Visualizing single-cell RNA-seq datasets with Similarity Weighted Nonnegative Embedding (SWNE)

Yan Wu[1], Pablo Tamayo[2,3], Kun Zhang[1]

[1]*Department of Bioengineering, University of California San Diego*

[2]*Moores Cancer Center, University of California San Diego*

[3]*School of Medicine, University of California San Diego*

Abstract

High throughput single-cell RNA-seq (scRNA-seq) has enabled the characterization of novel cell types, developmental trajectories, and responses to genetic perturbations. We developed a method for visualization and interpretation of scRNA-seq data, Similarity Weighted Nonnegative Embedding (SWNE), which captures both global and local structure, and embeds biological information directly onto the visualization. SWNE uses nonnegative matrix factorization to decompose the gene expression matrix into biologically relevant factors, embeds the cells, key genes, and factors in a 2D visualization, using a similarity matrix to smooth the embeddings. We evaluate SWNE on simulated data, and demonstrate SWNE on single-cell hematopoietic and brain datasets. The SWNE R package can be found at: www.github.com/yanwu2014/swne.

Background

Single cell gene expression profiling has enabled the quantitative analysis of many different cell types and states, such as human brain cell types[1,2] and cancer cell states[3,4], while also enabling the reconstruction of cell state trajectories during reprogramming and development[5–7]. Recent advances in droplet based single cell RNA-seq technology[2,8,9] as well as combinatorial indexing techniques[10,11] have improved throughput to the point where tens of thousands of single cells can be sequenced in a single experiment, creating an influx of large single cell gene expression

datasets. Numerous computational methods have been developed for latent factor identification[12], clustering[13], and cell trajectory reconstruction[6,7]. However, one of the most common visualization methods is still t-Stochastic Neighbor Embedding (t-SNE), a non-linear visualization method that tries to minimize the Kullback-Leibler (KL) divergence between the probability distribution defined in the high dimensional space and the distribution in the low dimensional space[14,15].

t-SNE very accurately captures local structures in the data, ensuring cells that are in the same cluster are close together[15]. This property enables t-SNE to find structures in the data that other methods, such as Principal Component Analysis (PCA)[16] and Multidimensional Scaling (MDS)[17], cannot[14]. However, t-SNE often fails to accurately capture global structures in the data, such as distances between clusters, possibly due to asymmetry in the KL divergence metric and the necessity of fine-tuning the perplexity hyperparameter[14]. Additionally, t-SNE visualizations do not provide biological context, such as which genes are expressed in which clusters, requiring additional plots or tables to interpret the data. While some newer methods such as UMAP address the issue of capturing global structures in the data, no methods, to our knowledge, allow for biological information to be embedded onto the visualization[18].

Results

We developed a method for visualizing high dimensional single cell gene expression datasets, Similarity Weighted Nonnegative Embedding (SWNE), which captures both local and global properties of the data, while enabling genes and relevant biological factors to be embedded directly onto the visualization. SWNE adapts the Onco-GPS NMF embedding framework[19] to decompose the gene expression matrix into latent factors, embeds both factors and cells in two dimensions, and smooths both the cell and factor embeddings by using a similarity matrix to ensure that cells which are close in the high dimensional space are also close in the visualization.

First, SWNE uses Nonnegative Matrix Factorization (NMF)[20,21] to create a parts based factor decomposition of the data (**Figure 1a**). The number of factors (k) is chosen by randomly selecting 20% of the gene expression matrix to be set to missing, and then finding the factorization that best imputes those missing values, minimizing the mean squared error (Euclidean distance) (**Figure S1a**). With NMF, the gene expression matrix (A) is decomposed into: (1) a *genes by factors* matrix (W), and (2) a *factors by cells* matrix (H) (**Figure 1a**). SWNE then uses the similarity matrix, specifically a Shared Nearest Neighbors (SNN) matrix[22], to smooth the H matrix, resulting in a new matrix H_{smoot} . SWNE calculates the pairwise distances between the rows of the H_{smooth} matrix, and uses Sammon mapping[23] to project the distance matrix onto two dimensions (**Figure 1a**). Next, SWNE embeds genes relative to the factors using the gene loadings in the W matrix, and embeds cells relative to the factors using the cell scores in the unsmoothed H matrix (**Figure 1a**). Finally, SWNE uses the SNN matrix to smooth the cell coordinates so that cells which are close in the high dimensional space are close in the visualization (**Figure 1a**). In the following analyses, we specifically use an SNN matrix although it is possible to use other types of similarity matrices.

To benchmark SWNE against PCA, t-SNE, and other visualization methods, we used the Splatter scRNA-seq simulation method[24] to generate two synthetic datasets. We generated a 2700 cell dataset with five discrete groups, where Groups 2 – 4 were relatively close and Groups 1 & 5 were further apart (**Figure 1b**). We also generated a simulated branching trajectory dataset with 2730 cells and four different paths, where Path 1 branches into Paths 2 & 3, and Path 4 continues from Path 3 (**Figure 1b**). For the discrete simulation, the t-SNE plot qualitatively distorts the cluster distances, making Groups 1 & 5 closer than they should be to Groups 2 – 4 **(Figure 1c**). The SWNE plot more accurately shows that Groups 1 & 5 are far from each other and Groups 2 – 4 **(Figure 1c)**. PCA **(Figure S2b, S2c)**, LLE **(Figure S2f)**, and MDS **(Figure S2g)** do a better job of accurately visualizing these cluster distances, but have

trouble separating Groups 2 – 4. For the branching trajectory simulation, the t-SNE plot incorrectly expands the background variance of the paths, while the SWNE plot does a much better job of capturing the important axes of variance, resulting in more clearly defined paths **(Figure 1d)**. PCA **(Figure S2i, S2j)**, LLE **(Figure S2m)**, and MDS **(Figure S2n)** again do a better job of capturing the trajectory-like structure of the data, but still inaccurately expand the background variance more than SWNE.

To quantitatively benchmark the visualizations, we developed metrics to quantify how well each embedding captures both the global and local structure of the original dataset. For the discrete simulation, we calculated the pairwise distances between the group centroids in the original gene expression space, and then correlated those distances with the pairwise distances in the 2D embedding space to evaluate the embeddings' ability to capture global structure **(Figure 1e)**. To evaluate local structure, we calculated the average Silhouette score[25], a measure of how well the groups are separated, for each embedding **(Figure 1e)**. SWNE outperforms t-SNE and performs nearly as well as PCA, MDS, and Diffusion Maps in maintaining global structure **(Figure 1e)**. Looking at the Silhouette scores, SWNE clearly outperforms every other embedding, including t-SNE, which comes in second **(Figure 1e)**. For the trajectory simulation, since we know the simulated pseudotime for each cell, we divide each path into groups (chunks) of cells that are temporally close **(Methods)**. We then evaluate global structure by calculating pairwise distances between each path-time-chunk in the original gene expression space and the 2D embedding space, and then correlating those distances **(Figure 1f)**. We can evaluate local structure by constructing a ground truth neighbor network by connecting cells from adjacent pseudotimes, and then calculating how well each cell's ground truth neighborhood matches up with its nearest neighbors in each 2D embedding to create a neighborhood score **(Methods, Figure 1f)**. SWNE outperforms t-SNE in capturing global structure, and is slightly outperformed by PCA, MDS, and LLE **(Figure 1f)**. For capturing neighborhood structure, SWNE again outperforms every other embedding, including t-SNE

(Figure 1f). Finally, both the qualitative and quantitative benchmarks show that SNN smoothing of the cell and factor embeddings is critical to SWNE's performance, especially for capturing local structure in the data **(Figure 1e, 1f)**.

We then applied SWNE to analyze the single cell gene expression profiles of hematopoietic cells at various stages of differentiation from Paul, et al[26] (**Figure 2a**). Briefly, single cells were sorted from bone marrow and their mRNA was sequenced with scRNA-seq[26] (**Figure 2a**). The differentiation trajectories of these cells were reconstructed using Monocle2[6], a method built to identify branching trajectories and order cells according to their differentiation status, or "pseudotime" (**Figure 2a**). The branched differentiation trajectories are shown in the tree in **Figure 2a**, starting from the monocyte and erythrocyte progenitors (MP/EP) and either moving to the erythrocyte (Ery) branch on the left, or the various monocyte cell types on the right[6]. To select the number of factors to use for NMF, we use our missing value imputation method and find the k that minimizes the reconstruction error (**Figure S1b**, **Methods**).

Qualitatively, the SWNE plot (**Figure 2b**) separates the different cell types at least as well as the t-SNE plot (**Figure 2c**). However, the SWNE plot does a much better job of capturing the two dominant branches: the erythrocyte differentiation branch and the monocyte differentiation branch, and shows that those two branches are the primary axes of variation in this dataset (**Figure 2b**). While the t-SNE plot captures the correct orientation of the cell types, it is not clear that there are two main branches in the data and that the main variation in this dataset is along these two branches (**Figure 2c**). Both SWNE and t-SNE do not accurately orient the different monocyte cell types in the monocyte branch, most likely because the variance is dominated by the erythrocyte – monocyte split, and the extent of differentiation. We also used Monocle2 to calculate differentiation pseudotime for the dataset, which is a metric that orders cells by how far along the differentiation trajectory they are[6]. We then overlaid the pseudotime score on the SWNE and t-SNE plots (**Figure 2d, 2e**). Again, we can see that in the SWNE plot, there's a clear gradient of cells at different stages of differentiation along the two

main branches (**Figure 2d**). The gradient in the t-SNE plot is not as visible because the t-SNE plot seems to be capturing the background variance in the highly differentiated cells (**Figure 2e**).

SWNE provides an intuitive framework to visualize how specific genes and biological factors contribute to the visual separation of cell types or cell trajectories by embedding factors and genes onto the visualization. We used the gene loadings matrix (W) to identify the top genes associated with each factor, as well as the top marker genes for each cell type, defined using Seurat[27] (**Methods**). We chose three factors and five genes that we found biologically relevant (**Figure S3a, S3b, S3c**). The five genes are: *Apoe, Flt3, Mt2, Sun2, and Pglyrp*. The three factors are: Antigen Presentation, Metal Binding, and Platelet Generation, and factor names were determined from the top genes associated with each factor (**Figure S3a**) (**Table S1**). The factors and genes enable a viewer to associate biological processes and genes with the cell types and latent structure shown in the data visualization. For example, dendritic cells (DC) are associated with Antigen Presentation, while erythrocytes (Ery) are associated with Heme metabolism and express *Mt2*, a key metal binding protein (**Figure 2b**). Additionally, the embedded factors and genes allow for interpretation of the overall differentiation process (**Figure 2d**). Undifferentiated progenitors (MP/EP) express *Apoe*, while more differentiated monocytes express *Sun2* and *Pglyrp1* (**Figure 2d, Figure S4a**).

We also applied SWNE to a single nucleus RNA-seq human brain dataset[2] from the visual cortex (13,232 cells) and the cerebellum (9,921 cells) (**Figure 3a**). Briefly, single nuclei were dissociated from the visual cortex and cerebellum of a single donor and sequenced using single nucleus Drop-seq[2]. We also applied SWNE to the subset of layer specific excitatory neurons in the visual cortex, where each layer has different functions[28–30] (**Figure 3a, inset**). For the SWNE plot, we selected the number of factors using the same missing value imputation method as for the hematopoiesis dataset (**Figure S1c, S1d**). We can see that both the SWNE plot (**Figure 3b**) and the t-SNE plot (**Figure 3d**) are able to visually separate the various brain

cell types. However, the SWNE plot is able to ensure that related cell types are close in the visualization, specifically that the inhibitory neuron subtypes (In1 – 8) are together in the top of the plot (**Figure 3b**). In the t-SNE plot the inhibitory neuron subtypes are separated by the Astrocytes (Ast) and the Oligodendrocytes (Oli) (**Figure 3d**).

SWNE again provides an intuitive framework to visualize the contributions of specific genes and factors to the cell types. We selected three factors (Myelin, Cell Junctions, and Immune Response) and 8 genes (*PLP1, GRIK1, SLC1A2, LHFPL3, CBLN2, NRGN, GRM1, FSTL5*) to project onto the SWNE plot using the cell type markers and gene loadings (**Figure S3d, Figure S3e, Figure S3f, Table S1**), adding biological context to the spatial placement of the cell types (**Figure 3b**). We can see that *CBLN2*, a gene known to be expressed in excitatory neuron types[31], is expressed in the visual cortex excitatory neurons and that *GRIK1*, a key glutamate receptor[32], is expressed in inhibitory neurons (**Figure 3b, Figure S4b**). Additionally, the Myelin biological factor is associated with Oligodendrocytes (Oli), consistent with their function in creating the myelin sheath[33] (**Figure 3b**). The Cell junction biological factor is very close to Endothelial cells (End), reinforcing their functions as the linings of blood vessels (**Figure 3b**).

Finally, SWNE has a unique advantage over t-SNE in capturing the local and global structure of the data, exemplified when we zoom into the layer specific excitatory neurons (**Figure 3c, 3e**). The SWNE plot visually separates the different neuronal layers, while also showing that the main axis of variance is along the six cortical layers of the human brain (**Figure 3c**). Each layer seems to branch off of a main trajectory, illustrating that SWNE can visualize the structure and function of the cortical layers (**Figure 3c**). The t-SNE plot can visually separate the layers, but the layer structure is not apparent and it is unclear that the main axis of variance is between the different layers (**Figure 3e**). Additionally, we selected five layer specific marker genes (*DAB1, NTNG1, DCC, HS3ST2, POSTN*) to project onto the SWNE plot (**Figure S2g, Figure 3c**). *DAB1*, a signal transducer for Reelin[34], is primarily expressed in Layer 2/3 excitatory

neurons, while NTNG1, a protein involved in axon guidance[35], is expressed in Layer 4 neurons (**Figure 3c, Figure S4c**).

Discussion

One important SWNE parameter is the number of factors (k) used for the decomposition (Figure 1a). We used a model selection method, suggested by the author of the NNLM[36] package, which uses NMF to impute missing values in the gene expression matrix (A) and tries to select the k that minimizes the imputation error (**Figure S1, Methods)**. However, oftentimes there is a range of k that is very close to the global minimum error, such as in **Figure S1a** where k could be anywhere from 12 to 16. We have found that the global minimum will also sometimes vary depending on the fraction of values set missing in A, and also the specific random sample of matrix elements. Because of this variability, we use our model selection method to first narrow down the range of possible k values, and then construct visualizations across this narrower range of k to pick the qualitatively best visualization (**Methods**). We calculated global and local evaluation metrics across a range of k values for the discrete and trajectory simulations **(Figure S1e, S1f)**. The SWNE performance was fairly stable across values of k, although it seems as if it is better to err on the side of higher values of k **(Figure S1e, S1f)**. One area of future work could be to develop an unbiased model selection method explicitly for creating the optimal visualization.

Selecting which genes and factors to embed onto the SWNE plot is an important process. Ideally one wants to select the best marker genes for cell types of interest, or the genes with the highest magnitude loadings for biologically relevant factors. Since embedding a gene as a single data point on the plot does not convey the same amount of information as overlaying the gene expression, we created feature plots for key genes in the hematopoiesis, cortex & cerebellum, and layer specific excitatory neuron datasets to demonstrate that cells

which are spatially close to an embedded gene actually express more of that gene (**Figure S4a, S4b, S4c**). For example, we can see in **Figure S4b** that the excitatory neurons close to the embedded CBLN2 point express more CBLN2. In cases where the gene selected is not a good cell type marker, then the gene's embedded coordinates should be near the center of the plot, equidistant to one or more groups of cells.

One additional highlight of SWNE is that that the underlying methodology is fairly simple, especially compared to methods such as UMAP and t-SNE[14,18]. The Onco-GPS based embedding, and subsequent similarity matrix weighting is very transparent, allowing users to understand how the visualization is being produced. For many users, methods such as UMAP and t-SNE can become a black box, which can result in the incorrect usage of key hyperparameters, such as t-SNE perplexity, and over-interpretation of potential computational artifacts.

The simplicity of SWNE also makes it possible to project additional data onto an existing SWNE plot, something that is difficult to do with non-linear methods like t-SNE and UMAP (**Methods, Figure S5a, S5b**). To demonstrate data projection, we used a 3,000 cell PBMC dataset generated by 10X genomics[37], and split the data into training and test datasets. We ran the standard SWNE embedding on the training dataset, and then projected the test dataset onto the training SWNE embedding (**Figure S5a, S5b**). We can see that the various clusters occupy the same general coordinates in the SWNE plot from training to test (**Figure S5a, S5b**). Additionally, we projected a 33,000 cell PBMC dataset with additional cell types onto the training SWNE embedding **(Figure S5c)**. The projected embedding obviously cannot distinguish between all the PBMC subtypes since they were not present in the training dataset. Nevertheless, SWNE does a fairly good job of making sure the new projected cell types are near similar cell types in the training dataset **(Figure S5c)**.

Finally we performed a runtime analysis for SWNE as we increase the number of cells, and the number of genes **(Figure S7a, S7b)**. It looks like SWNE scales linearly with the number

of cells, and can embed 50,000 cells using the top 3,000 overdispersed genes in around 17 minutes **(Figure S7a)**. However, SWNE seems to scale quadratically with the number of genes, implying that feature selection is vital to reducing runtimes **(Figure S7b)**.

Conclusion

Overall, we developed a visualization method, SWNE, which captures both the local and global structure of the data, and enables relevant biological factors and genes to be embedded directly onto the visualization. Capturing global structure enables SWNE to successfully capture differentiation trajectories, and layer-specific neuron structure that is not apparent in other visualizations such as t-SNE. Capturing local structure with the SNN smoothing enables SWNE to accurately visualize the key axes of variation, reducing the impact of background noise. Additionally, being able to embed key marker genes and relevant biological factors adds important biological context to the SWNE plot, facilitating interpretation. Finally, the simplicity of SWNE allows users to intuitively understand the embedding process, as well as project new data on existing SWNE visualizations. We applied SWNE to a hematopoiesis dataset, where it was able to capture the branched differentiation trajectory. We also applied SWNE to cells from the visual cortex and cerebellum, where it was able to visually separate different cell types while ensuring close cell types, such as different inhibitory neuron subtypes, are close together. SWNE was also able to capture the layer specific structure of excitatory neurons, demonstrating that SWNE can visualize both biological structure and function.

Future work could include examining how external methods for generating similarity matrices and for factor decomposition work within the SWNE framework. For example, one could use SIMLR[13] to create the similarity matrix instead of using an SNN matrix. There are also a variety of methods for decomposing gene expression matrices into latent factors, including f-scLVM and pagoda/pagoda2[12,38]. Both these methods can use pre-annotated gene sets to

guide the factor decomposition, which would allow the embedded factors to be even more easily interpreted.

Acknowledgements

We would like to acknowledge Dr. Prashant Mali for their feedback and advice, and Dinh Diep for their technical feedback. Additionally, we would like to acknowledge the Zhang Lab, the Tamayo Lab, and the Mali Lab for their help and support.

Funded in part by NIH grants R01HG009285 (KZ & PT), U01CA217885 (PT), and P30 CA023100 (PT), U01MH098977 (YW & KZ), R01HL123755 (YW & KZ)

Code and Data

The SWNE package is available at https://github.com/yanwu2014/swne. The scripts used for this manuscript are under the Scripts directory. The data needed to recreate the figures can be found here:

- ftp://genome-miner.ucsd.edu/swne_files/hemato_data.tar.gz (Hematopoiesis data)
- ftp://genome-miner.ucsd.edu/swne_files/neuronal_data.tar.gz (Neuronal data)

The raw data for the hematopoietic and neuronal cells can be found at the GEO accessions GSE72857 and GSE97930, respectively. The PBMC dataset can be found at the 10X genomics website: https://support.10xgenomics.com/single-cell-gene-expression/datasets/1.1.0/pbmc3k.

Conflicts of Interest

The authors declare no conflicts of interest.

Methods

Normalization, variance adjustment, and scaling

We normalize the gene expression matrix by dividing each column (sample) by the column sum and multiplying by a scaling factor. Batch effects were normalized by a simple model, adapted from pagoda2[38,39], that subtracts any batch specific expression from each gene. We used the variance adjustment method from pagoda[38] to adjust the variance of features, an important step when dealing with RNA-seq data. Briefly, a mean-variance relationship for each feature is fit using a generalized additive model (GAM) and each feature is multiplied by a variance scaling factor calculated from the GAM fit. Feature scaling is also performed using either a log-transform, or the Freeman-Tukey transform.

Feature Selection

We highly recommend doing some sort of selection of overdispersed genes before running SWNE, as the NMF algorithm scales poorly with the number of features. Both Pagoda2 and Seurat offer dispersion based feature selection methods, and we wrapped with Pagoda2 feature selection method in an SWNE function.

Nonnegative Matrix Factorization and model selection

We use the NNLM package[36] to run the Nonnegative Matrix Factorization (NMF). **Equation 1** shows the NMF decomposition:

$$A = WH \quad (1)$$

Where A is the (features x samples) data matrix, W is the (features x factors) feature loading matrix, and H is the (factors x samples) low dimensional representation of the data. The NMF initialization method can affect the embedding, and we offer an Independent Component Analysis (ICA) initialization, a Nonnegative-SVD (NNSVD) initialization, and a purely random initialization. ICA initialization works well with most datasets, and is set as the default option. We select the number of factors by setting a random subset of the data as missing, usually around

25% of the matrix, and then use the NMF reconstruction $(W\ x\ H)$ to impute the missing values across a range of factors. The number of factors, *k*, which minimizes the mean squared error, is typically the optimal number of factors to use. In some cases, there are multiple local minima, or there are multiple values of *k* that are very close to the global minimum, so we create SWNE visualizations for a subset of those values of *k* and pick the *k* which results in the qualitatively best visualization.

Generating the SNN matrix

In order to ensure that samples which are close to each other in the high dimensional space are close in the 2d embedding, we smooth the NMF embeddings with a Shared Nearest-Neighbors (SNN) matrix, calculated using code adapted from the Seurat package[27]. Briefly, we calculate the approximate k-nearest neighbors for each sample using the Euclidean distance metric (in the Principal Component space. We then calculate the fraction of shared nearest neighbors between that sample and its neighbors. We can then raise the SNN matrix, denoted here as S, to the exponent β: $S' = S^{\beta}$. If $\beta > 1$, then the effects of neighbors on the cell embedding coordinates will be decreased, and if $\beta < 1$, then the effects will be increased. Finally we normalize the SNN matrix so that each row sums up to one.

Weighted Factor Projection

We adapt the Onco-GPS[19] methodology to embed the NMF factors onto a two dimensional visualization. First, we smooth the H matrix with the SNN matrix using **Equation 2**: $H_{smooth} =$

$$H * S \quad (2)$$

We then calculate the pairwise similarities between the factors (rows of the H_{smoo} matrix) using either cosine similarity, or mutual information[40]. The similarity is converted into a distance with **equation 3**:

$$D = \sqrt{2(1 - R)} \quad (3)$$

Here, R is the pairwise similarity. We use Sammon mapping[23] to project the distance matrix into two dimensions, which represent the x and y coordinates for each factor. The factor coordinates are rescaled to be within the range zero to one.

Weighted Sample Embedding

Let F_{ix}, F_{iy} represent the x and y coordinates for factor i. To embed the samples, we use the sample loadings from the *unsmoothed* H matrix via **equations 4 & 5**:

$$L_{jx} = \frac{\sum_i (H_{ij} F_{ix})^{\alpha}}{\sum_i {H_{ij}}^{\alpha}} \quad (4)$$

$$L_{jy} = \frac{\sum_i (H_{ij} F_{iy})^{\alpha}}{\sum_i {H_{ij}}^{\alpha}} \quad (5)$$

Here, j is the sample index and i is iterating over the number of factors in the decomposition (number of rows in the H matrix). The exponent α can be used to increase the "pull" of the NMF components to improve separation between sample clusters, at the cost of distorting the data. Additionally, we can choose to sum over a subset of the top factors by magnitude for a given sample, which can sometimes help reduce noise. We end up with a $2\ x\ N$ matrix of sample coordinates, L.

To weight the effects of the SNN matrix on the samples, the sample coordinates L are smoothed using **equation 6**:

$$L_{smooth} = S * L \quad (6)$$

The smoothed sample coordinates (L_{smooth}) are then visualized. While we have found that an SNN matrix works well in improving the local accuracy of the embedding, other similarity matrices, such as those generated by scRNA-seq specific methods like SIMLR, could also work. In general, you should use whichever similarity or distance matrix you used for clustering.

Embedding features

In addition to embedding factors directly on the SWNE visualization, we can also use the gene loadings matrix (W) to embed genes onto the visualization. We simply use the W matrix to embed a gene relative to each factor, using the same method we used to embed the cells in the H matrix. If a gene has a very high loading for a factor, then it will be very close to that factor in the plot, and far from factors for which the gene has zero loadings.

Constructing the SNN matrix from different dimensional reductions

The SNN matrix can be constructed from either the original gene expression matrix (A), or on some type of dimensional reduction. We have found that constructing the SNN matrix from a PCA reduction tends to work well, especially in datasets where that follow a trajectory or trajectories (**Figure S6a**). Constructing the SNN matrix from the gene expression matrix is somewhat similar to using PCA; although the separation between cell types is not as clear (**Figure S6b**). However, using the NMF factors to build the SNN matrix oftentimes does not capture the primary axis or axes of variance, especially in cases where there is some type of smooth trajectory (**Figure S6c**). We believe this is due to PCA's ability to capture the axes of maximum variance, while NMF looks for a parts-based representation[16,20]. For datasets where there are discrete cell types, constructing the SNN matrix from the NMF factors is often similar to constructing the SNN matrix from PCA components. Thus, we default to building the SNN matrix from principal components.

Interpreting NMF components

In order to interpret the low dimensional factors, we look at the gene loadings matrix (W). We can find the top genes associated with each factor, in a manner similar to finding marker genes for cell clusters. Since we oftentimes only run the NMF decomposition on a subset of the overdispersed features, we can use a nonnegative linear model to project the all the genes onto

the low dimensional factor matrix. We can also run Geneset Enrichment Analysis[41] on the gene loadings for each factor to find the top genesets associated with that factor.

Projecting New Data

To project new data onto an existing SWNE embedding, we first have to project the new gene expression matrix onto an existing NMF decomposition, which we can do using a simple nonnegative linear model. The new decomposition looks like **equation 7**:

$$A' = WH' \quad (7)$$

Here, *A'* is the new gene expression matrix, and *W* is the original gene loadings matrix, which are both known. Thus, we can simply solve for *H'*. The next step is to project the new samples onto the existing SNN matrix. We project the new samples onto the existing principal components, and then for each test sample, we calculate the *k* closest training samples. Since we already have the kNN graph for the training samples, we can calculate, for each test sample, the fraction of Shared Nearest Neighbors between the test sample and every training sample. With the test factor matrix *H'*, and the test SNN matrix, we can run the SWNE embedding as previously described to project the new samples onto the existing SWNE visualization.

Simulations and quantitative evaluations

We used the Splatter [24] R package to generate a discrete dataset with five different clusters, estimating parameters from the 3k PBMC dataset published by 10X genomics. We generated five distinct clusters (groups), where Groups 1 and 5 had a differential expressed gene (DEG) probability of 0.3, while Groups 2 – 4 had a DEG probability of 0.15. Group 5 contains 1215 cells, Groups 2 – 4 contain 405 cells each, and Group 1 contains 270 cells. Thus, Groups 1 & 5 should be relatively distant and Groups 2 – 4 should be relatively close. To simulate a branching trajectory dataset, we estimated parameters from the hematopoiesis dataset from Paul et al. We generated four paths, where each path is parameterized by the number of cells in that path and

the number of “time-steps”, which essentially controls how long the path is. Path 1 branches into Paths 2 and Paths 3, and Path 3 continues onto Path 4. Paths 1 & 2 contained 819 cells each, and Paths 3 & 4 contained 546 cells each. Path 1 had 100 steps, Path 2 was the “longest” path with 200 steps, and Paths 3 & 4 had 50 steps each. Each cell is assigned to a path, and a time-step. For example, Cell2522 might belong to Path1 and time-step 68.

To evaluate how well each embedding maintained the global structure of the discrete simulation, we correlated the pairwise cluster distances in the 2D embedding with the pairwise cluster distances in the original gene expression space. We then calculated the average Silhouette score for each embedding, evaluating how well the visualization separates the clusters. For the trajectory simulation, we divided each path into “chunks” of five time-steps. We correlated the pairwise distances of each “path-time-chunk” in the embedding space, and the original gene expression space to evaluate how well the embeddings maintained the global structure. To evaluate the local structure, we constructed a “ground-truth” neighborhood graph by adding an edge between every cell in each path-time-step, and every cell in each neighboring path-time-step. For example, we would connect all the cells in Path1 at time-step 23, with all the cells in Path1 and time-step 24. We then created a nearest neighbors graph for each embedding, and took the Jaccard distance between each cell’s neighborhood in the embedding and the true neighborhood. We used the average Jaccard distance as our “neighborhood score”.

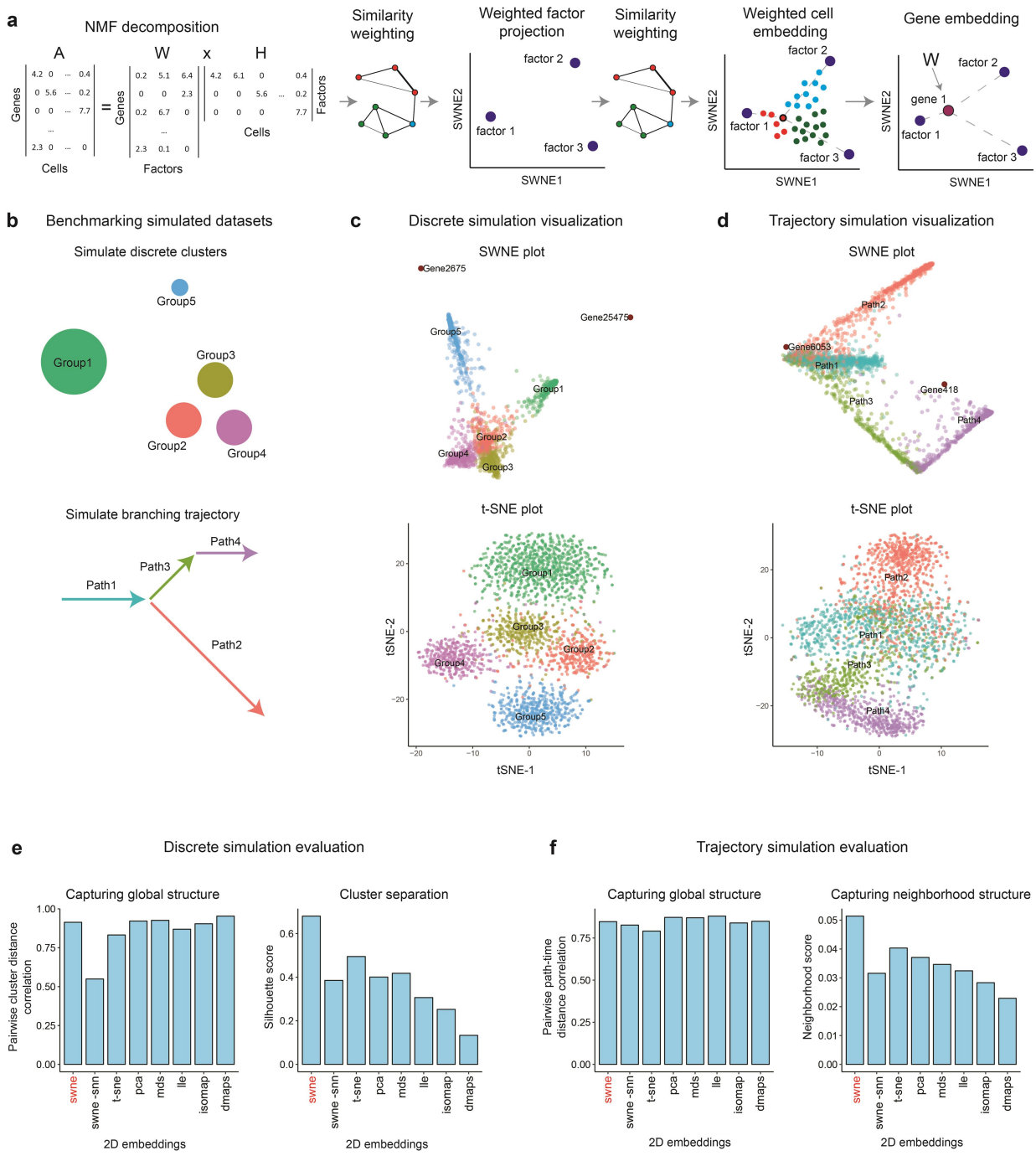

Figure 1: (a) Schematic overview of SWNE. The gene expression matrix (A) is decomposed into a gene loadings matrix (W) and a factor matrix (H) using Nonnegative Matrix Factorization (NMF), with the number of factors selected via minimizing imputation error. The factors are projected onto 2D via Sammon mapping by calculating pairwise distances between the factors (rows of H). Selected genes can be embedded relative to the factors using the gene loadings

(W) matrix. Finally, the cells are embedded relative to the factors using the cell scores in the H matrix and the cell coordinates are then refined using a similarity matrix. **(b)** Simulating a discrete dataset with five clusters, and a branching trajectory dataset with four paths. **(c)** SWNE and t-SNE plots of the discrete simulated dataset. **(d)** SWNE and t-SNE plots of the trajectory simulated dataset. **(e)** Quantitative evaluation of SWNE and existing visualization methods on the discrete simulation. Global structure is evaluated by correlating pairwise cluster distances in the 2D embeddings with cluster distances in the original gene expression space. Cluster separation is evaluated by calculating the Silhouette score for each 2D embedding. **(f)** Quantitative evaluation of SWNE and existing visualization methods on the trajectory simulation. Global structure is evaluated by dividing each path up into time steps, and correlating pairwise path-time-step distances in the embedding with distances in the original gene expression space. Local structure is evaluated by taking the Jaccard similarity of the nearest neighbors in the embeddings, with the true nearest neighbors.

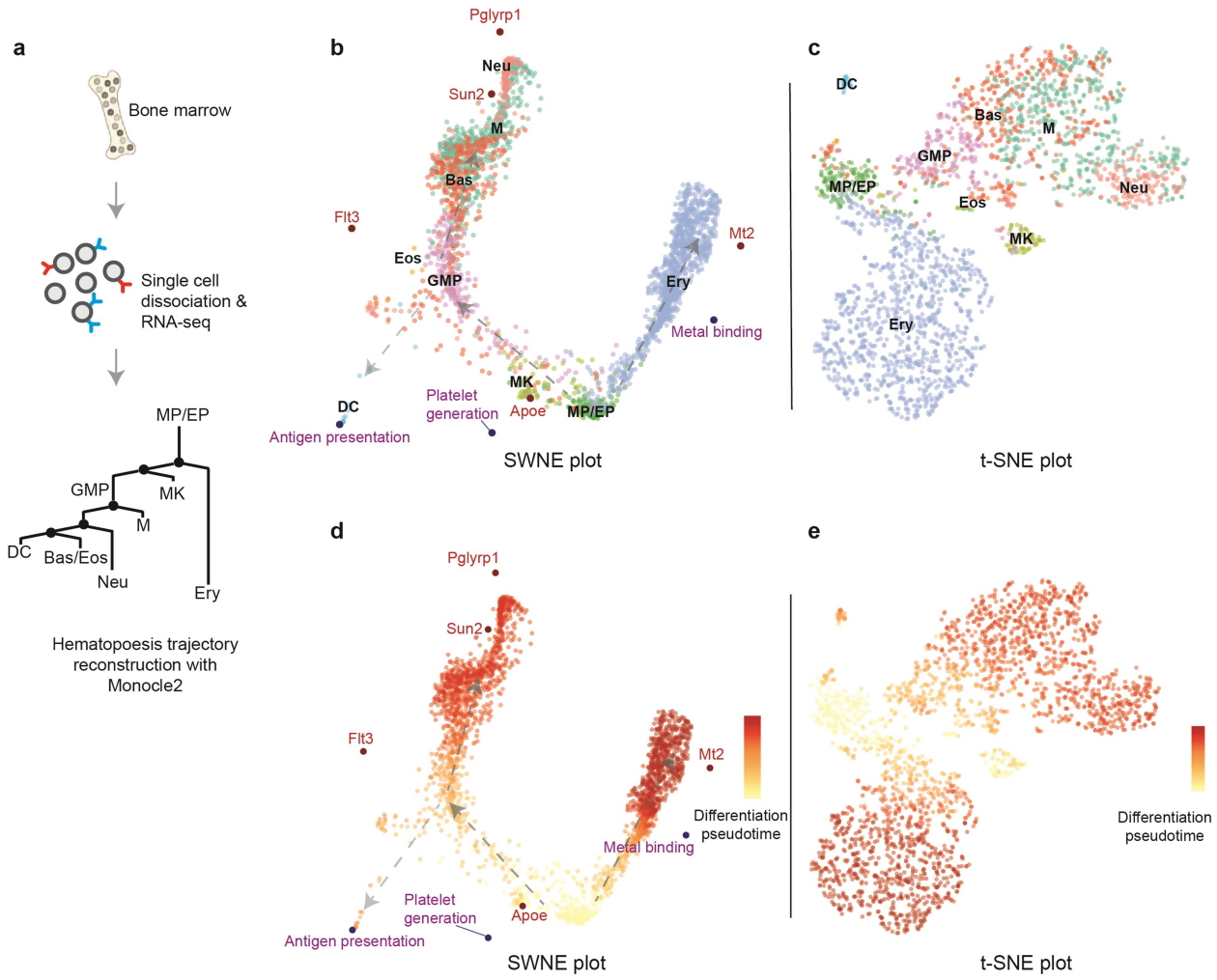

Figure 2: (a) Paul et al sorted single hematopoietic cells from bone marrow, sequenced them using scRNA-seq, and identified the relevant cell types. The hematopoiesis trajectories were reconstructed using Monocle2, and the cells were ordered according to their differentiation pseudotime. **(b)** SWNE plot of hematopoiesis dataset, with selected genes and biological factors displayed. **(c)** t-SNE plot of hematopoiesis dataset. **(d)** SWNE plot of hematopoiesis dataset, with developmental pseudotime calculated from Monocle2 overlaid onto the plot. **(e)** t-SNE plot of hematopoiesis dataset, with developmental pseudotime overlaid onto the plot.

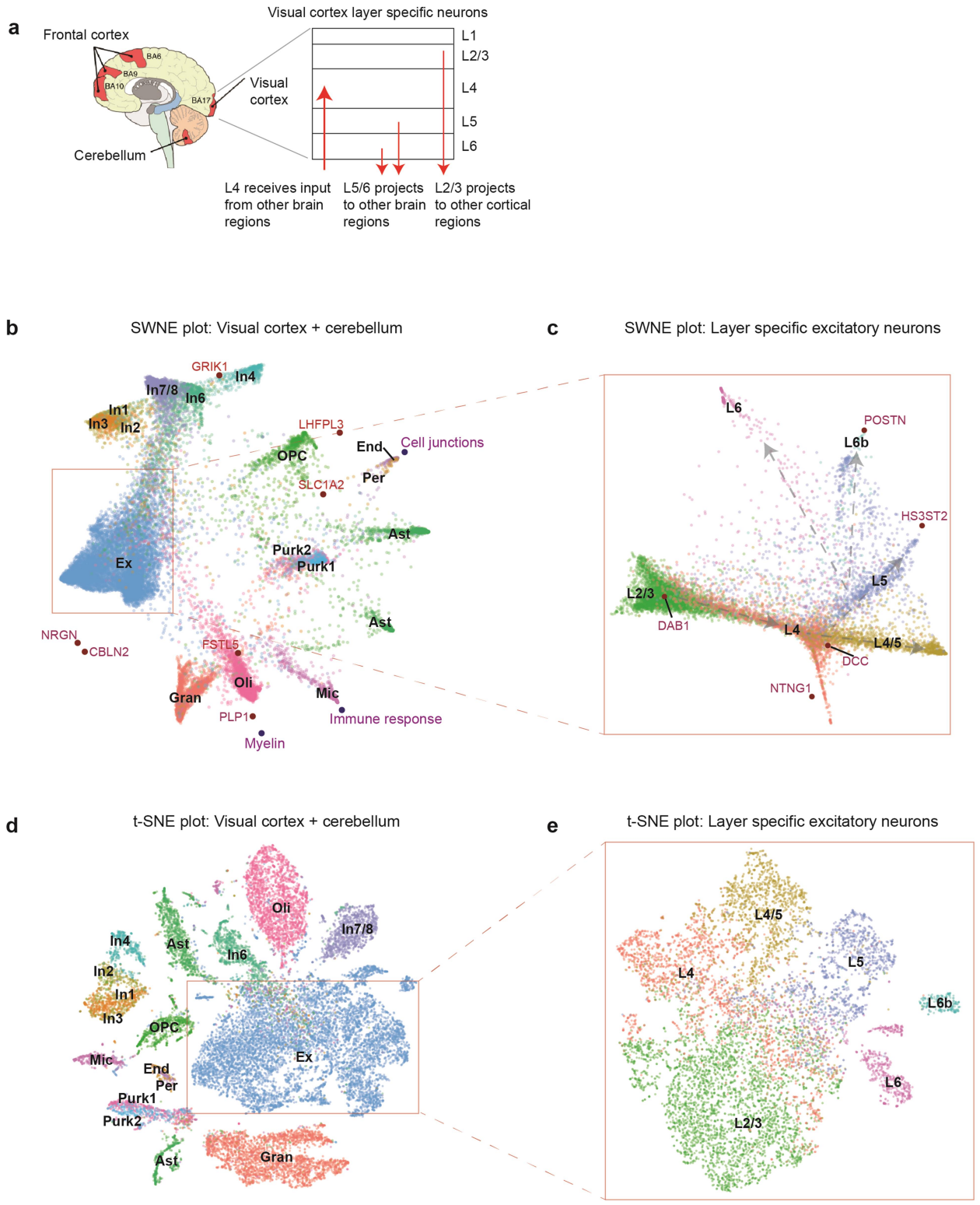

Figure 3: (a) Single nuclei were dissociated from the visual cortex and the cerebellum, and sequenced using single nucleus RNA-seq. The inset shows that the excitatory neurons from the

visual cortex are grouped into different spatial layers, each of which has different functions. **(b)** SWNE plot of single cells from the visual cortex and cerebellum, with selected genes and factors displayed. **(c)** Inset: SWNE plot of the excitatory neurons from the visual cortex only, colored by the spatial layer the excitatory neurons belong to. **(d)** t-SNE plot of single cells from the visual cortex and cerebellum. **(e)** t-SNE plot of the excitatory neurons from the visual cortex only, colored by the spatial layer the excitatory neurons belong to.

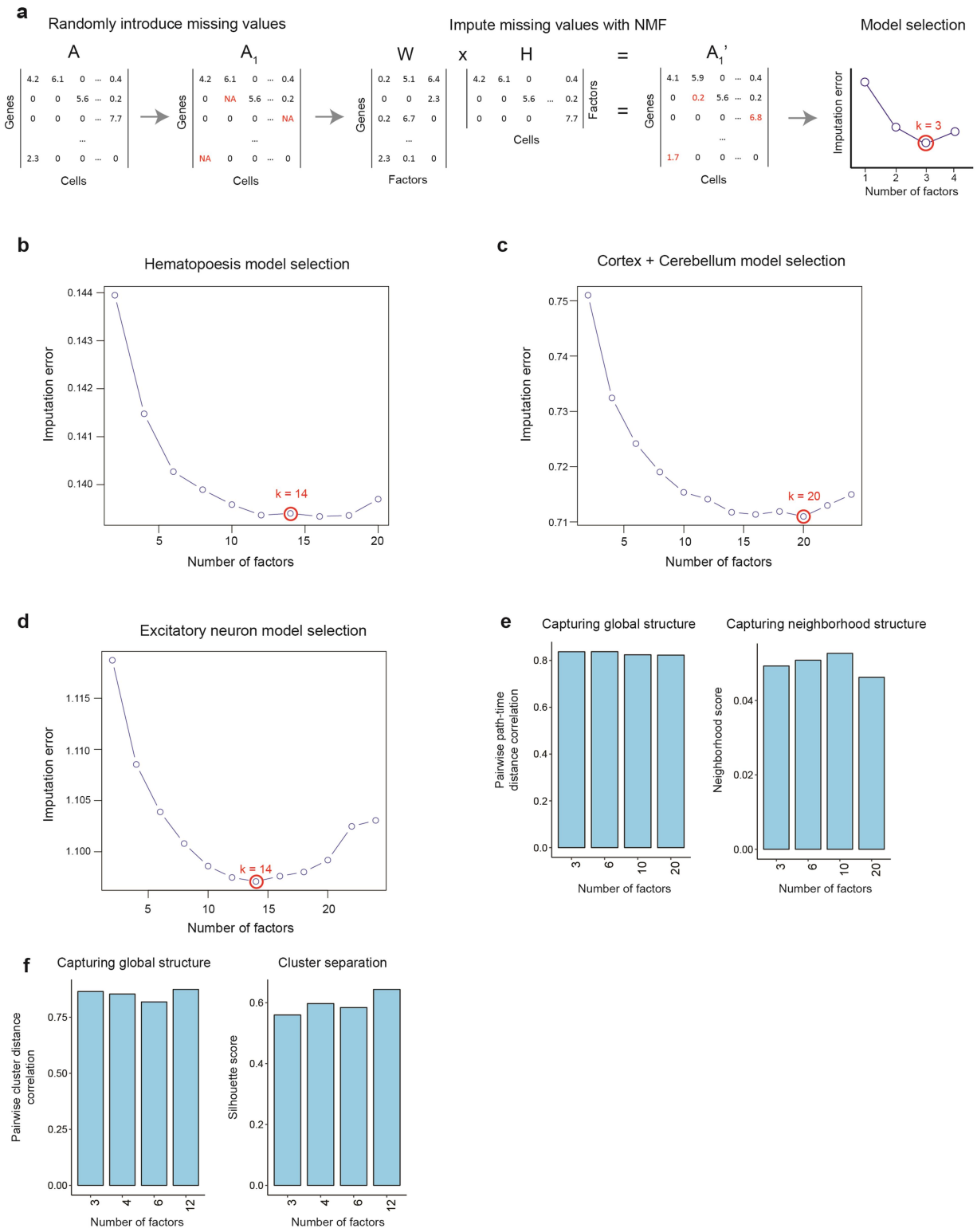

Figure S1: (a) A subset of the gene expression matrix is set to missing, and NMF is run across a range of factors, and the missing values are imputed. We then plot the imputation error vs

number of factors (*k*), and select the *k* values close to the minimum imputation error to create SWNE visualizations with. The *k* that gives the best visualization is selected. **(b)** Imputation error versus number of NMF factors for the hematopoiesis dataset **(c)** Imputation error versus number of NMF factors for the cortex & cerebellum dataset. **(d)** Imputation error versus number of NMF factors for the layer specific excitatory neurons.

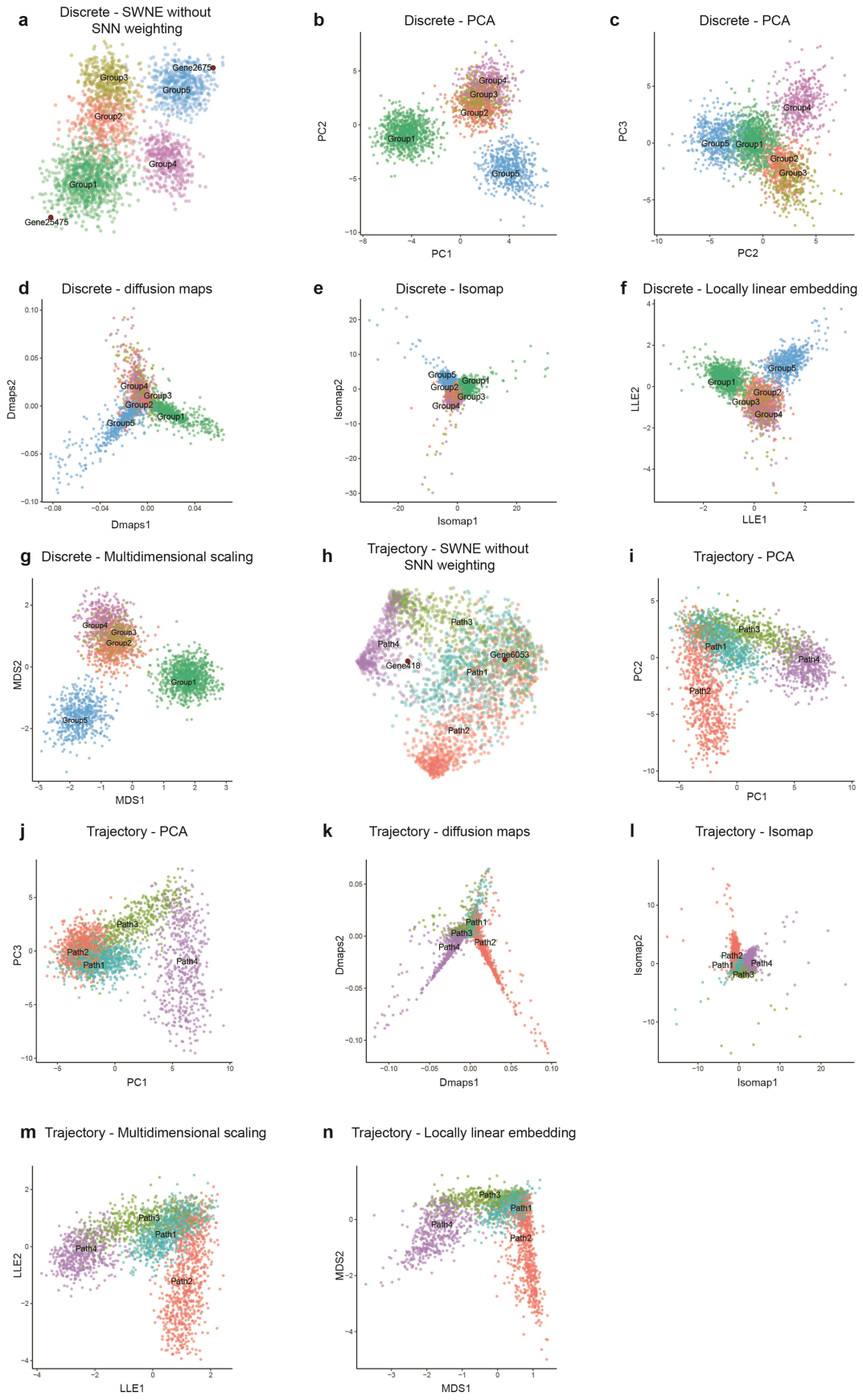
a
Discrete - SWNE without SNN weighting
Group3
Gene2675
Group5
Group2
Group4
Group1
Gene25475
b
Discrete - PCA
PC2
PC1
Group4
Group3
Group2
Group1
Group5
c
Discrete - PCA
PC3
PC2
Group4
Group5
Group1
Group2
Group3
d
Discrete - diffusion maps
Dmaps2
Dmaps1
Group4
Group3
Group2
Group1
Group5
e
Discrete - Isomap
Isomap2
Isomap1
Group5
Group1
Group2
Group3
Group4
f
Discrete - Locally linear embedding
LLE2
LLE1
Group5
Group1
Group2
Group3
Group4
g
Discrete - Multidimensional scaling
MDS2
MDS1
Group4
Group3
Group2
Group1
Group5
h
Trajectory - SWNE without SNN weighting
Path3
Path4
Gene418
Gene6053
Path1
Path2
i
Trajectory - PCA
PC2
PC1
Path3
Path1
Path4
Path2
j
Trajectory - PCA
PC3
PC1
Path3
Path2
Path1
Path4
k
Trajectory - diffusion maps
Dmaps2
Dmaps1
Path1
Path3
Path2
Path4
l
Trajectory - Isomap
Isomap2
Isomap1
Path2
Path1
Path4
Path3
m
Trajectory - Multidimensional scaling
LLE2
LLE1
Path3
Path1
Path4
Path2
n
Trajectory - Locally linear embedding
MDS2
MDS1
Path3
Path1
Path4
Path2

Figure S2: **(a)** Discrete simulation – SWNE without SNN weighting. **(b)** Discrete simulation – PC1 vs PC2. **(c)** Discrete simulation – PC2 vs PC3. **(d)** Discrete simulation – Diffusion maps. **(e)** Discrete simulation – Isomap. **(f)** Discrete simulation – Locally linear embedding. **(g)** Discrete simulation – Multidimensional scaling. **(h)** Trajectory simulation – SWNE without SNN weighting. **(i)** Trajectory simulation – PC1 vs PC2. **(j)** Trajectory simulation – PC2 vs PC3. **(k)** Trajectory simulation – Diffusion maps. **(l)** Trajectory simulation – Isomap. **(m)** Trajectory simulation – Locally linear embedding. **(n)** Trajectory simulation – Multidimensional scaling.

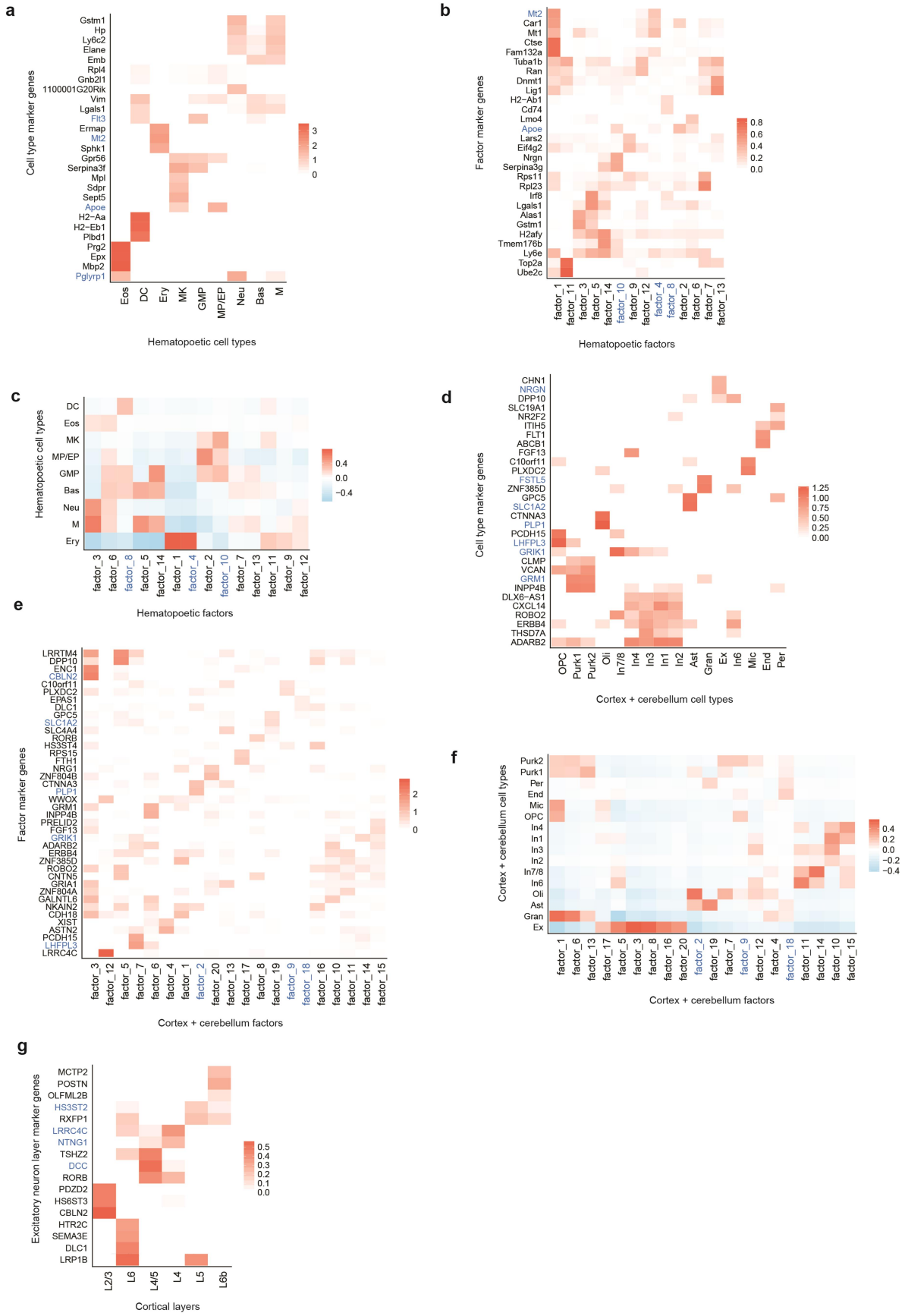
a
Cell type marker genes
Gstm1
Hp
Ly6c2
Elane
Emb
Rpl4
Gnb2l1
1100001G20Rik
Vim
Lgals1
Flt3
Ermap
Mt2
Sphk1
Gpr56
Serpina3f
Mpl
Sdpr
Sept5
Apoe
H2−Aa
H2−Eb1
Plbd1
Prg2
Epx
Mbp2
Pglyrp1
Eos
DC
Ery
MK
GMP
MP/EP
Neu
Bas
M
Hematopoetic cell types
b
Factor marker genes
Mt2
Car1
Mt1
Ctse
Fam132a
Tuba1b
Ran
Dnmt1
Lig1
H2−Ab1
Cd74
Lmo4
Apoe
Lars2
Eif4g2
Nrgn
Serpina3g
Rps11
Rpl23
Irf8
Lgals1
Alas1
Gstm1
H2afy
Tmem176b
Ly6e
Top2a
Ube2c
factor_1
factor_11
factor_3
factor_5
factor_14
factor_10
factor_9
factor_12
factor_4
factor_8
factor_2
factor_6
factor_7
factor_13
Hematopoetic factors
c
Hematopoetic cell types
DC
Eos
MK
MP/EP
GMP
Bas
Neu
M
Ery
factor_3
factor_6
factor_8
factor_5
factor_14
factor_1
factor_4
factor_2
factor_10
factor_7
factor_13
factor_11
factor_9
factor_12
Hematopoetic factors
d
Cell type marker genes
CHN1
NRGN
DPP10
SLC19A1
NR2F2
ITIH5
FLT1
ABCB1
FGF13
C10orf11
PLXDC2
FSTL5
ZNF385D
GPC5
SLC1A2
CTNNA3
PLP1
PCDH15
LHFPL3
GRIK1
CLMP
VCAN
GRM1
INPP4B
DLX6−AS1
CXCL14
ROBO2
ERBB4
THSD7A
ADARB2
OPC
Purk1
Purk2
Oli
In7/8
In4
In3
In1
In2
Ast
Gran
Ex
In6
Mic
End
Per
Cortex + cerebellum cell types
e
Factor marker genes
LRRTM4
DPP10
ENC1
CBLN2
C10orf11
PLXDC2
EPAS1
DLC1
GPC5
SLC1A2
SLC4A4
RORB
HS3ST4
RPS15
FTH1
NRG1
ZNF804B
CTNNA3
PLP1
WWOX
GRM1
INPP4B
PRELID2
FGF13
GRIK1
ADARB2
ERBB4
ZNF385D
ROBO2
CNTN5
GRIA1
ZNF804A
GALNTL6
NKAIN2
CDH18
XIST
ASTN2
PCDH15
LHFPL3
LRRC4C
factor_3
factor_12
factor_5
factor_7
factor_6
factor_4
factor_1
factor_2
factor_20
factor_13
factor_17
factor_8
factor_19
factor_9
factor_18
factor_16
factor_10
factor_11
factor_14
factor_15
Cortex + cerebellum factors
f
Cortex + cerebellum cell types
Purk2
Purk1
Per
End
Mic
OPC
In4
In1
In3
In2
In7/8
In6
Oli
Ast
Gran
Ex
factor_1
factor_6
factor_13
factor_17
factor_5
factor_3
factor_8
factor_16
factor_20
factor_2
factor_19
factor_7
factor_9
factor_12
factor_4
factor_18
factor_11
factor_14
factor_10
factor_15
Cortex + cerebellum factors
g
Excitatory neuron layer marker genes
MCTP2
POSTN
OLFML2B
HS3ST2
RXFP1
LRRC4C
NTNG1
TSHZ2
DCC
RORB
PDZD2
HS6ST3
CBLN2
HTR2C
SEMA3E
DLC1
LRP1B
L2/3
L6
L4/5
L4
L5
L6b
Cortical layers

Figure S3: Genes and factors highlighted in blue are embedded in the corresponding visualization. **(a)** Log fold-change heatmap for the top cell type specific markers in the hematopoiesis dataset (**Figure 2b, Figure 2d**). **(b)** Top gene loadings heatmap for NMF factors in the hematopoiesis dataset (**Figure 2b, Figure 2d**). **(c)** Mutual information heatmap between the cell types and NMF factors in the hematopoiesis dataset (**Figure 2b, Figure 2d**). **(d)** Log fold-change heatmap for the top cell type specific markers in the cortex & cerebellum dataset (**Figure 3b**). **(e)** Top gene loadings heatmap for NMF factors in the cortex & cerebellum dataset (**Figure 3b**). **(f)** Mutual information heatmap between the cell types and NMF factors in the cortex & cerebellum dataset (**Figure 3b**). **(g)** Log fold-change heatmap for the top layer specific marker genes in the visual cortex excitatory neuron layers (**Figure 3c**).

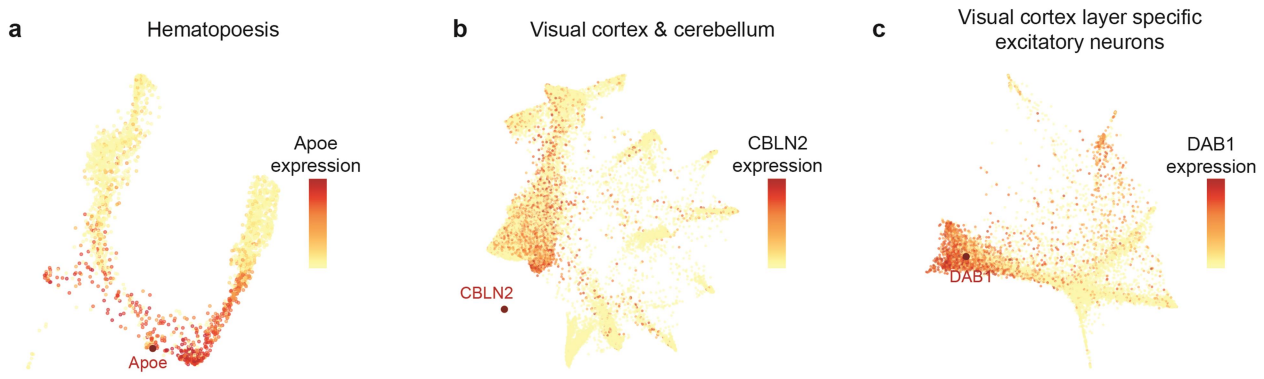

Figure S4: (a) Apoe projected onto the hematopoiesis SWNE plot with Apoe expression overlaid, validating the location of the Apoe projection. **(b)** CBLN2 projected onto the cortex & cerebellum SWNE plot with CBLN2 expression overlaid. **(c)** DAB1 projected onto the layer specific excitatory neuron SWNE plot with DAB1 expression overlaid.

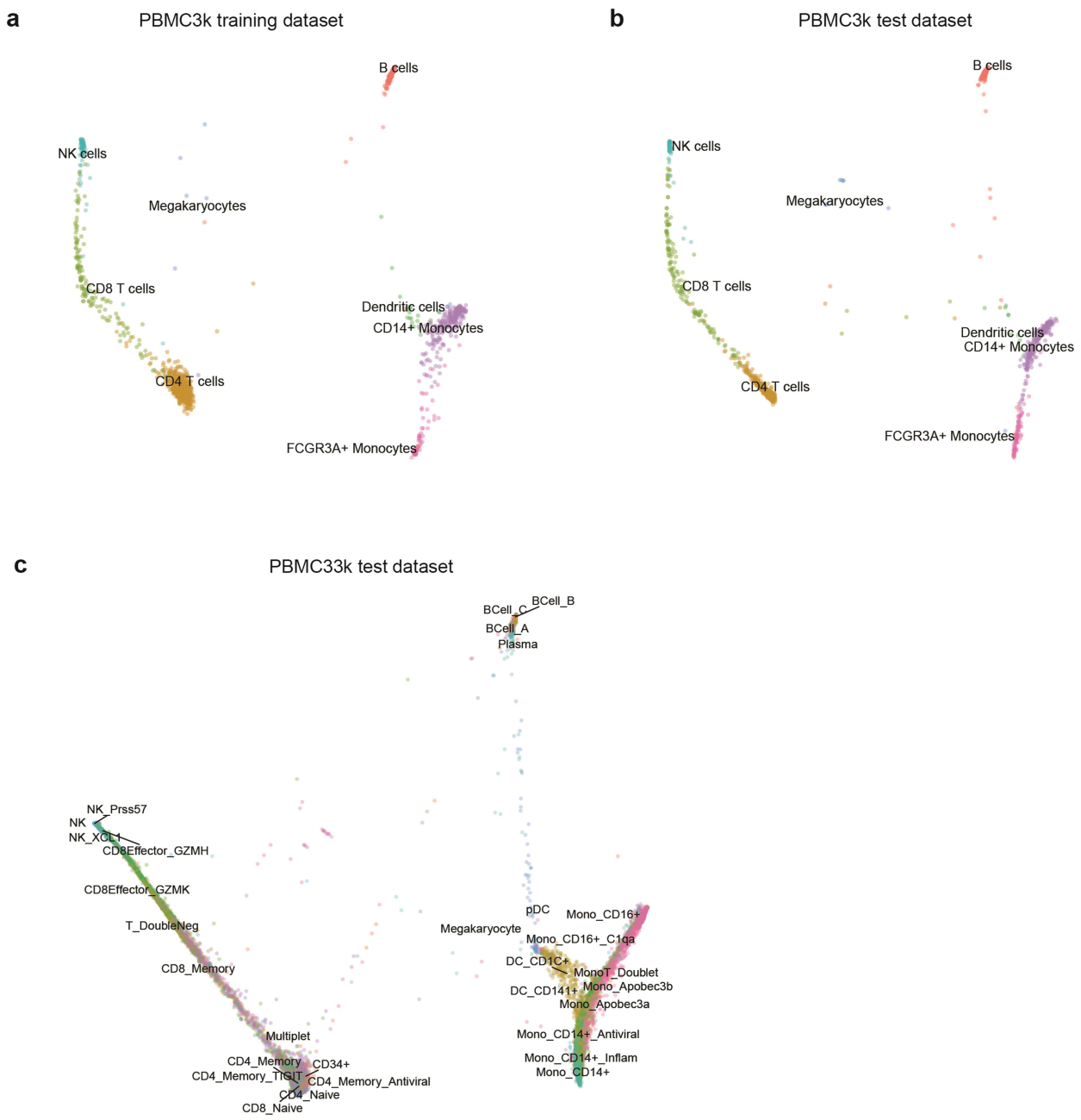

Figure S5: A 3,000 cell dataset of PBMCs was split in half to create training and test datasets. **(a)** SWNE plot of the training dataset only. **(b)** SWNE plot of the test dataset only, projected onto the training embedding. **(c)** SWNE plot of 33,000 PBMCs projected onto the training embedding.

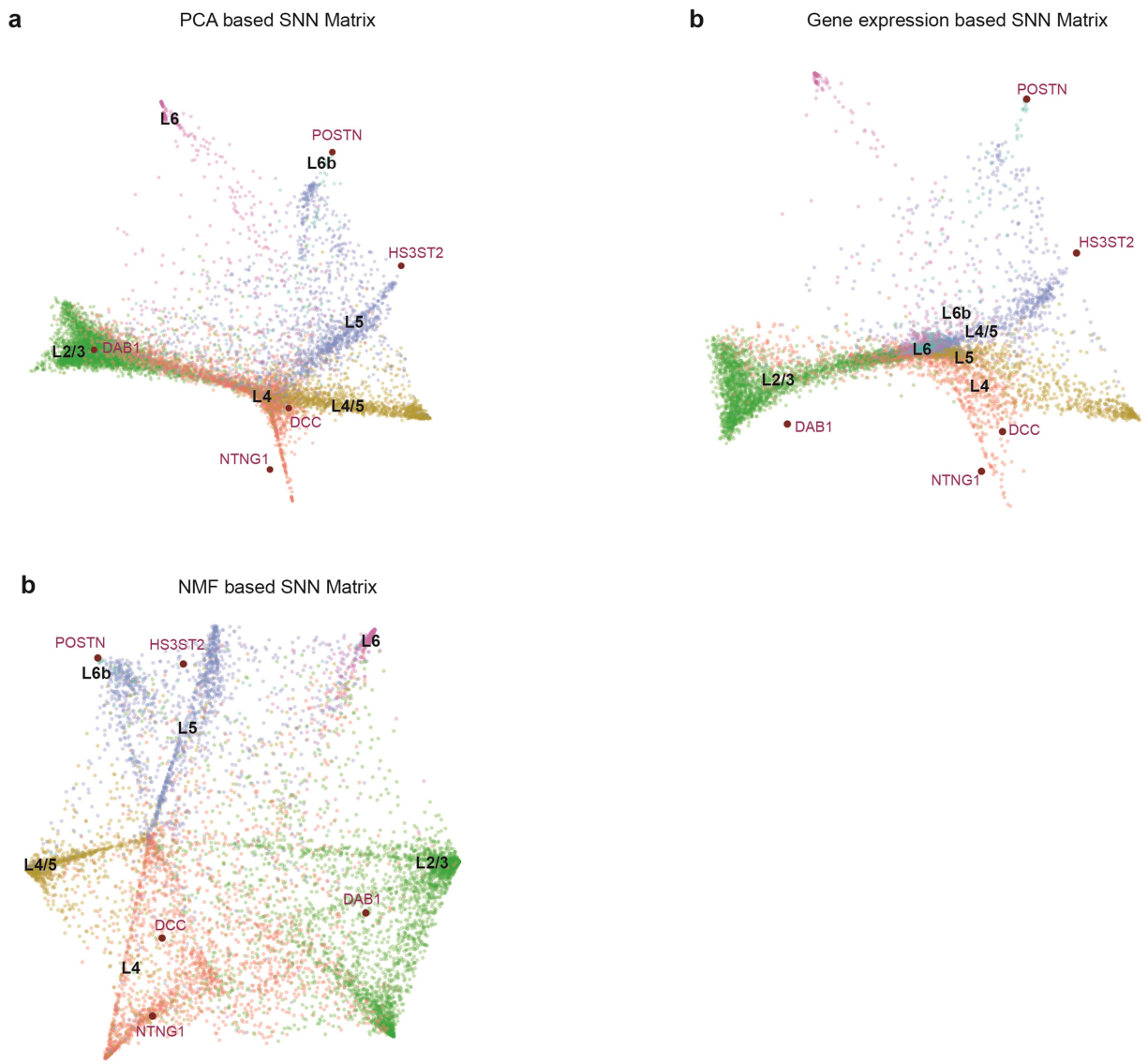

Figure S6: (a) SWNE plot of layer specific excitatory neurons with the SNN matrix constructed from PCA components (same as **Figure 3c**). **(b)** SWNE plot of layer specific excitatory neurons with the SNN matrix constructed from the gene expression matrix. **(c)** SWNE plot of layer specific excitatory neurons with the SNN matrix constructed from the NMF factors.

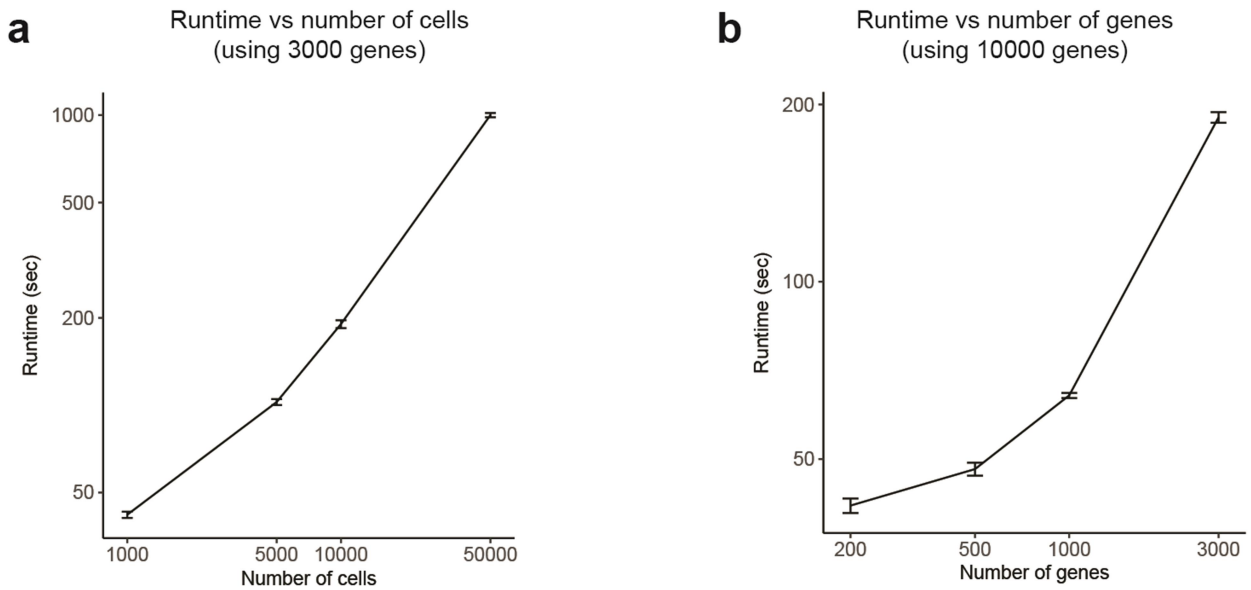

Figure S7: (a) SWNE runtime as a function of the number of cells. **(b)** SWNE runtime as a function of the number of genes.

References

1. Lake, B. *et al.* Neuronal subtypes and diversity revealed by single-nucleus RNA sequencing of the human brain. *Science (80-.).* **357,** 352–357 (2016).
2. Blue B. Lake1†, Song Chen1†, Brandon C. Sos1, 4†, Jean Fan2†, Yun Yung3, Gwendolyn E. Kaeser3, 4, Thu E. Duong1, 5, Derek Gao1, Jerold Chun3*, Peter Kharchenko2*, K. Z. Integrative single-cell analysis by transcriptional and epigenetic states in human adult brain. *Nat. Publ. Gr.* 1–3 (2017). doi:10.1101/128520
3. Tirosh, I. *et al.* Dissecting the multicellular ecosystem of metastatic melanoma by single-cell RNA-seq. *Science (80-.).* **352,** 189–196 (2016).
4. Puram, S. V *et al.* Single-Cell Transcriptomic Analysis of Primary and Metastatic Tumor Ecosystems in Head and Neck Cancer. *Cell* **172,** 1–14 (2017).
5. Trapnell, C. *et al.* The dynamics and regulators of cell fate decisions are revealed by pseudotemporal ordering of single cells. *Nat. Biotechnol.* **32,** 381–6 (2014).
6. Qiu, X. *et al.* Reversed graph embedding resolves complex single-cell trajectories. *Nat. Methods* **14,** 979–982 (2017).
7. Setty, M. *et al.* Wishbone identifies bifurcating developmental trajectories from single-cell data. *Nat. Biotechnol.* **34,** 1–14 (2016).
8. Macosko, E. Z. *et al.* Highly Parallel Genome-wide Expression Profiling of Individual Cells Using Nanoliter Droplets. *Cell* **161,** 1202–1214 (2015).
9. Klein, A. M. *et al.* Droplet Barcoding for Single-Cell Transcriptomics Applied to Embryonic Stem Cells. *Cell* **161,** 1187–1201 (2015).
10. Cao, J. *et al.* Comprehensive single cell transcriptional profiling of a multicellular organism by combinatorial indexing. *Science (80-.).* **667,** 1–35 (2017).
11. Rosenberg, A. B. *et al.* Scaling single cell transcriptomics through split pool barcoding. *Bioarxiv* (2017). doi:10.1101/105163
12. Buettner, F., Pratanwanich, N., McCarthy, D. J., Marioni, J. C. & Stegle, O. f-scLVM:

scalable and versatile factor analysis for single-cell RNA-seq. *Genome Biol.* **18,** 212 (2017).

13. Wang, B., Zhu, J., Pierson, E., Ramazzotti, D. & Batzoglou, S. Visualization and analysis of single-cell RNA-seq data by kernel-based similarity learning. *Nat. Methods* 1–6 (2017). doi:10.1038/nmeth.4207
14. Maaten, L. Van Der & Hinton, G. Visualizing Data using t-SNE. *J. Mach. Learn. Res.* **9,** 2579–2605 (2008).
15. van der Maaten, L. Accelerating t-sne using tree-based algorithms. *J. Mach. Learn. Res.* **15,** 3221–3245 (2014).
16. Abdi, H. & Williams, L. J. Principal component analysis. *Chemom. Intell. Lab. Syst.* **2,** 433–459 (2010).
17. Kruskal, J. B. Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrika* **29,** 1–27 (1964).
18. McInnes, L. & Healy, J. UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction. *arXiv* 1–18 (2018). at <http://arxiv.org/abs/1802.03426>
19. Kim, J. W. *et al.* Decomposing Oncogenic Transcriptional Signatures to Generate Maps of Divergent Cellular States. *Cell Syst.* **5,** 105–118.e9 (2017).
20. Lee, D. D. & Seung, H. S. Learning the parts of objects by non-negative matrix factorization. *Nature* **401,** 788–791 (1999).
21. Franc, V., Hlaváč, V. & Navara, M. Sequential coordinate-wise algorithm for the non-negative least squares problem. *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)* **3691 LNCS,** 407–414 (2005).
22. Houle, M. E., Kriegel, H. P., Kröger, P., Schubert, E. & Zimek, A. Can shared-neighbor distances defeat the curse of dimensionality? *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)* **6187 LNCS,** 482–500 (2010).
23. Sammon, J. W. A Nonlinear Mapping for Data Structure Analysis. *IEEE Trans. Comput.*

C-18, 401–409 (1969).

24. Zappia, L., Phipson, B. & Oshlack, A. Splatter: simulation of single-cell RNA sequencing data. *Genome Biol.* **18,** 174 (2017).

25. Rousseeuw, P. J. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *J. Comput. Appl. Math.* **20,** 53–65 (1987).

26. Paul, F. *et al.* Transcriptional Heterogeneity and Lineage Commitment in Myeloid Progenitors. *Cell* **163,** 1663–1677 (2015).

27. Satija, R., Butler, A. & Hoffman, P. Seurat: Tools for Single Cell Genomics. (2018). at <https://cran.r-project.org/package=Seurat>

28. Molyneaux, B. J., Arlotta, P., Menezes, J. R. L. & Macklis, J. D. Neuronal subtype specification in the cerebral cortex. *Nat. Rev. Neurosci.* **8,** 427–437 (2007).

29. Bernard, A. *et al.* Transcriptional Architecture of the Primate Neocortex. *Neuron* **73,** 1083–1099 (2012).

30. Hubel, D. H. *Eye, brain, and vision. Eye, brain, and vision.* (Scientific American Library/Scientific American Books, 1995).

31. Seigneur, E. & Sudhof, T. C. Cerebellins Are Differentially Expressed in Selective Subsets of Neurons Throughout the Brain. *J Comp Neurol* **525,** 3286–3311 (2017).

32. Sander, T. Allelic association of juvenile absence epilepsy with a GluR5 kainate receptor gene (GRIK1) polymorphism. *Am. J. Med. Genet. - Neuropsychiatr. Genet.* **74,** 416–421 (1997).

33. Bunge, R. P. Glial cells and the central myelin sheath. *Physiol. Rev.* **48,** 197 LP-251 (1968).

34. Trotter, J. *et al.* Dab1 Is Required for Synaptic Plasticity and Associative Learning. *J. Neurosci.* **33,** 15652–15668 (2013).

35. Lin, J. C., Ho, W. H., Gurney, A. & Rosenthal, A. The netrin-G1 ligand NGL-1 promotes the outgrowth of thalamocortical axons. *Nat. Neurosci.* **6,** 1270–1276 (2003).

36. Lin, X. & Paul C Boutros. NNLM: Fast and Versatile Non-Negative Matrix Factorization. (2016). at <https://cran.r-project.org/package=NNLM>

37. Zheng, G. X. Y. *et al.* Massively parallel digital transcriptional profiling of single cells. *Nat. Commun.* **8,** 1–12 (2017).

38. Fan, J. *et al.* Characterizing transcriptional heterogeneity through pathway and gene set overdispersion analysis. *Nat. Methods* **13,** 241–244 (2016).

39. Barkas, N. *et al.* pagoda2: A package for analyzing and interactively exploring large single-cell RNA-seq datasets. (2018). at <https://github.com/hms-dbmi/pagoda2>

40. Kim, J. W. *et al.* Characterizing genomic alterations in cancer by complementary functional associations. *Nat. Biotechnol.* **34,** 3–5 (2016).

41. Subramanian, A. *et al.* Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles. *Proc. Natl. Acad. Sci. U. S. A.* **102,** 15545–50 (2005).