

METHODOLOGY ARTICLE

Open Access

# Data reduction for spectral clustering to analyze high throughput flow cytometry data

Habil Zare<sup>1,2</sup>, Parisa Shooshtari<sup>1,2</sup>, Arvind Gupta<sup>3</sup>, Ryan R Brinkman<sup>2,4\*</sup>

## Abstract

**Background:** Recent biological discoveries have shown that clustering large datasets is essential for better understanding biology in many areas. Spectral clustering in particular has proven to be a powerful tool amenable for many applications. However, it cannot be directly applied to large datasets due to time and memory limitations. To address this issue, we have modified spectral clustering by adding an information preserving sampling procedure and applying a post-processing stage. We call this entire algorithm SamSPECTRAL.

**Results:** We tested our algorithm on flow cytometry data as an example of large, multidimensional data containing potentially hundreds of thousands of data points (*i.e.*, “events” in flow cytometry, typically corresponding to cells). Compared to two state of the art model-based flow cytometry clustering methods, SamSPECTRAL demonstrates significant advantages in proper identification of populations with non-elliptical shapes, low density populations close to dense ones, minor subpopulations of a major population and rare populations.

**Conclusions:** This work is the first successful attempt to apply spectral methodology on flow cytometry data. An implementation of our algorithm as an R package is freely available through BioConductor.

## Background

High throughput data analysis is a crucial step in research endeavours involving gene expression, protein classification, and flow cytometry. A classical approach for analysing biological data is to first group individual data points based on some similarity criterion, a process known as clustering, and then compare the outcome of clustering with the biological hypotheses. An example of this approach is in the analysis of flow cytometry data where populations of cells that express specific intracellular or surface proteins are identified. Flow cytometry is a technique for measuring physical, chemical and biological characteristics of individual microscopic particles such as cells and chromosomes. It has many applications in molecular and cell biology for both clinical diagnosis and research purposes [1]. In cytometers, cells are individually passed through a laser beam and the scattered light is captured to measure up to 19 characteristic of each cell [2]. As thousands of cells can be analyzed per second, cytometers can generate large-

sized datasets. Recently, sophisticated methods have been developed for automatic analysis of flow cytometry data [3-5]. The proposed clustering techniques include: mixture modeling approach [6], model-based cluster analysis [7], feature-guided clustering [8], density-based clustering [9], combining the curvature information with density information [10], and image processing [11]. The automatic techniques are useful in clinical and research applications such as: application of high-content flow cytometric screening (FC-HCS) to the problem of cellular signature definition for acute graft-versus-host-disease [12], vaccine trials [13], visualizing data in stem cell research [14], and immunophenotypic characterization of B-cell chronic lymphoproliferative disorders (B-CLPD) [15].

## Problem Statement

Automated identification of flow cytometry cell populations is complicated by overlapping and adjacent populations, especially when low and high density populations are close to each other. Analysing such data requires clustering methods that can separate these populations. Non-parametric methods include density clustering [16], real-time adaptive clustering [17], and Kohonen

\* Correspondence: rbrinkman@bccrc.ca

<sup>2</sup>Terry Fox Laboratory, BC Cancer Agency, 675 W 10th Ave., Vancouver, BC, Canada

Full list of author information is available at the end of the article

self-organizing maps [18]. The application of these methods is restricted since the first two are subjective due to a dependency on user-defined thresholds, and the latter one requires the number of clusters to be determined by the user. While accurately determining the number of clusters may not be a key issue in some clinical cytometry analysis [19], this requirement can be a critical obstacle for other analyses such as identifying novel populations for biomarker discovery [3].

Model-based clustering techniques such as FLAME [20], flowClust [21] and flowMerge [22] have been developed to improve results. flowMerge uses the flowClust framework to identify clusters based on a t-mixture model methodology, followed by a merging step to account for overestimation of the number of clusters by the Bayesian information criterion. FLAME uses a skew t-mixture model, which is in theory more robust to skew, because unlike t-distributions, skew t-distributions can be asymmetric [20]. However, the running time of this algorithm increases with the fourth degree of the number of dimensions. In practice this tends to make the algorithm impractical for more than five dimensions, while flow cytometry data can have up to 19 dimensions. Overall, the major drawback of these parametric methods is the requirement for assumptions on either the size of the clusters or the cluster distributions and shapes [23], which could result in incorrect identification of biologically interesting populations. In addition, one challenge for existing approaches is the identification of rare populations. Spectral clustering is a non-parametric clustering method that avoids the problems of estimating probability distribution functions by using a heuristic based on graphs [24]. It has proved useful in many pattern recognition areas [25-28]. Not only does it not require *a priori* assumptions on the size, shape or distribution of clusters, but it has features that make it particularly well-suited to clustering biological data:

- It is not sensitive to outliers, noise or shape of clusters;
- It is adjustable so that biological knowledge can be utilized to adapt it for a specific problem or dataset;
- There is mathematical evidence to guarantee its proper performance [29].

Two main challenges in applying spectral clustering algorithm on large data sets are the computationally expensive steps of constructing the normalized matrix and computing its eigenspace. For instance, for high throughput biological data containing one million data points (*i.e.*, vertices), it requires computing eigenspace of a million by million matrix, which is infeasible in terms of memory and time. Although there are some approximation methods for speeding up this computation [30,31], these could produce undesired errors in the final results. The problem of

applying this algorithm on large datasets has been studied in [32] using Nyström's method. They suggest a strategy of sampling data uniformly, clustering the sampled points and extrapolating this solution to the full set of points. However, sampling data uniformly can miss low-density populations entirely when the density of adjacent populations varies considerably, a situation that often arises for biologically interesting populations in flow cytometry data. Appendix 3 includes an experiment to explain the effect of uniform sampling in such cases.

Data reduction schemes have been developed to reduce the complexity of the flow cytometry data while preserving the information [33,34]. These methods reduce the dimensionality but not the size of the dataset, the latter being the more important bottleneck for spectral clustering.

### Our Approach

We hypothesized that spectral clustering could significantly improve high throughput biological data analysis. However, serious empirical barriers are encountered when applying this method to large data sets. Specifically, for  $n$  data points, the running time is  $O(n^3)$ , requiring  $O(n^2)$  units of memory. For instance, it would take 2 years and 5 terabytes of memory to analyze a typical flow cytometry sample with 300,000 events. We developed a novel solution for this problem through our non-uniform information preserving sampling. Our heuristic approach is specific to cytometry applications and made it possible for the first time, to apply spectral clustering method on flow cytometry data.

### Results

In this paper, we distinguish between the terms biological populations, clusters and components as follows. A *population* is a set of cells with similar functionality or molecular content. By a *cluster*, we mean a set of data points that are grouped together by spectral clustering algorithm. We incorporate a post-processing stage on spectral clusters to find the *connected components* intended to estimate the biological populations.

### Algorithm

#### Spectral Clustering

The first step is to build a graph. The vertices represent the  $n$  data points (*e.g.*, cells in flow cytometry data), and the edges between the vertices are weighted based on some *similarity* criterion. The adjacency matrix of the graph is then normalized using the following formula:

$$\tilde{A} = D^{-\frac{1}{2}} A D^{-\frac{1}{2}}, \quad (1)$$

where  $A$  is the adjacency matrix of the graph and  $D$  is a diagonal matrix where the  $(i, i)$  entry is equal to the

sum of the weights on the edges that are adjacent to vertex  $i$ .

The next step is to compute eigenspace of the normalized matrix. That is, all vectors  $V_i$  and values  $\lambda_i$  satisfying the following equation are computed:

$$\tilde{A}\bar{V}_i = \lambda_i\bar{V}_i. \quad (2)$$

In order to find  $k$  clusters, an  $n$  by  $k$  matrix is built using the  $k$  eigenvectors with highest eigenvalues. The rows of this matrix are normalized and finally  $k$ -means is used to cluster the rows.

However, the above method cannot be directly applied to flow cytometry data due to large number of data points (cells) per sample. Our solution for this problem is a data reduction scheme developed specifically for this purpose. This reduces the number of vertices significantly, but in a way such that biological information can be preserved by updating the weights on the edges.

#### Data Reduction Scheme

While data size can be reduced by known sampling methods [35], a very delicate method should be used to preserve biologically important information. From a high-level perspective, our data reduction scheme (Figure 1) consists of two major steps; first we sample the data in a representative manner to reduce the number of vertices of the graph (Figure 1b). Sample points cover the whole data space uniformly (Figures 2b), a property that aids in the identification of both low density and rare populations. In the second step as described below, we define a similarity matrix that assigns weights to the edges between the sampled data points. Higher weights are assigned to the edges between nodes in dense regions so that information about the density is preserved (Figure 1c).

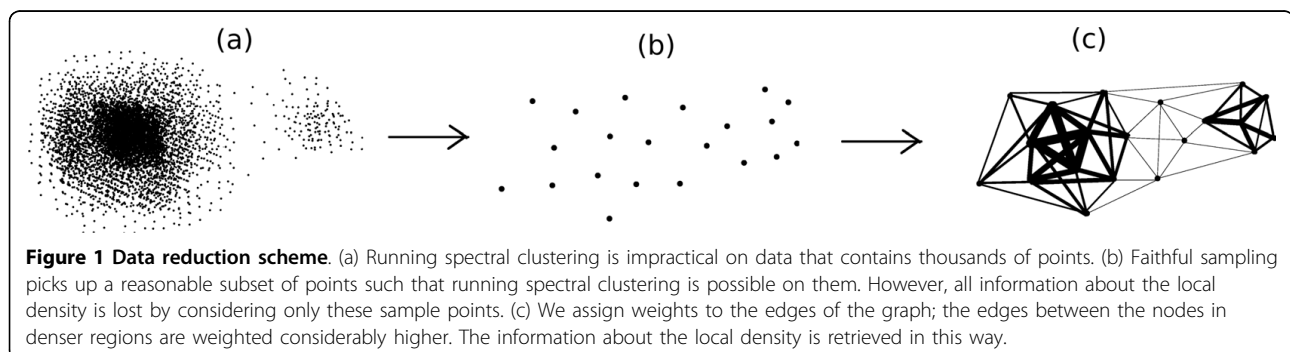
#### Faithful Sampling Algorithm

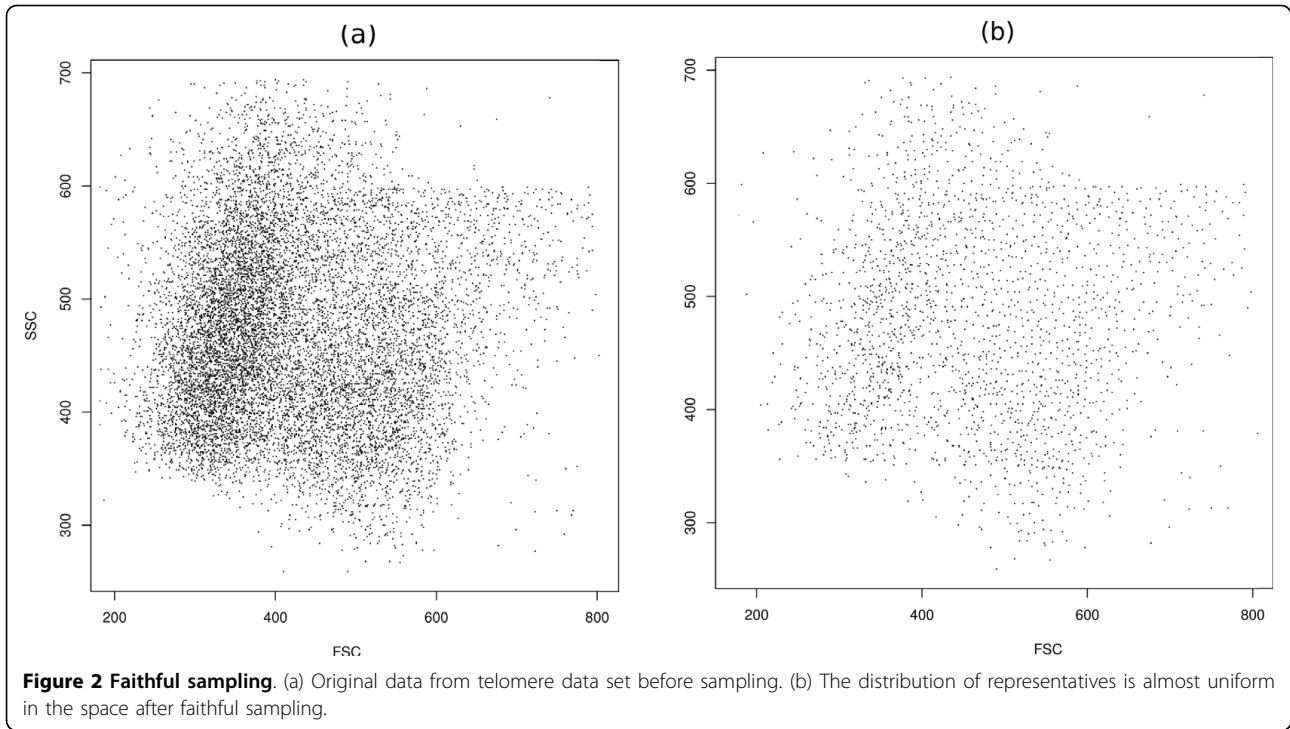
1. Label all data points as *unregistered*.
2. **repeat**
3. Pick a random unregistered point  $p$  {the representative of a new community}

4. Label all unregistered data points within distance  $h$  from  $p$  as *registering*
5. Put registering points in a set called *community p*
6. Relabel registering points as *registered*
7. **until** All points are registered
8. **return** All communities

After faithful sampling is completed, the set of all representatives can be regarded as a sample from the data. Reducing the value of parameter  $h$  will increase the number of sample points, resulting in increased computation time and required memory. Conversely, increasing  $h$  will result in fewer sample points that may lead to too low a resolution. In such a case, the computed spectral clusters may fail to estimate the real cell populations appropriately. In our implementation, we use an iterative procedure (explained in the overview of our algorithm) to adjust  $h$  automatically such that the number of representatives will be in range 1500-3000. As a result of this adjustment, the following two objectives are achieved. First, computing the eigenspace of a graph with a number of points in this range is feasible, (it takes less than one minute by a 2.7 GHz processor.) Second, the communities are “small” (Figure 2) and the resulting resolution is high enough such that no biologically interesting information is lost.

In the sampling stage, there is no preference in picking up the next data point, therefore, the final distribution of the sampled points will be uniform in the “effective” space. That is, the representatives are distributed almost uniformly in the space where data points were present (Figure 2). As a consequence, by repeating sampling procedure the final results of clustering will not change significantly. This observation is confirmed quantitatively in Appendix 1. By considering just the representatives, density information is effectively ignored so working directly with these representatives results in improper outcome. On the other hand, some biological information from the original data is preserved by the above algorithm that can be retrieved to guide the clustering algorithm. More precisely, for each sample point, we know the list of all points in its





neighbourhood (*i.e.*, the *members* of the corresponding community). In the next stage, we use this information to define the similarity between two sample points to modify the behaviour of spectral clustering. In this sense, our sampling scheme is *faithful*, meaning that the valuable biological information from the original data points is preserved even after sampling. We call the overall procedure, which consists of faithful sampling, computing modified similarity matrix and spectral clustering, SamSPECTRAL clustering.

#### Similarity Matrix

In this study, we use the following heat kernel formula [36] to compute the similarity between two vertices  $i$  and  $j$ :

$$s_{i,j} = e^{-\frac{\mathcal{D}^2(p_i, p_j)}{2\sigma^2}}, \quad (3)$$

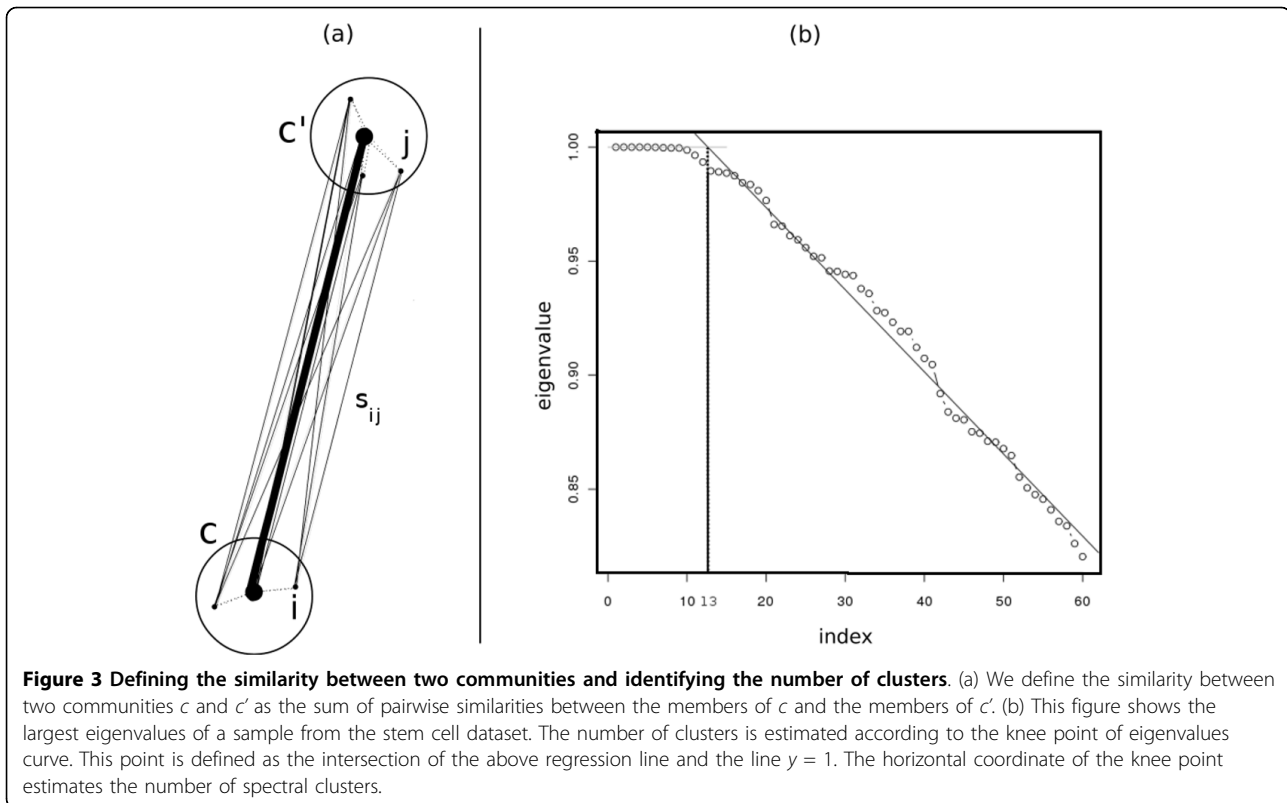
where  $\mathcal{D}(p_i, p_j)$  is the Euclidean distance between them.  $\sigma$  is a scaling parameter that controls how rapidly similarity between  $p_i$  and  $p_j$  falls off with increasing distance. We define the similarity between two communities  $c$  and  $c'$  as the sum of all pairwise similarities between all members of the first community and all members of the second community. That is,

$$S_{c,c'} = \sum_{i \in c} \sum_{j \in c'} s_{i,j}, \quad (4)$$

where  $i$  and  $j$  are members of  $c$  and  $c'$  respectively.

We do not normalize the similarity by dividing the above sum by the size of communities because we would lose valuable biological information that is supposed to be preserved. In short, the size of the communities determines the local density of the data points, which is biologically of great importance.

The above definition is motivated by the following intuition from potential theory that explains how biological information is preserved after faithful sampling by assigning similarities in this way. The eigenvectors of a graph are interpreted as potential functions on the electric network modeled by the graph [37]. Assuming the radius of each community is small enough, the potential values of the community members are almost the same. On the other hand, in potential theory, the equivalent conductance between a group of nodes  $\{v_i\}$  with equal potential values and another group of nodes  $\{w_j\}$  that also have equal potential values is computed by the summation of pairwise conductance between nodes  $v_i$  and  $w_j$  for all  $i$  and  $j$ . Since in our model, the similarity between two vertices is equivalent to the conductance between the corresponding electrical nodes, it is reasonable to sum up pairwise similarities



to estimate the equivalent similarity between communities (Figure 3a).

#### Number of Clusters

The number of clusters must be determined before running the spectral clustering algorithm [38]. To find this number automatically and in an efficient manner, we propose a method that is motivated by the following observation from spectral graph theory:

**Theorem** [39]: The number of connected partitions of a graph is equal to the number of eigenvectors with eigenvalue 1.

We observed that typically for flow cytometry data, if  $\sigma$  is adjusted properly as explained in the SamSPECTRAL package vignette [40], the first few eigenvalues are close to one and at a point we call *knee point* they start to decrease almost linearly. We compute the knee point by applying linear regression to the eigenvalues curve (Figure 3b) and use the horizontal coordinate of this point as a rough estimate for the number of spectral clusters.

#### Combining Clusters

Applying spectral clustering on sampled data results in graph partitioning, which is almost optimum in the sense of having minimum normalized cut [41,42]. However, in some cases, a biologically interesting population might be split into two or more smaller clusters by SamSPECTRAL. We addressed this issue by adding a

post-processing stage wherein the partitions of a population are combined based on known properties of flow cytometry cell populations. Typically, biologically meaningful cell populations in flow cytometry data have their highest density at the centre, and their density decreases towards the border of the population. Since higher density regions indicate communities with relatively more members, the conductance between them is expected to be relatively higher (Equation 4). Thus, similarity between communities is higher in regions with higher densities and the highest similarity is expected to be at the centre of the biological population. This observation forms the basis for our criterion for combining clusters. Specifically, similarity between communities determines the weight on graph edges and we define the maximum weight of the edges of a spectral cluster as *within similarity* of that cluster. Also, the maximum weight of the edges between two different spectral clusters is defined as *between similarity*. If the ratio of *between similarity* to *within similarity* is greater than a predefined threshold (*separation factor*), we conclude that these clusters are partitions of a single population, and should combine them to form a component. We repeat this stage until no two components can be combined. The final components computed in this way are called connected components of the data, and estimate the real biological

populations. With smaller separation factors, spectral clusters tend to combine more often.

#### Overview of SamSPECTRAL Algorithm

In summary, the stages of our algorithm are as follows, assuming the data contains  $n$  points in a  $d$  dimensional space of volume  $V$ , and the parameters  $m$  (max number of communities),  $\sigma$  (scaling parameter), and separation factor are set properly.

1. Sampling:

(a) Let  $h=12Vmd$ .

(b) **Repeat:**

- Run faithful (biological information preserving) sampling algorithm. Suppose  $m'$  communities are built.
- Update:  $h = h \left( \sqrt[d]{\frac{m'}{m}} \right)$ .

**Until**  $\frac{m}{2} \leq m' \leq m$ .

2. Compute the similarities between communities by adding pairwise similarities  $s_{i,j}$  defined by 3:

$$S_{c,c'} = \sum_{i \in c} \sum_{j \in c'} s_{i,j} \quad (5)$$

3. Build a graph wherein each community is a vertex. Put edges between all pairs of vertices and weight them by similarity between corresponding communities.

4. Analyze the spectrum of the above graph to find the clusters;

(a) Normalize the adjacency matrix of the graph according to Equation 1.

(b) Compute the eigenspace of the graph and set  $k$ , number of clusters, according to the knee point of eigenvalues curve.

(c) Run classical spectral clustering algorithm to find  $k$  clusters.

5. Combine the clusters to find connected components:

(a) Initiate the list of components equal to the list of spectral clusters.

(b) **Repeat:**

- For any pairs of components  $C_i, C_j$ , set:

$$\text{separation\_ratio} := \frac{\text{between\_similarity}(C_i, C_j)}{\text{within\_similarity}(C_i, C_j)} \quad (6)$$

- For each component  $C_i$ , compute:

$$M(i) := \max_{j \neq i} (\text{separation\_ratio}(C_i, C_j))$$

- If for all  $i$ ,  $M(i) \leq \text{separation\_factor}$ , **break**.
- Pick an  $i$  such that  $M(i) > \text{separation\_factor}$  and let:

$$j = \arg \max_{j \neq i} (\text{separation\_ratio}(C_i, C_j))$$

- Combine  $C_i$  and  $C_j$ , then update list of components.

**Until** number of components  $> 1$ .

In the sampling stage, we start with the initial value  $h = \frac{1}{2} \sqrt[d]{\frac{V}{m}}$  for the neighbourhood.  $m$  is a parameter that controls  $m'$ , the final number of sample points such that  $\frac{m}{2} \leq m' \leq m$ . Since in our implementation, we use Manhattan metric to measure the distance between points, the volume of a community can be estimated by  $(2h)^d = \frac{V}{m}$ . Therefore if the the data points were distributed uniformly in the space, we would get  $m$  sample points in the first run. However, in practice, we need to repeat the procedure after updating the neighbourhood value. According to our experiments, a few iterations are enough to fulfil the terminating condition  $\frac{m}{2} \leq m' \leq m$ . As the running time of this part of SamSPECTRAL is  $O(nm)$ , which is negligible compared to eigenspace computation time, we did not attempt to optimize the sampling loop.

#### Modified Markov Clustering Algorithm (MCL)

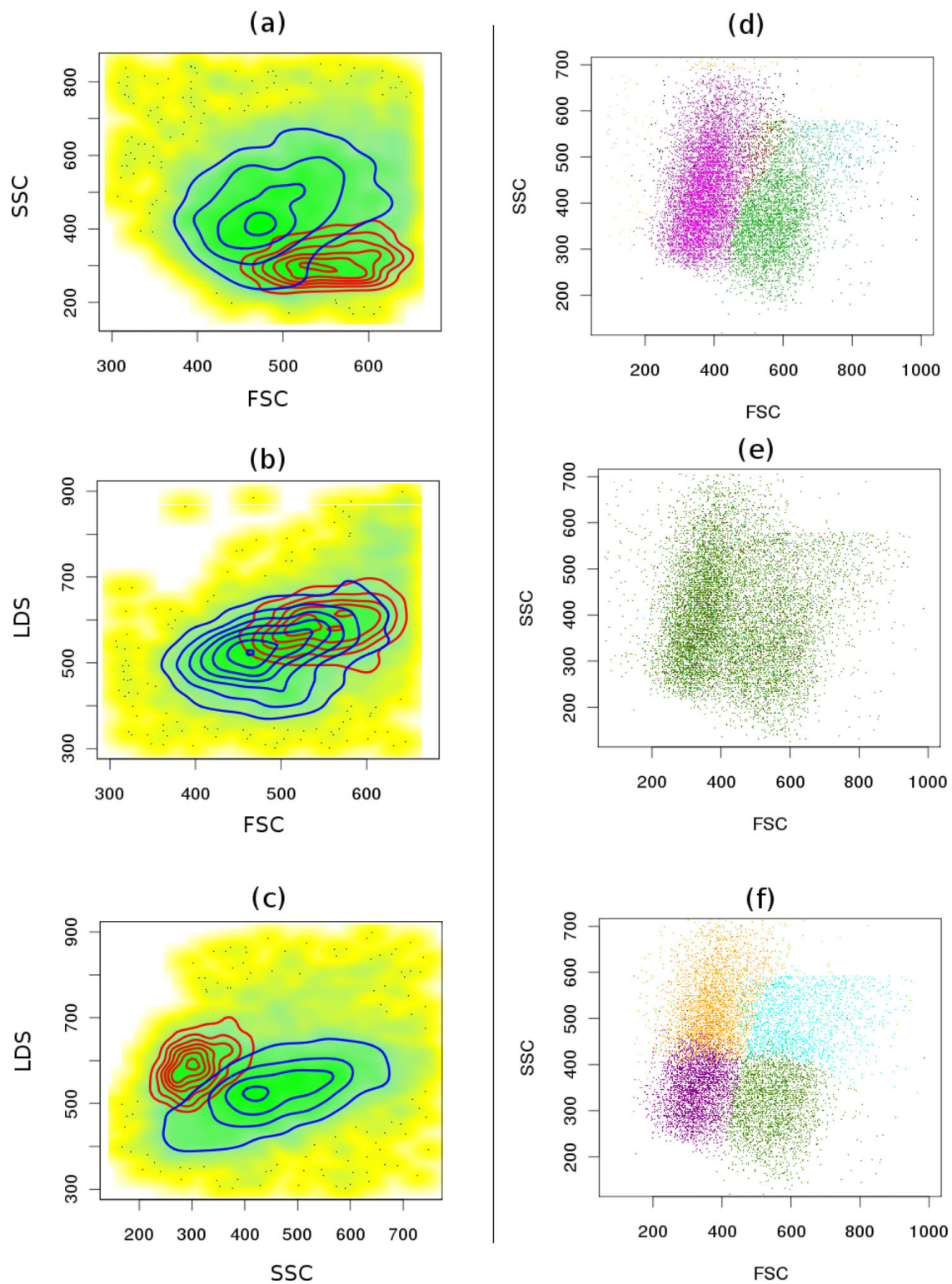
Step 4 in the above algorithm is the classic spectral clustering method. This step potentially could be substituted by any clustering algorithm for weighted graphs. To verify that our approach is extensible in this sense, we substituted classic spectral clustering with Markov Clustering (MCL) [43] keeping the rest of our algorithm, sampling and post-processing steps, unchanged.

MCL finds the partitions of a graph by simulating flow on the nodes. Simulation is done by iteratively multiplying two type of matrices that correspond to *expansion* and *inflation* operations [43]. Because flow and eigenspace of a graph are strongly related<sup>1</sup>, the outcome of this approach tends to be similar to spectral clustering through computing eigenspace.

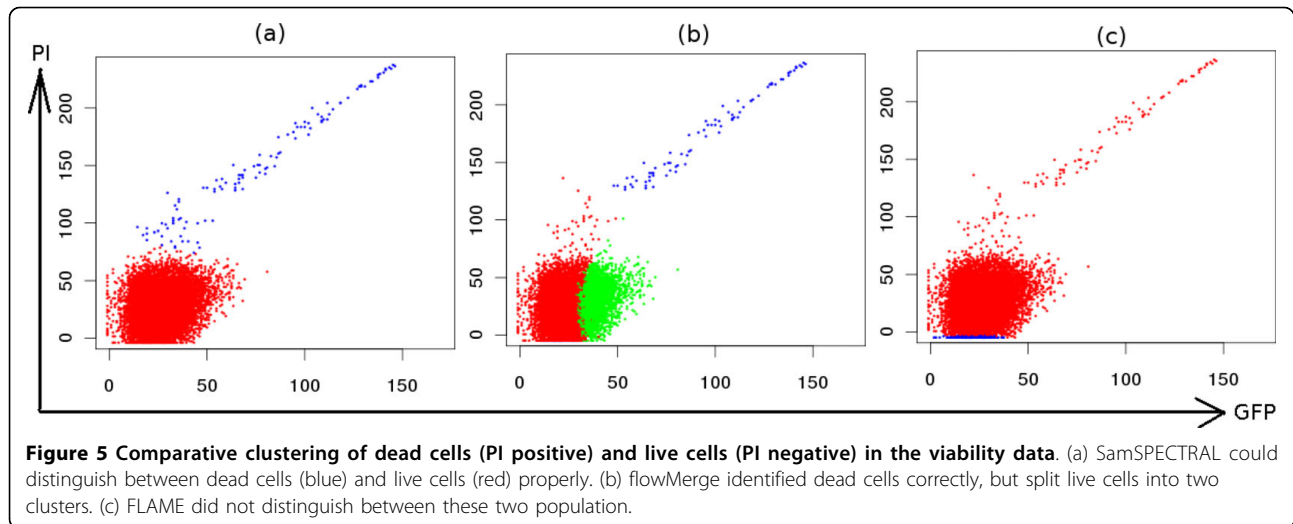
#### Testing

We implemented our algorithm with R, and applied it on four different datasets. We were able to identify some types of biologically interesting populations that were previously known to be hard to distinguish, including:

1. *Overlapping populations* (Figure 4a-c).
2. *Subpopulations of a major population* (Figure 4d-f).
3. *Non-elliptical shaped populations* (Figures 5 and Figure 6a-c).
4. *Low density populations close to dense ones* (Figures 6d-f and Figure 7).
5. *Rare populations comprising less than 2% of all data points* (Figure 8).



**Figure 4 Comparative clustering of the telomere dataset.** (a-c) Proper identification of overlapping populations. Although two populations shown by red and blue contours are overlapping in all bi-varient plots of this 3-dimensional sample, SamSPECTRAL can properly distinguish them by considering multiple parameters simultaneously.(d) SamSPECTRAL can also identify two major subpopulations of granulocytes correctly, as verified by expert analysis. (e) flowMerge does not distinguish between two populations of interest, and (f) FLAME improperly splits the same sample into several clusters.



Here, we demonstrate the capabilities of SamSPECTRAL in identifying biological populations in these cases and compare our results with two state of the art methods for clustering flow cytometry data, flowMerge (version 0.4.1) and FLAME (version 3), respectively obtained through BioConductor and GenePattern.

#### Overlapping Populations

Traditionally, identifying cell populations in flow cytometry data is accomplished by visualizing the multidimensional data as a series of bivariate plots, and separating interesting sections manually, in a process termed gating. Gating becomes challenging for high dimensional data since when the data is mapped to two dimensions, some clusters may overlap, resulting in the mixing of different populations. Consequently, even a trained operator cannot identify overlapping populations properly in all cases. However, our algorithm prevents this undesired error by considering all data dimensions together (Figure 4a-c). Model based multidimensional techniques also perform generally well in this regard.

#### Subpopulations of a Population

Figure 4d-f shows a major blood population (granulocytes) formed from two distinct subpopulations as verified by expert manual analysis. SamSPECTRAL could clearly distinguish between two subpopulations. flowMerge merged these two populations into one, while FLAME split both subpopulations.

#### Non-elliptical Shaped Populations

While most model based techniques have *a priori* assumptions on the shape of populations that resulted in mixing or splitting populations, our method worked relatively well on the samples with arbitrary shape populations. In Figure 5, the PI positive population (blue diagonal one) was clearly identified despite its non-

elliptical shape. flowMerge could also distinguish this population, but it incorrectly split the PI negative population into two parts. FLAME did not correctly distinguish the two populations. Figure 6a-c shows the output of the three algorithms on a four dimensional sample from GvHD dataset. While the red population has a complex shape, it could be identified with high accuracy by SamSPECTRAL. While FLAME produced a satisfactory result, flowMerge mixed this population with the one below it.

#### Low Density Populations Close to Dense Populations

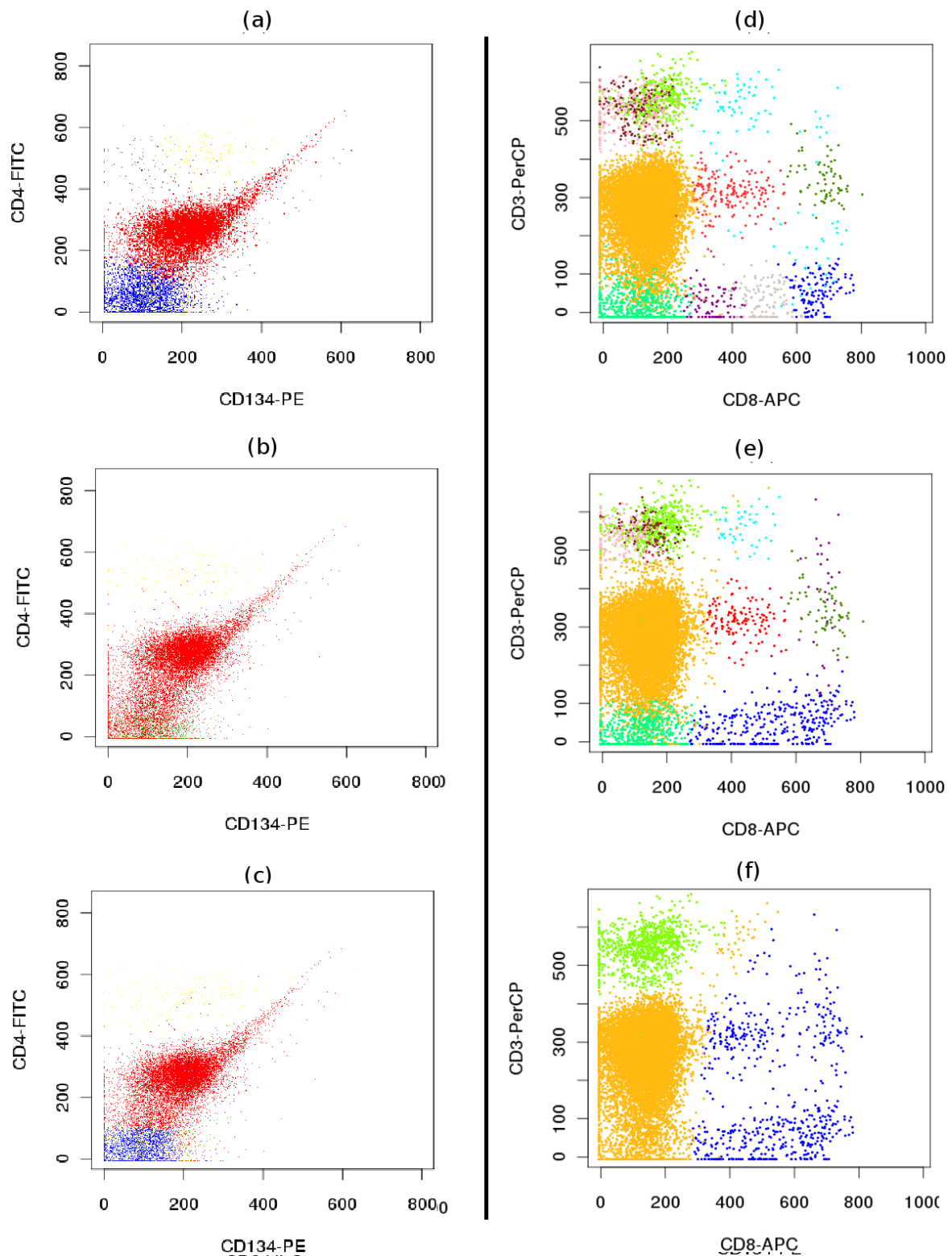
Figure 6d-f shows a sample from GvHD dataset containing a relatively low density and a high density population close together. SamSPECTRAL clearly distinguished the red population in the centre of the plot from the yellow dense population to its left. Moreover, it did not mix the red population with the other low density population to its right. FlowMerge also clustered this sample relatively well, requiring five times more processing time. The performance of FLAME was not satisfactory for this sample due to mixing the desired population with the other low density ones.

Figure 7 depicts a sample from the stem cell dataset containing a relatively low density population shown in blue. In each row, three 2-dimensional plots of the 3-dimensional data sample are presented. SamSPECTRAL could distinguish the blue population although it was surrounded by three relatively denser populations (the yellow, green and red ones). FlowMerge mixed this population with the yellow one, while FLAME mixed it with the red one.

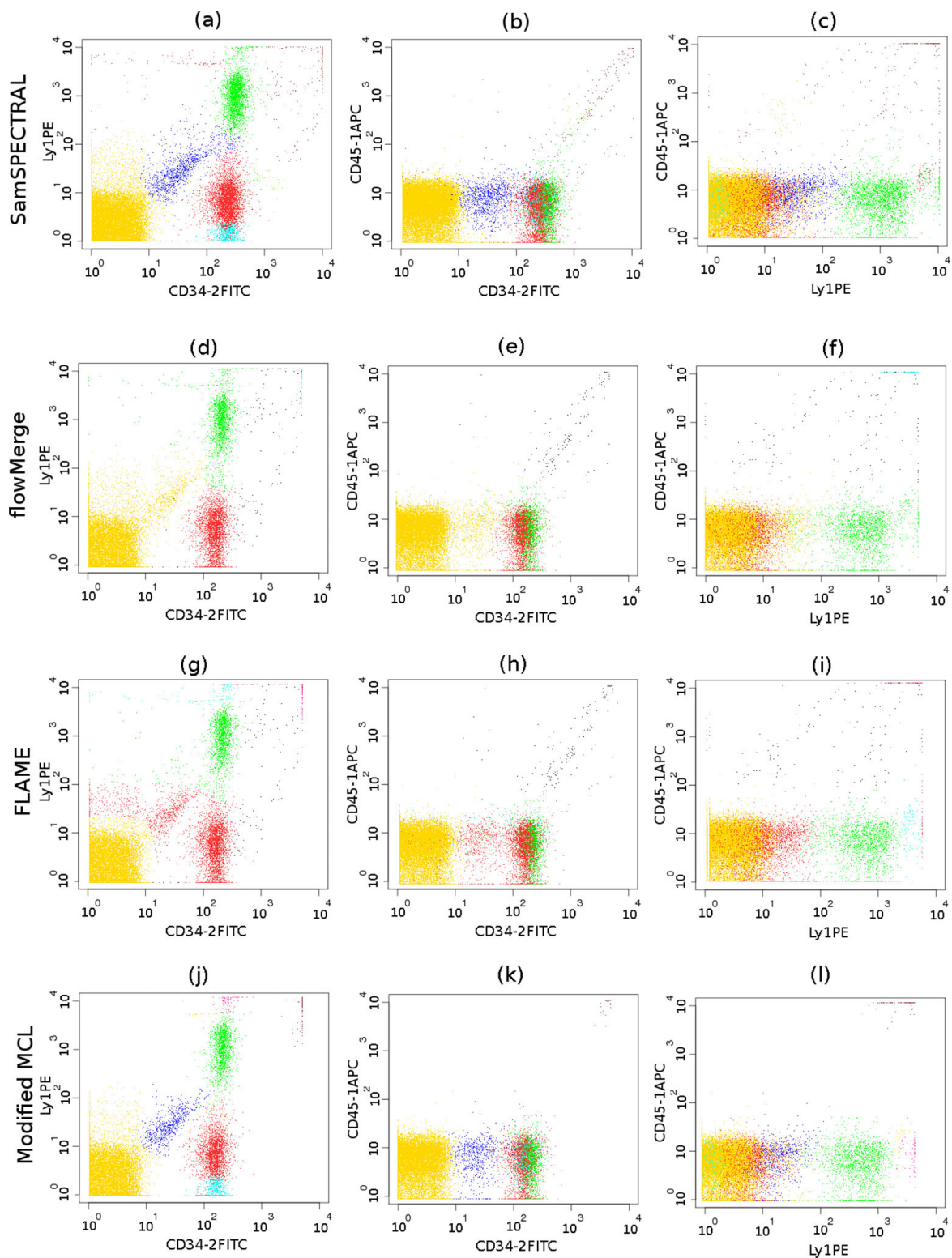
#### Rare Populations

Identifying rare populations has many significant applications in flow cytometry experiments including distinguishing cancer stem cells, hematopoietic stem cell transplantation, detection of fetal cells in maternal

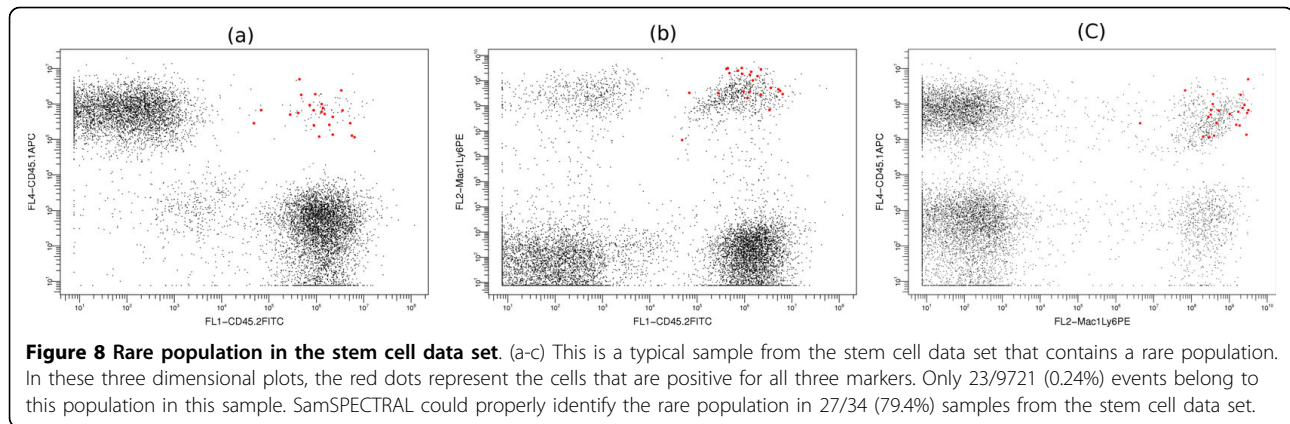




**Figure 6 Comparative clustering of the GvHD dataset.** (Left) Identification of non-elliptical shaped populations. (a) SamSPECTRAL could properly identify the red, non-elliptical population, while (b) flowMerge mixed this population with the one below it. (c) FLAME produced satisfactory results in identifying this population. (Right) Identification of low density populations close to dense populations. (d) SamSPECTRAL and (e) flowMerge could identify the low density population shown in red at the centre of the figure correctly, while (f) FLAME merged this population with the other ones surrounding it.



**Figure 7 Comparative identification of a low density population surrounded by much denser populations in the stem cell data set.** (a-c) SamSPECTRAL correctly identified the blue, low density population, while (d-f) flowMerge merged it to the yellow, high density population. (g-i) FLAME merged it to the red population. (j-l) The outcome of our modified MCL was similar to that obtained by SamSPECTRAL using classic spectral clustering. This shows that SamSPECTRAL is extensible by substituting classic spectral clustering with other clustering algorithms for weighted graph.



blood, detection of leukocytes in leukocyte-depleted platelet products, detection of injected cells for biotherapy and malaria diagnosis [44].

Figure 8 shows a typical sample from the stem cell data set that contains a rare population in red. This population is positive for all the three markers and in each sample, it comprises between 0.1% to 2% of total cells. We performed an experiment on 34 samples from the stem cell data set and compared the performance of SamSPECTRAL, flowMerge and FLAME. This rare population was distinguished manually and the result of manual gating was considered as the basis for our comparison. FLAME and flowMerge could identify this population only in 11 (32%) and 9 (26%) of samples, respectively.

SamSPECTRAL could distinguish this population in 27 (79%) samples including all the ones that were identified by FLAME and flowMerge. In the 7 (21%) samples that SamSPECTRAL failed, the rare population of interest contained less than 0.15% of all data points.

To measure the accuracy of SamSPECTRAL, we define sensitivity and specificity as follows. For each sample, we call a cell positive if it belongs to the rare population of interest, and it is negative otherwise. Sensitivity is defined to be the number of truly identified rare cells divided by the total number of rare cells. Accordingly, specificity is the number of cells identified as negative divided by the total number of truly negative cell. The 27 (79%) cases where SamSPECTRAL correctly identified the rare population, had a 0.83 mean sensitivity with a 0.26 standard deviation. The median sensitivity was .99. Specificity was 1 except for one sample. If we consider the samples with a rare population bigger than 0.2% of the total data, we obtained median = 1, mean = 0.93 and standard deviation of 0.15 for sensitivity. A detailed report of the results of this experiment is provided as a table in additional file 1.

#### SamSPECTRAL with MCL

Figure 7j-l depicts the output of MCL on a sample from stem cells dataset. We ran MCL on the sampled data

obtained by our faithful sampling algorithm and then the post-processing step was applied to the resulting clusters. This experiment showed there was no significant difference for SamSPECTRAL in clustering either through computing eigenvectors (Figure 7a-c) or by MCL (Figure 7j-l)<sup>2</sup>.

#### Discussion

Although spectral clustering algorithm is a powerful technique, it can not be directly applied to large datasets as it is computationally expensive both in time and memory. In this study, we developed a sampling method and combined it with spectral clustering by modifying the similarity matrix based on potential theory. As a result, for the first time, analysing flow cytometry data using spectral methods becomes possible and practical. We applied SamSPECTRAL to four different flow cytometry datasets to demonstrate its applicability on a broad spectrum of flow cytometry data, and compared its performance to two state of the art model-based clustering methods optimized for flow cytometry data.

Detecting rare populations is a challenging problem and in spite of its significant applications in medical and biological research, little progress has been achieved in automatic identification of such populations. Our data reduction scheme is delicate enough not to miss rare populations comprising between 0.2% to 2% of the total data. SamSPECTRAL can identify populations of relative size in this range with acceptable accuracy.

Since our method, SamSPECTRAL, is a multidimensional clustering approach, it can identify overlapping populations that are generally hard to identify by manual gating that uses sequential two dimensional visualizations of the data. SamSPECTRAL is the first method that has demonstrated the ability to correctly identify subpopulations of major flow cytometry cell populations.

An important challenge in analysing flow cytometry data is in clustering data files that contain populations that significantly differ in density. Model-based

techniques can produce errors in identifying a low density population close to denser populations because they typically make assumptions on the density of clusters [23]. Our experiments demonstrated that SamSPECTRAL can properly tackle this problem. Besides the practical observations, this capability is justified by the following observation. Spectral methodology clusters the graph such that the normal cut is “almost” optimum [41]. Now, assume that it can distinguish between two clusters when their densities are comparable. Then, if the size of the smaller cluster is reduced without change in its shape or distribution, the normal cut between them remains similar because the number of vertices and edges reduces almost proportionally to each other. Therefore, the clusters remain distinguishable. This explains why the overall performance of SamSPECTRAL is independent of cluster densities as long as their shapes are preserved.

Since parametric methods such as FLAME and flowMerge make *a priori* assumptions on the distribution or shape of the clusters [23], they may fail in identifying populations with arbitrary shapes. Although flowMerge attempts to solve this issue by finding more clusters than needed and then merging them together, it still does not produce satisfactory results when the shape of the cluster is complex. SamSPECTRAL has the capability of identifying arbitrary shape clusters since it is a non-parametric approach that makes no assumptions on the shape and distribution of clusters, and clusters data based only on similarity between data points. Compared to other non-parametric methods, our algorithm has the advantages of automatically identifying the number of clusters and having low sensitivity to the predefined thresholds. Therefore, users can adjust the parameters only once by running SamSPECTRAL on one or two random samples from a flow cytometry data set. Then, the algorithm can be run on the rest of data set without changing the parameters.

Not only does our sampling scheme increase the speed of spectral clustering without losing important biological information, but the resulting algorithm is faster than other methods considered in this study. More precisely, the running time of SamSPECTRAL is  $O(dmn) + O(m^3)$  where  $O(dmn)$  is the running time for building  $m$  communities from  $n$  points in  $d$  dimension and  $O(m^3)$  is the running time for computing the eigenspace. After this step, the k-means clustering runs very fast in time  $O(k m t)$  to find  $k$  clusters using eigenvectors by  $t$  iterations. In comparison, the time complexity of the original MCL method is  $O(nr^2)$  with no guarantee on upper bound for number of iterations  $r$ , other than  $n$ . Practically, for our model of flow cytometry data where all pairs of data points are connected, we could

not run MCL before applying our modification to it. Moreover, SamSPECTRAL running time is significantly less than model-based techniques. The running time of flowMerge is  $O(d^2k^2nt)$  and FLAME runs in time  $O(d^4kln t)$  where  $l$  is the number of times it runs to find the optimal number of clusters. In practice, we can keep  $m$  as small as 1500-3000 without losing important biological information, and consequently SamSPECTRAL ran at least 5-10 times faster than flowMerge and FLAME on the studied datasets. Furthermore, the time efficiency of our algorithm is more noticeable for higher dimensional data such as the one provided as additional file 2. This sample contains 100,000 events in 23 dimensions and SamSPECTRAL can analyze it in less than 25 minutes by a 2.7 GHz processor.

## Conclusions

Faithful sampling is based on potential theory. It reduces the size of input for spectral clustering algorithms and consequently they can now be efficiently applied on flow cytometry data in spite of its large size. Practically, our approach demonstrated significant advantages in proper identification of populations with non-elliptical shapes, low density populations close to dense ones, minor subpopulations of a major population, rare populations, and overlapping populations. No state of the art method can solve the challenges in identifying populations with the above properties simultaneously. Moreover, applying SamSPECTRAL to other biological data such as microarrays and protein databases may result in significant improvements in gene expression and protein classification.

Besides, our faithful sampling algorithm can have interesting applications by itself. For instance, it can be used appropriately to reduce the size of input for other clustering algorithms that are based on spectral graph theory such as Markov Clustering Algorithm (MCL), electrical circuit based clustering, and agent based graph clustering [45]. We have shown the extendibility of our approach in this sense by substituting classic spectral clustering with MCL, a method that has many applications in bioinformatics.

Other directions for future work include applying other schemes for estimating similarities between communities, combining clusters based on other combinatorial algorithms or biological criteria, and repeating the algorithm several times to obtain a more stable outcome.

## Methods

To run flowMerge and FLAME optimally, we used several settings for their parameters, finally selecting those that gave us the best results. For SamSPECTRAL algorithm, we set  $m = 3000$  to keep the running time below

1 minute by a 2.7 GHz processor and the obtained results remained satisfactory for all samples we analyzed. The separation factor and scaling parameter ( $\sigma$ ) are two main parameters that needed to be adjusted. Decreasing  $\sigma$  and increasing the separation factor will result in identifying more populations. In particular, if  $\sigma$  is decreased, then according to the heat kernel formula, the weights on the edges of the graph will decrease exponentially. Therefore, the graph will be sparser and tends to obtain more partitions. In consequence, the algorithm identifies more spectral clusters. This phenomenon can be useful in identifying rare populations. On the other hand, if separation factor is too high, a single population may be split into parts. In our experiments, we applied SamSPECTRAL on one or two random data samples of a data set and tried different values. Then, the selected parameters were fixed and used to apply SamSPECTRAL on the rest of data samples. The parameters values for the data sets presented in this paper are provided in additional file 3, 4, 5, and 6. The reader is referred to the SamSPECTRAL **Bioconductor** package vignette for more explanation on how to adjust parameters for a given data set<sup>3</sup>.

#### Datasets

We tested our algorithm on four different flow cytometry datasets as explained briefly here. The GvHD dataset is available in flowCore package through BioConductor, and the rest are available upon request.

#### Stem Cells

To investigate heterogeneity in the differentiation behaviour of hematopoietic stem cells, a subpopulation of adult mouse bone marrow was isolated and then each single stem cell was transplanted into one of 352 recipients [46]. 16 blood samples were taken from the recipients in biweekly intervals and were studied in a cytometer. The investigation contained hundreds of data files that needed to be analyzed to count the frequency of each subtype of white cells they contained.

#### Telomere

In all vertebrates, telomeres consist of tandem DNA repeats of the sequence d(TTAGGG) and associated proteins. Telomere length is known to be crucial elements in ageing and various diseases including cancer and it can be estimated by flow cytometry [47]. Since telomere length is different for various cell populations, these need to be distinguished before calculating telomere length.

#### GvHD

Acute graft versus host disease (GvHD) is a common outcome after bone marrow transplantation. It is difficult to diagnose in its early stages in order to provide timely treatment. To investigate how flow cytometry can help predict the development of GvHD, and to study its

advantages over microarrays, peripheral blood samples from 31 patients undergoing allogeneic blood and marrow transplant were analyzed [48]. The samples were taken at progressive time points post-transplant and were stained with four appropriate lymphocyte phenotypic and activation markers defining 121 different populations using six markers.

#### Viability

Propidium iodide (PI) is a widely used marker for determining viability of mammalian cells [49] because it has the capability of passing through only damaged cell membranes. However, depending on the complexity of the data, identifying dead cells automatically might still be difficult even if this marker is used. We tested the capability of our algorithm in identifying dead cells using PI marker on a dataset from the Terry Fox Laboratory.

### Appendix 1

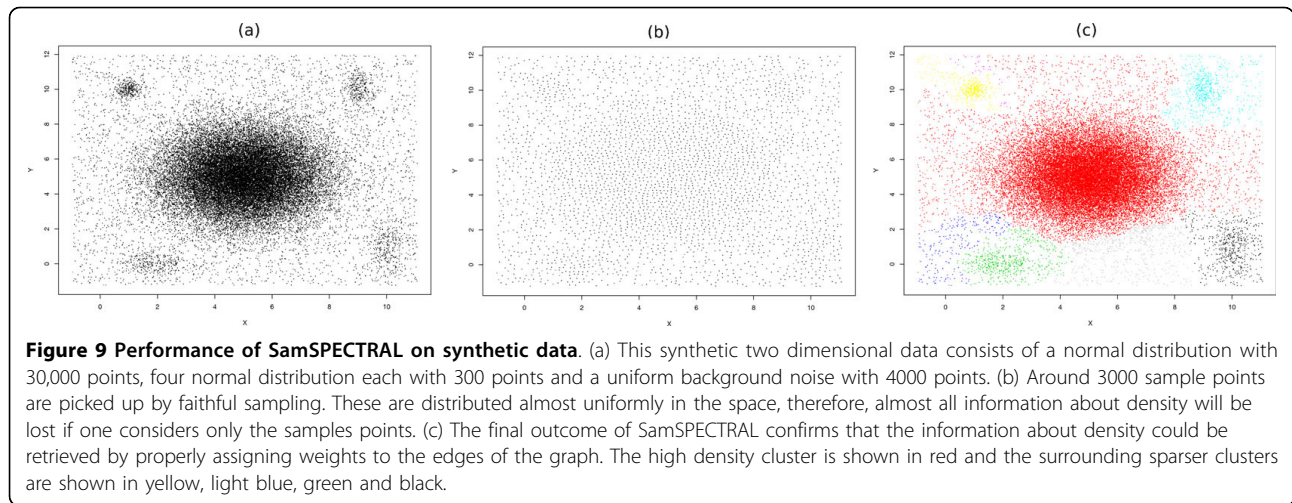
In the results section, we explained that the resolution of the sample points (Figure 2) is high enough such that by repeating the randomized faithful sampling procedure, the outcome of SamSPECTRAL does not vary significantly. The following experiment is performed to confirm this observation quantitatively. In this experiment we used F-measure, which is known to be appropriate for comparing clustering results of flow cytometry data [50]. F-measure varies in range 0-1 and reaches its best value at 1 when the two clustering results are identical. We ran SamSPECTRAL on a sample from the stem cell data set 20 times and compared the final results. The F-measure values obtained by pairwise comparison between the final results had mean = 0.98, median = 0.98 and standard deviation 0.0097.

### Appendix 2

We performed the following experiment to show the effect of edge weights on performance of spectral clustering. As shown in Figure 9, we produced synthetic data containing one normal distribution with relatively high density surrounded by four relatively small clusters with lower densities. The number of points in each small cluster is less than 0.01% of the whole data and noise is added to the data space uniformly (Figure 9a). For the central dense distribution, we set  $\sigma_{xx} = \sigma_{yy} = 2$ ,  $\sigma_{xy} = \sigma_{yx} = 0$  and the surrounding clusters are normal distributions with

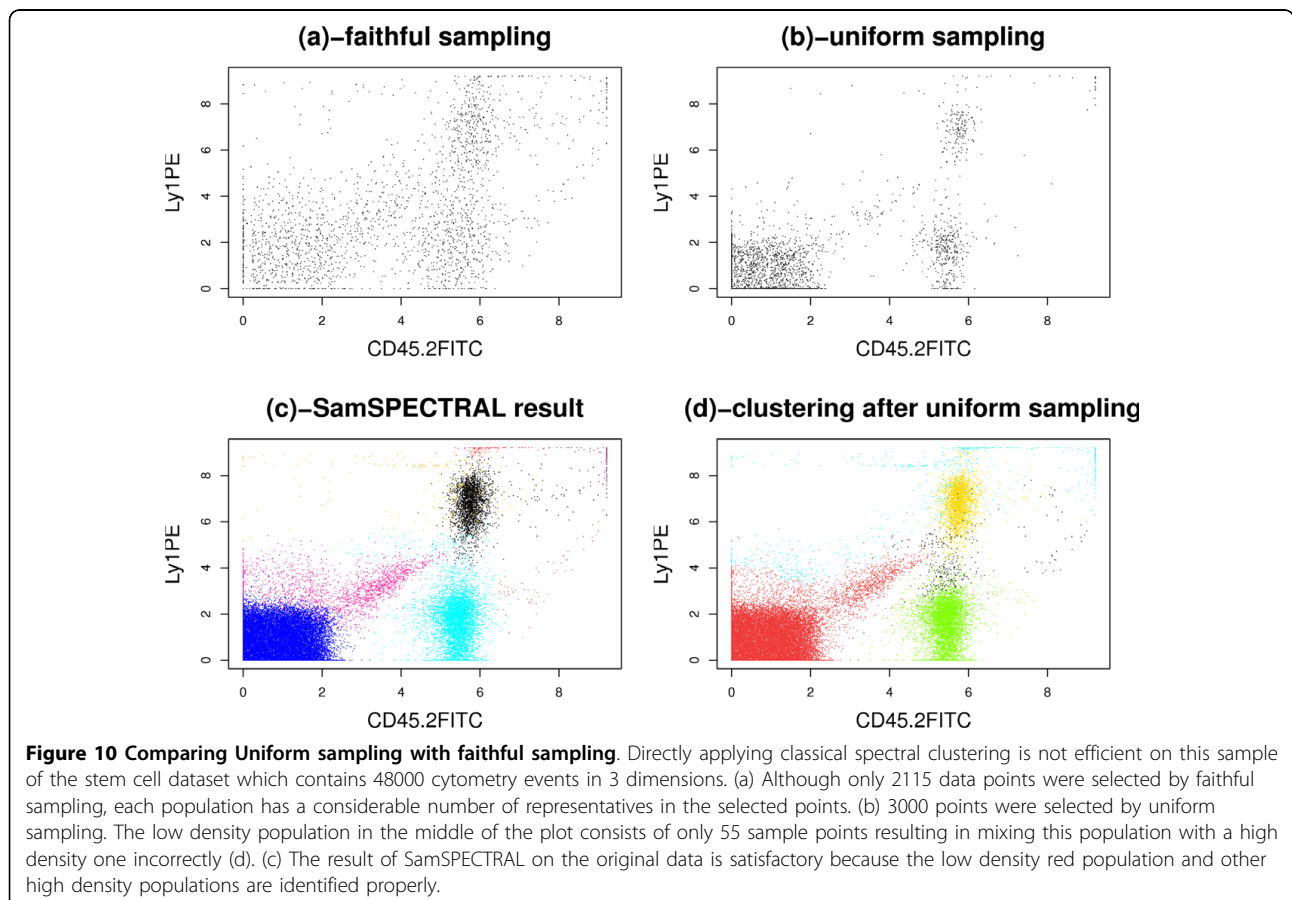
$$\begin{aligned}\sigma_{xx}^1 &= 0.08, \sigma_{yy}^1 = 0.30, \sigma_{xy}^1 = \sigma_{yx}^1 = 0, \\ \sigma_{xx}^2 &= 0.07, \sigma_{yy}^2 = 0.08, \sigma_{xy}^2 = \sigma_{yx}^2 = 0, \\ \sigma_{xx}^3 &= 0.50, \sigma_{yy}^3 = 0.10, \sigma_{xy}^3 = \sigma_{yx}^3 = 0, \\ \sigma_{xx}^4 &= 0.10, \sigma_{yy}^4 = 0.70, \sigma_{xy}^4 = \sigma_{yx}^4 = 0.\end{aligned}$$

The R code to produce this synthetic data and run SamSPECTRAL on it is provided in additional file 7.



After faithful sampling is done (Figure 9b), the sample points are distributed almost uniformly, and the information about the local density of original data is lost. However, faithful sampling provides us with more information than only the sample points. It will also return the members of each community and our data reduction scheme uses this information to assign

weight to the edges. According to formulas 3 and 4, the more populated and closer two communities are, the higher the weight between them will be (Figure 1c). According to Figure 9c, this strategy is successful in retrieving information about local density as all the five clusters are distinguished properly by SamSPECTRAL.



### Appendix 3

We observed that some low density populations disappeared entirely when simple uniform sampling was employed. To investigate the effect of this phenomenon on the final clustering results, we performed an experiment on a sample of the stem cell dataset that contained 48,000 events in 3 dimensions. First, 3,000 data points were selected uniformly at random. Then, we assigned a label to each of these selected points by applying classical spectral clustering on them. Finally, for each original data point, the label of the closest selected point was considered as its cluster label. Figures 10d and 10c show the results of this approach and SamSPECTRAL, accordingly. The red population that was distinguished by SamSPECTRAL correctly in Figure 10c consists of only 4% of the data. This population could not be distinguished properly by any setting of the parameters after uniform sampling (Figure 10d).

### Footnotes

<sup>1</sup> Cheeger inequality is an example of such a relation [41].

<sup>2</sup> The CD45+ cells that are considered as outliers by MCL are not plotted in Figure 7j-l.

<sup>3</sup> The vignette is located at: [http://bioconductor.org/packages/devel/bioc/vignettes/SamSPECTRAL/inst/doc/Clustering\\_by\\_SamSPECTRAL.pdf](http://bioconductor.org/packages/devel/bioc/vignettes/SamSPECTRAL/inst/doc/Clustering_by_SamSPECTRAL.pdf)

### Additional material

**Additional file 1: Report on identification of rare population.** The table contains the full detailed report on our comparative experiment for identifying rare populations.

**Additional file 2: High dimensional flow cytometry data.** This data file contains a matrix with 100,000 rows and 23 columns that represents a flow cytometry sample with 100,000 events. It can be directly loaded in R and analyzed by SamSPECTRAL. It takes less than 12 minutes to perform faithful sampling on this 23 dimensional data.

**Additional file 3: Parameters for GvHD data set.** These values are appropriate for running SamSPECTRAL on GvHD data set.

**Additional file 4: Parameters for stem cell data set.** These values are appropriate for running SamSPECTRAL on stem cell data set.

**Additional file 5: Parameters for telomere data set.** These values are appropriate for running SamSPECTRAL on telomere data set.

**Additional file 6: Parameters for viability data set.** These values are appropriate for running SamSPECTRAL on viability data set.

**Additional file 7: Simulation with synthetic data.** This R source code produces synthetic data with 5 clusters shown in Figure 5. The resulting data is passed to SamSPECTRAL to be clustered.

### Acknowledgements

The authors would like to thank Adrian Cortes, Connie Eaves, Peter Lansdorp and Keith Humphries for providing data, Bari Zahedi and Irma Vulto for their biological insight and manual data analysis, Josef Spidlen for assistance in running FLAME, Aaron Barsky and Nima Aghaeepour for their editorial comments, and Mani Ranjbar for programming guidance. This work was supported by NIH grants 1R01EB008400 and 1R01EB005034, the Michael

Smith Foundation for Health Research, the National Science and Engineering Research Council and the MITACS Network of Centres of Excellence.

### Author details

<sup>1</sup>Department of Computing Science, University of British Columbia, Vancouver, BC, Canada. <sup>2</sup>Terry Fox Laboratory, BC Cancer Agency, 675 W 10th Ave., Vancouver, BC, Canada. <sup>3</sup>Faculty of Science, University of British Columbia, Vancouver, BC, Canada. <sup>4</sup>Department of Medical Genetics, University of British Columbia, Vancouver, BC, Canada.

### Authors' contributions

The authors wish it to be known that, in their opinion, the first two authors, HZ and PS, should be regarded joint first authors. HZ designed and implemented the faithful sampling algorithm, and the method for computing similarity between communities. PS developed the idea of non-uniform sampling for spectral clustering and experimentally verified the stability of the algorithm. HZ and PS jointly worked on the method for estimating the number of clusters, and post-processing steps. HZ and PS performed the experiments. HZ, PS and RB wrote the paper. RB provided data and computing facilities. AG studied the convergence of faithful sampling. AG and RB supervised the project. All authors read, edited and approved the final manuscript.

Received: 21 December 2009 Accepted: 28 July 2010

Published: 28 July 2010

### References

1. Hawley TS, Hawley RG: *Flow Cytometry Protocols, Methods in Molecular Biology* Humana Press, 2 2005.
2. Peretto SP, Chattopadhyay PK, Roederer M: **Seventeen-colour flow cytometry: unravelling the immune system.** *Nat Rev Immunol* 2004, **4**(8):648-655.
3. Bashashati A, Brinkman R: **A survey of flow cytometry data analysis methods.** *Advances in Bioinformatics* 2009, **2009**:1-19.
4. Klinke D II, Brundage K: **Scalable analysis of flow cytometry data using R/Bioconductor.** *Cytometry Part A* 2009, **75**(8):699-706.
5. Lugli E, Roederer M, Cossarizza A: **Data analysis in flow cytometry: The future just started.** *Cytometry Part A* 2010, **77**(7):705-13.
6. Boedigheimer M, Ferbas J: **Mixture modeling approach to flow cytometry data.** *Cytometry Part A* 2008, **73**(5):421-429.
7. Simon U, Mucha H, Bruggemann R: **Model-based cluster analysis applied to flow cytometry data.** *Innovations in Classification, Data Science, and Information Systems* 2005, 69-76.
8. Zeng Q, Pratt J, Pak J, Ravnic D, Huss H, Mentzer S: **Feature-guided clustering of multi-dimensional flow cytometry datasets.** *Journal of Biomedical Informatics* 2007, **40**(3):325-331.
9. Scheuermann R, Qian Y, Wei C, Sanz I: **ImmPort FLOCK: Automated cell population identification in high dimensional flow cytometry data.** *The Journal of Immunology* 2009, **182**(Meeting Abstracts 1):42-17.
10. Naumann U, Luta G, Wand M: **The curvHDR method for gating flow cytometry samples.** *BMC bioinformatics* 2010, **11**:44.
11. Jeffries D, Zaidi I, de Jong B, Holland M, Miles D: **Analysis of flow cytometry data using an automatic processing tool.** *Cytometry Part A* 2008, **73**(9):857-867.
12. Naumann U, Wand M: **Automation in high-content flow cytometry screening.** *Cytometry Part A* 2009, **75**(9):789-797.
13. Shulman N, Bellew M, Snelling G, Carter D, Huang Y, Li H, Self S, McElrath M, De Rosa S: **Development of an automated analysis system for data from flow cytometric intracellular cytokine staining assays from clinical vaccine trials.** *Cytometry Part A* 2008, **73**(9):847-856.
14. Preffer F, Dombkowski D: **Advances in complex multiparameter flow cytometry technology: Applications in stem cell research.** *Cytometry Part B: Clinical Cytometry* 2009, **76**(5):295-314.
15. Pedreira C, Costa E, Barrena S, Lecrevisse Q, Almeida J, van Dongen J, Orfao A, et al: **Generation of flow cytometry data files with a potentially infinite number of dimensions.** *Cytometry Part A* 2008, **73**(9):834-846.
16. Conrad MP: **A rapid, non-parametric clustering scheme for flow cytometric data.** *Pattern Recognition* 1987, **20**(2):229-35.
17. Fu L, Yang M, Braylan R, Benson N: **Real-time adaptive clustering of flow cytometric data.** *Pattern Recognition* 1993, **26**(2):365-373.

18. Boddy L, Wilkins M, Morris C: **Pattern recognition in flow cytometry.** *Cytometry* 2001, **44**(3):195-209.
19. Finn WG, Carter KM, Raich R, Stoolman LM, Hero AO: **Analysis of clinical flow cytometric immunophenotyping data by clustering on statistical manifolds: Treating flow cytometry data as high-dimensional objects.** *Cytometry Part B: Clinical Cytometry* 2009, **76B**:1-7.
20. Pyne S, Hu X, Wang K, Rossin E, Lin Ti, Maier LM, Baecher-Allan C, McLachlan GJ, Tamayo P, Hafler DA, De Jager PL, Mesirov JP: **Automated high-dimensional flow cytometric data analysis.** *Proceedings of the National Academy of Sciences* 2009, **106**(21):8519-8524.
21. Lo K, Hahne F, Brinkman R, Gottardo R: **flowClust: a Bioconductor package for automated gating of flow cytometry data.** *BMC Bioinformatics* 2009, **10**:1-145.
22. Finak G, Bashashati A, Brinkman RR, Gottardo R: **Merging Mixture Components for Cell Population Identification in Flow Cytometry.** *Advances in Bioinformatics* 2009, **2009**:1-12.
23. Chan C, Feng F, Ottinger J, Foster D, West M, Kepler TB: **Statistical mixture modeling for cell subtype identification in flow cytometry.** *Cytometry Part A* 2008, **73A**(8):693-701.
24. von Luxburg U: **A tutorial on spectral clustering.** *Statistics and Computing* 2007, **17**(4):395-416.
25. Ng AY, Jordan MI, Weiss Y: **On Spectral Clustering: Analysis and an algorithm.** *Advances in Neural Information Processing Systems 14* MIT Press 2001, 849-856.
26. Azran A, Ghahramani Z: **Spectral Methods for Automatic Multiscale Data Clustering.** *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on* 2006, 1:190-197.
27. Pentney W, Meila M: **Spectral Clustering of Biological Sequence Data.** *AAAI* 2005, 845-850.
28. Bach FR, Jordan MI: **Learning Spectral Clustering, With Application To Speech Separation.** *J Mach Learn Res* 2006, **7**:1963-2001.
29. von Luxburg U, Belkin M, Bousquet O: **Consistency of Spectral Clustering.** *Annals of Statistics* 2008, **36**(2):555-586.
30. Trefethen LN, Bau D: *Numerical Linear Algebra* SIAM 1997.
31. Cullum JK, Willoughby RA: *Lanczos Algorithms for Large Symmetric Eigenvalue Computations* SIAM 2002, 1.
32. Fowlkes C, Belongie S, Chung F, Malik J: **Spectral Grouping Using the Nystrom Method.** *IEEE Transactions on Pattern Analysis and Machine Intelligence* 2004, **26**(2):214-225.
33. Carter KM, Raich R, Finn WG, Hero AO: **Information Preserving Component Analysis: Data Projections for Flow Cytometry Analysis.** *IEEE Journal of Selected Topics in Signal Processing* 2009, **3**(1):148-158.
34. Mann RC: **On multiparameter data analysis in flow cytometry.** *Cytometry A* 1987, **8**(2):184-189.
35. Yan D, Huang L, Jordan M: **Fast Approximate Spectral Clustering.** *Tech. Rep. UCB/EECS-2009-45* EECS Department, University of California, Berkeley 2009.
36. Kondor RI, Lafferty J: **Diffusion Kernels on Graphs and Other Discrete Structures.** *In Proceedings of the ICML 2002*, 315-322.
37. Biyikoglu T, Leydold J, Stadler PF: **Laplacian eigenvectors of graphs.** *Lecture notes in mathematics; 1915* Springer 2007.
38. Xiang T, Gong S: **Spectral clustering with eigenvector selection.** *Pattern Recogn* 2008, **41**(3):1012-1029.
39. Biggs N: *Topics in Algebraic Graph Theory: Encyclopedia of Mathematics and its Applications* 2007, **16**:171-172.
40. **SamSPECTRAL package at BioConductor.** [<http://www.bioconductor.org/packages/devel/bioc/html/SamSPECTRAL.html>].
41. Chung FRK: *Spectral Graph Theory (CBMS Regional Conference Series in Mathematics, No. 92).* American Mathematical Society 1997.
42. Shi J, Malik J: **Normalized Cuts and Image Segmentation.** *IEEE Transactions on Pattern Analysis and Machine Intelligence* 1997, **22**:888-905.
43. Dongen SV: **Graph Clustering Via a Discrete Uncoupling Process.** *SIAM Journal on Matrix Analysis and Applications* 2008, **30**:121-141.
44. Donnenberg AD, Donnenberg VS: **Rare-Event Analysis in Flow Cytometry.** *Clinics in Laboratory Medicine* 2007, **27**(3):627-652.
45. Gil Alterovitz MR, Benson Roseann: **Automation in proteomics and genomics: an engineering case-based approach.** Wiley 2009.
46. Dykstra B, Kent D, Bowie M, McCaffrey L, Hamilton M, Lyons K, Lee S, Brinkman R, Eaves C: **Long-term propagation of distinct hematopoietic differentiation programs in vivo.** *Cell Stem Cell* 2007, **1**(2):218-229.
47. Baerlocher G, Vulto I, de JG, Lansdorp P: **Flow cytometry and FISH to measure the average length of telomeres (flow FISH).** *Nature protocols* 2006, **1**(5):2365..
48. Brinkman R, Gasparetto M, Lee S, Ribickas A, Perkins J, Janssen W, Smiley R, Smith C: **High-content flow cytometry and temporal data analysis for defining a cellular signature of graft-versus-host disease.** *Biol Blood Marrow Transplant* 2007, **13**(6):691-700.
49. Yeh C, Hsi B, Lee S, Faulk W: **Propidium iodide as a nuclear marker in immunofluorescence. II. Use with cellular identification and viability studies.** *J Immunol Methods* 1981, **43**(3):269-275.
50. Aghaeepour N, Khodabakhshi AH, Brinkman RR: **An Empirical Study of Cluster Evaluation Metrics using Flow Cytometry Data.** *Proceedings of NIPS workshop "Clustering: Science or Art"* 2009.

doi:10.1186/1471-2105-11-403

**Cite this article as:** Zare et al.: Data reduction for spectral clustering to analyze high throughput flow cytometry data. *BMC Bioinformatics* 2010 11:403.

**Submit your next manuscript to BioMed Central and take full advantage of:**

- Convenient online submission
- Thorough peer review
- No space constraints or color figure charges
- Immediate publication on acceptance
- Inclusion in PubMed, CAS, Scopus and Google Scholar
- Research which is freely available for redistribution

Submit your manuscript at  
[www.biomedcentral.com/submit](http://www.biomedcentral.com/submit)

