SPRING: a kinetic interface for visualizing high dimensional single-cell expression data

Caleb Weinreb [1],*, Samuel Wolock [1] and Allon Klein [1]* [1]Department of Systems Biology, Harvard Medical School, Boston, MA 02115, USA.

Supplemental material:

1. Guide for uploading data using the online webserver

The SPRING software includes preprocessing code for creating a graph structure and a web-based browser for viewing the graph and overlaying other data. Though the preprocessing routines are available online (https://github.com/AllonKleinLab/SPRING/), non-computational users can upload their data and parameter choices to an online server that does the preprocessing automatically and generates a link to the SPRING interface with data pre-loaded (Figure 1)

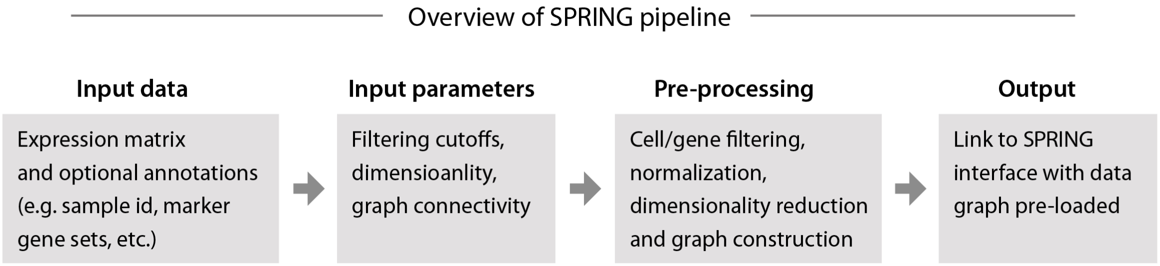
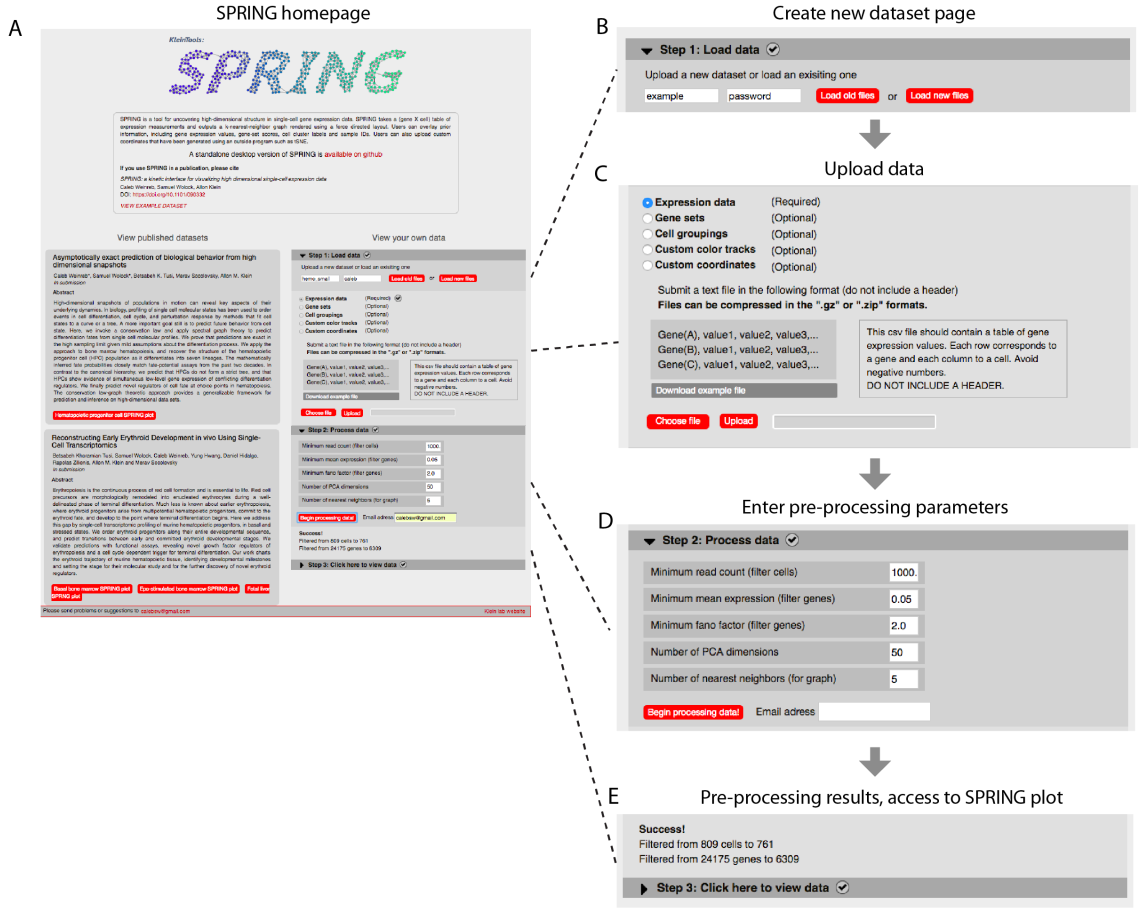

Overview of SPRING pipeline
Input data
Expression matrix and optional annotations (e.g. sample id, marker gene sets, etc.)
Input parameters
Filtering cutoffs, dimensioanlity, graph connectivity
Pre-processing
Cell/gene filtering, normalization, dimensionality reduction and graph construction
Output
Link to SPRING interface with data graph pre-loaded

Figure 1: Overall workflow for the data upload webserver

To use the online server (https://kleintools.hms.harvard.edu/tools/spring.html; Figure 2), follow these steps:

1) Create a dataset name and password (Figure 2B). These can be used to reload a project and change preprocessing parameters without uploading the data every time. Then click "Load new files".
2) A data upload panel will appear (Figure 2C). In addition to a gene expression matrix, which is required, other optional data types can be uploaded, such as cluster/sample labels for each cell, gene sets, etc. To load a given data type, click the corresponding radio-button, click "choose file" and then click "upload". A checkmark will appear when the upload is complete.
3) After an expression matrix has been uploaded, clicking the gray "Process data" bar will expand a parameter entry panel (Figure 2D). Default parameters are pre-loaded. Section (2) of the supplement provides guidance for choosing parameters.
4) After parameter entry, click "Begin processing data" to initiate preprocessing. When this is complete, an email will be sent with a link to the SPRING interface with the dataset pre-loaded. Section (3) gives instructions for using the SPRING interface.


A
SPRING homepage
B
Create new dataset page
C
Upload data
D
Enter pre-processing parameters
E
Pre-processing results, access to SPRING plot

KleinTools: SPRING

SPRING is a tool for uncovering high-dimensional structure in single-cell gene expression data. SPRING takes a (gene X cell) table of expression measurements and outputs a k-nearest-neighbor graph rendered using a force directed layout. Users can overlay prior information, including gene expression values, gene-set scores, cell cluster labels and sample IDs. Users can also upload custom coordinates that have been generated using an outside program such as tSNE.

A standalone desktop version of SPRING is available on github

If you use SPRING in a publication, please cite

SPRING: a kinetic interface for visualizing high dimensional single-cell expression data
Caleb Weinreb, Samuel Wolock, Allon Klein
DOI: https://doi.org/10.1101/090332

VIEW EXAMPLE DATASET

View published datasets

Asymptotically exact prediction of biological behavior from high dimensional snapshots

Caleb Weinreb*, Samuel Wolock*, Betsabeh K. Tusi, Merav Socolovsky, Allon M. Klein
In submission

Abstract

High-dimensional snapshots of populations in motion can reveal key aspects of their underlying dynamics. In biology, profiling of single cell molecular states has been used to order events in cell differentiation, cell cycle, and perturbation response by methods that fit cell states to a curve or a tree. A more important goal still is to predict future behavior from cell state. Here, we invoke a conservation law and apply spectral graph theory to predict differentiation fates from single cell molecular profiles. We prove that predictions are exact in the high sampling limit given mild assumptions about the differentiation process. We apply the approach to bone marrow hematopoiesis, and recover the structure of the hematopoietic progenitor cell (HPC) population as it differentiates into seven lineages. The mathematically inferred fate probabilities closely match fate-potential assays from the past two decades. In contrast to the canonical hierarchy, we predict that HPCs do not form a strict tree, and that HPCs show evidence of simultaneous low-level gene expression of conflicting differentiation regulators. We finally predict novel regulators of cell fate at choice points in hematopoiesis. The conservation law-graph theoretic approach provides a generalizable framework for prediction and inference on high-dimensional data sets.

Reconstructing Early Erythroid Development in vivo Using Single-Cell Transcriptomics

Betsabeh Khoramian Tusi, Samuel Wolock, Caleb Weinreb, Yung Hwang, Daniel Hidalgo, Rapolas Zilionis, Allon M. Klein and Merav Socolovsky
In submission

Abstract

Erythropoiesis is the continuous process of red cell formation and is essential to life. Red cell precursors are morphologically remodeled into enucleated erythrocytes during a well-delineated phase of terminal differentiation. Much less is known about earlier erythropoiesis, where erythroid progenitors arise from multipotential hematopoietic progenitors, commit to the erythroid fate, and develop to the point where terminal differentiation begins. Here we address this gap by single-cell transcriptomic profiling of murine hematopoietic progenitors, in basal and stressed states. We order erythroid progenitors along their entire developmental sequence, and predict transitions between early and committed erythroid developmental stages. We validate predictions with functional assays, revealing novel growth factor regulators of erythropoiesis and a cell cycle dependent trigger for terminal differentiation. Our work charts the erythroid trajectory of murine hematopoietic tissue, identifying developmental milestones and setting the stage for their molecular study and for the further discovery of novel erythroid regulators.

Basal bone marrow SPRING plot
Epo-stimulated bone marrow SPRING plot
Fetal liver SPRNG plot

View your own data

Step 1: Load data ✓

Upload a new dataset or load an exisiting one

hemo_small caleb

Please send problems or suggestions to calebsw@gmail.com

Klein lab website

Step 1: Load data ✓

Upload a new dataset or load an exisiting one

example password

● **Expression data** (Required)
○ **Gene sets** (Optional)
○ **Cell groupings** (Optional)
○ **Custom color tracks** (Optional)
○ **Custom coordinates** (Optional)

Submit a text file in the following format (do not include a header)
Files can be compressed in the ".gz" or ".zip" formats.

Gene(A), value1, value2, value3,...
Gene(B), value1, value2, value3,...
Gene(C), value1, value2, value3,...

This csv file should contain a table of gene expression values. Each row corresponds to a gene and each column to a cell. Avoid negative numbers.
DO NOT INCLUDE A HEADER.

Step 2: Process data ✓

Minimum read count (filter cells)	1000.
Minimum mean expression (filter genes)	0.05
Minimum fano factor (filter genes)	2.0
Number of PCA dimensions	50
Number of nearest neighbors (for graph)	5

Email adress

Success!
Filtered from 809 cells to 761
Filtered from 24175 genes to 6309

Step 3: Click here to view data ✓

Figure 2: Screenshot of the SPRING data upload webserver

2. Guide for choosing parameters

The default parameters on the SPRING webserver are optimized for droplet-based single-cell sequencing approahces (e.g. inDrops, Drop-seq or 10X Genomics), with gene counts based on unique molecular identifiers and an average of several thousand total UMIs detected per cell. Data from other single-cell sequencing platforms and non-sequecing based approaches such as single-cell qPCR can still be visualized in SPRING, but changes in parameters may be required. In general, the appearance of a dataset is relatively robust to the choice of parameters (Figure 3). However, the following guidelines in our experience may be used to maximize the qualiy of the SPRING visualization.

Minimum read count: Used to filter cells. Unless familiar with the correct cutoff by prior inspection of the distribution of total reads, this cutoff can initially be set to zero and raised as needed to filter out putative cell fragments and other non bona-fide cells.

Minimum mean expression: Used to filter low-expressed genes. This cutoff should be set so that several thousand genes pass. The default value of 0.05 is usually good for UMI-based methods, but a larger cutoff may be needed for non-UMI sequencing based approaches.

Minimum Fano factor: Used to filter low variability genes. This cutoff should be set so that in combination with the minimum expression filter, a few thousand (500 – 5000) genes pass.

Number of PCA dimensions: Should be set to roughly 1-2X the total number of expected subpopulations in the sample. The visualization is generally robust to this parameter.

Number of nearest neighbors: Used to determine the graph conectivity. May be set lower than the default of 5 when very few cells are present or when the graph appears too densely inter-connected.

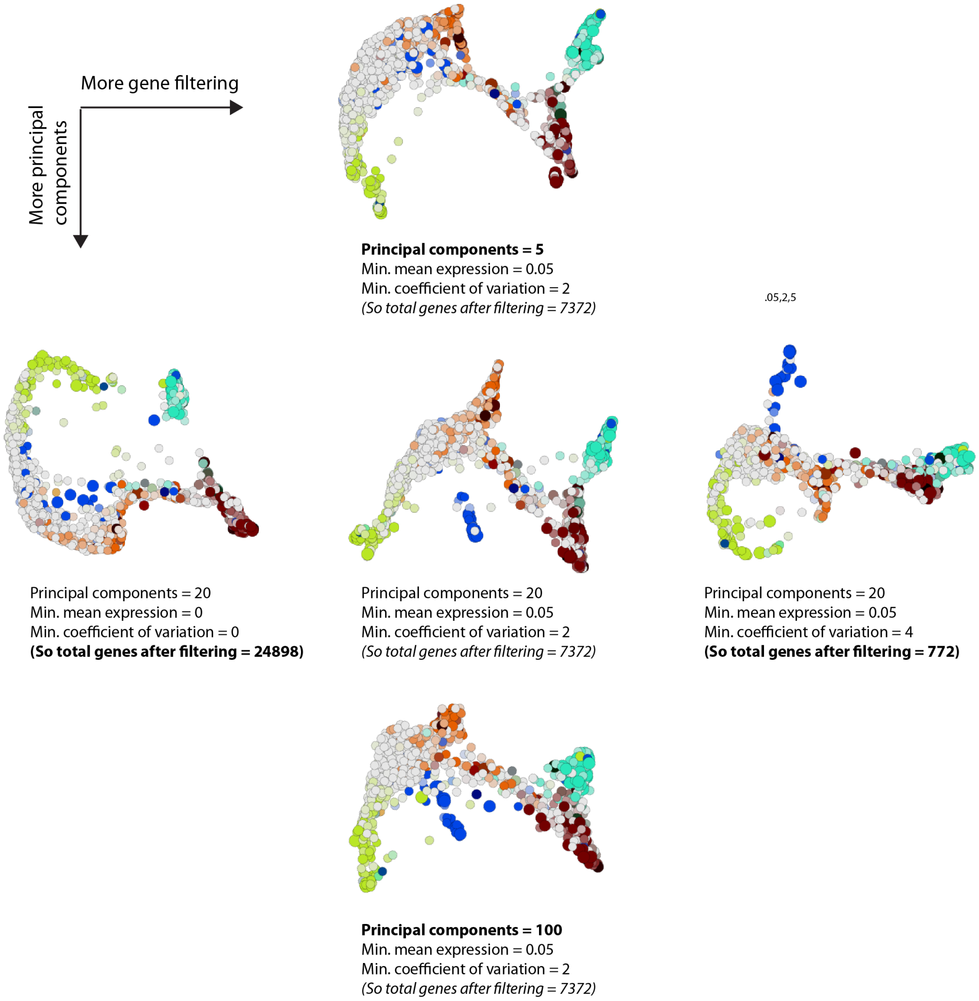
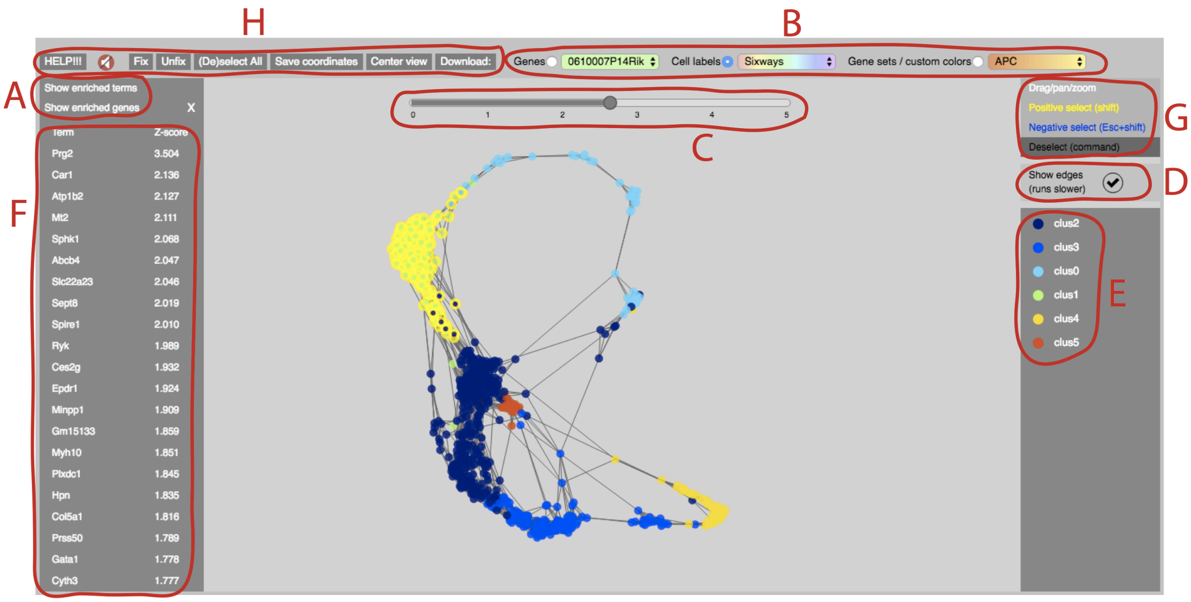


Figure 3: SPRING plots of the data from Figure 1D (top) in the main text with different parameters.

3. How to use the interactive SPRING interface

Quick start

- Click and drag nodes to move the graph.
- Drag the background to pan; mouse-scroll to zoom.
- To select and deselect cells, choose a selection mode (Figure 4G) and drag a box around the cells, or use keyboard shortcuts.
- to find markers enriched in a cell Selection, click “show enriched terms/genes” (Figure 4A).
- Use radio buttons (Figure 4B) to switch between color modes. Within a color mode, choose a specific color track from the drop-down menu.
- Drag the colored range slider (Figure 4C) to adjust color saturation point.
- Turn off “show edges” (Figure 4D) to make SPRING run faster.



Genes ○ 0610007P14Rik
Cell labels ● Sixways
Gene sets / custom colors ○ APC

Term	Z-score
Prg2	3.504
Car1	2.136
Atp1b2	2.127
Mt2	2.111
Sphk1	2.068
Abcb4	2.047
Slc22a23	2.046
Sept8	2.019
Spire1	2.010
Ryk	1.989
Ces2g	1.932
Epdr1	1.924
Minpp1	1.909
Gm15133	1.859
Myh10	1.851
Plxdc1	1.845
Hpn	1.835
Col5a1	1.816
Prss50	1.789
Gata1	1.778
Cyth3	1.777

Figure 4: Screen shot of the SPRING interface

Menu Bar (Figure 4H)

HELP!!!: Links to this page.

Fix/Unfix: Can be used to fix and unfix the positions of nodes. To fix nodes, select them and click "Fix". To set nodes free, select them and click "Unfix". SPRING plots with saved/uploaded coordinates automatically load with all nodes fixed.

(De)select All: If no nodes are selected, this button selects all nodes. If some nodes are already selected, this button deselects all nodes.

Save coordinates: Clicking this button saves the current node coordinates so that they are automatically reloaded when you next open the SPRING plot. To export coordinates for outside use, see "Download:" section.

Center view: Automatically centers and rescales the SPRING plot.

Download:

- **Coordinates:** Exports csv file storing the coordinates of the current view. Each line contains the index of a node (base-0 numbering) and its x- and y- coordinates.
- **Selected cells:** Exports a text file that lists the index of a currently selected cell on each line (0-based numbering).
- **Enriched terms:** Exports file showing terms enriched in the selected population of cells. The terms can be either genes or gene-sets/custom-coloring-tracks. When enriched terms are being viewed in the left sidebar (Figure 4F), then they are

exported. Otherwise, the choice of genes vs. gene-sets/custom-coloring-tracks depends on which marker type is currently coloring the cells. The exported file contains a list of selected cells on the top line (comma-delimited) followed by a blank-line, and then a list of terms and their enrichment Z-score (see "Show enriched terms" section)

- **Edge list:** Exports a list of edges in the graph. Each line contains one edge with the source and target indexes separated by a comma (based-0 numbering).
- **Screenshot (.png)/(.svg):** Exports an image of the current plot. The exported image will have a white background. All on-screen items such as the color-slider and enrichment sidebars are hidden in the exported image. The ".svg" option sometimes fails for very large datasets.

Coloring nodes

Nodes can be colored using three different modes, with the current mode set by radio buttons (Figure 4B)

Genes: Each node is colored by normalized gene expression. Choose genes using the drop-down menu. You can jump straight to a gene in the menu by typing its name.

Gene sets / custom colors: Each line in this drop-down menu corresponds to a user-uploaded custom color track or gene set. When a gene set is selected, the color of a node reflects the sum of Z-scores of genes in the set.

Cell labels: Each line in this drop-down menu corresponds to a different labeling of cells. When you select a labeling, a sidebar (Figure 4E) will appear on the right side of the screen showing the color for each label. Click a label to select the corresponding nodes.

Show enriched terms

To find marker genes, enriched within a population of cells, select the cells and click "Show enriched genes" (Figure 4A). Similarly to find gene sets / custom coloring tracks enriched in a selection of cells, click "Show enriched terms" (also Figure 4A). In both cases, a sidebar (Figure 4F) will appear that lists the terms and their enrichment score. The enrichment score is the sum of Z-scores for the given track across all cells in the selection. Users can scroll through tracks in the sidebar. To color nodes with a track, just click it.

In addition to finding markers enriched in a population, it is also possible to directly compare two populations using negative selection. To negatively select cells, use Shift+Esc+drag or click "Negative select" in the SPRING interface (Figure 4G). If cells are negatively selected, then enriched genes/terms are ranked and scored by the formula:

enrichment_score(positive selection) – enrichment_score(negative selection)

4. Down-sampling method for large datasets

The SPRING interface is designed to handle at most 10,000 cells. Beyond that number, the interface becomes very slow and difficult to use. To cope with large numbers of cells, a coarse-graining strategy can be used, where the original cells are partitioned into groups and averaged together to make ‘pseudo-cells’. The resulting graph structure has fewer nodes than the original dataset, but the pseudo-cells carry more information since averaging over many real cells allows for a less sparse gene expression profile. We provide code for coarse graining on github (https://github.com/AllonKleinLab/SPRING/) and detail the algorithm in pseudo-code below.

We applied the coarse-graining method to a dataset of 8,000 peripheral blood mononuclear cells from 10X Genomics (https://support.10xgenomics.com/single-cell-gene-expression/datasets/2.0.1/pbmc8k). Figure 5 shows that there is a minimal loss of structure in the dataset even after coarse-graining to two orders of magnitude.

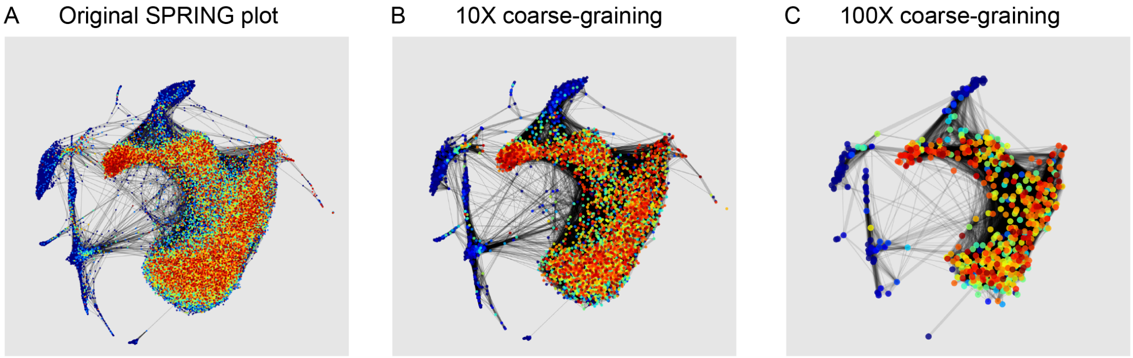


Figure 5: Coarse-graining of a dataset of 8,000 peripheral blood mononuclear cells

Coarse-graining pseudocode

```
input : A graph $G = (V, E)$ consisting of vertices $V = \{i\}$ and edges $E = \{(i, j)\}$.
        A gene expression profile $x_i$ for each vertex.
        The target number $N$ of vertices in the coarse-grained graph.
output: A coarse grained graph $G' = (V', E')$ with expression profiles $x'_i$.

$representativeNodes \leftarrow N$ unique integers between 1 and $|V|$
$nodeLabels \leftarrow$ dictionary mapping $i$ to $-1$ for each $i \in V$
for $i \in \{1, ..., N\}$ do
    $nodeLabels[representativeNodes[i]] = i$
end
while There exists $i$ such that $nodeLabels[i] = -1$ do
    for $(i, j) \in E$ do
        if $i = -1$ AND $j \neq -1$ then
            $nodeLabels[i] \leftarrow nodeLabels[j]$
        end
        if $i \neq -1$ AND $j = -1$ then
            $nodeLabels[j] \leftarrow nodeLabels[i]$
        end
    end
end
$V' \leftarrow \{1, ..., N\}$
for $i \in \{1, ..., N\}$ do
    $x'_i \leftarrow \mathrm{mean}(\{x_j \mid nodeLabels[j] = i\})$
end
$E' \leftarrow$ knn graph($\{x'_i\}$)
```

5. Extended Methods

Generating the tSNE plot in Figure 1B We selected genes based on their average expression level (keeping the top 25%) and their above-poisson noise (keeping the top 25%). Of these genes, we further restricted to the 532 principal variable genes (as in Klein *et al.* (2015)). tSNE was performed on Z-score normalized expression data, which was first reduced to 55 dimensions using PCA. We used a tSNE perplexity of 30.

Generating the diffusion map in Figure 1C We generated diffusion maps using the destiny package (Angerer *et al.*, 2016) in R. The DiffusionMap function with euclidean distance similarity was applied to Z-score normalized expression data after filtering genes as described in Methods (expression cutoff = 0.1, CV cutoff = 2).

Generating the tSNE plots in Figure 1D Gene filtering and PCA were performed on expression data from three human donors using the standard SPRING pipeline (see Methods; expression cutoff = 0.05, CV cutoff = 2, number of principal components = 50) and then processed with a tSNE perplexity of 30.

References

Angerer, P., Haghverdi, L., Büttner, M., Theis, F. J., Marr, C., and Buettner, F. (2016). destiny: diffusion maps for large-scale single-cell data in r. *Bioinformatics*, **32**(8), 1241–1243.

Klein, A. M., Mazutis, L., Akartuna, I., Tallapragada, N., Veres, A., Li, V., Peshkin, L., Weitz, D. A., and Kirschner, M. W. (2015). Droplet barcoding for single-cell transcriptomics applied to embryonic stem cells. *Cell*, **161**(5), 1187–1201.