Algorithmic Tools for Mining High-Dimensional Cytometry Data

Cariad Chester and Holden T. Maecker

_J Immunol_ 2015; 195:773-779; doi: 10.4049/jimmunol.1500633

http://www.jimmunol.org/content/195/3/773

Supplementary Material
http://www.jimmunol.org/content/suppl/2015/08/03/195.3.773.DCSupplemental

References
This article cites 25 articles, 4 of which you can access for free at:
http://www.jimmunol.org/content/195/3/773.full#ref-list-1

Why _The JI_? Submit online.

• Rapid Reviews! 30 days* from submission to initial decision
• No Triage! Every submission reviewed by practicing scientists
• Fast Publication! 4 weeks from acceptance to publication

*average

Subscription
Information about subscribing to _The Journal of Immunology_ is online at:
http://jimmunol.org/subscription

Permissions
Submit copyright permission requests at:
http://www.aai.org/About/Publications/JI/copyright.html

Email Alerts
Receive free email-alerts when new articles cite this article. Sign up at:
http://jimmunol.org/alerts

_The Journal of Immunology_ is published twice each month by The American Association of Immunologists, Inc., 1451 Rockville Pike, Suite 650, Rockville, MD 20852
Copyright © 2015 by The American Association of Immunologists, Inc. All rights reserved.
Print ISSN: 0022-1767 Online ISSN: 1550-6606.
Algorithmic Tools for Mining High-Dimensional Cytometry Data

Cariad Chester*† and Holden T. Maecker*

The advent of mass cytometry has led to an unprecedented increase in the number of analytes measured in individual cells, thereby increasing the complexity and information content of cytometric data. Although this technology is ideally suited to the detailed examination of the immune system, the applicability of the different methods for analyzing such complex data is less clear. Conventional data analysis by manual gating of cells in biaxial dot plots is often subjective, time consuming, and neglectful of much of the information contained in a highly dimensional cytometric dataset. Algorithmic data mining has the promise to eliminate these concerns, and several such tools have been applied recently to mass cytometry data. We review computational data mining tools that have been used to analyze mass cytometry data, outline their differences, and comment on their strengths and limitations. This review will help immunologists to identify suitable algorithmic tools for their particular projects. *The Journal of Immunology, 2015, 195: 773–779.

Single-cell cytometry techniques permit phenotypic and functional analyses of large numbers of individual immune cells and have provided numerous insights into basic, translational, and clinical immunology. Historically, software facilitating manual gating via biaxial plots and bar graphs has been the predominant platform for exploring cytometry data. In “manual” gating, cell subsets of interest are identified from parent populations via visual inspection of dot plots displaying individual cells’ fluorescence intensities. Despite considerable efforts to harmonize immunophenotyping and gating strategies for multicenter studies (1), this approach suffers from individual user bias when delineating population boundaries and requires prior knowledge of the cell type of interest. The increasing efforts in systems-level immunology and biomarker-driven research are not well served by this historical approach alone. Analyses by manual gating focus on specific populations, which often represent only a fraction of the total information contained in a cytometric dataset (2). Relationships between populations can be overlooked and because biases and a priori knowledge dictate analysis, discovery of meaningful, but yet undefined, populations is difficult. Additionally, manual gating is not scalable; as the number of parameters increases, analyzing higher-dimensional data by manual gating quickly becomes impractical.

The advent of mass cytometry enables the measurement of an unprecedented number of parameters. Single-cell analyses of > 40 parameters are now feasible (3, 4). However, the complexity of mass cytometry data complicates analysis: to visually analyze all combinations for a 40-parameter dataset would necessitate examining 780 two-dimensional dot plots. Clearly, manual gating alone is insufficient for exploring the full complexity of mass cytometry data in systematic and exhaustive ways.

In response to the limitations of manual gating, the last decade has witnessed the development and application of computational methods to analyze cytometry data. Most existing algorithms for flow cytometry data analysis automatically identify cell populations based on unsupervised clustering according to their marker expression profiles, allowing an unbiased investigation of cytometry data (5). Beyond that, some algorithms provide the capacity to identify rare populations, match cell populations across samples, and statistically compare features between different populations (6–8). Once established, workflows that include algorithmic analyses are less labor intensive than manual gating and can consider multidimensional relationships within the data. Algorithms also provide an unsupervised analysis, allowing an unbiased investigation of cytometry data. Although unsupervised data analysis can be useful to identify aberrations of the immune system without knowing the target phenotype, the success of the approach still depends on the chosen analytes for an experiment and the quality of the input data.

In this review, computational approaches are divided into dimensionality-reduction techniques, clustering-based analyses, and a trajectory detection algorithm (Table I). Although we have not tried to compare the algorithms in a direct competition, example outputs of the most accessible algorithms are shown in Supplemental Fig. 1. Despite the applicability of many previously developed algorithms to mass cytometry data, we focus on algorithms that have been explicitly applied to mass cytometry.
data. Conversely, the algorithms that we discuss are all applicable to data generated on fluorescence-based flow cytometers. We attempt to provide descriptions of not just the relevant algorithms, but also the underpinning statistical techniques that they use. As such, this review functions as both a primer on working with high-dimensional data and a guide to the current suite of algorithms available for immunologic research using mass cytometry.

**Dimensionality reduction**

The aim of dimensionality reduction is to display and analyze high-dimensional data (e.g., 40 surface markers) in a lower-dimensional space, using surrogate dimensions. The surrogate dimensions facilitate plotting of data in two or three dimensions and aim to preserve the significant information in the high-dimensional data. The lower-dimensional plot (two or three dimensions) provides an accessible visualization of the structure of multidimensional data. Two tools for dimensionality reduction that have been applied to mass cytometry data are principal component analysis (PCA) and stochastic neighbor embedding (SNE).

**Principal component analysis.** PCA is a widely used technique for reducing dimensionality of multivariate data by condensing it to its principal components (PC) (9). The resulting PC represent a new set of variables that recapitulate the variance in the original data, which are ordered by the amount of variance they explain. In an artificial/hypothetical three-dimensional dataset that can be visualized as an ellipsoid cloud of data points (analogous to a cell sample analyzed using three Abs), the first PC runs through the longest axis of the ellipsoid (Fig. 1). The second PC is perpendicular to the first PC and runs through the second longest axis. The third PC runs through the shortest axis of the ellipsoid and remain orthogonal to the first and second PC. In higher-dimensional mass cytometry data, the approach is the same but iterates through more dimensions. By relying on the first two or three PC and ignoring later PC, the underlying structure of the data can be summarized in significantly fewer dimensions than exist in the initial data. However, PCA does not explicitly assign cells to discrete clusters. Any observed separation between populations is the result of single cells segregating when graphed along the PC axes.

**Table I. Current computational methods applied to mass cytometry data**

<table>
<thead>
<tr>
<th>Method</th>
<th>Common Application</th>
<th>Advantages</th>
<th>Limitations</th>
<th>Availability</th>
</tr>
</thead>
<tbody>
<tr>
<td>PCA</td>
<td>Visualize relationships in multidimensional data</td>
<td>Familiar and established</td>
<td>Linear dimensionality-reduction technique</td>
<td>Scripts available for MATLAB, Stata, R, SPSS, and so forth</td>
</tr>
<tr>
<td>viSNE</td>
<td>Visualize relationships in multidimensional data</td>
<td>Nonlinear dimensionality-reduction technique</td>
<td>Difficult to compare groups of samples</td>
<td>Part of CYT: <a href="http://www.c2b2.columbia.edu/danapeerlab/html/cyt-download.html">http://www.c2b2.columbia.edu/danapeerlab/html/cyt-download.html</a></td>
</tr>
<tr>
<td>ACCENSE</td>
<td>Identify clusters within multidimensional data without losing single-cell resolution</td>
<td>Nonlinear dimensionality-reduction technique</td>
<td>Difficult to compare groups of samples</td>
<td>After installation, plug-and-play GUI <a href="http://www.cellaccense.com/">http://www.cellaccense.com/</a></td>
</tr>
<tr>
<td>SPADE</td>
<td>Visualize signaling data on an MST of clustered cells</td>
<td>Visualization of fold change data</td>
<td>Loss of single-cell resolution</td>
<td>Cytobank <a href="http://www.cytobank.org">www.cytobank.org</a> FlowJo</td>
</tr>
<tr>
<td>FlowSOM</td>
<td>Cluster cells and visualize marker expression</td>
<td>Computationally undemanding</td>
<td>No stand-alone application</td>
<td>R code available on GitHub, script editing required for use <a href="https://github.com/SofieVG/FlowSOM">https://github.com/SofieVG/FlowSOM</a></td>
</tr>
<tr>
<td>Citrus</td>
<td>Identify cell populations that are associated with an experimental end point</td>
<td>Built-in regression analysis to compare groups of samples and identify clusters that are different between end points</td>
<td>Need ~10+ samples/group for valid comparisons</td>
<td>R code available on GitHub <a href="https://github.com/nolanlab/citrus">https://github.com/nolanlab/citrus</a></td>
</tr>
</tbody>
</table>

![FIGURE 1. PCA of a three-dimensional point cloud. The PC are ordered by their capacity to explain the variance in the data.](http://www.jimmunol.org/)
To illustrate this point, we consider a multidimensional dataset of only B cells and T cells. The first PC might comprise mutually exclusive lineage-restricted markers, such as CD2, CD3, CD7, CD19, CD20, and CD22, each of which would similarly allow for distinguishing B and T cells, therefore being redundant in terms of variance in the dataset. When the data are graphed along an axis made from this PC, the two populations can be separated based on lineage markers. The second PC may be based on expression levels of molecules shared between B and T cells but present at different frequencies or signal intensities, such as CD45RA, CD27, and certain chemokine receptors (e.g., CCR7, CXCR5, and CXCR3), effectively allowing the differentiation of different cell subsets. By graphing PC1 versus PC2 in a two-dimensional plot, distinct populations would be resolved. The loading of the PC indicates how much each original measurement parameter contributed to each new PC. In our example, the loading would reveal how well each marker explains the variance in the data. If the first and second PC only capture a small percentage of the variance, then the usefulness of the PCA is limited; the majority of the variation is hidden in other PC.

PCA has been used to analyze mass cytometry data of human CD8+ T cells (10). Three-dimensional PCA was combined with PyMOL, a molecular visualization program, to create a spatial map of the phenotypes within the CD8+ T cell compartment: the data formed a folded-Y pattern, with naive T cells at the base and short-lived effector cells and central memory cells located terminally along the arms of the Y. The projection was used to illustrate the continuum of T cell differentiation between canonical T cell subsets. The investigators also used the PCA projection to identify unique phenotypical niches occupied by virus-specific cells: EBV-specific CD8+ T cells had phenotypes similar to effector memory T cells, whereas influenza-specific CD8+ T cells demonstrated phenotypes similar to central memory T cells. PCA can be performed in most statistical software (e.g., SPSS, Stata) or in R using publicly available scripts.

PCA performs linear transformations to reduce dimensionality, and biological systems can contain many nonlinear relationships. This conflict can result in the production of misleading results by some PCA. Specifically, linear mapping preserves large pairwise distances in the data; two points that are far apart in multidimensional space appear proximate when linearly reduced to two dimensions. Prioritizing large pairwise distances when translating multidimensional data to low dimensions can create inaccurate representations. This caveat to PCA-driven analyses is illustrated in a nonlinear manifold, like a Swiss roll (Fig. 2). In this example, the Euclidean distance between two points on the Swiss roll does not accurately reflect their dissimilarity. The pairwise distance between the two points is low, suggesting that the points are similar, but if the entire structure of the manifold is considered, the points are far apart (11).

Nonlinear dimensionality reduction through t-distributed SNE

To overcome the limitations of linear transformations inherent in PCA, nonlinear dimensionality-reduction techniques can be used. A recently developed tool for nonlinear dimensionality reduction, t-distributed SNE (t-SNE), provides a more accurate representation of multidimensional data during the translation to low dimensionality (12). t-SNE is the foundational algorithm in two tools for analyzing CyTOF data: viSNE and automatic classification of cellular expression by nonlinear stochastic embedding (ACCENSE).
ACCENSE. ACCENSE combines t-SNE mapping with a built-in feature that helps to identify discrete clusters of cells (16). ACCENSE was developed in Python and released as an easily navigated graphical user interface (GUI) (http://www.cellaccense.com/). In the GUI, t-SNE maps can be colored by marker expression, like in viSNE. Additionally, ACCENSE can partition the t-SNE map into clusters by applying a peak-detection algorithm that identifies local maxima within the two-dimensional t-SNE data. The number of clusters is controlled by a user-specified threshold p value. Identifying clusters within dimensionality-reduced projections of single-cell data is difficult because of the continuity of phenotypes and disjointed protein-expression patterns. The density-partitioned, color-coded clusters produced by ACCENSE make visualizing cellular subpopulations easier than in viSNE or a standard t-SNE scatter plot.

In its original application, ACCENSE was used to examine the CD8+ T cell compartment in specific pathogen–free and germ-free mice (16). ACCENSE successfully recovered populations corresponding to naive, central memory, and effector memory CD8+ T cells, but it also revealed phenotypic heterogeneity within these populations. In both specific pathogen–free and germ-free mice, ACCENSE identified a subpopulation with the same or proximate clusters, and different observations are localized in separate clusters. In a cellular analysis, cells with similar phenotypes are grouped together. The primary drawbacks of clustering approaches are the loss of single-cell resolution and the occasional requirement for prespecification of the number of target clusters desired, introducing bias regarding a quantity that is rarely known.

Clustering approaches

Clustering algorithms begin the process of visualizing high-dimensional data by defining and mapping discrete clusters of cells. Clustering operates by grouping observations into distinct clusters so that similar observations are confined within the same or proximate clusters, and different observations are localized in separate clusters. In a cellular analysis, cells with similar phenotypes are grouped together. The primary drawbacks of clustering approaches are the loss of single-cell resolution and the occasional requirement for prespecification of the number of target clusters desired, introducing bias regarding a quantity that is rarely known.

Spanning-tree progression analysis of density-normalized events. The first tool created to help visualize mass cytometry data was spanning-tree progression analysis of density-normalized events (SPADE) (17). SPADE, available on Cytobank (http://cytobank.org/) and as the CytoSPADE source code, implemented clustering and a minimum spanning tree (MST) projection to reduce multidimensional cytometry data to two-dimensional representations where expression of functional markers could be visualized (18). SPADE clustering is an example of hierarchical, agglomerative clustering (Fig. 3). In agglomerative clustering, each point in multidimensional space is initially its own cluster. Then, the most similar clusters merge into an agglomerative, parent cluster. This procedure then iterates until the target number of clusters is reached. Both SPADE and Citrus use agglomerative clustering.

Unlike pure clustering algorithms, like SWIFT (19), by incorporating an MST SPADE recapitulates the underlying continuity of phenotypes that is inherent in cellular differentiation. The SPADE algorithm also preserves rare populations in the initial data by performing a density-dependent downsampling before clustering initializes. Density-dependent downsampling reduces the amount of data points that are fed into the clustering based on a user-defined percentage value. It helps to preserve rare populations by ensuring more equal representation of cells from dense populations and rare populations.

One advantage of SPADE is its color-coded visualization of signal intensity (expression levels) and fold changes, referring to a user-defined sample as a control. This makes it an appropriate tool for visualizing signaling data. In the original description of SPADE, a combination of surface markers and intracellular markers was used to systematically measure and compare the induction of phosphorylation in human bone marrow samples in response to in vitro LPS stimulation (4). Recently, a modification of the SPADE algorithm, named FLOW-MAP, was released (20). FLOW-MAP is similar to SPADE, but the SPADE MST has been replaced with a highly connected graph structure. The introduction of the graph structure decreases the intrarun variability and leads to a more reproducible layout of clusters. The investigators also extended the FLOW-MAP algorithm to incorporate data from multiple time points. Cells from each separate time point are clustered and added to the two-dimensional FLOW-MAP graph sequentially, allowing users to analyze cellular progression along a time course. Initially, FLOW-MAP was applied to mouse embryonic fibroblasts to study cellular reprogramming, but it provides a useful tool for studying differentiation in a variety of experimental systems.

FlowSOM. FlowSOM is a recently released algorithm that provides a similar visualization capacity to SPADE but requires significantly less computation time and allows multiple parameter visualization on the same MST (21). The decreased algorithmic runtime is achieved by using a self-organizing map (SOM) instead of the hierarchical clustering used in SPADE. In hierarchical clustering, all cells simultaneously undergo an agglomerative fusion with their neighbors in a-dimensional space. To avoid losing rare populations as cells cluster, the SPADE algorithm incorporates density-dependent downsampling. In contrast, FlowSOM relies on the properties of the SOM, a type of artificial neural network in which nodes can be represented in a two-dimensional grid (Fig. 4). Through successive iterations of training, each multidimensional data point is assigned to the node that it most closely resembles. When assignment completes, the
SOM contains topological information, and the nodes have different numbers of cells, preserving rare populations in distinct nodes. The SOM training process eliminates the need for density-dependent downsampling, resulting in a significant improvement in the duration of computation relative to SPADE.

The incorporation of star charts into the presentation of the MST is another feature that separates FlowSOM from SPADE. Star charts allow the mean expression of chosen markers to be visualized on a node-by-node basis, allowing more information to be contained in a single plot relative to SPADE. Importantly, star charts simplify the process of identifying cellular populations because they eliminate the need to examine multiple plots to determine the expression of different markers. More of the relevant information for a sample can be contained in one figure. FlowSOM also provides the option of including a meta-plotting of the nodes from the SOM. The meta-plotting functions to group nodes together based on their phenotypic similarity and, via background shading of the MST, can define large cell populations like B cells, CD4+ T cells, and monocytes.

SPADE and FlowSOM are powerful visualization tools, but both rely on stochastic MST, which results in nonidentical MST with potentially different branching structures each time SPADE or FlowSOM is run. This can complicate preliminary data analysis. Also, although both algorithms have the capacity to detect rare populations, this capacity is dependent on having an adequate number of clusters, which is a user-defined parameter.

Cluster identification, characterization, and regression. Cluster identification, characterization, and regression (Citrus) is a useful tool for identifying cellular populations or traits that are associated with an experimental end point (22). Experimental end points are derived from external information used to categorize samples. Common end points include before or after therapeutic intervention, good or poor disease outcome, and healthy or diseased patients. The ability to identify the phenotypically similar cell clusters that are statistically different between experimental end points makes Citrus uniquely appealing as an analytic approach to multidimensional cytometry data.

Citrus begins by hierarchically clustering phenotypically similar cells. This clustering, similar to SPADE’s, is agglomerative and iteratively reduces the number of nodes by pairing the closest nodes in high-dimensional space. The size of the clusters that can be analyzed is limited by the minimum cluster size threshold, which can be user-defined and by default is set at 5% of input events. As clusters are formed, descriptive parameters are translated to a cluster property matrix, which contains information on the proportion of a sample’s cells in each cluster (abundance) and the median expression level of each functional marker. The cluster property matrix is then used to ascertain relationships between clusters and experimental end points using either a correlation or regression models.

Significance analysis of microarrays is the correlational model used to identify relationships between parameter measurements and a response variable or end point (e.g., treated versus untreated or survival time) (22). Initially developed for analyzing gene expression data from microarray experiments, significance analysis of microarrays generates a list of parameters that are correlated with the response variable. The user chooses the cutoff for significance from three levels of significance (0.1, 0.05, or 0.01); at a higher cutoff, a greater number of significant relationships is reported. As the cutoff is made more stringent, the reported number of relationships decreases, but the confidence in the correlations increases. The flexibility in significance threshold provides a way to generate multiple potentially significant relationships that can be investigated further. Citrus’s predictive models, prediction analysis for microarrays and L1-penalized regression (glmnet), are regularized supervised learning algorithms that identify the subset of cluster features that best predicts a sample’s assignment to one of the response variables. By plotting the accuracy of the classification model via cross-validation and the regularization threshold, Citrus outputs a graph that enables investigators to assess the quality of the results and only use the results if the Citrus analysis achieves an acceptable error rate. In the output graph, the ideal model has a low cross-validation error rate and incorporates multiple model features before the addition of features becomes redundant, leading to increased error rates.

Citrus was recently used to analyze blood leukocyte signaling, and it identified a signature that correlated with postoperative patient recovery after hip replacement (23). Surgical patients completed questionnaires that measured clinical parameters, including pain and functional impairment, and donated peripheral blood samples. Performing Citrus on the samples identified a correlation between clinical parameters of surgical recovery and signaling responses in subsets of CD14+ monocytes. The work of Gaudillière et al. (23) represents an elegant implementation of Citrus to analyze multiparameter relationships in high-dimensional parameter space.

**Developmental trajectory detection: Wanderlust**

The growth, differentiation, and morphogenesis of cells within our tissues follow developmental trajectories that are tightly coordinated by extrinsic forces and internal signaling. Traditionally, these trajectories have been represented by an ordered series of distinct, usually relatively major cell subsets, whereas transitional states, represented by less frequent or not easily classifiable cells, remained less well explored. Mass cytometry provides the highly dimensional analytic platform necessary to capture many of the known relevant phenotypical traits of developmental processes. Although the true time dimension of cell differentiation cannot be elucidated by this approach, sequentially ordering cells according to their developmental trajectory provides

![FIGURE 4. SOM with star charts. A SOM is a type of artificial neural network used for clustering. In SOM-based clustering, each cell is assigned to a phenotypically similar node in the network. Characteristics of each node can be visualized via star charts. Star charts provide a way to visualize the mean intensities of the markers for all cells in the dataset assigned to a specific node. The height of each wedge within the star chart indicates the marker’s intensity (i.e., if the wedge reaches the border of the circle, the cells have a high expression of that marker).](attachment:figure4.png)
Further considerations. Algorithmic analysis of mass cytometry data can add insight beyond that found by manual gating. However, the power of computational tools is constrained by the quality of the input data and is influenced by user-specified parameters. Elevated background staining, artifacts such as doublets, broad or narrow signal intensity coefficients of variation, and instrument variability can confound computational tools and compromise their usefulness. Additionally, algorithms that produce outputs with run-to-run inconsistency warrant special attention to ensure reproducible results.

Computational analyses are based on signal-intensity data available for each cell. The consequences of differing signal spread and logarithmic distribution of signals are ameliorated by Z normalization and data transformations, relatively easy features to build into an algorithm. High background staining is more difficult to accommodate computationally and endangers the identification of rare subsets. For example, background staining for an intracellular cytokine may be at a level that is similar or even above the specific staining of dim markers like PD-1 or ICOS. Currently, no computational tool exists for quantifying the quality of data and identifying potentially problematic samples. Before feeding data into an algorithm, it is important to perform a preliminary analysis to confirm that nonspecific staining and processing and acquisition issues have not artificially distorted the data.

Using computational tools that implement stochastic processes introduces concerns regarding the reproducibility of achieved results. For example, although PCA delivers a consistent result when the input data remain constant, other tools involve random downsampling of the input data or transformations of measured data into probabilities, both of which can lead to different results on a run-to-run basis. This poses an apparent problem to the scientific demand for result reproducibility. A potential solution is to use all of the available events in a sample instead of downsampling. However, this approach can be highly computationally demanding. In our experience, PCA, ACCENSE, SPADE, FlowSOM, and Wanderlust did not produce run times exceeding a few hours when examining FCS files containing <100,000 events on a MacBook Pro with a 2.4-GHz, i5 processor and 8 GB of memory. When analyzing files exceeding 100,000 events or performing Citrus analyses, we used machines with multicore processors or accepted overnight run times. Adjustments in the number of parameters included in the analysis did not significantly affect algorithmic run times. Avoiding downsampling in all analyses would necessitate powerful computational infrastructure.
Another solution to the varying reproducibility of computational tools is wider sharing of data and outputs. The Cytobank implementation of SPADE and viSNE is valuable in this regard. Results achieved with these tools can be made directly accessible to the public with the accompanying data, encouraging further analysis. The question of reproducibility also emphasizes the importance of manual gating in confirming and complementing algorithmic analyses. Broader experience with computational tools outside of projects tailored to showcase algorithmic features will determine to what degree algorithms might replace manual gating approaches in the future.

Conclusions

Computational approaches provide powerful tools for exploring both flow cytometry and mass cytometry data and can complement traditional manual gating. Each of the algorithmic tools described above has strengths and limitations, and many are designed to serve distinct purposes. Some perform only dimensionality reduction, others further define cell clusters, and still others provide statistical testing or infer differentiation trajectories. Therefore, understanding the functionality of different algorithms is important for researchers to select the optimal tool for the question being asked in their research. As the popularity of algorithmic approaches increases, newer tools are becoming faster and include more features. The combination of computational approaches and increasing parameters will provide insights into the true complexity of biological systems.

Acknowledgments

We thank Henrik Mei and Michael Leipold for helpful discussions and editing.

Disclosures

The authors have no financial conflicts of interest.

References

Supplementary Figure 1. Example outputs from algorithmic analysis of mass cytometry data. S1a: viSNE analysis (Cytobank implementation) of a peripheral blood mononuclear cell sample stained with 33 extracellular markers. S1b: ACCENSE analysis of the same PBMC sample from S1a. At a significance threshold of 1e-14, the peak-detection algorithm in ACCENSE partitions the single-cell data into 16 distinct clusters. S1c: SPADE analysis of STAT1 phosphorylation signaling in CD45+ cells from a sample of whole blood fixed in the Smart Tube fixation/stabilization buffer. S1d: Citrus analysis of 12 PBMC samples with monocytes with low CD33 expression versus 14 PBMC samples with normal monocyte CD33 expression. As expected, Citrus detects the difference in CD33 expression between experimental groups and shades the corresponding clusters of cells accordingly.