— Supplementary Materials —

Extracting a Cellular Hierarchy from High-dimensional Cytometry Data with SPADE

Peng Qiu[1,4], Erin F. Simonds[2], Sean C. Bendall[2], Kenneth D. Gibbs Jr.[2], Robert V. Bruggner[2], Michael D. Linderman[3], Karen Sachs[2], Garry P. Nolan[2], Sylvia K. Plevritis[1]

[1]Department of Radiology, [2]Department of Microbiology and Immunology, [3]Computer Systems Laboratory, Stanford University; [4]Department of Bioinformatics and Computational Biology, University of Texas, MD Anderson Cancer Center

S1. Additional figures for SPADE result on mouse bone marrow data

To generate the 8-parameter flow cytometry data, we used normal mouse bone marrow, stained for 8 markers (c-kit, Sca-1, CD150, CD11b, B220, TCRβ, CD4, and CD8), and collected data using the LSR2 flow cytometer. The 8 markers were chosen so that we can capture the major cell types in mouse hematopoiesis (Figure 2 (a) in the main text). The raw intensity values were transformed using the inverse hyperbolic sine transformation, which is linear for low intensity values and logarithmic for high intensity values. All subsequent analyses were based on the transformed data. The data were initially gated using FSC and SSC to exclude doublets and debris (as shown in Figure S1).

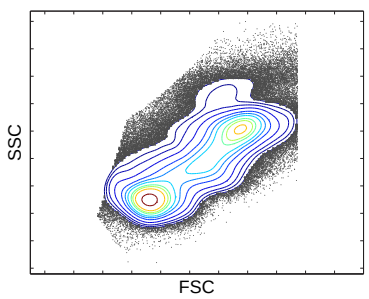


Figure S1: The mouse bone marrow data after an initial gating based on FSC and SSC.

After the initial gating, the total number of cells was ~500,000. Data of the 8 markers across

these cells were the input data to SPADE. The input parameters included: markers used to build the SPADE tree, outlier density, target density, and desired number of clusters. In this analysis, we chose to use all 8 markers to build the SPADE tree; the outlier and target densities were chosen to be the 1st and 5th percentile of local densities of all the cells; the desired number of clusters was 50.

SPADE downsampled the data in a density-dependent fashion to remove the density variation. After downsampling using the above parameters, the number of remaining cells was ∼30,000. SPADE then performed agglomerative clustering and minimum spanning tree construction. The resulting SPADE tree is shown in Figure 2 (a) in the main text. In Figure S2, we re-drew the SPADE trees colored by each of the 8 markers, and overlaid our annotations of the SPADE tree.

The annotations were derived manually, based on the colored trees. For example, when colored by c-kit, the upper branch in the middle of the SPADE tree showed clear pattern of c-kit+. Therefore, we drew a circle around that branch of the SPADE tree and annotated it as c-kit+. Similarly, by looking at the SPADE tree colored by CD11b, we drew a circle around the left branch, because this branch is CD11b+. The other annotations were drawn similarly, based on B220, TCRβ, CD4, and CD8. Gating was not used to derive the boundaries.

SPADE trees colored by CD150 and Sca-1 were not shown in the main text, because trees colored by these two markers did not show clear patterns. However, subtle features of these colored trees were consistent with traditional gating analysis. For example, in the comparison shown in Figure 2 (b) in the main text, we can observe that one tree node in the c-kit+ branch is enriched with cells in the hematopoietic stem and progenitor cells (HSPC) gate, characterized by c-kit+ and Sca-1+, in the CD11b- B220- TCRβ- gate. From the bottom right plot of Figure S2, we can see that the same node does show higher Sca-1 intensity compared to other nodes in the c-kit+ branch.

The manual gating analysis in Figure 2 (b) of the main text was actually performed right after the data were generated, prior to the SPADE analysis. When the data were generated, we expected to find the major cell types in the upper left panel of Figure 2 (a) of the main text, and manual gating analysis was performed to confirm that we had captured these cell types. Manual gating did not identify the dendritic cells, because it is a subjective approach that relies on our prior knowledge, and we did not plan to find dendritic cells. If we had looked for the dendritic cells using a 2D plot defined by B220 and CD4, we would have observed a subpopulation that were B220+ and CD4+, as shown in Figure S3. In contrast, the SPADE analysis was able to identify this cell type, even though we had not expected to see it.

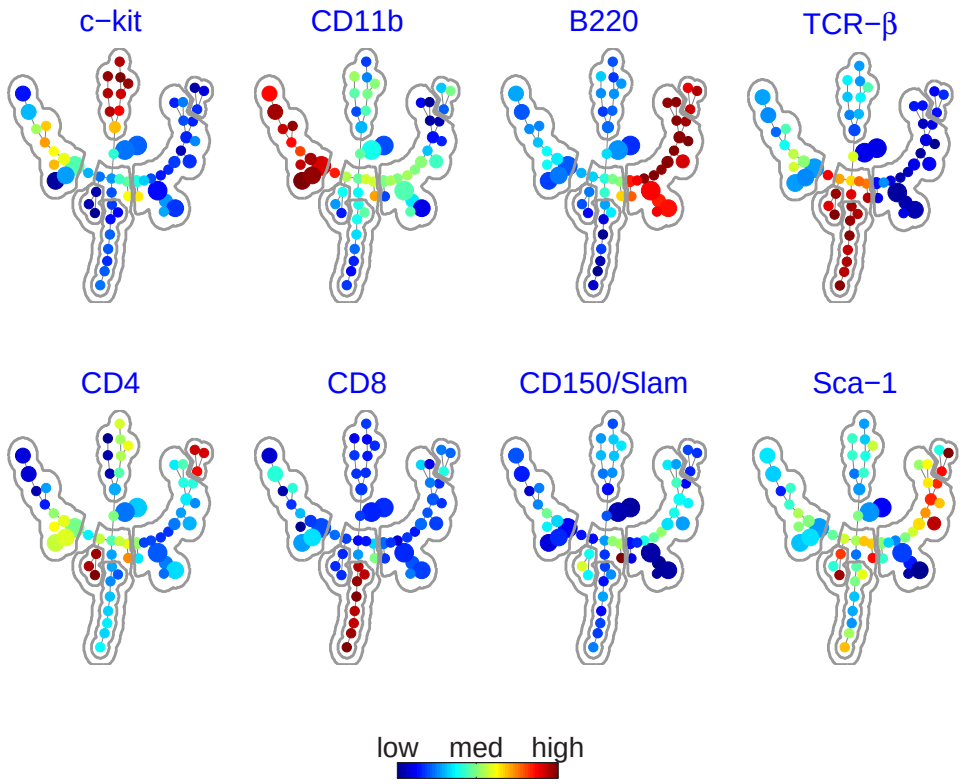


Figure S2: SPADE tree derived from mouse bone marrow flow data. Each tree is colored by one marker.

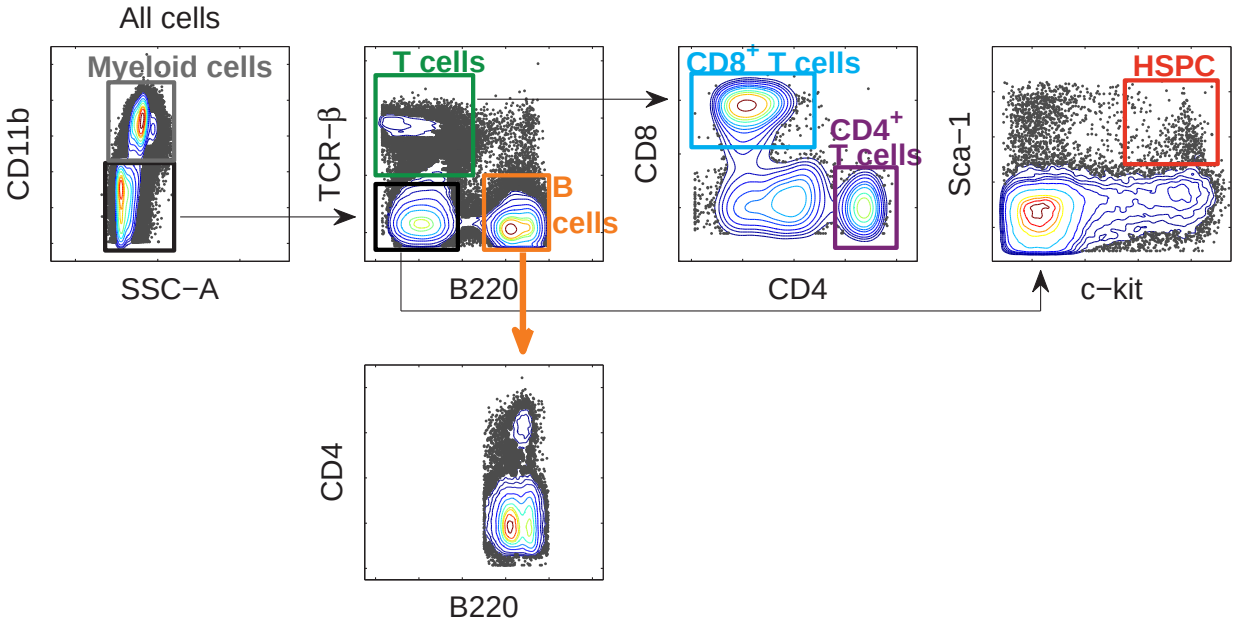


Figure S3: The top row is gating analysis of the mouse bone marrow data, which was performed before SPADE analysis. To investigate the B220+ CD4+ subpopulation identified by SPADE, we drew the bottom plot, and noticed the CD4+ subpopulation among the B220+ cells.

The comparison in Figure 2 (b) of the main text illustrated general agreement between gating and SPADE. If we focus on the middle region of the SPADE tree, we can observe that one node in the T-cell region contained many cells in the myeloid gate (bottom-left panel of Figure 2(b) in main text). To investigate this inconsistency between the two methods, we displayed cells in the myeloid gate (CD11b+) using another biaxial plot defined by B220 and TCRβ (bottom-left plot in Figure S4). We observed that the majority of cells in the myeloid gate are B220- and TCRβ-, as expected. However, there were two unexpected subpopulations:

(1) One subpopulation was B220+ and TCRβ+, which contributed 75% of one node in the middle region of the SPADE tree. When SPADE constructed the tree, the minimum spanning tree algorithm connected this node to its nearest neighbors, which was near the boundary between B220+ cells and TCRβ+ cells. In contrast, gating worked in a hierarchical fashion. Since CD11b was the first marker considered in the gating process, the myeloid gate was primarily defined based on CD11b+, which included the CD11b+ B220+ TCRβ+ cells.

(2) The other subpopulation was B220+ TCRβ-. Cells in this population were clustered into several nodes in the B220+ branch, because these cells show high B220 expression. From the scale of the color bar in the bottom-right plot of Figure S4, we can see that these cells only contributed to a small percentage (<6%) of a few nodes in the B220+ branch. Therefore, when the SPADE tree was colored by median marker intensities or cell frequency, this subpopulation did not produce any visible distinguishing pattern.

These 'improbable' subpopulations were likely the result of doublet/triplet cell events that were not removed by the FSC/SSC gate, or myeloid cells with high levels of the Fc receptor which bind with the reporter antibodies against TCR and B220. Either way, the events were "real" in the data and highlight how SPADE could potentially identify new cell populations easily missed by manual gating.

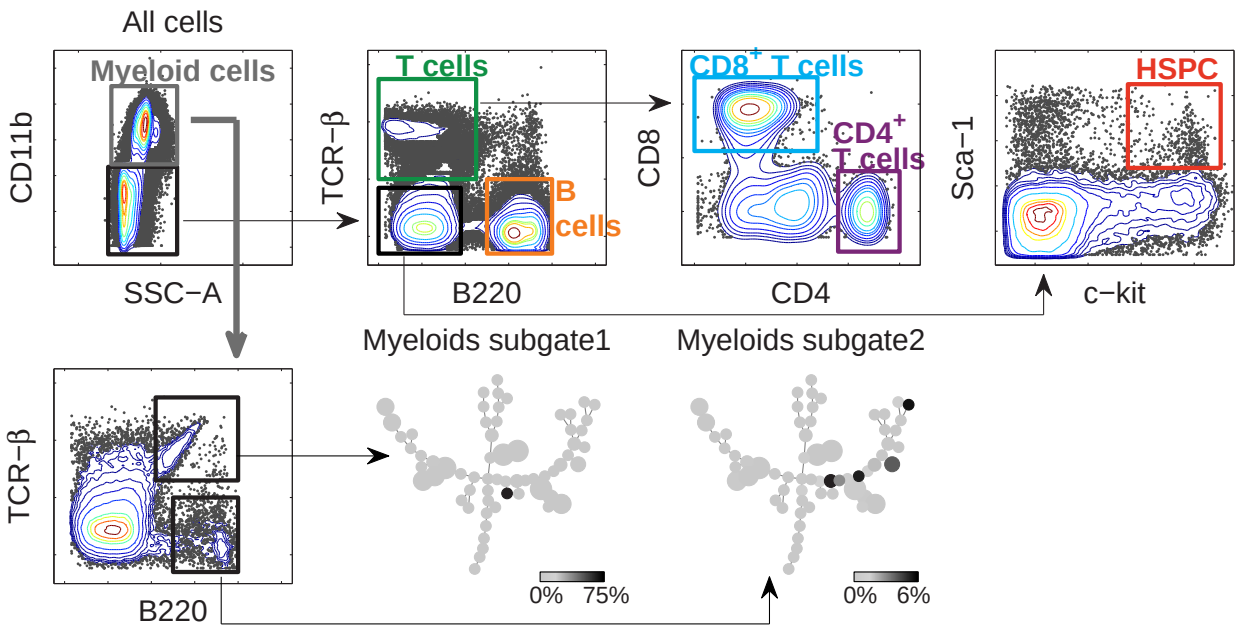


Figure S4: Analysis of the inconsistency between SPADE and gating. Description of this figure can be found in the text above.

S2. Sensitivity and robustness of SPADE with respect to markers used in the analysis

In the main text and section S1 above, SPADE was applied to an 8-marker dataset of mouse bone marrow, and recovered a meaningful cellular hierarchy that is consistent with the known differentiation hierarchy in mouse hematopoiesis. A geometric interpretation of SPADE is that: due to the complex correlations among the 8 protein markers (plus FSC and SSC), in the high-dimensional space defined by these markers, the shape of the cell cloud resembles the underlying cellular hierarchy, and SPADE is able to recover the shape of the cell cloud.

Intuitively, we would expect that if we choose a different set of markers for the SPADE analysis, we are effectively examining the cell cloud in a different high-dimensional space, where the shape of the cloud and the SPADE tree may be different. Therefore, it is of interest to examine how the choice of used markers impacts the SPADE result.

In this section, we demonstrate the sensitivity and robustness of SPADE with respect to the choice of used markers. We used the mouse bone marrow data. First, we performed the SPADE algorithm on just one of the 8 markers and interpreted the resulting tree. Then, we added a second marker and repeated the analysis to examine how the SPADE result changed. We incrementally added more markers, until the all measured markers were used. The purpose of this analysis is to show that when the additional marker added enough new "information" to change the shape of the cloud defined by the markers already included, the SPADE result changed. On the other hand, if the additional marker was highly correlated to the ones that were already included, adding this additional marker did not change the SPADE tree.

SPADE applied to only one marker: CD11b

First, we used only one marker (CD11b) for the SPADE analysis. The other input parameters of the algorithm were the same as the mouse bone marrow analysis in the main text (the outlier and target densities were 1st and 5th percentiles of local densities; the number of clusters was 50).

The resulting SPADE tree is shown in Figure S5. When only one marker was used, each cell was one point in a 1-dimensional space, and thus the shape of the point cloud was a 1-dimensional line. From Figure S5, we can see that the topology of the SPADE tree was a chain without branchings, which correctly reflected the shape of the 1-dimensional cloud.

Each tree was colored by one marker. Take the tree colored by CD11b for example. If the color

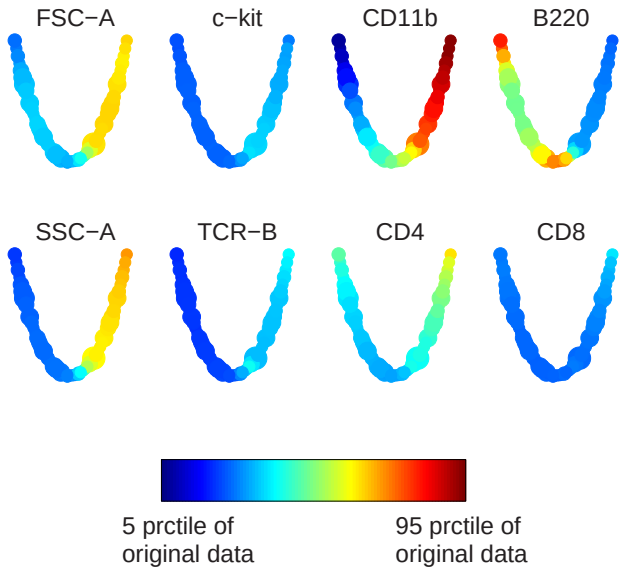


Figure S5: SPADE analysis of the mouse BM data using only one marker, **CD11b**. Each tree was colored by one marker, scaled according to the 5th and 95th percentiles of intensities of this marker on individual cells.

of a node is dark red, the median CD11b intensity of cells in this node is equal to or greater than the 95th percentile of CD11b intensities of individual cells. If the color of a node is dark blue, the median CD11b intensity of this node is equal to or smaller than the 5th percentile of CD11b intensities of individual cells. In this tree colored by CD11b, the color ranged from dark blue to dark red, indicating that SPADE successfully clustered cells into CD11b+ nodes (red) and CD11b- nodes (blue). The red and blue nodes corresponded to the two gates in the first gating plot in Figure S3. The green/yellow nodes in the center of the tree corresponded to the boundary and space between the two gates.

When the SPADE tree was colored by markers that were not used to construct the tree, few nodes were red, indicating that SPADE did not accurately identify the subpopulations that were highly positive for unused markers. Nevertheless, the FSC, SSC and B220 trees showed color patterns similar to the CD11b tree, because these markers are correlated with CD11b (correlation coefficient $cc = 0.60, 0.73, -0.46$). Although the correlation between CD4 and CD11b was weak ($cc = 0.26$), the CD4 tree showed an interesting color pattern, because CD4 and CD11b were nonlinearly correlated.

In summary, when applied to only one marker CD11b, SPADE accurately identified the myeloid cells (red nodes). Although correlated color patterns of a few other markers (FSC, SSC, B220 and CD4) were observed, SPADE did not accurately cluster the subpopulations positive for these markers, because CD11b did not contain enough information to distinguish these cell types.

SPADE applied to two markers: CD11b and TCRβ

We then added another marker to the analysis, and built a SPADE tree using a total of two markers (CD11b and TCRβ). When two markers were considered, the cell cloud was a 2-dimensional cloud, which might contain branches. Since TCRβ was able to separate the CD11b- population into at least two subtypes (CD11b-TCRβ+ and CD11b-TCRβ-), adding TCRβ into the analysis brought new information to change the shape of the cloud. We should expect to see at least three major branches (CD11b+, CD11b-TCRβ+ and CD11b-TCRβ+). In Figure S6, the topology of the resulting tree was indeed different from the previous case.

We observed dark red nodes in the trees colored by CD11b and TCRβ. This observation indicated that SPADE identified the subpopulations that were positive for CD11b and TCRβ respectively, which was expected because these two markers were used. Although the TCRβ+ nodes formed two branches (bottom of the tree), according to the CD4 and CD8 trees, these two branches did not separate the two T cell subtypes. We did not observe signs for the dendritic cells and ckit+ cells.

Interestingly, the B220 tree also contained dark red nodes, meaning that the majority of cells in these nodes were B220+. Thus, SPADE accurately identified at least a subset of B cells without using the B-cell specific marker B220. This result was contributed by the correlation between B220 and TCRβ, their mutually exclusive expression in the lymphoid lineage. We hypothesize that, further including B220 in the analysis would not significantly alter the SPADE tree.

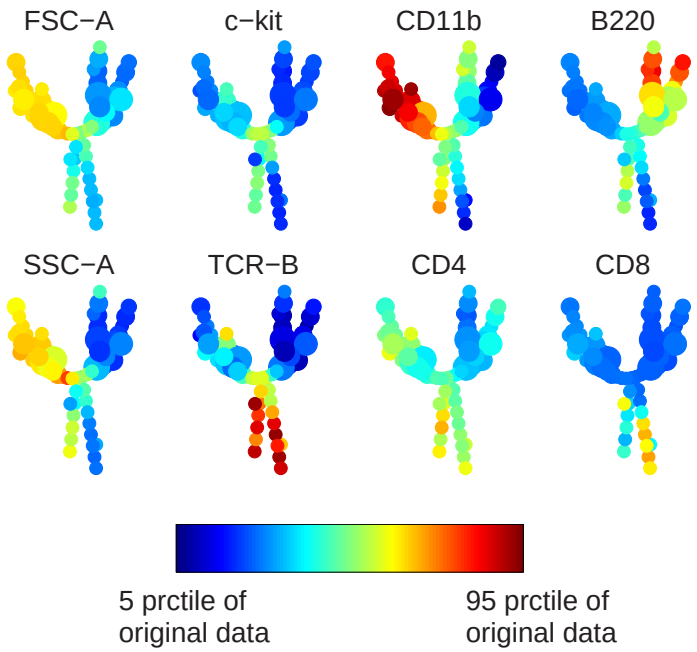


Figure S6: SPADE analysis of the mouse BM data using 2 markers, **CD11b and TCRβ**.

SPADE applied to three markers: CD11b, TCRβ and B220

We further added a third marker B220 into the analysis. The SPADE result is shown in Figure S7. As expected from our previous 2-marker analysis with C11b and TCRβ, the topology of the tree was not significantly changed when B220 was added.

In the B220 tree in Figure S7, we observed more dark red nodes than those in the previous page. After including B220, SPADE more accurately clustered the B220+ cells.

Although inclusion of B220 did not significantly change the overall topology, this marker did alter the two TCRβ+ branches (bottom of the tree). From the CD4 and CD8 colored trees in Figure S7, we observed that the two TCRβ+ branches showed differential expression of CD4 and CD8. One of them was more enriched by CD4+ T cells, and the other was more enriched by CD8+ T cells. SPADE defined these two branches because of the subtle differences in CD11b and B220. It appeared that such subtle differences was able to separate the two T cell subtypes to some extent. This result resonates with the NK example in the human bone marrow analysis discussed in the main text and section S6 in the supplement.

In this analysis, we still did not observe signs for the dendritic cells and ckit+ cells. Identification of these two cell types required more markers.

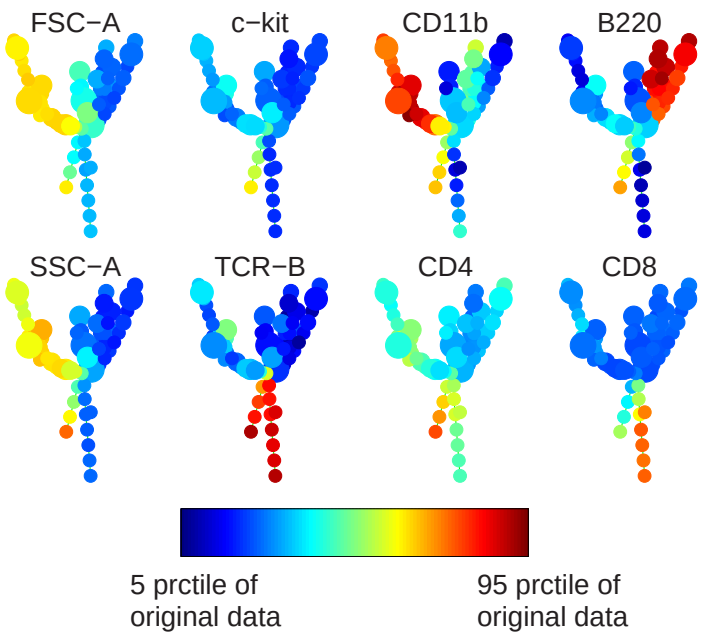


Figure S7: SPADE analysis of the mouse BM data using 3 markers, **CD11b, TCRβ and B220**.

SPADE applied to six markers: CD11b, TCRβ, B220, CD4, CD8 and ckit

We added three additional markers to the analysis, CD4, CD8 and ckit. The SPADE tree is shown in Figure S8. Since CD4 and CD8 were used, SPADE identified the two T cell subtypes, which corresponded to the two bottom branches, respectively (see the CD4 tree and CD8 tree). In addition, CD4 helped SPADE to identify the B220+CD4+ dendritic cells, located in the upper right corner of the tree. Inclusion of ckit resulted in a new branch near the center of the tree, which was positive for ckit+.

Using these six markers, SPADE identified a topology similar to the one reported in Figure 2 of the main text (which was based on all markers). This means including the remaining markers (FSC, SSC, Sca-1 and CD150) will not change the SPADE result, because (1) FSC and SSC were correlated with the six markers used here; (2) Sca-1 and CD150 did not contain strong signal in this dataset.

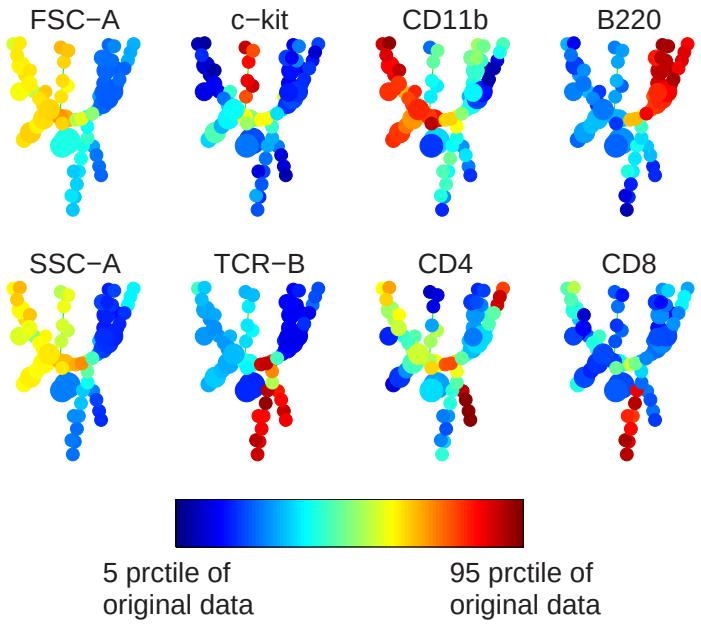


Figure S8: SPADE analysis of the mouse BM data using 6 markers, **CD11b, TCRβ, B220, CD4, CD8 and ckit**.

S3. Robustness of SPADE with respect to noise

Since the downsampling and clustering steps of SPADE both are both inherently stochastic, the robustness of the algorithm needs to be evaluated. In this section, we evaluate the robustness of SPADE with respect to noisy data.

The SPADE result contains three pieces of information, (i) the clustering of cells, (ii) mapping between cell clusters and cell types, and (iii) the topology and interconnectivity among cell types. A rigorous robustness analysis of SPADE that simultaneously accounts for all of these three pieces of information has not been established. While many distance metrics are available for comparing two sets of clustering results, optimal methods to compare two topologies are still unresolved. Therefore, in our current analysis, we pursued an ad hoc approach to demonstrate the robustness of SPADE.

We started with the mouse bone marrow flow cytometry data after inverse hyperbolic sine (arcsinh) transformation (the same data that was analyzed by SPADE), added Gaussian noise to the data, applied SPADE to analyze the new data which is more noisy than the original data, and qualitatively evaluated the SPADE results. We preformed this analysis at three noise levels 5%, 10% and 20%. The noise level was defined as the ratio between the standard deviation of added noise and the standard deviation of the original data. The standard deviation of noise was calculated on a per-channel basis. The SPADE algorithm parameters were set to be the same as the mouse bone marrow analysis in the main text. The SPADE results are shown in Figures S9, S10 and S11.

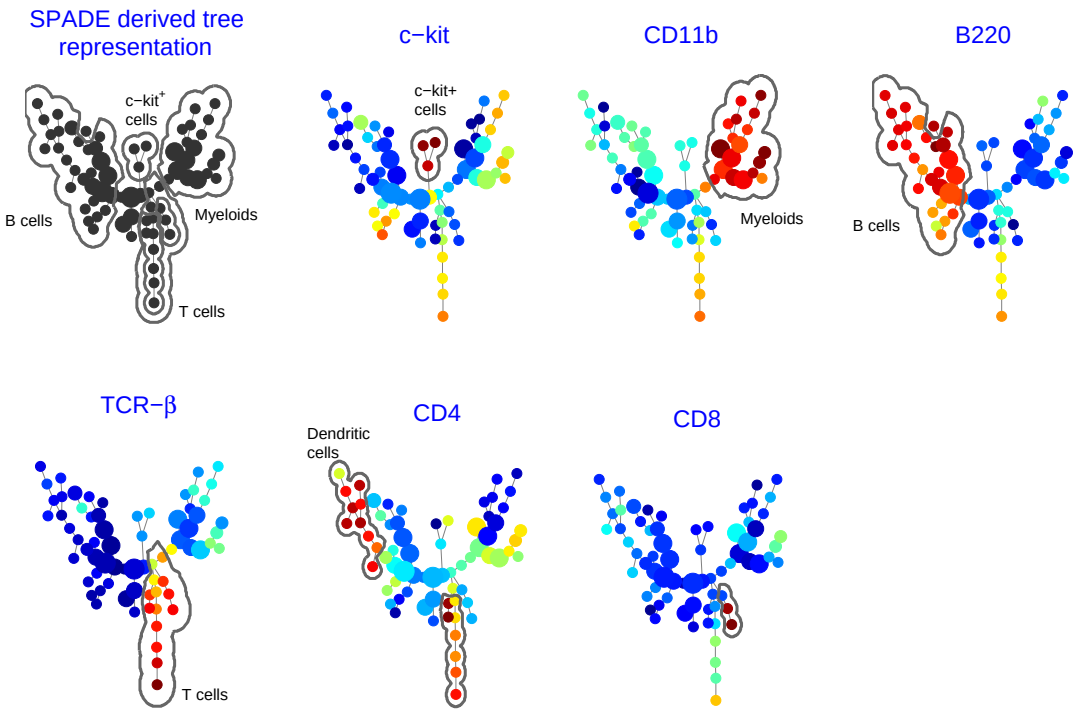


Figure S9: SPADE robustness analysis at noise level 5%.

We applied SPADE to a simulated mouse bone marrow dataset with a noise level 5%. SPADE derived the tree shown in Figure S9, which was very similar to the result derived from the original dataset (Figure 2 in the main text) in the following ways: (1) in the colored trees, cell clusters (nodes) of the same phenotype (color) were close to each other; (2) CD4+ T cells and CD8+ T cells were co-localized; (3) B220+ CD4+ dendritic cells were clearly observed; (4) the overall topology that connected different cell types was almost identical to the original result in Figure 2 of the main text. On the other hand, we also observed the following differences: (1) the number of c-kit+ clusters was different; (2) the number of clusters corresponding to the dendritic cells was different; (3) a few clusters are negative in every marker, and they were connected to a slightly different place on the tree. These differences were due to the added noise and the stochasticity in clustering and downsampling.

Despite the differences observed in this analysis, we considered the SPADE result robust. SPADE was intended to derive a tree structure that represented the cellular heterogeneity underlying the data. This tree facilitated visualization and interpretation of the data. Although there were a few differences between the simulated results in Figure S9 and the original result in Figure 2 of the main text, as long as the overall topologies were similar, the interpretations of the results remained the same.

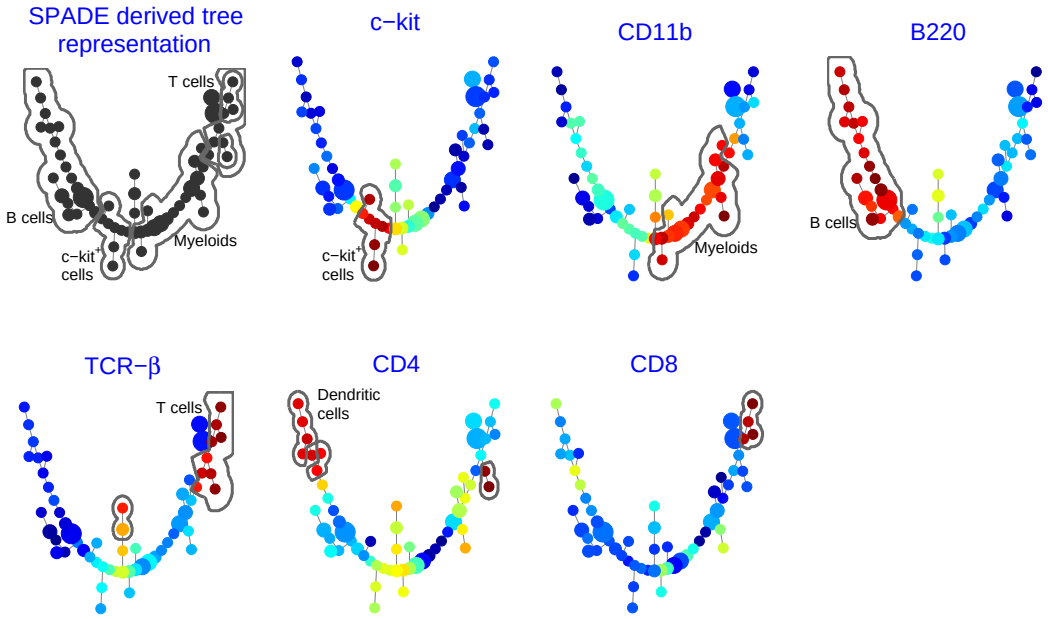


Figure S10: SPADE robustness analysis at noise level 10%.

When applied to a simulated mouse bone marrow dataset with noise level 10%, the SPADE result showed more differences than the previous case. As shown in Figure S10, cell clusters that shared the same phenotype were still close to each other. CD4+ T cells and CD8+ T cells were co-localized. Dendritic cells were clearly defined. However, the overall topology showed some differences:

- Instead of forming its own branch, the myeloid cells were located in-between c-kit+ cells and T cells.
- Based on the tree colored by TCRβ, the two cell clusters highlighted in the middle appeared to be T cells (because they are TCRβ+), but they were far away from the majority of the T cells.

These differences were due to the added noise, which altered the shape of the point cloud of the cells, and blurred the boundaries between branches/arms of the cloud. Since SPADE was designed to extract the shape of the point cloud, the result at this noise level was different from the original result in Figure 2 of the main text. We hypothesize that at a higher noise level, the topology of the SPADE result will look even more different.

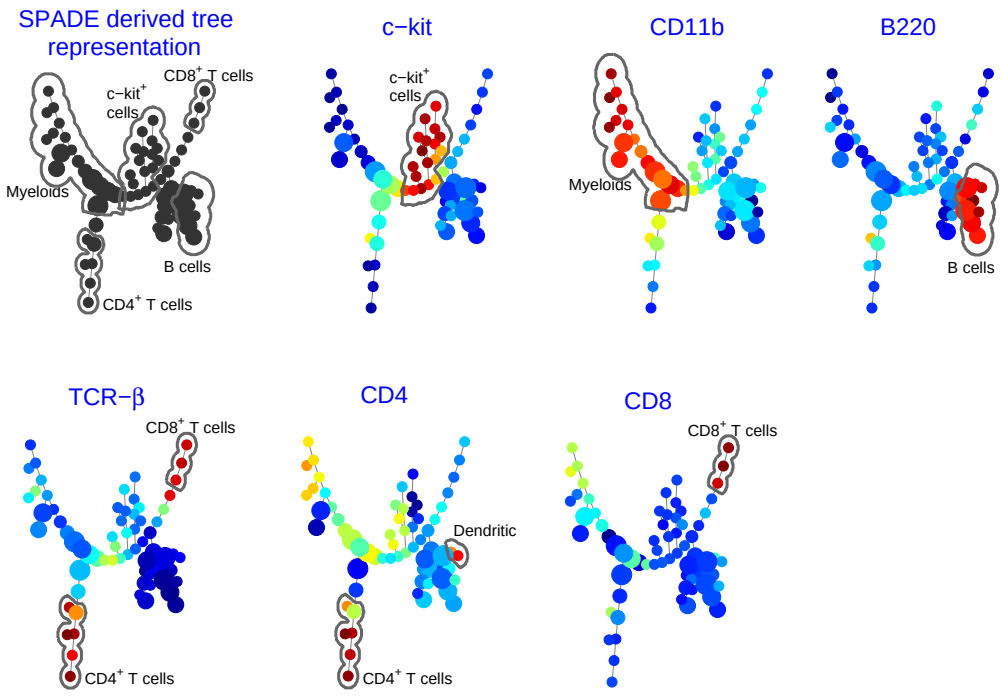


Figure S11: SPADE robustness analysis at noise level 20%.

When SPADE was applied to a simulated dataset with 20% noise, the result showed more differences than the previous analyses at 5% and 10% noise level, as expected. The two subtype of T cells (CD4+ T cells and CD8+ T cells) were no longer located close to each other.

Note that the 20% noise level was added on top of a real flow cytometry dataset which already contained experiment noise. Despite the presence of large noise, the SPADE result still indicated major cell types in the data, i.e. c-kit+ cells, myeloids, B cells and T cells.

In summary, if we consider the data as a point cloud and regard each cell type as a compartment/region/part of this cloud, adding noise to the data is effectively expanding the volume of all compartments. As the noise level increased, the volume of all compartments increased, and the boundaries among different compartments in the high-dimensional space became less clear. When the noise was large enough to change the shape of the point cloud, the topology of the SPADE tree changed.

S4. Comparison of the clustering component of SPADE vs. hierarchical clustering

For the clustering component of SPADE, we proposed a variation of agglomerative clustering algorithm. In this section, we compared SPADE's clustering component with the widely-used hierarchical clustering algorithm.

To compare these two clustering algorithms, we created synthetic datasets containing measurements of 4 markers and 10,000 cells. In the synthetic data, cells were sampled from 5 clusters that were partially overlapped. Figure S12 showed a typical example dataset. Markers 1, 2 and 3 contained signals that defined the 5 clusters. Marker 4 was zero-mean-unit-variance Gaussian noise.

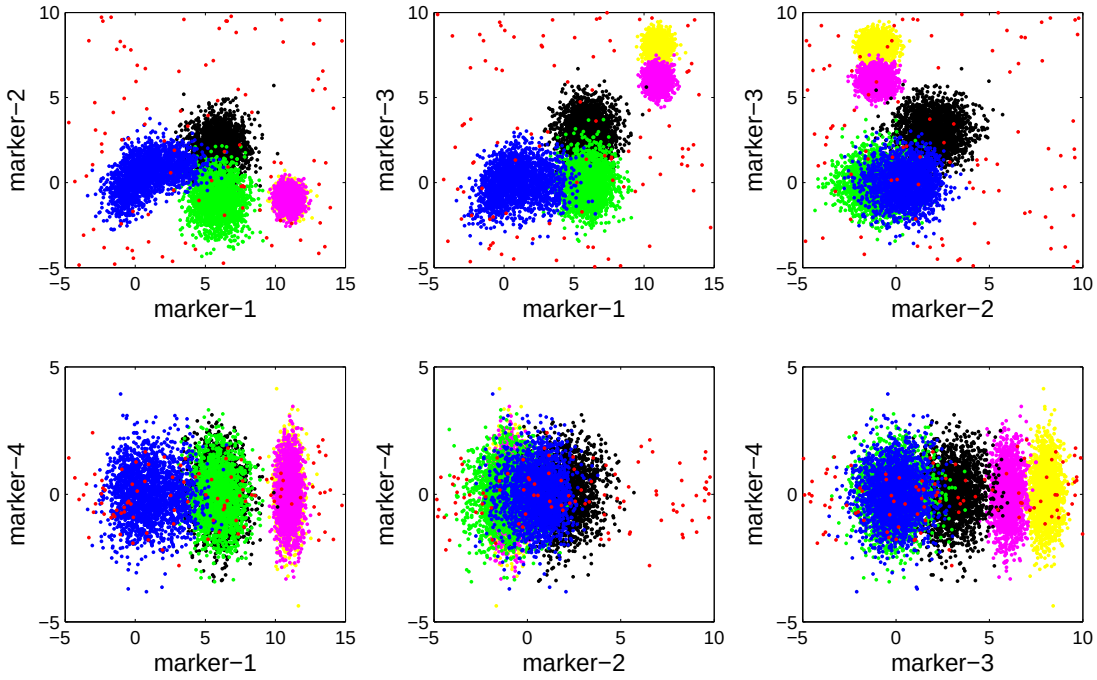


Figure S12: An example of synthetic datasets considered in this section S4.

To evaluate clustering performance, the error metric we used was defined as:

$$error(C,T) = \frac{\sum_{I_C(i,i')=1} I_T(i,i')}{\sum_{I_C(i,i')=1} 1} \tag{1}$$

where C corresponded to the partition of cells obtained by a clustering algorithm, and $I_C(i,i')$ was an indicator function of whether cells i and i' were in the same cluster according to C. Similarly, $I_T(i,i')$ was an indicator function of whether cells i and i' belonged to the same cluster according to the true clusters. This error metric reflected the purity of the resulting clusters. More precisely, among the i

and i' pairs that were assigned to the same cluster by a clustering algorithm, how many percent did not belong to the same true cluster.

We generated 100 datasets, whose compositions of the 5 clusters were unbalanced. The probability distribution of the 5 clusters was: 60%, 25%, 10%, 2.5% and 1.5% (leaving 1% for noise cells). Each simulated dataset contained measurements of 4 markers on 10,000 cells. For each dataset, we applied SPADE's clustering component and hierarchical clustering, and computed error rates for different number of resulting clusters. The mean and standard deviation of clustering errors were shown in Figure S13 (a). We observed an expected trend that as the number of clusters increased, the clustering error reduced. In this simulation setting, hierarchical clustering outperformed SPADE's clustering component, in terms of both mean and standard deviation of the clustering error.

We then generated another 100 datasets, with balanced cluster sizes. The probability distribution of the 5 clusters was: 20%, 20%, 20%, 20% and 19% (leaving 1% for noise cells). Again, for each simulated dataset, we applied the two algorithms and computed error rates for different number of clusters. When the sizes of the true clusters were balanced, SPADE's clustering component showed better performance, as shown in Figure S13 (b). This was because the SPADE's clustering component tried to reduce the number of clusters by half in each of its iterations, and thus encouraged the resulting agglomerative hierarchical tree to be relatively balanced.

Since the density-dependent downsampling removes the density/distribution variation in the data, SPADE's clustering component is more suitable for clustering the downsampled data. This was the motivation why we developed this variation of agglomerative clustering for SPADE.

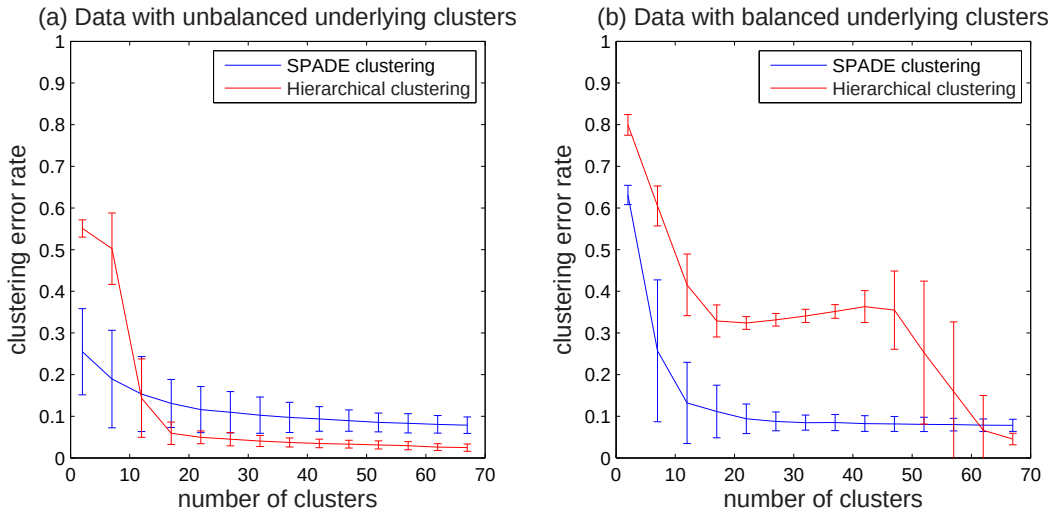


Figure S13: Clustering error comparison of SPADE clustering and hierarchical clustering under two simulation settings.

S5. SPADE tree on human bone marrow data, colored by 13 core surface markers.

SPADE was applied to a mass cytometry dataset of human bone marrow, obtained from Bendall et al., Science, 2011. Single-cell data from 30 individual stimulatory conditions were obtained. In the first tube, an unstimulated aliquot of the bone marrow sample was measured with an immunophenotyping panel of 31 cell surface antibodies. The remaining 29 tubes, comprising of 5 unstimulated samples and 24 samples under different perturbations, were measured by a functional staining panel of 13 core surface markers (CD3, 4, 8, 11b, 19, 20, 33, 34, 38, 45, 45RA, 90, 123, from the previous panel) and 18 intracellular targets that reflect intracellular signaling states.

The data provided by Bendall et al. was gated for singlets based on cell length, DNA content and viability. Therefore, the input data of SPADE were already gated to exclude doublets and debris. The input parameters were:

- Markers for constructing the SPADE tree: 13 core surface markers. Since we wanted to combine multiple dataset with overlapping staining panels, we needed to choose the overlapping markers.
- Outlier density: 1st percentile of the local densities of all cells.
- Target density: we chose the target density such that 20,000 cells would survive the downsampling step. Since we are pooling multiple datasets, we varied the target density for each of the 30 data files such that each experiment contributed equal number of cells.
- Desired number of clusters was set to be 300. Since the underlying cellular heterogeneity is more complex than that in the mouse bone marrow, the desired number of clusters is larger than that in the mouse bone marrow analysis.

Using the above parameters, SPADE derived the tree structure shown in Figure 3 in the main manuscript. The tree structure is also shown in Figures S14 and S15, where nodes are colored by each of the 13 core surface markers. From these figures, we can observe how surface markers behave across the tree. Boundaries of the annotations were manually derived based on the colored trees.

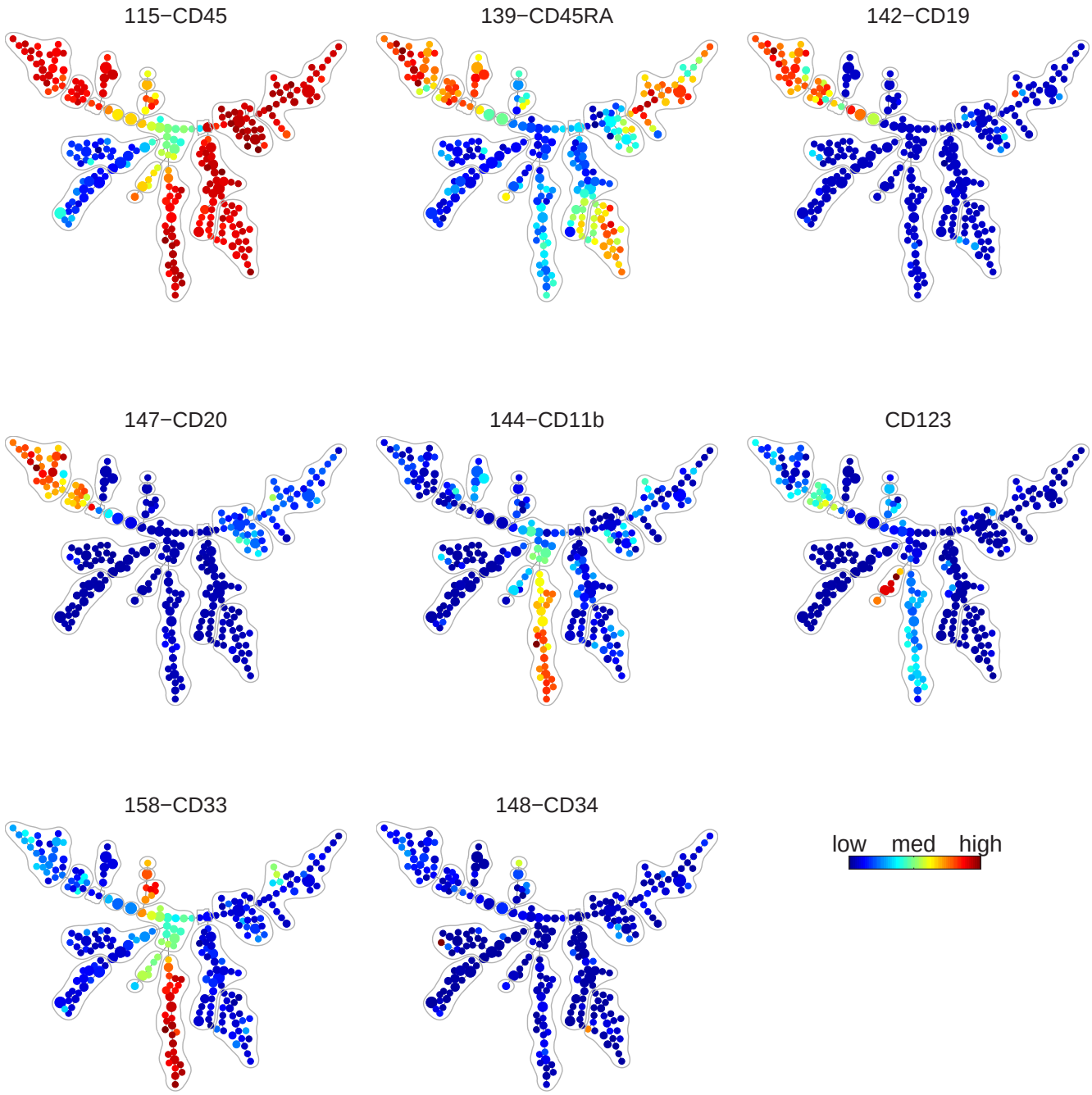


Figure S14: SPADE tree derived from human bone marrow data, color-coded by CD45, CD45RA, CD19, CD20, CD11b, CD123, CD33 and CD34.

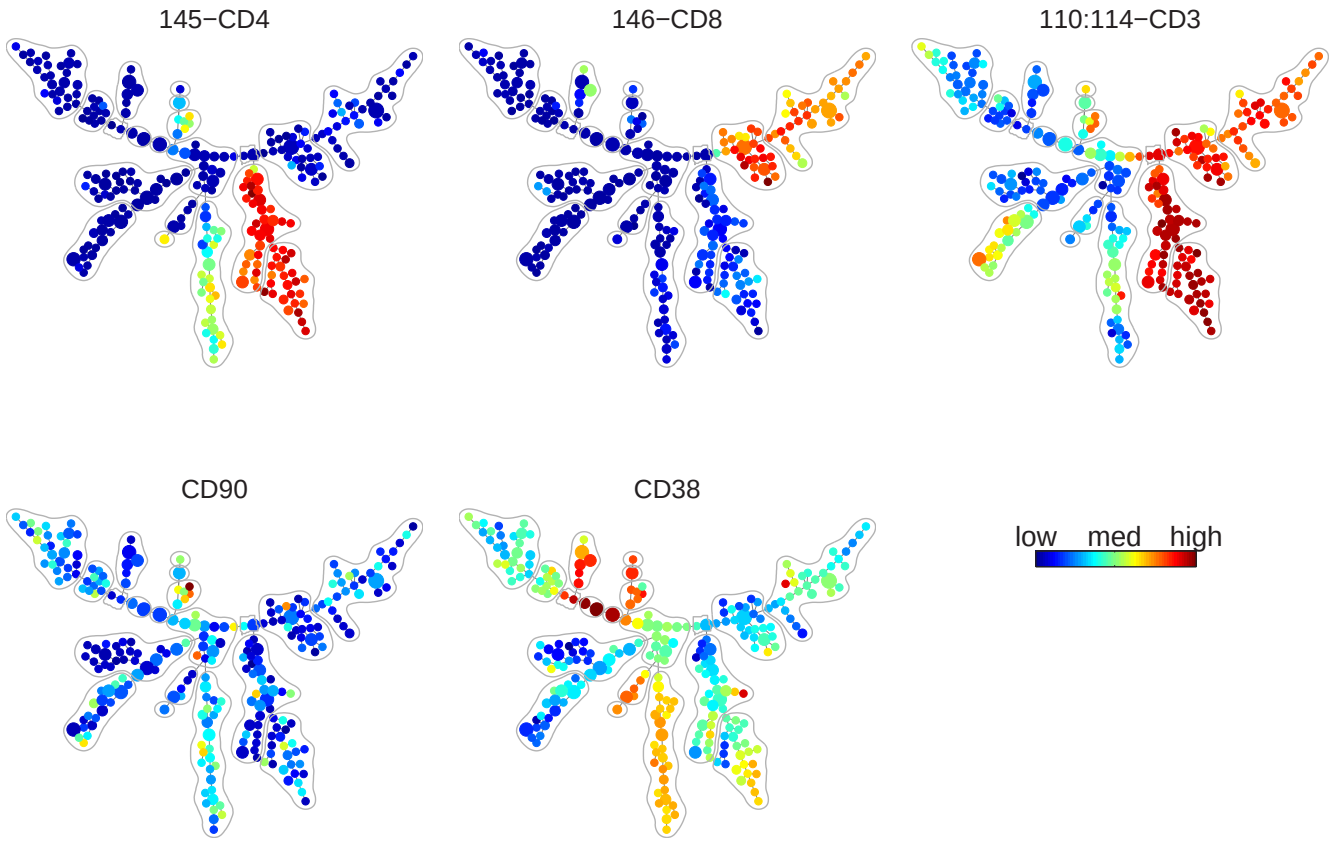
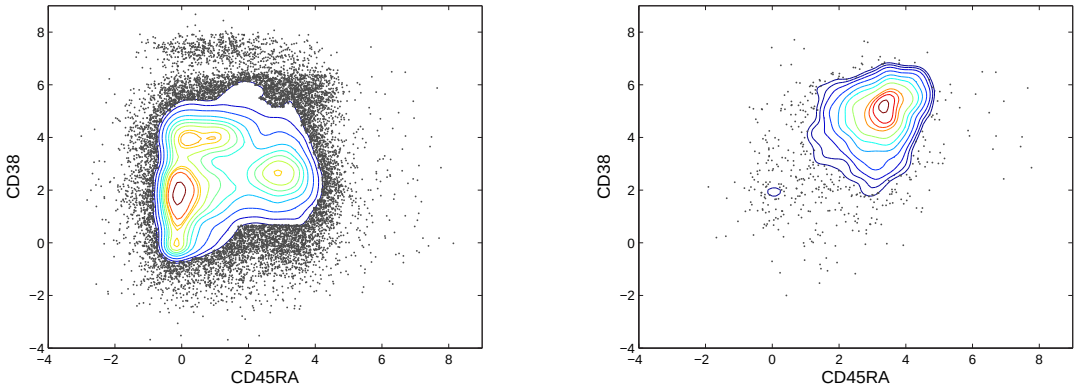


Figure S15: SPADE tree derived from human bone marrow data, color-coded by CD4, CD8, CD3, CD90 and CD38.

S6. Scatter plot of NK cells identified from the human bone marrow mass cytometry dataset

The human bone marrow dataset contains 30 tubes with two different but overlapping staining panels. The overlap consists of 13 surface markers (CD3, 4, 8, 11b, 19, 20, 33, 34, 38, 45, 45RA, 90, 123). The tree structure in Figure 3 of the main text was derived based on the above 13 surface markers. Although none of these markers are NK-specific, SPADE clustered NK cells into one branch of the SPADE tree. This is because NK cells are CD45+ CD45RA+ CD38+ CD19-, as shown in the two figures below. The profile jointly defined by these four markers distinguishes NK cells from all other cells, which is the reason why SPADE was able to capture the NK cells.



(a) (b)

Figure S16: Contour plots of (a) all pooled downsampled cells, (b) SPADE-identified NK cells.

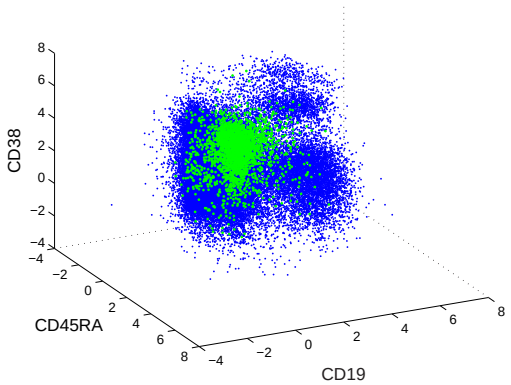


Figure S17: Scatter plot of all pooled downsampled cells, with SPADE-identified NK cells in green.

S7. Validation of the human bone marrow SPADE tree based on functional markers not used to build the tree

In the SPADE analysis of the human bone marrow CyTOF data, the 13 core markers were used to build the tree structure and derive the manual annotations, shown in Figure 3 of the main text and Figures S14 and S15. The functional markers were not used in the analysis. In Figure 5 of the main text, when we colored the SPADE tree using changes of functional markers in response to perturbations, we observed color patterns consistent to the annotations, meaning that the activities of the functional markers supported the annotations derived from the surface markers. Such observation can be viewed as a functional validation of the SPADE tree (using markers that were not used to generate the tree).

Figure 5 of the main text only showed four examples of functional markers' response to perturbations. Each tree showed the log-fold-change of one functional marker before and after a perturbation. Since this dataset contained measurements of 18 function markers across 24 different perturbation conditions, we could draw a total of 18*24=432 such colored trees. In this section, we derived a summary of these 432 trees, in terms of how much they supported the SPADE tree derived from the surface markers.

We considered annotations (bubbles) that had more than 3 nodes (13 bubbles satisfied this criterion). For each of the 432 trees, we computed the standard deviation of log-fold-change among nodes within each bubble. Therefore, we obtained 13*432 (number of bubbles * number of trees) standard deviation values that described the variance of activities within the bubbles. In order to derive a null distribution to evaluate the statistical significance, we randomly switched nodes among the bubbles and re-calculated the within-bubble standard deviation. In Figure S18, the distribution of within-bubble log-fold-change standard deviation was shown in blue, and the red curve represented the distribution based on randomly reshuffled data. From this figure, we observed that the within-bubble variances of the functional markers' activities were significantly smaller than random (two-sample student t-test $p < 10^{-25}$). Therefore, the SPADE tree and annotations derived from surface markers were supported by the activities of the functional markers.

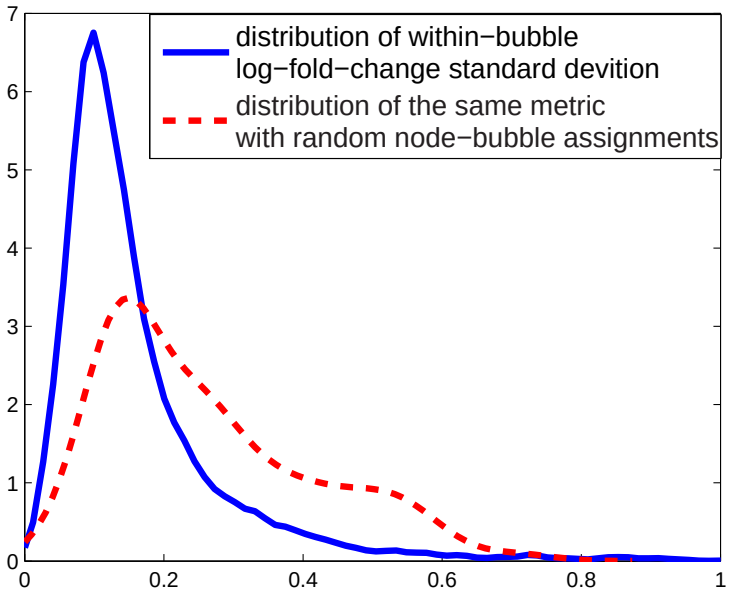


Figure S18: Blue curve is the distribution of within-bubble standard deviation of functional markers' log-fold-change in response to perturbations. Red curve is a distribution of the same metric based on randomly re-shuffled data. The blue distribution is centered around smaller values than the random case (red curve), indicating that the activities of the function markers supported the SPADE tree and annotations (bubbles) derived from surface markers. Two-sample student t-test of these two distributions showed a significant p-value ($p < 10^{-25}$).

S8. Scatter plot of CD123++ that respond to TPO through phosphorylation of STAT5

In the SPADE analysis of the human bone marrow mass cytometry data, Figure 5 (c) of the main text, we showed the induction of phosphorylated STAT5 after stimulation with thrombopoietin (TPO). This activity was expected in HSCs and earlier myeloid progenitors, but not necessarily in the CD123++ population indicated in Figure 3 (b) of the main text.

To investigate the activity of this CD123++ population, we inspected the raw data using gating analysis. We gated on CD3- CD4- CD8- CD19- CD34- CD38+ CD33mid, to exclude cells that are irrelevant to the CD123++ populations as much as possible. The remaining events are shown in the Figure S19, where we can observe two well-defined populations that express CD123, separated by the expression of CD45RA. For each of the two populations, we showed the pSTAT5 activity in Basel condition, IL-3 stimulation and TPO stimulation. The upper left population corresponded to the CD123++ annotation in the SPADE tree, whose pSTAT5 activity responded to both IL-3 and TPO stimulations. The upper right population corresponded to the Plasmacytoid DC annotation in the SPADE tree, whose pSTAT5 activity only responded to IL-3 but not TPO. Although the upper left population (CD123++ annotation) does not match any reported immunological population with the markers at hand, it may be a form of plasmacytoid dendritic cell progenitor, which has been previously described to exhibit enhanced in vitro expansion and maturation into plasmacytoid dendritic cells when TPO is added to the traditional Flt3-containing growth media (Chen et al, Blood, 2004).

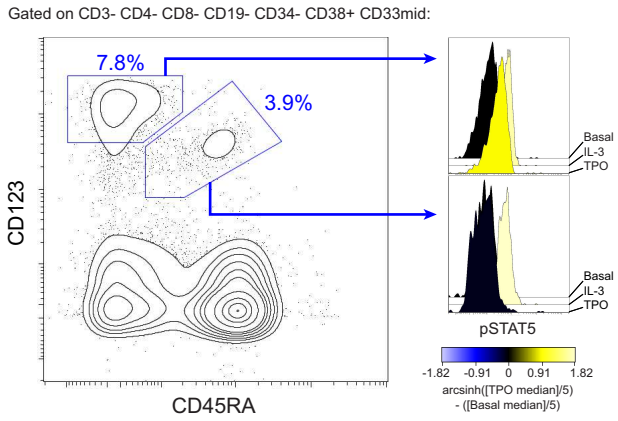


Figure S19: Gating of the CD123 high cells that showed TPO induced phosphorylation of STAT5.

S9. Pseudo-code for SPADE analysis

Gather user input: List of fcs files (i.e. 30 fcs files in the human bone marrow data)
Gather user input: List of markers to use (i.e. 13 core surface markers)
Gather user input: Outlier density, target density, desired number of clusters

% Step 1: density-dependent downsampling
for {each fcs file to be analyzed} **do**
  read data matrix from one fcs file, transform data if needed
  estimate the median minimum distance between a cell and its nearest neighbor, *med_min_dist* (randomly pick 2000 cells, compute the distance of each one to its nearest neighbor, and take the median of these distances)
  define $dist\_threshold = med\_min\_dist * \alpha$ (default $\alpha = 5$)
  compute *local_density* for each cell, which is the number of cells within a neighborhood defined by *dist_threshold*
  downsample the data according to Equation (1) in main text
end for

% Step 2: pool downsampled data together
read all the cells in the downsampled data files created above
pool all these cells into one single dataset, the "pooled downsampled data"

% Step 3: clustering
apply the agglomerative clustering method in the main text to group pooled downsampled cells into clusters

% Step 4: minimum spanning tree (MST) construction
represent each cluster by its center (average intensities of measure markers, averaging across all cells in this cluster)
construct the MST that connects the centers, using the algorithm in the main text

% Step 5: upsampling
for {each original fcs file} **do**
  read data matrix from this original fcs file, transform data if needed
  for {each cell in this original fcs file} **do**
    compute its distance to all the cells in the pooled downsampled data
    find its nearest neighbor
    assign it to the cluster to which the nearest neighbor belongs
  end for
end for

% Step 6: visualization
compute the average intensity for each marker, each cluster, and each file
display the MST and color-code each node using the average intensity of one marker (i.e. Figures 2a and 4 in the main text)
display the MST and color-code each node using the ratio between average intensity of one marker across different files (i.e. Figure 5 in the main text)