Published in final edited form as:

IEEE/ACM Trans Comput Biol Bioinform. 2014 ; 11(6): 1045–1051. doi:10.1109/TCBB.2014.2321403.

Unfold high-dimensional clouds for exhaustive gating of flow cytometry data

Peng Qiu*

Department of Biomedical Engineering, Georgia Institute of Technology and Emory University, Atlanta, GA, USA

Abstract

Flow cytometry is able to measure the expressions of multiple proteins simultaneously at the single-cell level. A flow cytometry experiment on one biological sample provides measurements of several protein markers on or inside a large number of individual cells in that sample. Analysis of such data often aims to identify subpopulations of cells with distinct phenotypes. Currently, the most widely used analytical approach in the flow cytometry community is manual gating on a sequence of nested biaxial plots, which is highly subjective, labor intensive, and not exhaustive. To address those issues, a number of methods have been developed to automate the gating analysis by clustering algorithms. However, completely removing the subjectivity can be quite challenging. This paper describes an alternative approach. Instead of automating the analysis, we develop novel visualizations to facilitate manual gating. The proposed method views single-cell data of one biological sample as a high-dimensional point cloud of cells, derives the skeleton of the cloud, and unfolds the skeleton to generate 2D visualizations. We demonstrate the utility of the proposed visualization using real data, and provide quantitative comparison to visualizations generated from principal component analysis and multidimensional scaling.

Introduction

Flow cytometry is a technology that can simultaneously measure the expressions of multiple proteins at the single-cell level [1]. The flow cytometry data of one biological sample can be presented in the form of a tall thin matrix, where each column corresponds to one protein marker, and each row corresponds to one individual cell in the sample. Modern flow cytometers are able to simultaneously measure up to 12 proteins routinely, and the capacity of the next-generation mass cytometer is more than 30 [2,3]. The total number of cells varies depending on the experiment design, and is typically on the order of a hundred thousand. Such single-cell data contains information on the cellular heterogeneity of the sample, which is of great biological interests [4].

The goal of flow cytometry data analysis is often to identify subtypes of cells with distinct phenotypes, which is essentially a clustering problem. Currently, the most widely used approach in the flow cytometry and immunology communities is manual gating [5, 6]. A

*Corresponding author peng.qiu@bme.gatech.edu.

Competing interests The author declares that he has no competing interests.

manual gate is a user-defined region in a biaxial plot of two protein markers, which is used to select cells with a desired phenotype. Cells inside one gate are visualized in other biaxial plots, in which further gates are drawn to refine the selection. The result of manual gating is a hierarchy of gates in a user-defined sequence of nested biaxial plots. Each gate and the cells in it are annotated with a different phenotype according to prior knowledge. Manual gating is subjective because the sequence of biaxial plots relies on user's prior knowledge and interpretation of the biological system underlying the data. Moreover, manual gating is not exhaustive, because the manual gates usually do not cover all the cells, leaving a non-trivial amount of the cells unannotated.

To remove the subjectivity and achieve exhaustive gating, a number of methods have been developed to automate the gating analysis using clustering algorithms, such as K-means [7–9], mixture models, [10–13], density-based clustering [14–17], and spectral analysis [18]. Many of those methods include mechanisms for both determining the number of clusters and clustering the cells, which remove the subjectivity from gating analysis. However, it is difficult to tune a clustering algorithm to identify all the subpopulations that a human expert would define, due to the fact that cell counts of abundant and rare subpopulations are usually quite unbalanced.

In this paper, we take an alternative approach. Instead of automating the gating analysis, we propose to develop novel visualizations to facilitate manual exploration of the data. We believe that many disadvantages of manual gating are caused by the poor visualization of biaxial plots which only display pairwise relationships. Better visualizations that encode higher order relationships can greatly facilitate manual analysis. A recent method, SPADE, illustrated that tree diagrams can be used to approximate high-dimensional relationships among clusters of cells in cytometry data [19,20]. Motivated by SPADE, we propose a new 2D visualization that captures high-dimensional relationships among individual cells. The purpose is to enable manual gating analysis on one 2D visualization, rather than a sequence of nested biaxial plots in traditional manual gating. The proposed method views a flow cytometry dataset as a high-dimensional point cloud, applies density-dependent downsampling to balance the sizes of abundant and rare subpopulations, derives the skeleton of the downsampled cloud by K-means clustering and minimum spanning tree, and unfolds the skeleton to generate a 2D visualization. A motivating example and details of the method are described in the following sections.

Materials and Methods

Motivating example

This section describes manual gating analysis of a real flow cytometry dataset, which will also be used to illustrate the proposed method in the Results section. The data is derived from a healthy mouse bone marrow sample, and is publicly available [19]. The total number of cells is 500,000. The measured protein markers include: c-kit, CD11b, B220, TCR β , CD4 and CD8, which can be used to delineate major cell types in mouse bone marrow [21]. Immature progenitor cells are c-kit positive. Myeloid cells are positive for CD11b, whereas lymphoid cells are negative for this marker. Within the lymphoid population, B cells express B220 but not TCR β , and the majority of T cells express TCR β but not B220. Mature T cells

can be divided into two subtypes, characterized by mutually exclusive expression of CD4 and CD8. In addition to the protein markers, this dataset contains measurements of forward and side scatters (FSC and SSC) for each cell, which reflect cell size and granularity.

Manual gating of the mouse bone marrow dataset is shown in Figure 1 (a). The analysis starts with the first biaxial plot which uses SSC and CD11b to visualize all 500,000 cells. Contours are used to visualize the density of cells in the biaxial plot. The contours suggest that there are two major subpopulations, and two gates are manually drawn to separate them. The upper gate contains cells that are CD11b positive, and is annotated as myeloid cells according to prior knowledge of mouse hematopoiesis. Cells in the lower gate are further visualized in the second biaxial plot using B220 and TCR β . Three gates are drawn for the subpopulations observed in the second plot, and two of them are annotated as B cells and T cells because of their mutually exclusive expression of B220 and TCR β . Following the same logic, the third plot uses CD4 and CD8 to identify the two subtype of T cells. The forth plot reveals a B220⁺CD4⁺ subpopulation within the B cell gate, which is annotated as dendritic cells. Finally, the last biaxial plot shows that a c-kit positive subpopulation exists in the B220⁻TCR β ⁻ gate in the second plot. During the gating analysis, a set of gates are manually drawn on a user-defined sequence of nested biaxial plots. Each gate contains cells with a distinct phenotype.

If we consider the data as a high-dimensional point cloud of cells, manual gating can be viewed as a user-guided process to understand the shape of the cloud. A cartoon illustration of this intuition is shown in Figure 1 (b). In the first biaxial plot, the point cloud is projected onto a 2D subspace defined by SSC and CD11b, in which the cloud appears to be composed of two arms. Cells in the lower arm are projected onto another 2D subspace defined by B220 and TCR β in the second biaxial plot, which shows that the lower arm can be divided into three smaller arms. Similarly, each subsequent biaxial plot focuses on one arm of the cloud and reveals more detailed structures within that arm. The last plot in Figure 1 (b) conceptualizes the shape of the point cloud revealed by the gating process. Since the point cloud itself is a high-dimensional object which cannot be visualized, the 2D illustration in Figure 1 (b) can be viewed as an unfolded version of the cloud. This idea motivates the following question: is it possible to computationally unfold a high-dimensional point cloud so that the unfolded version fits in a 2D plane, similar to unfolding an origami back to a piece of paper? The following section describes an answer to this question.

Unfold high-dimensional point clouds

The proposed method views a flow cytometry dataset as a high-dimensional point cloud of cells, and derives a 2D visualization of the shape of the cloud. In Figure 2, the proposed method is shown using a flowchart: density-dependent downsampling, K-means clustering, minimum spanning tree (MST) construction, 2D visualization by unfolding the MST, and mapping of individual cells to the unfolded MST.

Density-dependent downsampling—If we view a flow cytometry dataset as a point cloud and examine the density of points in different regions of the cloud, we typically observe that the density can vary dramatically across different regions. Dense regions

correspond to abundant cell types, whereas sparse regions correspond to rare cell types and noise. One critical challenge in clustering those cells is the extremely unbalanced sizes of different cell types. Since the primary goal here is to visualize the shape of the cloud, density-dependent downsampling is applied to remove the density variation within the cloud while preserving its shape [19].

Density-dependent downsampling stochastically removes cells according to two user-defined parameters, target density (TD) and outlier density (OD):

$$p(\text{remove } i) = \begin{cases} 1, & \text{if } LD_i \leq OD \\ 0, & \text{if } OD < LD_i \leq TD \\ 1 - \frac{TD}{LD_i}, & \text{if } LD_i > TD \end{cases}$$

where LD_i is the local density for cell i , defined by the number of cells within its neighborhood. An empirical choice of the neighborhood size is 5 times the median distance from a randomly chosen cell to its nearest neighbor. During density-dependent downsampling, cells with local densities $< OD$ are considered as noise and removed. Cells whose local densities are between OD and TD are not removed. Cells in dense regions ($> TD$) are stochastically removed, so that the densities reduce to TD after the downsampling process.

K-means clustering—After density-dependent downsampling, standard K-means clustering is applied to partition the downsampled cloud into small pieces. Since the density variation in the cloud is removed in the previous step, the resulting clusters tend to share similar size, in terms of both their volumes and cell counts. The key setting of this clustering step is that the desired number of clusters k should be much larger than the expected number of subpopulations in the data. Although such a setting inevitably over-partitions the data, it enables the subsequent steps to recover detailed topology of the shape of the point cloud. The author's default setting for k is 150. More discussions of this parameter is available in Results.

MST construction—By representing each cell cluster using its center, the pairwise distances between clusters can be defined by the Euclidean distances between their centers. Using this distance definition, the Boruvka's algorithm [22] is applied to construct an MST to connect all k cluster centers by $k - 1$ edges with minimum total edge length. The MST approximates the skeleton of the cloud.

Unfold MST to 2D—Since the cluster centers are points in the high-dimensional space that the cloud lives in, the MST is also a high-dimensional object that cannot be directly visualized. To automatically generate a 2D visualization for the MST, a modified Fruchterman-Reingold algorithm is applied [23, 24]. The algorithm first places the longest path in the tree horizontally, and then visualizes the remaining tree nodes one by one. When a new node is appended, its 2D position is determined by simulating two forces: (i) repelling forces between the new node and the nodes that are already visualized, and (ii) an attracting force along the new edge. This algorithm encourages smaller branches to be oriented

perpendicularly to the bigger branches to which they are attached. Although the algorithm does not explicitly prohibit edges to cross each other, its resulting layout typically does not contain edge crossings. After this step, each cluster center is assigned to one position in the 2D visualization space.

Map cells to the unfolded MST—Denote the positions of two cluster centers in the data space as $\mathbf{d}_i = [d_{i1}, d_{i2}, d_{i3}, \dots, d_{in}]^T$, $\mathbf{d}_j = [d_{j1}, d_{j2}, d_{j3}, \dots, d_{jn}]^T$, and denote their corresponding positions in the 2D visualization space as $\mathbf{v}_i = [v_{i1}, v_{i2}, 0, \dots, 0]^T$, $\mathbf{v}_j = [v_{j1}, v_{j2}, 0, \dots, 0]^T$. Since any two line segments can be related by a rotation operation and a scaling operation, there exist a non-negative scalar α and a rotation matrix $\mathbf{R}$ that satisfy,

$$\mathbf{v}_j - \mathbf{v}_i = \alpha \mathbf{R} (\mathbf{d}_j - \mathbf{d}_i) \quad (1)$$

The value of α can be uniquely determined, which is the ratio between the length of the two segments. Determining the rotation matrix $\mathbf{R}$ is not as straightforward, and $\mathbf{R}$ is typically not unique. An algorithm for finding one rotation matrix that satisfies equation (1) is provided in Table 1. Using α and $\mathbf{R}$, a data point near the $(\mathbf{d}_i, \mathbf{d}_j)$ segment in the data space can be mapped to a position near the $(\mathbf{v}_i, \mathbf{v}_j)$ segment in the visualization space.

For one point $\mathbf{d}_x$ in the data space, assume $\mathbf{d}_i$ is its nearest cluster center and $(\mathbf{d}_i, \mathbf{d}_j)$ is its nearest tree edge in the data space. This data point is visualized by the following equation:

$$\mathbf{v}_x = \alpha \mathbf{R} (\mathbf{d}_x - \mathbf{d}_i) + \mathbf{v}_i \quad (2)$$

where α and $\mathbf{R}$ are a non-negative scalar and a rotation matrix that satisfy equation (1), and the visualization position of the data point is determined by the first two elements of $\mathbf{v}_x$.

Results

Unfold the mouse bone marrow dataset

To illustrate the proposed visualization method, we apply it to the mouse bone marrow dataset in Figure 1. The input to the proposed method is single-cell the measurements of 6 protein markers on 500,000 individual cells, which can be viewed as a point cloud of 500,000 points in a 6-dimensional space. For density-dependent downsampling, the parameter *OD* is set to be the 5th percentile of the local densities of all cells, meaning that 5% of cells with the lowest local densities are considered as noise and removed. The parameter *TD* is chosen such that 20,000 cells survive the downsampling process. The K-means clustering step requires a pre-specified number of clusters, which is chosen to be 150. After clustering, the proposed method constructs an MST to connect the clusters, unfolds the MST to a 2D visualization space, and maps individual cells to the unfolded MST. Figure 3 shows the 2D visualization of the MST and the 20,000 downsampled cells. The longest path of the MST is placed horizontally, and smaller branches are oriented outwards from the longest path. Individual cells (shown in green) follow and surround the MST, which approximates the skeleton of the point cloud in the 6-dimensional data space.

To demonstrate the effect of density-dependent downsampling, contour plots of cell densities in the visualization space are shown in Figure 4, where the outermost and innermost

contours in both plots represent 10% and 90% of the highest density. Figure 4 (a) shows that the 20,000 downsampled cells are relatively evenly distributed in the visualization space. The density variation is mainly because the cluster centers are not evenly spaced. In contrast, the distribution of all 500,000 cells shown in Figure 4 (b) exhibits much larger density variation in the visualization space. Since the outermost contour only covers half of the visualization, the density of half of the cloud is smaller than 10% of the highest density. Although the contour plots in the 2D visualization space do not fully reflect the effect of downsampling in the 6-dimensional data space, Figure 4 does provide side-evidence that density-dependent downsampling reduces the density variation within the point cloud in the data space.

Contour plots can also be used to visualize the expressions of protein markers. To generate such a plot for one protein marker, a 100×100 grid is applied to the 2D visualization space, and each of the downsampled cells is assigned to its nearest grid point. By defining a “height” above each grid point using the average marker intensity of the downsampled cells assigned to it, a contour plot can be created. In Figure 5, each plot visualizes the expression of one protein marker. From these plots, all gated cell types in Figure 1 can be identified. Based on the coloration of CD11b, the right half of the visualized cloud is positive for CD11b, which corresponds to the myeloid gate. According to the plot colored by B220, the left one third of the cloud expresses B220 and corresponds to the B cell gate. Within the B220⁺ region, the CD4⁺ arm represents the dendritic cells. Plots colored by TCR β , CD4 and CD8 show that the bottom arm of the cloud is TCR β ⁺ and contains two parts that exhibit mutually exclusive expression of CD4 and CD8. This arm corresponds to the three T cell gates. The center region of the visualized cloud is positive for c-kit, which contains cells in the c-kit⁺ gate.

Using the visualization in Figure 5, manual gating can be performed in a new way, by specifying a desired combination of multiple protein markers. For example, to examine whether the data contains cells that express a particular marker combination B220⁺CD4⁺TCR β ⁻, one can specify a set of criteria (i.e. B220>3, CD4>3, TCR β <2) and highlight the regions that satisfy the criteria. The thresholds chosen here are consistent to those used in the gating plots in Figure 1. As highlighted in Figure 6, there exists one subpopulation that express the specified marker combination, which corresponds to the dendritic cells in the gating analysis. By specifying other marker combinations, one can easily manually explore what known phenotypes exist in the data.

Since all phenotypes that exist in the data are reflected in the proposed 2D visualization, after marker combinations of known phenotypes are used to highlight the visualization, regions that do not satisfy any prior knowledge can be easily observed. Marker combinations for those regions can form hypotheses of novel phenotypes that are not known a priori. This is the main advantage of the proposed visualization, *enabling exhaustive gating*.

Comparison with PCA and MDS

For comparison, two well-established dimension reduction methods are applied to the mouse bone marrow data, principal component analysis (PCA) and multidimensional scaling (MDS). PCA is implemented using the “princomp” function in the MATLAB statistical

toolbox. MDS is also implemented using a MATLAB function, “mdscale”, whose default setting performs non-metric MDS using Euclidean dissimilarity and Kruskal's normalized stress criterion [25]. The 20,000 downsampled cells are analyzed with both methods, which produce two sets of 2D coordinates for each of the downsampled cells.

Similar to the previous subsection, contour plots are used to display the 2D visualizations and the variation of each protein marker with respect to it. Results for PCA and MDS are shown in Figures 7 and 8, respectively. In the PCA visualization in Figure 7, a relatively small region is positive for TCR β , and the known mutually exclusive expression of CD4 and CD8 in TCR β^+ cells cannot be observed. A closer look at the principal components reveals that TCR β and CD8 contribute little to the first two principal components. Therefore, the PCA visualization is not able to effectively distinguish TCR $\beta^{+/-}$ cells, and the CD8 $^{+/-}$ cells. Qualitatively, the non-metric MDS shows similar performance as the proposed method, because all major phenotypes in this dataset are visible in the visualization in Figure 8.

To quantitatively compare PCA, MDS and the proposed method, we propose a variation of the trustworthiness measure [26]. During the interpretation of a visualization, proximities in the visualization space are the most salient. When examining a cell, the first perceptions are which other cells are nearby. A high-quality trustworthy visualization should satisfy that: if two cells are visualized to be similar, they should also be similar in the original high-dimensional data space. To evaluate the trustworthiness of one visualization, we apply K-means to cluster the downsampled cells in the visualization space and evaluate the quality of the clusters in the data space. In this analysis, K ranges from 5 to 50, and K-means is performed 100 times with random initializations for each choice of K. For one particular run of K-means, we compute the standard deviation of each marker within each cluster and record the average of the K*6 standard deviation values. This average measures the quality of the clusters derived from the visualization space. Figure 9 shows the mean and error-bar of the quality measures of the three visualizations, where we can see that the proposed method is more trustworthy and consistently outperforms PCA and MDS.

Discussion

When manually exploring the proposed visualization, one needs to specify marker combinations and thresholds. The appropriate thresholds are often data-dependent (preprocessing, transformation and normalization). A simple way to identify appropriate thresholds is by examining the biaxial plots in Figure 1. Therefore, the proposed visualization should be used in conjunction with biaxial plots, rather than replacing them. Similar to the traditional manual gating, the proposed method requires user's prior knowledge. However, as mentioned above, the main advantage of the proposed visualization is that it enables exhaustive gating.

In the K-means clustering step, the desired number of clusters is a tuning parameter. As stated in Materials and Methods, we intentionally over-partition the point cloud, by setting the number of clusters to be much larger than the number of expected subpopulations. If this parameter is too small, the subsequent MST will not be able to approximate the skeleton of the cloud. If this parameter is too large, the resulting tree will contain many noisy branches

and is difficult to interpret. Although MST is sensitive to this parameter, after the individual cells are visualized, small noisy branches tend to be buried in the final visualization (i.e., Figures 5). Empirically, 100 ~ 300 is a good choice for the desired number of clusters. Defining the optimal number of clusters for tree-based skeletonisation is a challenging problem. Another interesting future direction is how to identify the skeleton of a point cloud without clustering.

The robustness of the proposed visualization is difficult to quantify. Density-dependent downsampling and K-means clustering both involve randomness. Since MST is known to be sensitive to noise, the randomness will affect the tree structure and the final visualization of individual cells. However, in multiple runs of the proposed analysis, the main phenotypes (marker combinations) are always separated and reflected by different regions in the 2D visualization. Therefore, for the purpose of biological interpretation, the proposed visualization method is robust. To support this claim and enable the community to test the proposed method, we provide the raw data and MATLAB code for our analysis at <http://odin.mdacc.tmc.edu/~pqiu/software/UNFOLD/index.html>.

Acknowledgements

The project described was supported by Award Number R01 CA163481 from the National Cancer Institute. The content is solely the responsibility of the author and does not necessarily represent the official views of the National Cancer Institute or the National Institutes of Health.

References

1. Chattopadhyay P, Price D, Harper T, Betts M, Yu J, Gostick E, Perfetto S, Goepfert P, Koup R, De Rosa S, Bruchez M, Roederer M. Quantum dot semiconductor nanocrystals for immunophenotyping by polychromatic flow cytometry. *Nature Medicine*. 2006; 12(8):972–977.
2. Bandura D, Baranov V, Ornatsky O, Antonov A, Kinach R, Lou X, Pavlov S, Vorobiev S, Dick J, Tanner S. Mass Cytometry: Technique for Real Time Single Cell Multitarget Immunoassay Based on Inductively Coupled Plasma Time-of-Flight Mass Spectrometry. *Analytical Chemistry*. 2009; 81(16):6813–6822. [PubMed: 19601617]
3. Bendall S, Simonds E, Qiu P, Amir E, Krutzik P, Finck R, Bruggner R, Melamed R, Ornatsky O, Balderas R, Plevritis S, Sachs K, Pe'er D, Tanner S, Nolan G. Single Cell Mass Cytometry of Differential Immune and Drug Responses Across the Human Hematopoietic Continuum. *Science*. 2011; 332(6030):687–696. [PubMed: 21551058]
4. Newell E, Sigal N, Bendall S, Nolan G, Davis M. Cytometry by Time-of-Flight Shows Combinatorial Cytokine Expression and Virus-Specific Cell Niches within a Continuum of CD8 + T Cell Phenotypes. *Immunity*. 2012; 36:142–152. [PubMed: 22265676]
5. Herzenberg L, Tung J, Moore W, Herzenberg L, Parks D. Interpreting flow cytometry data: a guide for the perplexed. *Nature Immunology*. 2006; 7(7):681–685. [PubMed: 16785881]
6. Hahne F, LeMeur N, Brinkman R, Ellis B, Haaland P, Sarkar D, Spidlen J, Strain E, Gentleman R. FlowCore: a Bioconductor package for high throughput flow cytometry. *BMC Bioinformatics*. 2009; 10:106. [PubMed: 19358741]
7. Murphy RF. Automated identification of subpopulations in flow cytometric list mode data using cluster analysis. *Cytometry*. 1985; 6(4):302–309. [PubMed: 4017796]
8. Aghaeepour N, Nikolic R, Hoos H, RR B. Rapid cell population identification in flow cytometry data. *Cytometry A*. 2011; 79:6–13. [PubMed: 21182178]
9. Ge Y, Sealfon S. flowPeaks: a fast unsupervised clustering for flow cytometry data via K-means and density peak finding. *Bioinformatics*. 2012; 28(15):2052–2058. [PubMed: 22595209]

10. Lo K, Brinkman R, Gottardo R. Automated gating of flow cytometry data via robust model-based clustering. *Cytometry A*. 2008; 73(4):321–332. [PubMed: 18307272]
11. Boedigheimer M, Ferbas J. Mixture modeling approach to flow cytometry data. *Cytometry A*. 2008; 73(5):421–429. [PubMed: 18383311]
12. Chan C, Feng F, Ottinger J, Foster D, West M, Kepler T. Statistical mixture modeling for cell subtype identification in flow cytometry. *Cytometry A*. 2008; 73(8):693–701. [PubMed: 18496851]
13. Pyne S, Hu X, Kang K, Rossin E, Lin T, Maier L, Baecher-Allan C, McLachlan G, Tamayo P, Hafler D, De Jager P, Mesirov J. Automated high-dimensional flow cytometric data analysis. *Proceedings of the National Academy of Science*. 2009; 106(21):8519–8524.
14. Walther G, Zimmerman N, Moore W, Parks D, Meehan S, Belitskaya I, Pan J, Herzenberg L. Automatic Clustering of Flow Cytometry Data with Density-Based Merging. *Advances in Bioinformatics*. 2009; 2009:686759. [PubMed: 20069107]
15. Qian Y, Wei C, Lee F, Campbell J, Halliley J, Lee J, Cai J, Kong Y, Sadat E, Thomson E, Dunn P, Seegmiller A, Karandikar N, Tipton C, Mosmann T, Sanz I, Scheuermann R. Elucidation of seventeen human peripheral blood B-cell subsets and quantification of the tetanus response using a density-based method for the automated identification of cell populations in multidimensional flow cytometry data. *Cytometry Part B: Clinical Cytometry*. 2010; 78B(S1):S69–S82.
16. Naumann U, Luta G, Wand M. The curvHDR method for gating flow cytometry samples. *BMC Bioinformatics*. 2010; 11:44. [PubMed: 20096119]
17. Sugar I, Sealfon S. Misty Mountain clustering: application to fast unsupervised flow cytometry gating. *BMC Bioinformatics*. 2010; 11:502. [PubMed: 20932336]
18. Zare H, Shoostari P, Gupta A, Brinkman R. Data reduction for spectral clustering to analyze high throughput flow cytometry data. *BMC Bioinformatics*. 2010; 11:403. [PubMed: 20667133]
19. Qiu P, Simonds E, Bendall S, Gibbs K Jr, Bruggner R, Linderman M, Sachs K, Nolan G, Plevritis S. Extracting a cellular hierarchy from high-dimensional cytometry data with SPADE. *Nature Biotechnology*. 2011; 29(10):886–891.
20. Qiu P. Inferring phenotypic properties from single-cell characteristics. *PLoS One*. 2012; 7(5):e37038. [PubMed: 22662133]
21. Bryder D, Rossi D, Weissman I. Hematopoietic Stem Cells: The Paradigmatic Tissue-Specific Stem Cell. *Am J Pathol*. 2006; 169(2):338–346. [PubMed: 16877336]
22. Pettie S, Ramach V. An Optimal Minimum Spanning Tree Algorithm. *Journal of the ACM*. 1999; 49:49–60.
23. Fruchterman T, Reingold E. Graph Drawing by Force-directed Placement. *Software Practice and Experience*. 1991; 21:1129–1164.
24. Qiu P, Plevritis SK. TreeVis: A MATLAB-based tool for tree visualization. *Computer Methods and Programs in Biomedicine*. 2013; 109:74–76. [PubMed: 23036855]
25. Cox, T.; Cox, M. *Multidimensional Scaling*, Second Edition. Chapman & Hall/CRC; 2000.
26. Kaski S, Nikkila J, Oja M, Venna J, Toronen P, Castren E. Trustworthiness and metrics in visualizing similarity of gene expression. *BMC Bioinformatics*. 2003; 4(48):e24592.

Figure 1. Manual gating analysis

Analysis of a mouse bone marrow flow cytometry dataset. (a) The sequence of biaxial plots used in manual gating, and the gates that identify subpopulations with different phenotypes.

(b) Illustration of the point cloud analogy: the shape of the data as understood from the manual gating process.

Figure 2. Flowchart of the proposed method

Computational pipeline for unfolding a multi-dimensional point cloud to a 2D visualization.

Figure 3. Unfold the mouse bone marrow data

2D visualization of the 6-dimensional mouse bone marrow dataset. The MST is shown in blue. The green dots represent the 20,000 cells after downsampling.

Figure 4. Contour plots of cell density in the visualization space

Contour plots of distribution of cells in the visualization space for (a) the 20,000 cells after downsampling, and (b) all the 500,000 cells in the original dataset.

Figure 5. Contour plots of protein expression

Contour plots drawn according to the expression of protein markers. Each plot is colored by one protein marker, where red indicates regions that contain cells positive for the marker, and blue indicates regions consist of cells that are negative for the marker.

Highlight the population with $B220 > 3$, $CD4 > 3$, $TCR\beta < 2$

Figure 6. Gate by a specific marker combination

Contour plot that highlights a population with user-specified phenotype: $B220 > 3$, $CD4 > 3$, $TCR\beta < 2$. This population corresponds to the dendritic cells in the gating analysis.

Figure 7. Visualization by PCA

Dimension reduction and visualization by PCA. Each contour plot is colored by one protein marker, where red indicates regions that contain cells that highly express the marker, and blue indicates regions consist of cells that are negative for the marker.

Figure 8. Visualization by MDS
Dimension reduction and visualization by MDS.

Figure 9. Comparison of Unfold, PCA and MDS

Comparison of trustworthiness of the three visualizations in Figures 5, 7 and 8, derived from the mouse bone marrow data using PCA, MDS and the proposed method UNFOLD.

Table 1

Algorithm for finding one rotation matrix $\mathbf{R}$ such that $\frac{\mathbf{v}}{\|\mathbf{v}\|} = \mathbf{R} \frac{\mathbf{d}}{\|\mathbf{d}\|}$. This algorithm constructs $\mathbf{R}$ by a sequence of 2D rotation operations.

-
- 1: Give two column vectors:
 $\mathbf{v} = [v_1, v_2, \dots, v_n]^T, \mathbf{d} = [d_1, d_2, \dots, d_n]^T$
 - 2: $\mathbf{v} = \frac{\mathbf{v}}{\|\mathbf{v}\|}, \mathbf{d} = \frac{\mathbf{d}}{\|\mathbf{d}\|}$
 Normalize to unit norm:
 - 3: Initialize $\mathbf{R}$ as identity matrix: $\mathbf{R} = \mathbf{I}_{n \times n}$
 - 4: **while** $(\sum_{i=1, \dots, n} 1(v_i \neq d_i)) \geq 2$ **do**
 - 5: Find elements to operate on:
 $a = \arg \min_{i, v_i \neq d_i} |v_i|$
 $b = \max \left\{ i: v_a^2 \leq d_a^2 + d_i^2, v_i \neq d_i \right\}$
 - 6: Find θ that solves

$$\begin{bmatrix} \frac{va}{\sqrt{d_a^2 + d_b^2 - v_a^2}} \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} d_a \\ d_b \end{bmatrix}$$
 - 7: Construct an intermediate rotation matrix:
 $\mathbf{R}_{tmp} = \mathbf{I}_{n \times n}, \mathbf{R}_{tmp}(a, a) = \cos \theta, \mathbf{R}_{tmp}(b, b) = \cos \theta, \mathbf{R}_{tmp}(a, b) = -\sin \theta, \mathbf{R}_{tmp}(b, a) = \sin \theta$
 - 8: Update: $\mathbf{d} = \mathbf{R}_{tmp} \mathbf{d}, \mathbf{R} = \mathbf{R}_{tmp} \mathbf{R}$
 - 9: **end while**
 - 10: **if** $(\sum_{i=1, \dots, n} 1(v_i \neq d_i)) = 1$ **then**
 - 11: $a = \{i: v_i \neq d_i\}$
 - 12: $\mathbf{R}_{tmp} = \mathbf{I}_{n \times n}, \mathbf{R}_{tmp}(a, a) = -1$
 - 13: $\mathbf{d} = \mathbf{R}_{tmp} \mathbf{d}, \mathbf{R} = \mathbf{R}_{tmp} \mathbf{R}$
 - 14: **end if**
-