cover. Note that a vertex $u \in V_k$, where $k = 1$ or 2, and $\mu(u)$ is the minimum weight among the edges of G incident on u, will always be an outlier if $\mu(u) \geq 2\lambda$, and can never be an outlier if $\mu(u) < \lambda$. Otherwise, it will be an outlier if and only if it is not matched to a vertex in G' during step 3 of the algorithm.

For a graph with n vertices and m edges, an edge cover of minimum weight can be computed in time $O(n(n + m \log n))$ [13]. In this context, since there are at most K clusters in each patient, $n \leq 2K$, and $m \leq K(K-1)/2$, and thus the time complexity of pairwise comparison to identify outliers is $O(K^3 \log K)$.

3.3 Comparing Multiple Clusters

Formation of Coherent Groups. We now consider an approach that compares multiple diseased samples to identify clusters common to them but not belonging to the Wild Type. A *group* of clusters S is a set of distinct clusters from each patient, $S = \{u_1, u_2 ... u_N\}$, with $u_i \in P^i$. In Sec. 4.5 we relax this to form groups that do not cover all patients. To identify common outliers we find such groups that exhibit high similarity among their clusters while being dissimilar to the Wild Type.

A graph representation of a group S of clusters is a clique consisting of one cluster from each patient. The cost of a group S is the average weight of all the edges of the corresponding clique, which is the Cohesion Index $CI(S)$. It is easy to show that finding a group with minimum CI score is NP-hard via a reduction from MAX-CLIQUE. To identify groups with low CI score we use a branch and bound technique, which provides good performance for a reasonable number of clusters and patients. We omit the details here due to space considerations.

The branch and bound procedure is called once with each cluster as seed, and it finds a group with minimum cost $CI(S)$ containing the seed cluster, resulting in NK groups in total. The method works very well in practice, although it has a worst-case running time exponential in N. Since the distance measure is not metric, no obvious approximation guarantee exists.

Once we have obtained coherent groups of clusters with small CI scores, we calculate the DI and CC scores for those sets. Since we have a single Wild Type sample with K clusters we can find the minimum Divergence Index for a group S in $O(NK)$ time. The decision about distinctive and common outliers is based on the following rules:

Distinctive Outliers: If a group has high value for CI, then declare the seed cluster of that set as a distinctive outlier, since it fails to form a close group with clusters from other patients.

Common Outliers: Among the remaining groups with small CI scores, find those groups having large CC values. These sets are close to one another while being distinct from any WT cluster.

4 Results

4.1 Clustering Results

We cluster each normalized sample using a DPM clustering algorithm and the results are shown in Table 1. All subsequent downstream analysis is built upon these clustering results.

Table 1. Clustering the flow cytometry data for two datasets. WT represents the Wild Type, P^i denotes a leukemic mouse, and H is a pre-leukemic mouse with an oncogene expressed.

Dataset-1				Dataset-2			
Sample	Dimension	#Cells	#Clusters	Sample	Dimension	#Cells	#Clusters
WT	6	115,407	18	WT	7	49,316	21
H	6	131,850	23	H	7	68,886	22
P^1	6	107,299	22	P^1	7	78,406	21
P^2	6	131,575	28	P^2	7	6,050	12
P^3	6	236,392	31	P^3	7	48,998	21

The computational cost of the clustering step using the Matlab DPM code is about six to ten hours depending on the dataset size, while the edge cover and branch and bound computations run under a minute on a 3 GHz PC. The DPM clustering code should be much faster when it is implemented efficiently in a non-interpreted environment, but it would still be the dominant cost of the current computation. Improving its performance was not the scope of this work.

4.2 Pairwise Comparison Results

The generalized edge cover approach compares leukemic mouse samples with WT and identifies outliers depending on a cut-off value λ. The optimal cut-off value is dependent on the Kullback-Leibler divergences of the clusters involved. The number of outliers is inversely related to λ in that a large value of λ yields very few outliers, and vice versa. A plot is shown in Figure 2.

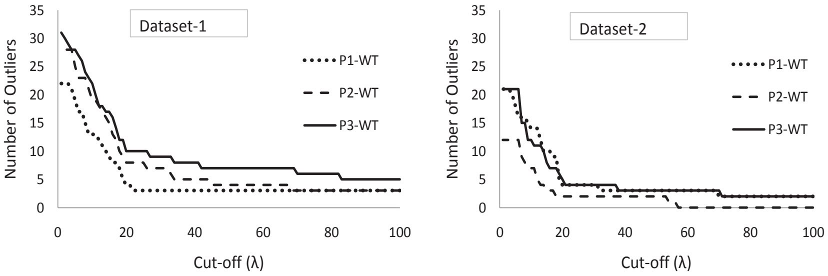

Fig. 2. Outliers from pairwise comparison between WT and leukemic samples with different cut-off (λ) values for two datasets.

The outlier profile in both datasets shows a sharp change approximately at $\lambda = 20$, which we choose to be a good cut-off value for the detection of extreme outliers. Table 2 shows all the outliers obtained for two different values of λ. Note that pairwise comparison of a leukemic sample with the WT sample cannot distinguish between distinctive and common outliers.

Table 2. Outlying clusters in leukemic samples (Dataset 1) for two cut-off λ values

Sample	Outliers at $\lambda = 20$	Outliers at $\lambda = 10$
P^1	2,10,13,14	2,3,10,12,13,14,16,17,18,19,21
P^2	4,12,17,18,20,24,25,28	2,3,4,11,12,16,17,18,19,20,21,22,24,25,26,27,28
P^3	11,12,13,16,17,18,19,20,21,29	8,9,11,12,13,15,16,17,18,19,20,21,23,24,26,28,29,31

4.3 Coherent Groups

Every cluster in each sample is used as a seed cluster to construct a group S with a minimum value of the Cohesion Index. The significance of the CI scores of the identified groups can be assessed using the permutation test [8]. We randomly select one cluster from each leukemic mouse to form a group and construct $N_{perm} = 100,000$ random groups in total. For any (non-randomly constructed) group S, let N_S be the number of random groups (S_{rand}) having $CI(S_{rand}) \leq CI(S)$. The significance measure, the p-value of S, can then be calculated as $p(CI(S)) = (N_S + 1)/(N_{perm} + 1)$. Groups with small $p(CI(S))$ values are significant since the chance of finding them at random is small. The histogram of the N_{perm} permutations is shown in the left subfigure in Figure 3 with the broken vertical line indicating 5% confidence level. We observe that most of the non-random groups fall within the 5% confidence interval.

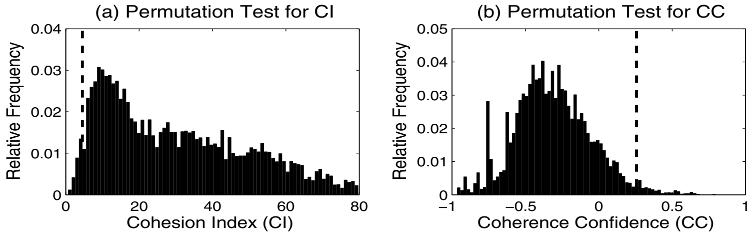

Fig. 3. Histogram of the permutation tests for CI (left) and CC (right) scores from Dataset 1. Groups at a 5% confidence level are to the left of the broken vertical line for CI scores, and to the right of the broken vertical line for CC scores.

Table 3. Groups with CI scores and p-values of CI scores. Seed clusters are shown in grey and distinctive outliers are shown in boxed squares.

Dataset-1					Dataset-2				
P1	P2	P3	CI	$p(CI)$	P1	P2	P3	CI	$p(CI)$
1	7	8	0.62	0.00024	3	6	4	1.232	0.00064
2	$\boxed{4}$	18	296.7	-	$\boxed{6}$	8	9	24.495	-
4	7	8	1.404	0.0002	8	10	14	1.204	0.00043
$\boxed{10}$	26	18	154.658	-	10	10	7	0.627	0.00012
11	13	14	1.27	0.00013	$\boxed{11}$	4	12	44.63	-
$\boxed{14}$	1	30	74.604	-	12	5	15	3.397	0.00815
15	11	28	3.031	0.00335	13	5	15	3.918	0.01196
15	11	$\boxed{21}$	49.91	-	14	4	$\boxed{10}$	107.167	-
17	21	17	1.675	0.00046	18	5	15	3.326	0.0076
18	26	31	3.054	0.00345	21	12	21	7.234	0.053099

Several representative groups with their p-values are presented in the Table 3, where seed clusters are highlighted in grey. Notice that multiple seeds may construct the same group (e.g., $\{4, 7, 8\}$ in dataset-1). Such groups are usually tight with low p-values. Also notice the three groups in $\{12, 5, 15\}$, $\{13, 5, 15\}$, $\{18, 5, 15\}$ in Dataset-2, where the same clusters from P^2 and P^3 are grouped with different clusters from P^1 with similar CI scores. If clusters 12, 13, 18 from P^1 all have small KL divergence from each other, then merging these three clusters produces a unified cluster. Thus we can use the group formation approach to refine clusters obtained from the clustering algorithm.

4.4 Distinctive and Common Outliers

Seed clusters that fail to form a group with significantly low CI scores are declared as *distinctive outliers* and are shown in boxed squares in Table 3. The p-values for such groups will be large, bearing little significance in this context.

Common outliers are groups of clusters with high Cohesion Confidence (CC) values, and do not have a distinctive outlier as a member. We performed the permutation test on the CC value to assess its significance in the same way as we did for the CI score. However, we are now interested in the confidence limit to the right side of the broken vertical line in the histogram in the right subfigure of Figure 3. Again, we find that groups with high CC values are significant since the chance of finding them at random is small. We report distinctive and common outliers discovered by the clique approach in Table 4. All distinctive and common outliers identified by this approach were also identified by the pairwise comparison approach (Table 2), but the converse is generally not true. Hence the clique approach is more powerful in detecting and classifying outliers than the edge cover approach.

Table 4. Distinctive and Common Outlying clusters identified by the weighted clique approach. Here u_i is a cluster belonging to a leukemic mouse P^i.

Distinctive Outliers				Common Outliers					
Dataset-1		Dataset-2		Dataset-1			Dataset-2		
Sample	Clusters	Sample	Clusters	$\{u_1, u_2, u_3\}$	CC	$p(CC)$	$\{u_1, u_2, u_3\}$	CC	$p(CC)$
P^1	2,10,14,16	P^1	6,11	17,21,17	0.64	0.00007	10,10,7	0.78	0.00013
P^2	4,20	P^2	4	1,7,8	0.63	0.00011	4,9,1	0.64	0.00078
P^3	12,16,19,20,21	P^3	5,10	9,5,3	0.58	0.00015	17,2,6	0.612	0.0016

4.5 Probabilistic Model for Groups

We describe a probabilistic model to refine the groups by relaxing our initial requirement that a group must necessarily include one cluster from each patient. Let f_{ij} be the number of times two clusters u_i and u_j are grouped together, and let $f_i = \sum_{j \neq i} f_{ij}$ be the number of times cluster u_i appears in any group. Then for a group $S = \{u_1, u_2 ... u_N\}$ the probability of u_i being a member of S, $P(u_i|S)$, and the probability of the whole group, $P(S)$ can be calculated by

$$P(u_i|S) = \frac{\sum_{\substack{u_j \in S \\ i \neq j}} f_{ij}}{f_i}, \quad \text{and} \quad P(S) = \prod_{\substack{u_i \in S \\ 1 \leq i \leq N}} P(u_i|S). \tag{8}$$

Within a group, a low $P(u_i|S)$ value and high $P(u_j|S)$ value for all $j \neq i$, suggests that u_i is a member with weak cohesion to S. We can refine the group S by deleting the weak member u_i, and inserting a gap in that position. A high $P(u_i|S)$ and low $P(u_j|S)$, for all $j \neq i$, also indicates a weakly formed group. In this case, we allow u_i to form a group by itself deleting all other members of S, making u_i a distinctive outlier. We present several representative groups from Dataset-1 in Table 5, where clusters with high support are marked in grey.

Consider the groups in rows 2 and 3 of the Table. Each has one weak member(u_1 and u_3, respectively) that can be safely removed from the corresponding groups. However, rows 4 and 5 show groups with only one strong member (u_2 and u_1, respectively), which can be declared as a distinctive outlier by removing all other members from the groups. The group in the sixth row has a low probability.

Table 5. Group-uniqueness probabilities for Dataset-1. Clusters in grey have the highest probabilities of belonging to the group.

u_1	$P(u_1\|S)$	u_2	$P(u_2\|S)$	u_3	$P(u_3\|S)$	$P(S)$
11	1	13	1	14	1	1
17	.29	16	1	6	1	.29
19	.83	22	.83	28	.33	.23
2	.33	4	1	18	.33	.11
21	1	6	.20	27	.25	.05
18	.17	19	.23	27	.25	.01

4.6 Effect of APL on Bone Marrow Cells

Wojiski et al. [15] compared the populations of a number of cell types in the bone marrow of WT, leukemic and pre-leukemic (with oncogene PML-RARα expressed in the latter two groups) mice. They reported that WT and pre-leukemic (H) mice had similar cell populations of hematopoietic stem cells (LSKs), common myeloid progenitor cells (CMPs), granulocyte-monocyte progenitor cells (GMPs), and megakaryocyte erythrocyte progenitor cells (MEPs); but in leukemic mice (P) cell populations of LSKs, CMPs and MEPs are reduced and GMPs are increased, relative to the WT and pre-leukemic mice. They also found that mature granulocytes were increased in pre-leukemic mice relative to WT.

In a pairwise comparison of flow cytometry data from WT and H using the edge cover approach, we found that of the 18 clusters in WT and 23 clusters in H in the first data set, only 3 clusters from each set were left uncovered when a value $\lambda = 20$ was used. Similar results were obtained for the second data set also, confirming the general correspondence of populations of various cell types in these two kinds of mice. Generally, a group of clusters from leukemic mice that has a high value of CC (hence it is distant from any cluster in the WT) also has a high value of CC when clusters from a pre-leukemic mouse are used in the place of WT. However, we found some clusters in the pre-leukemic mouse that were closer to the leukemic mice rather than the WT. We performed this experiment by treating the pre-leukemic sample as an additional leukemic sample, and using the branch and bound algorithm to identify sets with high CC values. In Dataset-1, we found the clusters $\{8, 14, 15, 5\}$ and $\{17, 21, 17, 18\}$; and in Dataset-2, we found the clusters $\{4, 9, 1, 4\}$ and $\{10, 10, 7, 8\}$; here in each set the first three clusters are from leukemic mice and the last is from the pre-leukemic mouse, and these clusters are all distant from any cluster in the WT. Identifying these specific cell types through further experimental work could shed light on disease progression in the murine model of APL.

Acknowledgments. This research was supported through a PRF grant from the College of Science at Purdue, NSF grant CCF-0830645, and Department of Energy grant DE-FC02-08ER25864 (CSCAPES Institute).

References

1. Antoniak, C.E.: Mixtures of Dirichlet processes with applications to Bayesian nonparametric problems. Annals of Statistics 2(6), 1152–1174 (1974)
2. Bashashati, A., Brinkman, R.: A survey of flow cytometry data analysis methods. In: Advances in Bioinformatics, pp. 1–19 (December 2009)
3. Boedigheimer, M., Ferbas, J.: Mixture modeling approach to flow cytometry data. Cytometry A 73, 421–429 (2008)
4. Chan, C., Feng, F., Ottinger, J., et al.: Statistical mixture modeling for cell subtype identification in flow cytometry. Cytometry A 73(A), 693–701 (2008)
5. Herzenberg, L., Tung, J., Moore, W., et al.: Interpreting flow cytometry data: A guide for the perplexed. Nature Immunology 7(7), 681–685 (2006)
6. Kullback, S.: Information Theory and Statistics. Dover Publications Inc., Mineola (1968)
7. Meur, N., Rossini, A., Gasparetto, M., Smith, C., Brinkman, R., Gentleman, R.: Data quality assessment of ungated flow cytometry data in high throughput experiments. Cytometry A 71A, 393–403 (2007)
8. Moore, D., McCabe, G.: Introduction to the Practice of Statistics. W. H. Freeman & Co., New York (2006)
9. Neal, R.: Markov chain sampling methods for Dirichlet process mixture models. Journal of Computational and Graphical Statistics 9, 249–265 (2000)
10. Pyne, S., Hu, X., Wang, K., et al.: Automated high-dimensional flow cytometric data analysis. PNAS 106(21), 8519–8524 (2009)
11. Rasmussen, C.E.: The infinite Gaussian mixture model. In: Solla, S., Leen, T., Muller, K.R. (eds.) Advances in Neural Information Processing Systems, vol. 12. MIT Press, Cambridge (2000)
12. De Rosa, S., Brenchley, J., Roederer, M.: Beyond six colors: A new era in flow cytometry. Nature Medicine 9(1), 112–117 (2003)
13. Schrijver, A.: Combinatorial Optimization — Polyhedra and Efficiency, Volume A: Paths, Flows, Matchings. Algorithms and Combinatorics, vol. 24. Springer, New York (2003)
14. Teh, Y.W.: DPM Software (2010), `http://www.gatsby.ucl.ac.uk/~ywteh/research/software.html`
15. Wojiski, S., Gubal, F.C., Kindler, T., et al.: PML-RARα initiates leukemia by conferring properties of self-renewal to committed promyelocytic progenitors. Leukemia 23, 1462–1471 (2009)