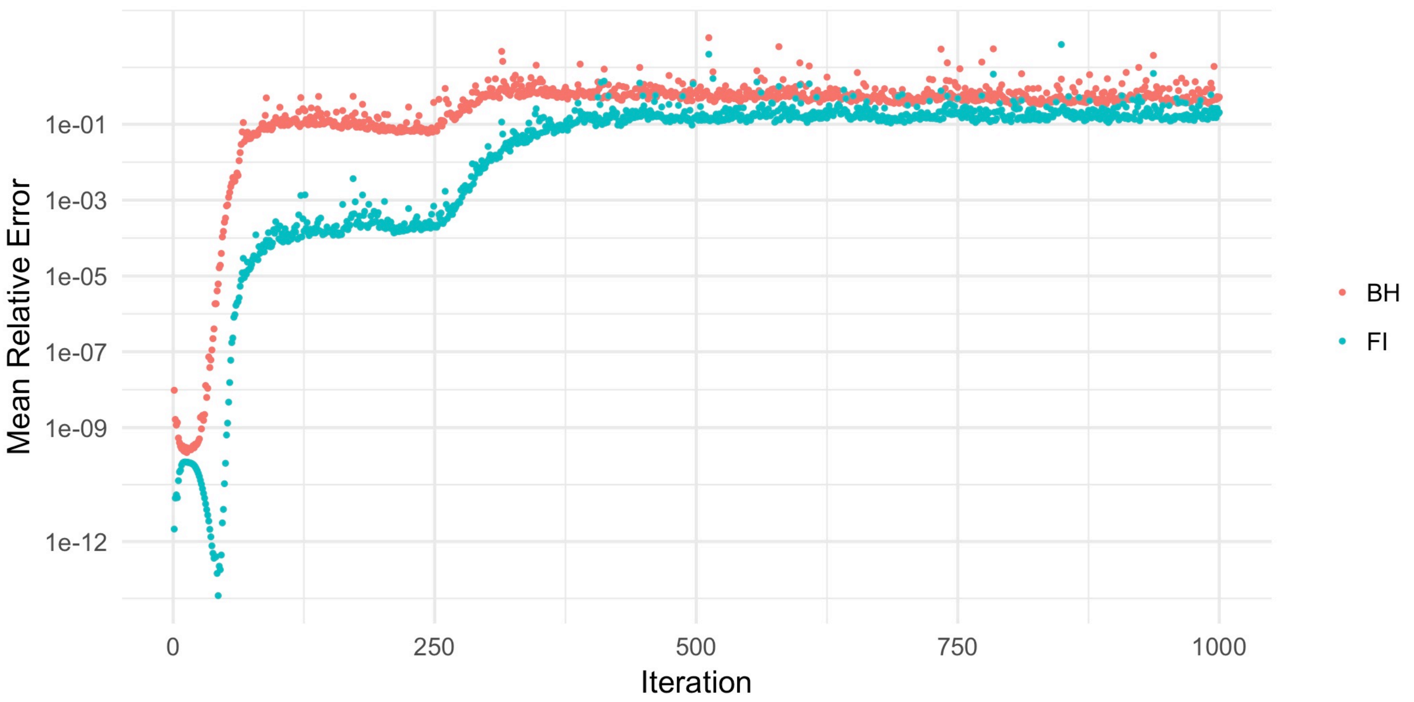
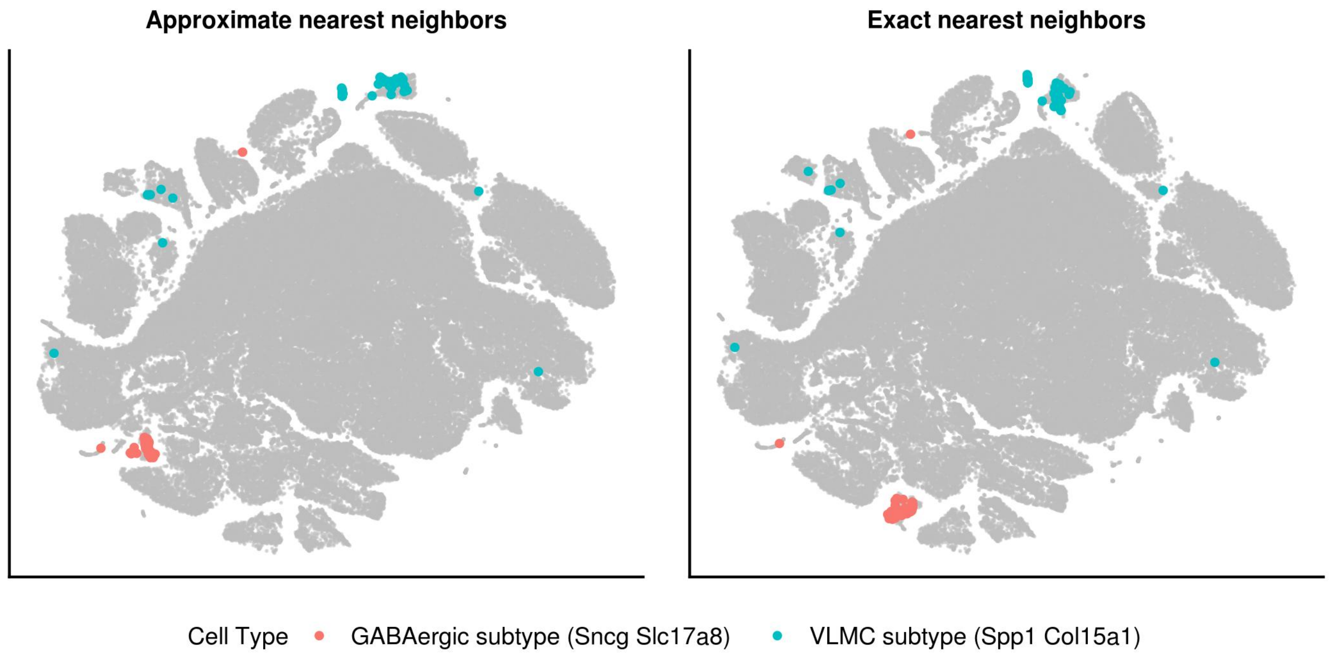


Supplementary Figure 1

Accuracy of approximation to the repulsive term.

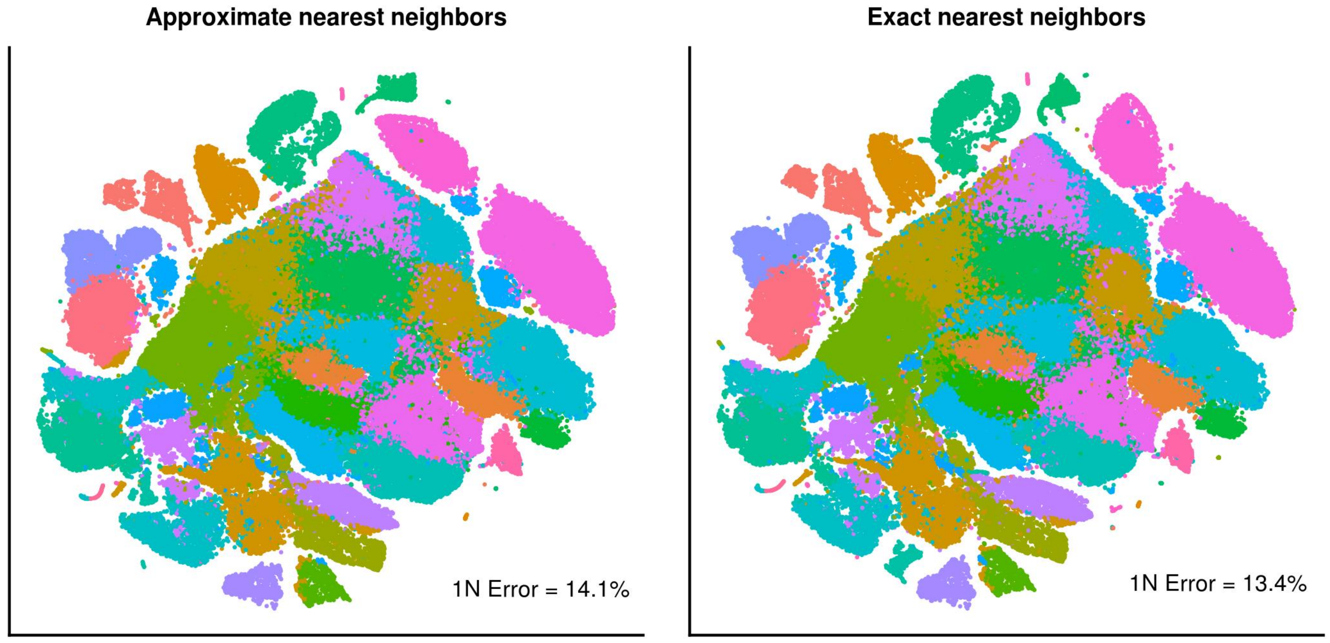
Accuracy of computing $F_{rep,i}$ using FFT-accelerated Interpolation-based (FI) t-SNE as compared to the Barnes-Hut (BH) t-SNE implementation over 1000 iterations.



Supplementary Figure 2

The populations identified in Figure 1 are apparent by embedding using exact nearest neighbors (VP trees) and approximate nearest neighbors (ANN).

1.3 million mouse brain cells are embedding using FIt-SNE with ANN and VP trees; a random 100,000 sized subset of the embedded cells is shown.



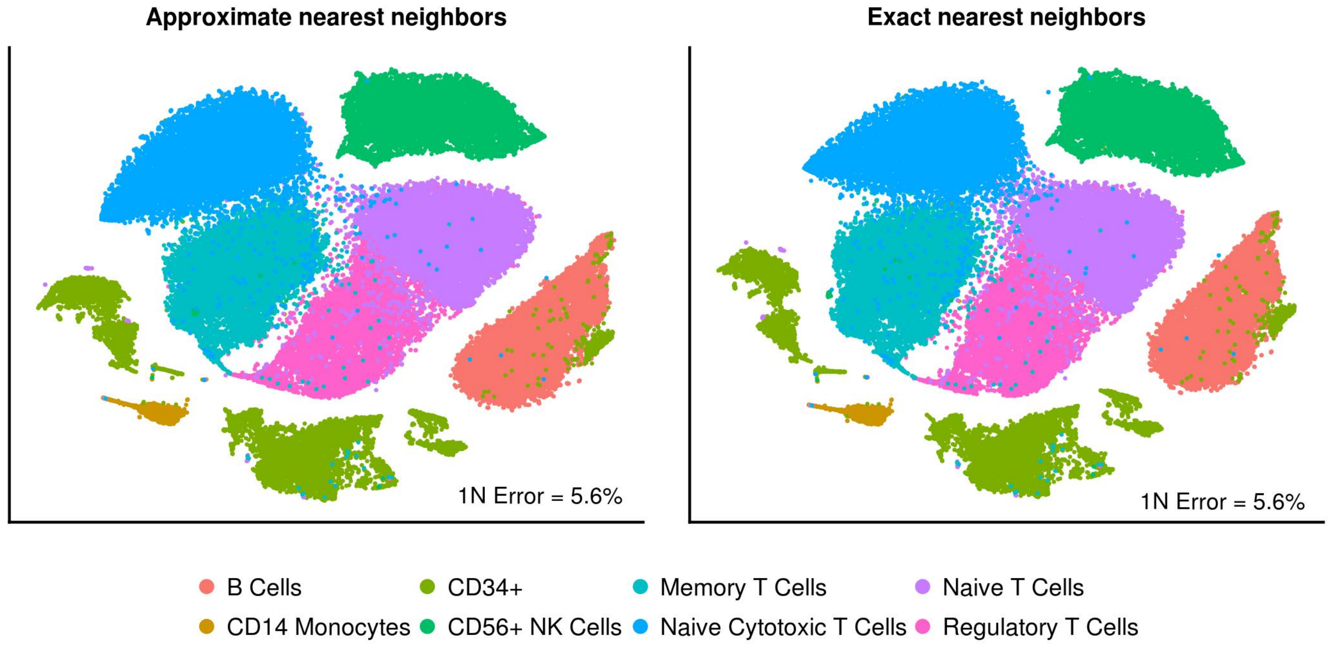
Approximate nearest neighbors

Exact nearest neighbors

Supplementary Figure 3

FIt-SNE of 1.3 million mouse brain cells using exact nearest neighbors (VP trees) vs. FIt-SNE of same cells using approximate nearest neighbors (ANNOY).

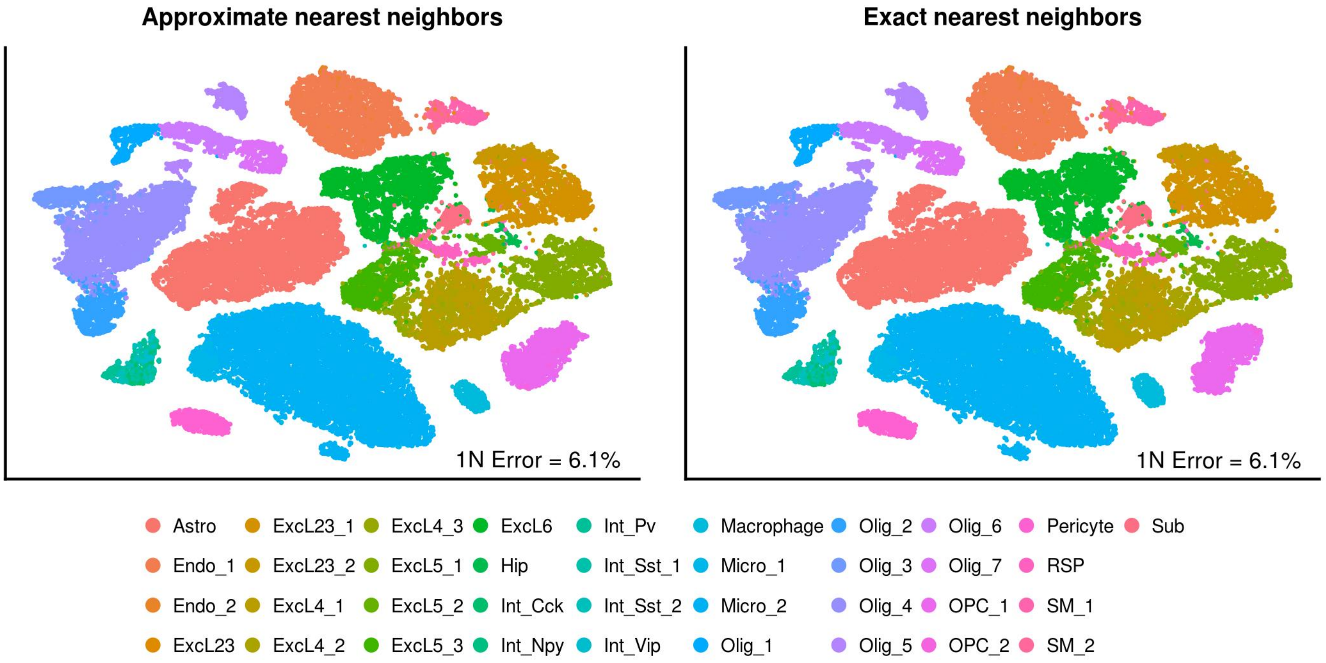
1.3 million mouse brain cells are embedding using t-SNE with ANN and VP trees; a random 100,000 sized subset of the embedded cells is shown, colored by Louvain clustering in the original high dimensional space. The 1N error is computed as the proportion of cells for which the nearest neighbor in the embedding is a member of the same cluster.



Supplementary Figure 4

FIt-SNE of purified peripheral blood monocyte cell (PBMC) populations using exact nearest neighbors (VP trees) vs. FIt-SNE of same cells using approximate nearest neighbors (ANNOY).

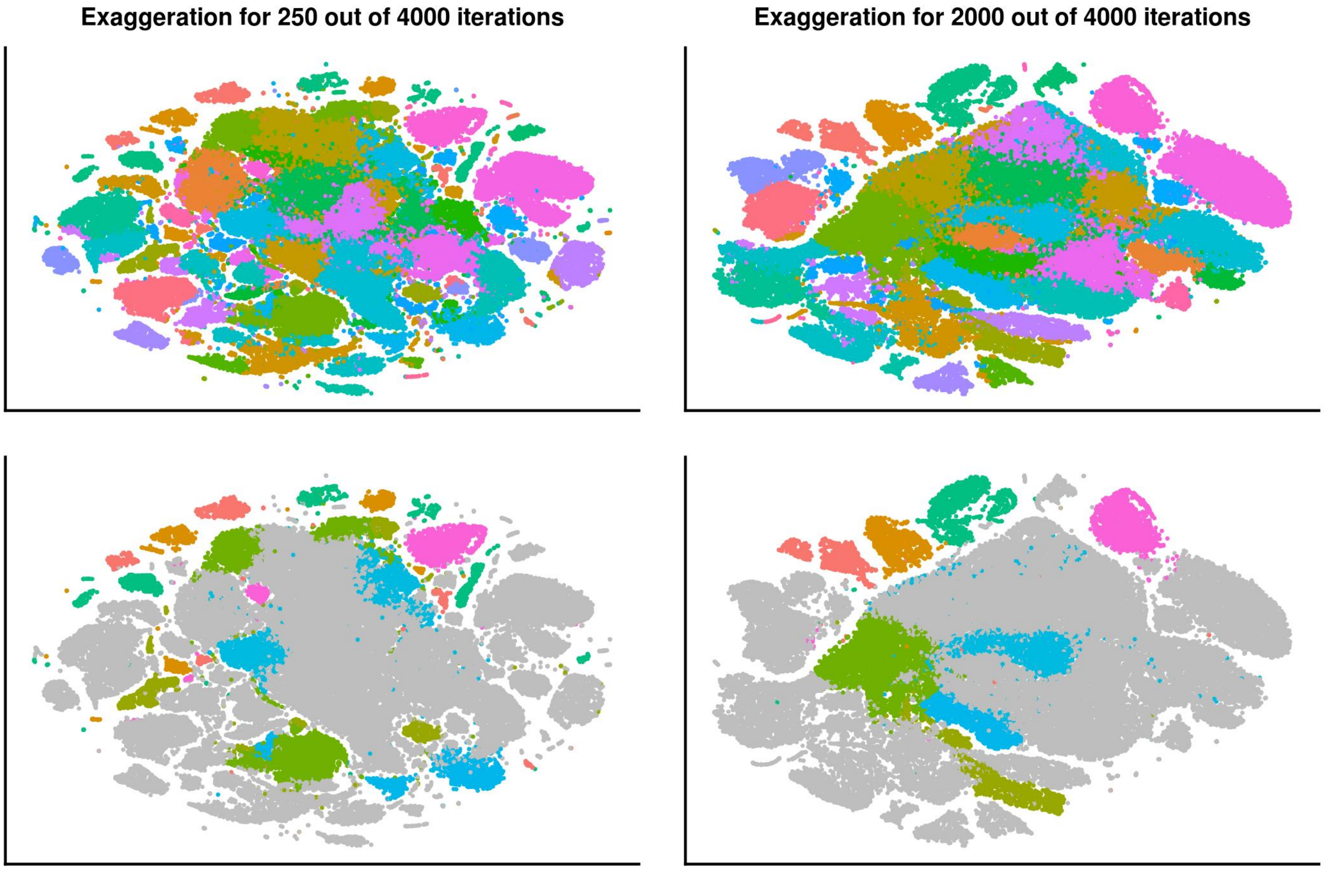
64,664 purified PBMCs of Zheng et al. (2017) are embedding using t-SNE with ANN and VP trees. The 1N error is computed as the proportion of cells for which the nearest neighbor in the embedding is a member of the same population.



Supplementary Figure 5

FIt-SNE of mouse cortical cells using exact nearest neighbors (VP trees) vs. FIt-SNE of same cells using approximate nearest neighbors (ANNOY).

48,266 cells from Hrvatin et al. (2018) are embedding using t-SNE with ANN and VP trees and labelled as the subtypes in that paper. The 1N error is computed as the proportion of cells for which the nearest neighbor in the embedding is a member of the same subtype.



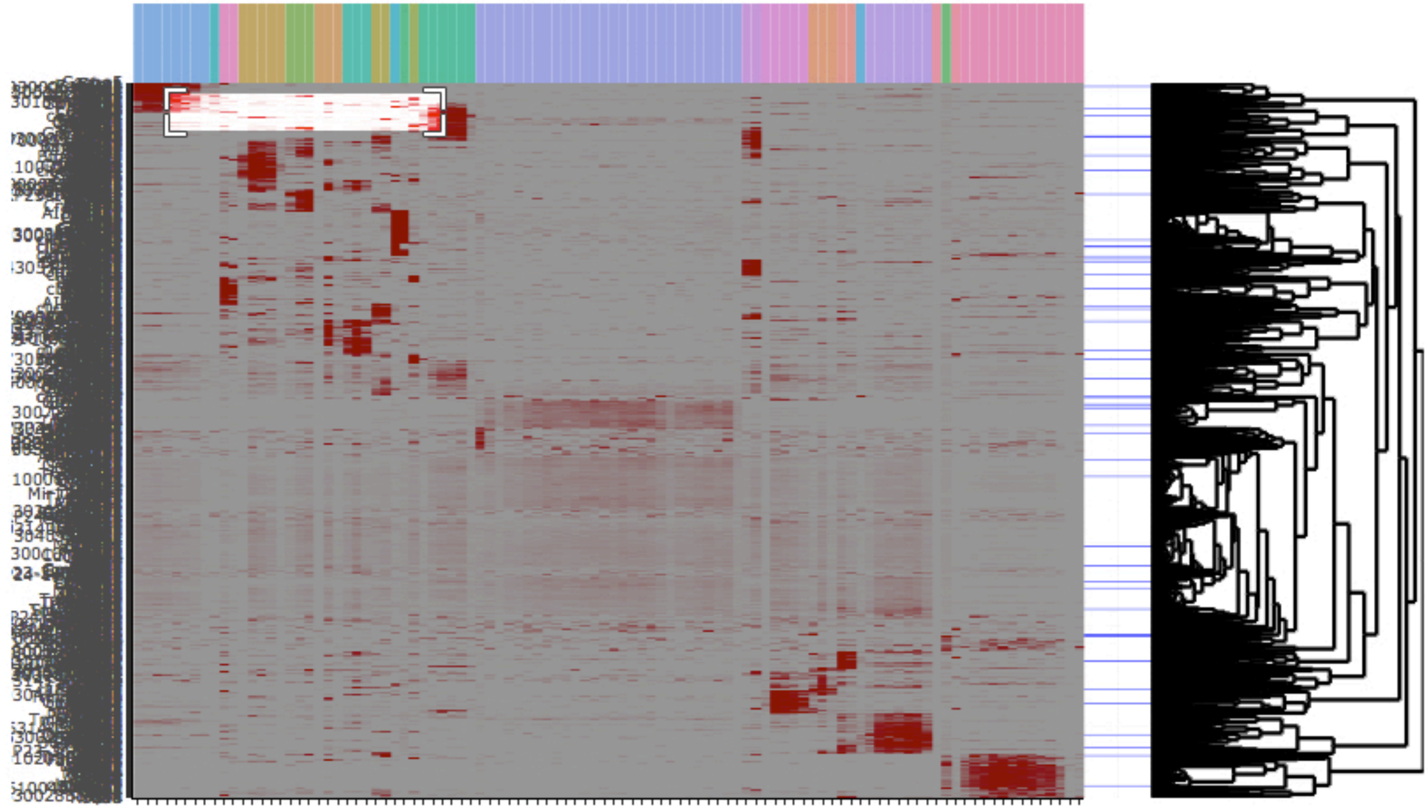
Exaggeration for 250 out of 4000 iterations

Exaggeration for 2000 out of 4000 iterations

Supplementary Figure 6

The importance of early exaggeration when embedding large datasets.

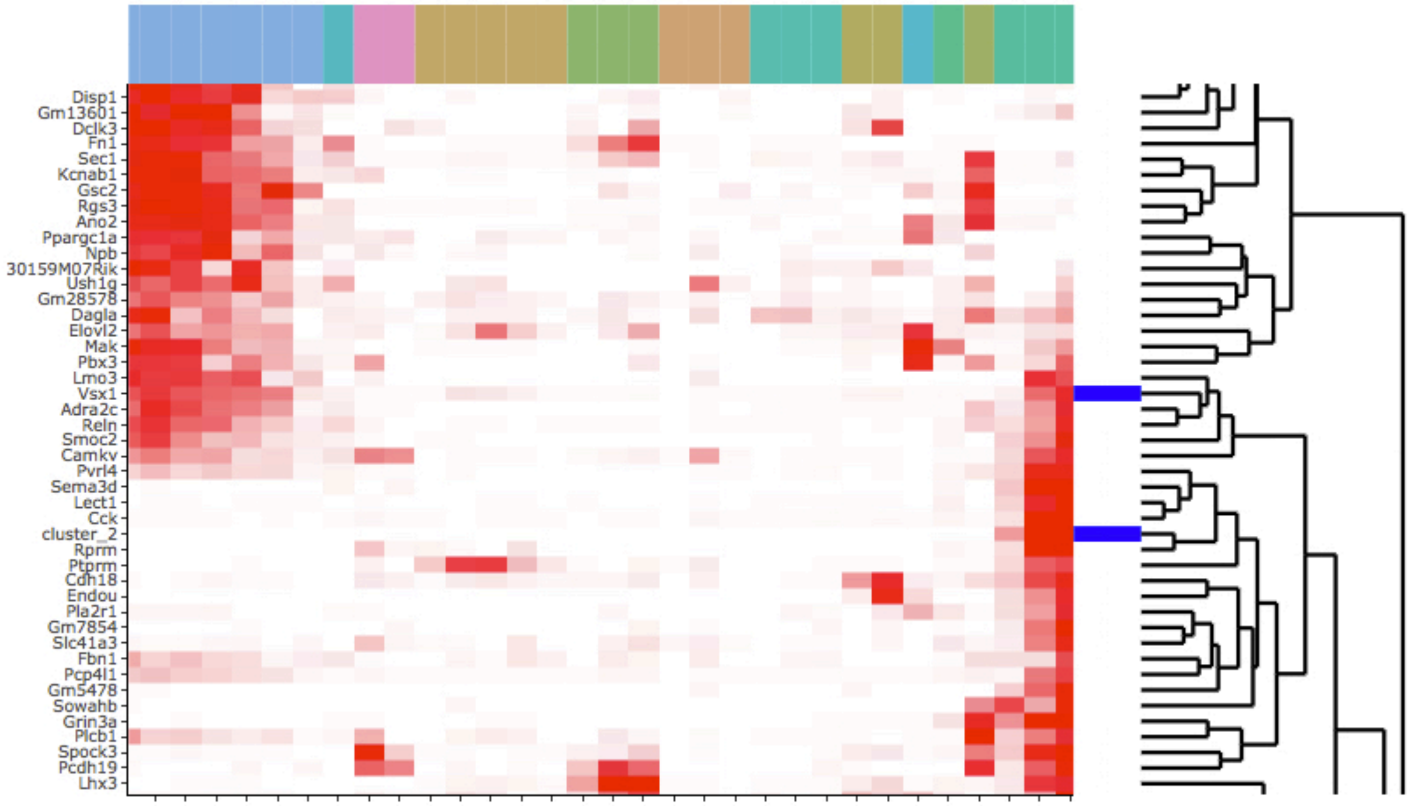
1.3 million mouse brain cells are embedded using default early exaggeration setting of 250 (left) and also embedded using setting of 2000 (right). Cells are colored by Louvain clustering in the original high dimensional space (independent of the t-SNE). Many clusters are broken up when the number of early exaggeration iterations is insufficient, e.g. the 6 clusters highlighted (bottom).



Supplementary Figure 7

t-SNE heatmap of retinal bipolar cells from Shekhar et al. (2016).

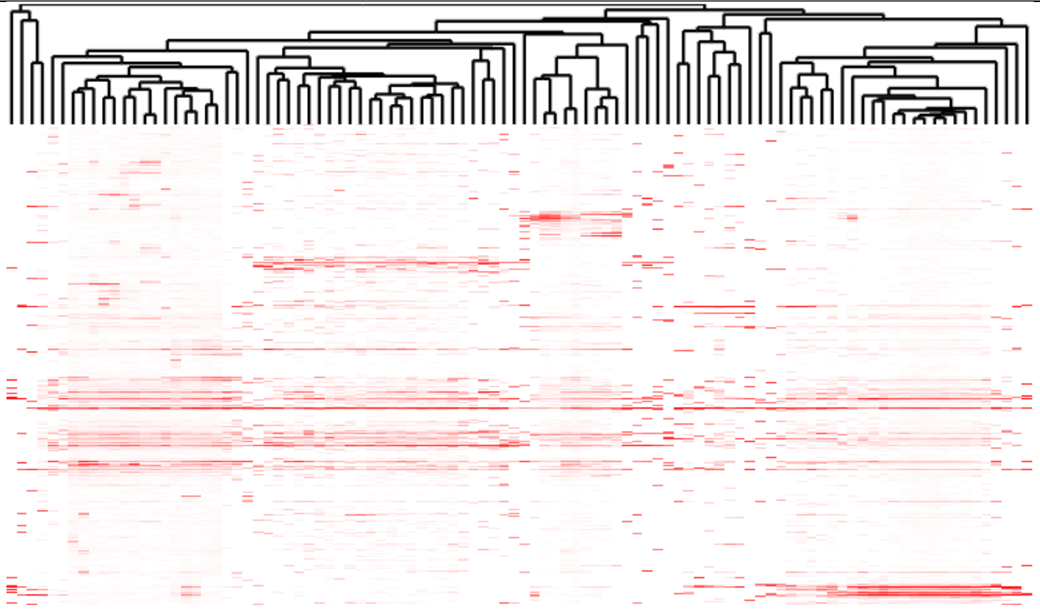
Genes presented are the 25 genes most associated with each marker gene and cluster metagene (denoted by blue). The heatmap is interactive, allowing users to zoom into a region of interest (see Supplemental Figure 8).



Supplementary Figure 8

t-SNE heatmap of retinal bipolar cells from Shekhar et al. (2016), zoomed into region of interest.

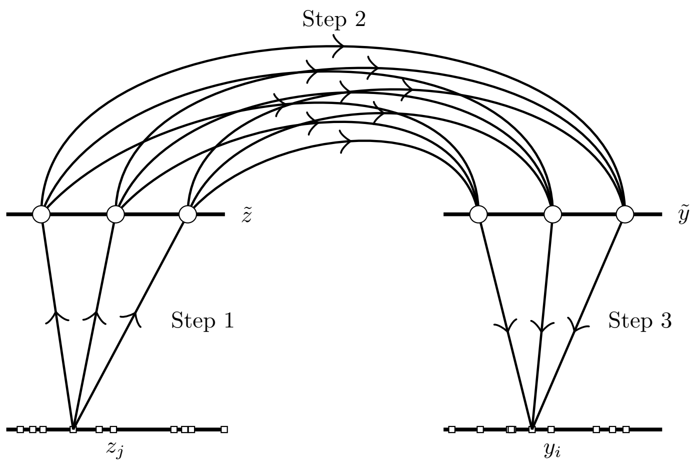
Zooming into a section of the t-SNE heatmap in Supplementary Figure 7.



Supplementary Figure 9

Standard heatmap of retinal bipolar cells from Shekhar et al. (2016).

Using the same genes (rows) in the same ordering as Figure 2e and Supplementary Figure 7, cells were clustered using hierarchical clustering (columns), for comparison to Figure 2e.



Supplementary Figure 10

An illustration of the algorithm

Both the intervals on the left are (z_0, z_0 + R), and both the intervals on the right are (y_0, y_0 + R). In the lower intervals, the white squares denote the locations z_i and y_i, and in the upper intervals the white circles indicate the locations of the equispaced nodes $\tilde{z}_\imath$ and $\tilde{z}_\jmath$. The arrows illustrate how a point z_i communicates with a point y_i.

Supplementary Table 1. Time taken for 1000 iterations of the gradient descent phase of 2D t-SNE using Barnes-Hut t-SNE (BH t-SNE) compared to our implementation (FIt-SNE), as compared on a server for a given number of points N. See Online Methods for more details.

N	**BH t-SNE**	**FIt-SNE**
10,000	1 min.	< 1 min.
100,000	26 min.	1 min.
500,000	4 hr. 12 min.	6 min.
1,000,000	8 hr. 45 min.	11 min.

Supplementary Table 2. Time taken to compute input similarities in Barnes-Hut t-SNE (vptree) compared to FIt-SNE using either multithreaded vantage-point trees (vptreeMT) or a multi-threaded approximate nearest neighbor (annMT) approach on a server for a given number of points N.

	50 Dimensions			**100 Dimensions**		
N	**vptree**	**vptreeMT**	**annMT**	**vptree**	**vptreeMT**	**annMT**
10,000	< 1 min.	< 1 min.	< 1 min.	< 1 min.	< 1 min.	< 1 min.
100,000	2 min.	< 1 min.	< 1 min.	3 min.	< 1 min.	< 1 min.
500,000	45 min.	6 min.	3 min.	1 hr. 8 min.	13 min.	3 min.
1,000,000	3 hr. 23 min.	25 min.	5 min.	5 hr. and 7 min.	39 min.	9 min.