Semisoft clustering of single-cell data

Lingxue Zhu[a], Jing Lei[a], Lambertus Klei[b], Bernie Devlin[b], and Kathryn Roeder[a,c,1]

[a]Department of Statistics and Data Science, Carnegie Mellon University, Pittsburgh, PA 15213; [b]Department of Psychiatry, University of Pittsburgh School of Medicine, Pittsburgh, PA 15213; and [c]Department of Computational Biology, Carnegie Mellon University, Pittsburgh, PA 15213

Edited by Haiyan Huang, University of California, Berkeley, CA, and accepted by Editorial Board Member Charles F. Stevens November 27, 2018 (received for review October 24, 2018)

Motivated by the dynamics of development, in which cells of recognizable types, or pure cell types, transition into other types over time, we propose a method of semisoft clustering that can classify both pure and intermediate cell types from data on gene expression from individual cells. Called semisoft clustering with pure cells (SOUP), this algorithm reveals the clustering structure for both pure cells and transitional cells with soft memberships. SOUP involves a two-step process: Identify the set of pure cells and then estimate a membership matrix. To find pure cells, SOUP uses the special block structure in the expression similarity matrix. Once pure cells are identified, they provide the key information from which the membership matrix can be computed. By modeling cells as a continuous mixture of *K* discrete types we obtain more parsimonious results than obtained with standard clustering algorithms. Moreover, using soft membership estimates of cell type cluster centers leads to better estimates of developmental trajectories. The strong performance of SOUP is documented via simulation studies, which show its robustness to violations of modeling assumptions. The advantages of SOUP are illustrated by analyses of two independent datasets of gene expression from a large number of cells from fetal brain.

single-cell RNA-seq | soft clustering | developmental trajectories | neuronal lineages

Development often involves pluripotent cells transitioning into other cell types, sometimes in a series of stages. For example, early in development of the cerebral cortex (1), one progression begins with neuroepithelial cells differentiating to apical progenitors, which can develop into basal progenitors, which will transition to neurons. Moreover, there are diverse classes of neurons, some arising from distinct types of progenitor cells (2, 3). By the human midfetal period there are myriad cell types and the foundations of typical and atypical neurodevelopment are already established (4). While the challenges for neurobiology in this setting are obvious, some of them could be alleviated by statistical methods that permit cells to be classified into pure or transitional types. We develop such a method here. Similar scenarios arise with the development of bone-marrow–derived immune cells, cancer cells, and disease cells (5); hence we envision broad applicability of the proposed modeling tools.

Different types of cells have different transcriptomes or gene expression profiles (4). Thus, they can be identified by these profiles (6), especially by expression of certain genes that tend to have cell-specific expression (marker genes). Characterization of these profiles has recently been facilitated by single-cell RNA sequencing (scRNA-seq) techniques (7, 8), which seek to quantify expression for all genes in the genome. For single cells, the number of possible sequence reads is limited and therefore the data can be noisy. Nonetheless, cells of the same and different cell types can be successfully clustered using these data (6, 9–12).

What is missing from the clustering toolbox is a method that recognizes development, with both pure type and transitional cells. In this paper, we develop an efficient algorithm for semisoft clustering with pure cells (SOUP). SOUP intelligently recovers the set of pure cells by exploiting the block structures in a cell–cell similarity matrix and also estimates the soft memberships for transitional cells. We also incorporate a gene selection procedure to identify the informative genes for clustering. This selection procedure is shown to retain fine-scaled clustering structures in the data and substantially enhances clustering accuracy. Incorporating soft-clustering results into methods that estimate developmental trajectories yields less biased estimates of developmental courses.

We first document the performance of SOUP via extensive simulations. These show that SOUP performs well in a wide range of contexts; it is superior to natural competitors for soft clustering; and it compares quite well, if not better, than other clustering methods in settings ideal for hard clustering. Next, we apply it to two single-cell datasets from fetal development of the prefrontal cortex of the human brain. In both settings SOUP produces results congruent with known features of fetal development.

Results

Model Overview. Suppose we observe the expression levels of n cells measured on p genes and let $X \in \mathbb{R}^{n \times p}$ be the cell-by-gene expression matrix. Consider the problem of semisoft clustering, where we expect the existence of both (*i*) pure cells, each belonging to a single cluster and requiring a hard cluster assignment, and (*ii*) mixed cells (transitional cells) that are transitioning between two or more cell types and hence should obtain soft assignments. With K distinct cell types, to represent the soft membership, let $\Theta \in {\mathbb{R}_+}^{n \times K}$ be a nonnegative membership matrix. Each row of the membership matrix,

Significance

Growth typically involves differentiation of cells from progenitors into more specialized descendants, often involving lineages of pure and transitional cells to achieve final form. Recent technology has enabled estimation of gene expression profiles of single cells and these profiles theoretically differentiate pure cell types. What is missing from the analytical toolbox is an efficient technique to classify pure and transitional cells from their profiles. Here we propose semisoft clustering with pure cells (SOUP). This algorithm performs well in the hard-clustering problem for pure cell types and excels at identifying transitional cells with soft memberships. Moreover, SOUP provides an estimate of the developmental trajectories based on the estimated cell type membership that naturally adapts to cells in transition.

Author contributions: L.Z., J.L., B.D., and K.R. designed research; L.Z., J.L., B.D., and K.R. performed research; L.Z., J.L., and K.R. contributed new reagents/analytic tools; L.Z., L.K., B.D., and K.R. analyzed data; and L.Z., J.L., L.K., B.D., and K.R. wrote the paper.

The authors declare no conflict of interest.

This article is a PNAS Direct Submission. H.H. is a guest editor invited by the Editorial Board.

Data deposition: The R package of SOUP is available on GitHub (https://github.com/lingxuez/SOUPR).

[1]To whom correspondence should be addressed. Email: roeder@andrew.cmu.edu.

This article contains supporting information online at www.pnas.org/lookup/suppl/doi:10.1073/pnas.1817715116/-/DCSupplemental.

Published online December 26, 2018.

$\Theta_i := (\theta_{i1}, \ldots, \theta_{iK})$, contains nonnegative numbers that sum to one, representing the proportions of cell i in K clusters. In particular, a pure cell in type k has $\theta_{ik} = 1$ and zeros elsewhere.

Let $C \in \mathbb{R}^{p \times K}$ denote the cluster centers, which represent the expected gene expression for each pure cell type. When a cell is developing or transitioning from one category to another, it may exhibit properties of both subcategories, which is naturally viewed as a combination of the two cluster centers. Weights in the membership matrix reflect the stage (early or late) of the transition. Here we formulate a simple probability model that is convenient for analysis and highly robust to expected violations of the assumptions. Let

$$X = \Theta C^T + E, \qquad [1]$$

where $E \in \mathbb{R}^{n \times p}$ is a zero-mean noise matrix with $\mathbb{E}(EE^T) = \sigma^2 I$. It follows directly that the cell–cell similarity matrix takes a convenient form,

$$A := \mathbb{E}\left[XX^T\right] = \Theta Z \Theta^T + \sigma^2 I, \qquad [2]$$

where $Z = C^T C \in \mathbb{R}^{K \times K}$ represents the association among different cell types.

In practice, many genes will not follow the developmental trajectory described by Eq. **1**; however, it is expected that the expression of many marker genes and other highly informative genes will transition smoothly between cluster centers during development (for example, the genes featured in ref. 13). In particular, one can empirically check the plausibility of Eq. **1** for marker genes; see *Case Studies* below for details. Moreover, because SOUP's inferences are based on the empirical cell–cell similarity matrix $\hat{A}$, it is sufficient that $\hat{A}$ approximately follows the form specified in Eq. **2**, a weaker assumption than Eq. **1**. Indeed, similar assumptions are implicit in many algorithms that estimate developmental trajectories (14–17). Gene expression is also likely to have nonconstant variance, depending on gene and cell type. However, our pure cell search algorithm does not depend on the diagonal entries of A, and our estimate of Θ is based on spectral decomposition of A, so the method remains robust to moderate fluctuation of diagonal entries of A unless the magnitude of noise is unrealistically large.

As a graphical illustration of the SOUP model, we simulate an example with a developmental trajectory of type1 → type2 → type3. A fraction of the genes were chosen to have differential expression across cell types, and of these a fraction change nonlinearly between cell types (Fig. 1*A*). Regardless of the violations of Eq. **1**, the cells depict a smooth transition between cell types (Fig. 1*B*).

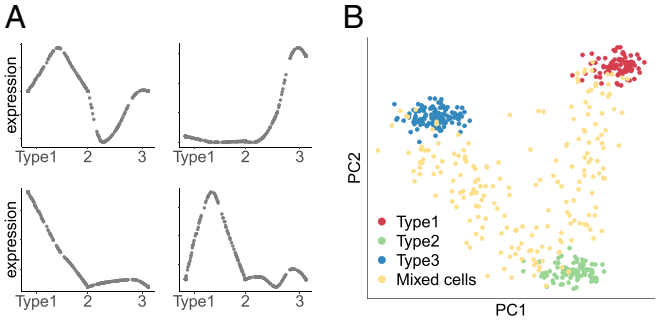

Fig. 1. Illustration of the SOUP framework for three cell types with simulated developmental trajectory of type1 → type2 → type3. (*A*) Example of four differentially expressed genes along the developmental trajectory, with potentially nonlinear differentiation patterns. (*B*) Simulation of 300 pure cells and 200 mixed cells, visualized in the leading principal component space.

Similar factorization problems to that of Eq. **2** have appeared in previous literature under different settings. The most popular are the mixed-membership stochastic block model (MMSB) (18) and topic modeling (for example, refs. 19–21). However, it is nontrivial to extend these algorithms to our scenario. A similar formulation also appeared in nonnegative matrix factorization (NMF), where nonnegative rank-K matrices Θ and C are estimated such that $X \approx \Theta C^T$, for example, by minimizing the Euclidean distance (22). However, traditional NMF differs from our setting in two important ways: (*i*) The NMF problem is nonidentifiable without introducing nontrivial assumptions, and (*ii*) SOUP does not rely on the nonnegativeness of C, which makes it more broadly applicable to scRNA-seq data after certain preprocessing steps, such as batch-effect corrections, which can result in negative values. Recent work in ref. 23 considered the problem of overlapping variable clustering under latent factor models. Despite the different setup, the model comes down to a problem similar to Eq. **2**, and the authors proposed the latent-model approach to overlapping clustering (LOVE) algorithm to recover the variable allocation matrix, which can be treated as a generalized membership matrix. LOVE consists of two steps: (*i*) finding pure variables and (*ii*) estimating the allocations of the remaining overlapping variables. Both steps rely on a critical tuning parameter that corresponds to the noise level, which can be estimated using a cross-validation procedure. When we applied the LOVE algorithm to our single-cell datasets, however, we found it sensitive to noise, leading to poor performance (*SI Appendix*). Nonetheless, inspired by the LOVE algorithm, SOUP works in a similar two-step manner, while adopting different approaches in both parts. Most importantly, SOUP parameters are intuitive to set, and it is illustrated to have robust performance in both simulations and real data.

SOUP algorithm. The SOUP algorithm involves finding the set of pure cells and then estimating Θ. Pure cells play a critical role in this problem. Intuitively, they provide valuable information from which to recover the cluster centers, which further guides the estimation of Θ for the mixed cells. In fact, it has been shown in ref. 23 that the existence of pure cells is essential for model (2) in ref. 23 to be identifiable, and we restate the theorem below.

Theorem 1 (Identifiability). *Model* (2) *is identifiable up to the permutation of labels, if* (*a*) Θ *is a membership matrix*; (*b*) *there exist at least two pure cells per cluster*; *and* (*c*) Z *is full rank.*

These assumptions are minimal, because in most single-cell datasets, it is natural to expect the existence of at least a few pure cells in each type, and Z usually has larger entries along the diagonal.

The details of SOUP are presented in *Methods* and *SI Appendix*. As an overview, to recover the pure cells the key is to notice the special block structure formed by the pure cells in the similarity matrix A. SOUP exploits this structure to calculate a purity score for each cell. This calculation requires two tuning parameters: ϵ, the fraction of most similar neighbors to be examined for each cell, and γ, the fraction of cells declared as pure after ranking the purity scores. After selection, the pure cells are partitioned into K clusters, by standard clustering algorithms such as K-means. The choice of K is guided by empirical investigations, including a sample splitting procedure (*SI Appendix*).

To recover Θ, consider the top K eigenvectors of the similarity matrix A, denoted as $V \in \mathbb{R}^{n \times K}$. There exists a matrix $Q^* \in \mathbb{R}^{K \times K}$, such that $\Theta = VQ^*$. If we have identified the set of pure cells $\mathcal{I}$ and their partitions $\{\mathcal{I}_k\}$, we essentially know their memberships, $\Theta_{\mathcal{I}}$.. Then it is straightforward to recover

the desired Q^* from the submatrix $\Theta_{\mathcal{I}\cdot} = V_{\mathcal{I}\cdot}Q^*$, which further recovers the full membership matrix $\Theta = VQ^*$ (*Theorem 2*). In practice, we plug in the sample similarity matrix $\hat{A}$ to obtain an estimate $\hat{\Theta}$, and we can further estimate $\hat{C}$ by minimizing $\|X - \hat{\Theta}C^T\|_F^2$.

Theorem 2 (SOUP clustering). *In model* (2), *let* $V \in \mathbb{R}^{n\times K}$ *be the top* K *eigenvectors of* A *and* $\mathcal{I}$ *be the set of pure cells. Under the same assumptions as those of Theorem 1, the optimization problem*

$$\min_{Q\in\mathbb{R}^{K\times K}} \|\Theta_{\mathcal{I}\cdot} - V_{\mathcal{I}\cdot}Q\|_F^2 \qquad [3]$$

has a unique solution Q^* *such that* $\Theta = VQ^*$.

The majority membership probability is $\max_j \theta_{ij}$, and the majority type is the class that achieves the maximum.

Developmental trajectories. SOUP provides two outcomes not available from hard-clustering procedures such as in refs. 24–26: soft membership probabilities, $\hat{\Theta}$, and soft cluster centers, $\hat{C}$. The next step is to estimate one or more developmental trajectories from the cells. Various algorithms have been developed that can identify multibranching developmental trajectories in single-cell data (14–17, 27), and one successful direction is to estimate the lineages from cell clusters, usually by fitting a minimum spanning tree (MST) to the cluster centers in a low-dimensional space (15–17) and then fitting a smooth branching curve to the inferred lineages (17). It is straightforward to extend this idea to SOUP, where we identify the MST using SOUP-estimated soft cluster centers, $\hat{C}$. Following the common practice, $\hat{C}$ can be projected to a low-dimensional space for MST estimation. Notably, soft clusters provide an alternative input for Slingshot (17), which yields more refined insights into development by providing less-biased estimates of cluster centers in developing cells.

Performance Evaluation.

Simulations. There are no direct competitors of SOUP for semisoft clustering in the single-cell literature, and here we use the following three candidates for comparison: NMF, where we use the standard algorithm from ref. 22 to solve for nonnegative $(\hat{\Theta}, \hat{C})$; Fuzzy C-Means (FC) (28), a generic soft-clustering algorithm; and DIMMSC (Dirichlet mixture model for clustering droplet-based single cell) (29), a probabilistic clustering algorithm for single-cell data based on Dirichlet mixture models. All algorithms are applied to the log-transformed data, except for DIMMSC, which is developed under a multinomial model for count data. NMF can be applied to the raw count data as well, which usually has slightly worse performance.

Although SOUP is derived from a linear model, it is robust and applicable to general scRNA-seq data. To illustrate this, we use the splat algorithm in the Splatter R package (30) to conduct simulations. Splatter is a single-cell simulation framework that generates synthetic scRNA-seq data with hyperparameters estimated from a real dataset. The algorithm incorporates expected violations of the model assumptions (*SI Appendix*). We simulate 500 genes and 300 pure cells from four clusters. Mixed cells are simulated along a developmental path and the number varies from 100 to 500.

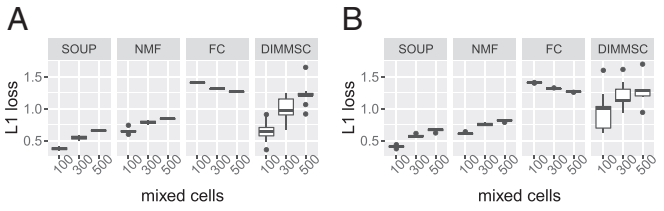

Fig. 2. Boxplot of the average L_1 losses of estimating Θ in 10 repetitions. Using the splat algorithm in the Splatter package, expression levels of 500 genes are simulated for 300 pure cells from four clusters, as well as {100, 300, 500} mixed cells along the trajectory of type1 → type2 → {type3 or type4}. (*A*) Without dropout. (*B*) With dropout.

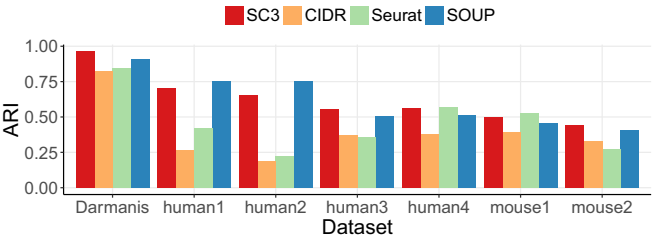

Fig. 3. ARI on seven labeled public datasets (6, 10), using (*i*) SC3, (*ii*) CIDR, (*iii*) Seurat, and (*iv*) SOUP.

For comparable evaluation across different scenarios with different cell numbers, we present the average L_1 loss per cell, i.e., $\frac{1}{n}\|\hat{\Theta} - \Theta\|_1$, where $\|\cdot\|_1$ is the usual vector L_1 norm after vectorization. SOUP achieves the best performance under all scenarios (Fig. 2*A*). In particular, with 100, 300, and 500 mixed cells, the true proportions of pure cells in the data are 75%, 50%, and 37.5%, respectively. Note that we always set $\gamma = 0.5$ for SOUP, which represents a prior guess of 50% pure cells, and we see that SOUP remains stable even when the given γ clearly overestimates or underestimates the pure proportion.

One of the biggest challenges in single-cell data is the existence of dropouts (31), where the mRNA for a gene fails to be amplified before sequencing, producing a "false" zero in the observed data. We see that SOUP remains robust and outperforms all other algorithms (Fig. 2*B*).

SOUP as hard clustering. Although SOUP aims at recovering the full membership matrix Θ, it can also be used as a hard-clustering method by labeling each cell as the majority type. We benchmark SOUP as a hard-clustering method on seven labeled public single-cell datasets (refs. 6 and 10; details in *SI Appendix*, Table S6). We compare SOUP to three popular single-cell clustering algorithms: (*i*) SC3, or single-cell consensus clustering (24); (*ii*) CIDR, or clustering through imputation and dimensionality reduction (25); and (*iii*) Seurat, named for Georges Seurat (26). Because we aim at hard clustering, here we set $\gamma = 0.8$ for SOUP. We give the true K as input to SC3, CIDR, and SOUP. For Seurat, we follow the choices in ref. 32 and set the resolution parameter to be 0.9 and use the estimated number of principal components (nPC) from CIDR. Even for hard clustering, SOUP is among the highest [Fig. 3, showing adjusted Rand index (ARI)]. Finally, when using the default choice of $\gamma = 0.5$, SOUP also achieves sensible performance, sometimes with even higher ARI (*SI Appendix*, Table S6).

Case Studies.

Fetal brain cells I. We apply SOUP to a fetal brain scRNA-seq dataset, with 220 developing fetal brain cells between 12 and 13 gestational weeks (GW) (9). Guided with marker genes, these single cells are labeled with seven types in the original paper: two subtypes of apical progenitors (AP1, AP2), two subtypes of basal progenitors (BP1, BP2), and three subtypes of neurons (N1, N2, N3). We refer to these as Camp labels after the lead author of ref. 9. At this age many cells are still transitioning between different types, providing valuable information regarding brain development. Therefore, instead of the traditional hard-clustering methods, SOUP can be used to recover the fine-scaled soft-clustering structure.

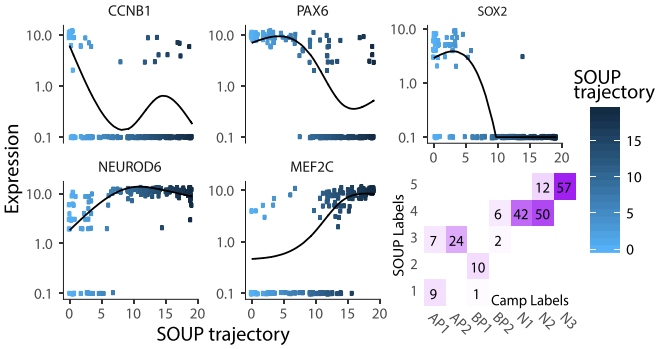

Fig. 4. Expression levels of five anchor genes, visualized in log scale, where the 220 fetal brain cells are ordered by a SOUP unilineal developmental trajectory.

We run SOUP with $K = 2, 3, \ldots, 7$ on the log-transformed transcript counts and examine the clusters of cells, initially treating this as a hard-clustering problem and focusing on the dominating type for each cell. For $K = 6$ and 7, some clusters have no cells assigned to them, which is indicative of a misspecified K. For $K = 5$, the algorithm identifies cell types that correspond to A1, A2, B1, N2, and N3 in Camp's nomenclature (Fig. 4 and *SI Appendix*, Fig. S6*A*). For these data, when cells are in various developmental stages, hard clustering appears to overfit the data.

Next, we examine the soft assignments. For each cluster k, we label it by an anchor gene, which is the marker gene defined in ref. 9 that has the largest anchor score, $[C_{gk} - \max\{C_{g,(-k)}\}]/sd(C_{g,(-k)})$, where $C_{g,(-k)}$ represents the center values of gene g on the (K-1) clusters other than k. The expression levels of the five anchor genes along the SOUP trajectory vary smoothly over developmental time (Fig. 4), consistent with Eq. **1**. In the top three PCs space, the cells show a smooth developmental trajectory between clusters (Fig. 5*A*), which is also consistent with Eqs. **1** and **2**.

To model the developmental trajectories we plot the cluster centers determined directly by SOUP (softSOUP) and by hard clustering (hardSOUP). Fitting a MST to the cluster centers, softSOUP identifies two lineages, AP-BP-N and AP-N (Fig. 5*A*), both of which were previously described in ref. 9, while hardSOUP identifies less intuitive BP-AP-N and AP-N lineages (Fig. 5*B*). Using Slingshot to fit smooth branching curves to these lineages via simultaneous principal curves, hardSOUP recovers AP-N and BP-N transitions, and the artificial BP1–AP2 transition in the initial MST fit is dropped (Fig. 5*D*). However, the AP–BP transition is still missing. softSOUP MST successfully reveals AP-N and AP-BP-N transitions (Fig. 5 *A* and *C*), thus capturing the true transition of cell types leading to neurons by accounting for the soft membership structures.

Fetal brain cells II. We next applied SOUP to a richer dataset with 2,309 single cells from human embryonic prefrontal cortex (PFC) from 8 GW to 26 GW (33). Using the Seurat package (26) the authors identified six major clusters: neural progenitor cells (NPC), excitatory neurons (EN), interneurons (IN), astrocytes (AST), oligodendrocyte progenitor cells (OPC) and microglia (MIC), which are referred to as Zhong labels after the lead author of ref. 33. Our objective is to evaluate the developmental trajectories of the major cell types, after excluding IN and MIC, which are known to originate elsewhere and migrate to the PFC (33). After several iterations of hard clustering by SOUP to remove IN and MIC cells (*SI Appendix*, Tables S1–S3) 1,503 cells remain, and they cluster into $K = 7$ types. These types correspond fairly well with the Zhong labels (Fig. 6*A*); however, many cells have low majority membership probabilities (*SI Appendix*, Fig. S8) and do not strongly favor a particular cluster (*SI Appendix*, Table S4). To illustrate this feature we display cells assigned to clusters 3 (NPC) and 7 (EN), color coded by the majority membership probability (Fig. 6*B*). The two clusters divide the PC space evenly, with the pure cells identifying the cluster centers, while many nonpure cells can be best described as transitioning between clusters. SOUP captures the transitional nature by soft clustering.

The SOUP trajectories reveal two developmental paths (Fig. 7): a neuronal lineage showing NPCs evolving to ENs (clusters: $4 \to 3 \to 7 \to 6 \to 5$) and a glial lineage showing NPCs evolving to OPCs and then to ASTs. Projecting the cells onto the lineages can provide pseudotime estimates of development. The lineages correspond roughly with sampled GWs (*SI Appendix*, Table S4). Our results are similar to those in ref. 33; however, we found that NPCs evolve to OPCs and then to ASTs (clusters: $4 \to 3 \to 1 \to 2$). The latter transitional step, which differs from the published analysis, is consistent with the literature (34). Finally, cluster 5, which consists of a mixture of cells Zhong labeled as EN and NPC, is placed at the end of the neuronal lineage, suggesting that some of the NPC labels are incorrect and that this cluster constitutes a distinct class of ENs.

Additional strengths of SOUP are highlighted by analyses described in *SI Appendix*, which investigate gene expression as a function of cell membership to cluster and the proximity of cells to the neuronal trajectory (Fig. 7 and *SI Appendix*, Fig. S9). In particular, we evaluate the final clusters of the neuronal lineage, clusters 5 and 6. In terms of gene expression, cells in cluster 6 shows all of the hallmarks of neuronal development, including low expression of neuronal markers in immature neurons and much higher expression in maturing neurons. There is also some evidence of heterogeneity of expression of genes marking neurons in some cells, consistent with differentiation into different neuronal subtypes. For cells from cluster 5, the evidence is far less clear: The majority of cells manifest neuronal markers at high levels, consistent with maturing neurons; yet, there is also expression of a substantial set of NPC markers in these neurons, a puzzling feature that could be either a technical artifact or an unanticipated developmental feature of deep-layer projection neurons.

Discussion

We develop SOUP, a semisoft clustering algorithm for single-cell data. SOUP fills the gap of modeling uncertain cell labels, including cells that are transitioning between cell types, which is ubiquitous in single-cell datasets. SOUP outperforms generic

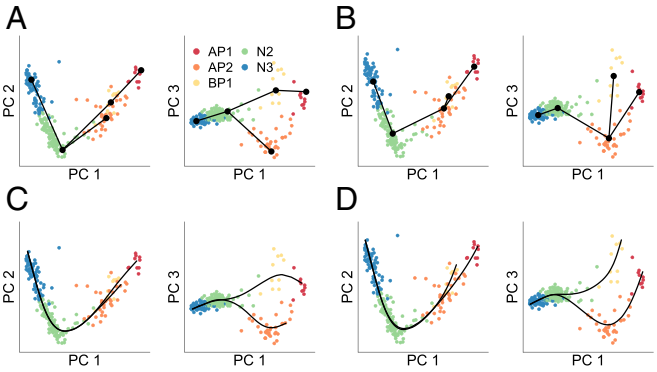

Fig. 5. Two hundred twenty fetal brain cells, cluster centers, lineages, and branching curves in the top three PCs space. Cells are colored according to their SOUP major types, but annotated using Camp labels based on the largest overlap (Fig. 4). (*A* and *B*) MST of softSOUP and hardSOUP cluster centers. (*C* and *D*) Smooth branching curves fitted by Slingshot based on MST in *A* and *B*, respectively.

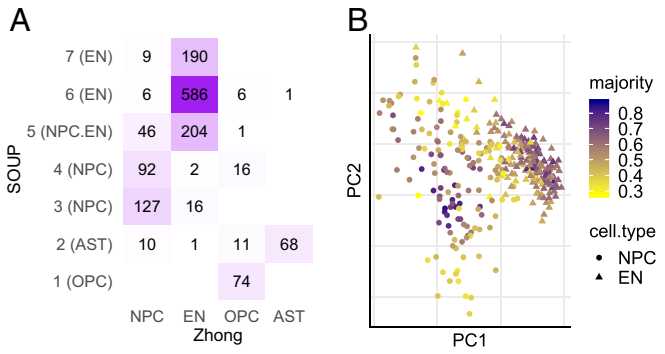

Fig. 6. (*A*) Contingency table of Zhong labels and major SOUP labels excluding IN and MIC. (*B*) Distribution of cluster 7 (EN) and cluster 3 (NPC) cells and their majority membership probabilities.

soft-clustering algorithms and, if treated as hard clustering, it also achieves comparable performance to that of state-of-the-art single-cell clustering methods. By using soft-clustering input, it can provide an estimate of developmental trajectories that is less biased and these results reflect valuable information regarding developmental patterns. We present the results from two case studies based on expression of human fetal brain cells and find SOUP reveals patterns of development not apparent in prior published analyses.

As is typical for clustering algorithms, selecting the optimal number of clusters, K, is challenging. We recommend balancing input from several empirical approaches and iterating over a range of K to determine a good choice. Notably, applying SOUP to different numbers of clusters reveals hierarchical structure among the cell types. To determine fine-scale structure within major cell types, SOUP can be applied iteratively to subsets of cells.

Using SOUP to obtain soft membership probabilities and then estimate developmental trajectories provides two complementary views of the data. Some cells can be reliably assigned to a cluster and these cells constitute pure types, which can be highly informative. Other cells are transitioning and estimated membership will fall within two, or even more, cell types. Examining the membership probabilities, and the placement on a developmental trajectory, provides critical information about the developmental processes and offers a parsimonious and scientifically meaningful alternative to estimating a large number of discrete cell types.

Notably, although SOUP is derived under a generic additive noise model and does not explicitly model the technical noise such as dropouts, we find it to be robust when applied to realistic simulations and to a variety of single-cell datasets. Moreover, it is computationally efficient. SOUP takes less than 15 min for 3,600 cells and 20,000 genes, benchmarked on a Linux computer equipped with an AMD Opteron Processor 6320 at 2.8 GHz. Therefore, SOUP is a versatile tool for single-cell analyses.

Methods

SOUP. Our SOUP algorithm contains two steps: (*i*) Find the set of pure cells and (*ii*) estimate Θ. Pure cells play a critical role in this problem. Intuitively, they provide valuable information from which to recover the cluster centers, which further guides the estimation of Θ for the mixed cells. Once the pure cells are identified, then the algorithm proceeds as described in *Results*.

Find Pure Cells. Denote the set of pure cells in cluster k as

$$\mathcal{I}_k = \{1 \le i \le n : \theta_{ik} = 1 \text{ and } \theta_{il} = 0, \forall l \ne k\} \quad [4]$$

and the set of all pure cells as $\mathcal{I} = \cup_{k=1}^{K} \mathcal{I}_k$. To recover $\mathcal{I}$, the key is to notice the special block structure formed by the pure cells in the similarity A. In particular, under Eq. **2**, the pure cells form K blocks in A, where the entries in these blocks are also the maxima in their rows and columns, ignoring the diagonal. Specifically, define $m_i = \max_{j \ne i} |A_{ij}|$, $S_i = \{j \ne i : |A_{ij}| = m_i\}$, and we call S_i the extreme neighbors of cell i. It can be shown that if cell i is pure, then $|A_{ij}| = m_i = m_j$ for all $j \in S_i$. On the contrary, for a mixed cell i, there exist some cells $j \in S_i$ where $m_j > |A_{ij}|$. Inspired by these observations, we define a purity score of each cell, $p_i = \frac{1}{|S_i|} \sum_{j \in S_i} |A_{ij}|/m_j$, and then naturally $p_i \in [0, 1]$. Furthermore, the pure cells have the highest purity scores; that is, $\mathcal{I} = \{i : p_i = 1\}$ (*SI Appendix*, Theorem S1).

In practice, we plug in the sample similarity matrix $\hat{A} = XX^T$ and estimate S_i and p_i by

$$\hat{S}_i = \{j \ne i : \text{the top } \epsilon \text{ percent with the largest } |\hat{A}_{ij}|\},$$
$$\hat{p}_i = \frac{1}{|\hat{S}_i|} \sum_{j \in \hat{S}_i} \frac{|\hat{A}_{ij}|}{\hat{m}_j}, \text{ where } \hat{m}_i = \max_{j \ne i} |\hat{A}_{ij}|, \quad [5]$$

and we estimate $\mathcal{I}$ with the top γ percent of cells: $\hat{\mathcal{I}} = \{i : \text{the top } \gamma \text{ percent with the largest } \hat{p}_i\}$. Finally, these pure cells are partitioned into K clusters, $\{\hat{\mathcal{I}}_k\}$, by standard clustering algorithms such as K-means. The complete algorithm is summarized in *SI Appendix*.

Tuning Parameters. The two tuning parameters of SOUP are the quantiles, ϵ and γ, both intuitive to set. The quantile γ should be an estimate of the proportion of pure cells in the data, of which we usually have prior knowledge. In practice, we find that SOUP remains stable even when γ is far from the true pure proportion, and it is helpful to use a generous choice. Throughout this paper, we always set $\gamma = 0.5$ and obtain sensible results. As for ϵ, it corresponds to the smallest proportion of per-type pure cells, and it suffices if $\epsilon \le \min_k |\mathcal{I}_k|/n$, so that $\hat{S}_i \subseteq S_i$ for pure cells. This choice does not need to be exact, as long as ϵ is a reasonable lower bound. In practice, we find it often beneficial to use a smaller ϵ that corresponds to less than 100 pure cells per type. By default, we use $\epsilon = 0.1$ for datasets with less than 1,000 cells, $\epsilon = 0.05$ for 1,000–2,000 cells, and $\epsilon = 0.03$ for even larger datasets.

Gene Selection. It is usually expected that not all genes are informative for clustering. For example, housekeeping genes are unlikely to differ across

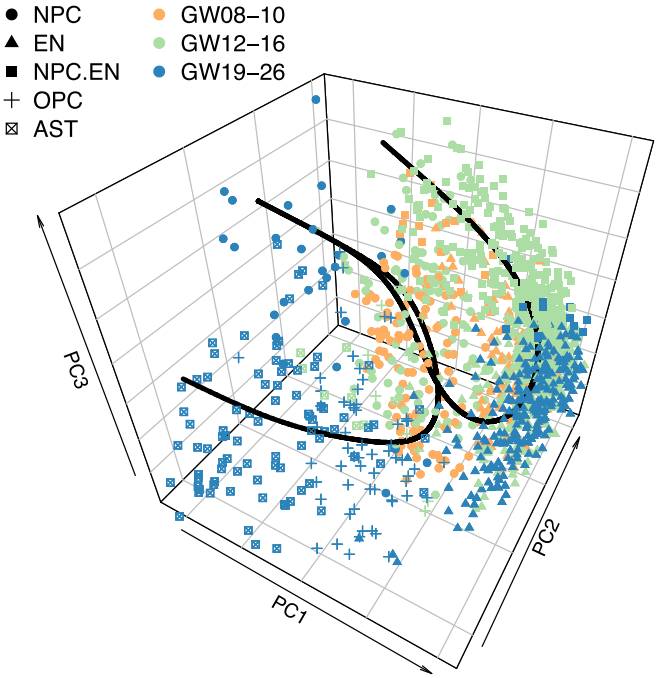

Fig. 7. Developmental trajectories of 1,503 Zhong cells delineate glial and neuronal pathways. Cluster labels are defined in Fig. 6*A*.

cell types and hence provide limited information for clustering other than introducing extra noise. Therefore, it is desirable to select a set of informative genes before applying SOUP clustering. Here, we combine two approaches for gene selection: (*i*) the DESCEND algorithm proposed in ref. 35 based on the Gini index and (*ii*) the Sparse PCA (SPCA) algorithm (36) (*SI Appendix*).

The R package of SOUP is available at https://github.com/lingxuez/SOUPR.

ACKNOWLEDGMENTS. This work was supported by National Institute of Mental Health Grants R37MH057881 and R01MH109900 and the Simons Foundation Grants SFARI 402281 and 367561.

1. Kowalczyk T, et al. (2009) Intermediate neuronal progenitors (basal progenitors) produce pyramidal-projection neurons for all layers of cerebral cortex. *Cereb Cortex* 19:2439–2450.
2. Jones EG (2009) The origins of cortical interneurons: Mouse versus monkey and human. *Cereb Cortex* 19:1953–1956.
3. Nadarajah B, Alifragis P, Wong ROL, Parnavelas JG (2003) Neuronal migration in the developing cerebral cortex: Observations based on real-time imaging. *Cereb Cortex* 13:607–611.
4. Silbereis JC, Pochareddy S, Zhu Y, Li M, Sestan N (2016) The cellular and molecular landscapes of the developing human central nervous system. *Neuron* 89:248–268.
5. Keren-Shaul H, et al. (2017) A unique microglia type associated with restricting development of Alzheimer's disease. *Cell* 169:1276–1290.e17.
6. Darmanis S, et al. (2015) A survey of human brain transcriptome diversity at the single cell level. *Proc Natl Acad Sci USA* 112:7285–7290.
7. Tang F, et al. (2009) mRNA-seq whole-transcriptome analysis of a single cell. *Nat Methods* 6:377–382.
8. Ramsköld D, et al. (2012) Full-length mRNA-seq from single-cell levels of RNA and individual circulating tumor cells. *Nat Biotechnol* 30:777–782.
9. Camp JG, et al. (2015) Human cerebral organoids recapitulate gene expression programs of fetal neocortex development. *Proc Natl Acad Sci USA* 112:15672–15677.
10. Baron M, et al. (2016) A single-cell transcriptomic map of the human and mouse pancreas reveals inter- and intra-cell population structure. *Cell Syst* 3:346–360.e4.
11. Zeisel A, et al. (2015) Brain structure. Cell types in the mouse cortex and hippocampus revealed by single-cell RNA-seq. *Science* 347:1138–1142.
12. Tasic B, et al. (2016) Adult mouse cortical cell taxonomy revealed by single cell transcriptomics. *Nat Neurosci* 19:335–346.
13. Trapnell C, et al. (2014) The dynamics and regulators of cell fate decisions are revealed by pseudotemporal ordering of single cells. *Nat Biotechnol* 32:381–386.
14. Bendall SC, et al. (2014) Single-cell trajectory detection uncovers progression and regulatory coordination in human B cell development. *Cell* 157:714–725.
15. Shin J, et al. (2015) Single-cell RNA-seq with waterfall reveals molecular cascades underlying adult neurogenesis. *Cell Stem Cell* 17:360–372.
16. Ji Z, Ji H (2016) TSCAN: Pseudo-time reconstruction and evaluation in single-cell RNA-seq analysis. *Nucleic Acids Res* 44:e117.
17. Street K, et al. (2018) Slingshot: Cell lineage and pseudotime inference for single-cell transcriptomics. *BMC Genomics* 19:477.
18. Mao X, Sarkar P, Chakrabarti D (2017) On mixed memberships and symmetric nonnegative matrix factorizations. *Proceedings of the 34th International Conference on Machine Learning*. Available at proceedings.mlr.press/v70/mao17a.html. Accessed December 18, 2018.
19. Arora S, Ge R, Moitra A (2012) Learning topic models–going beyond SVD. *2012 IEEE 53rd Annual Symposium on Foundations of Computer Science (FOCS)*. Available at https://ieeexplore.ieee.org/document/6375276. Accessed December 18, 2018.
20. Arora S, et al. (2013) A practical algorithm for topic modeling with provable guarantees. Proceedings of the 30th International Conference on Machine Learning. Available at proceedings.mlr.press/v28/arora13.html. Accessed December 18, 2018.
21. Huang K, Fu X, Sidiropoulos ND (2016) Anchor-free correlated topic modeling: Identifiability and algorithm. *Advances in Neural Information Processing Systems*, eds Lee DD, Sugiyama M, Luxburg UV, Guyon I, Garnett R (Curran Associates, Inc., Red Hook, NY), Vol 29, pp 1786–1794.
22. Lee DD, Seung HS (2001) Algorithms for non-negative matrix factorization. *Advances in Neural Information Processing Systems*, eds Leen TK, Dietterich TG, Tresp V (MIT Press, Cambridge, MA), Vol 13, pp 556–562.
23. Bing X, Bunea F, Ning Y, Wegkamp M (2017) Sparse latent factor models with pure variables for overlapping clustering. arXiv:1704.06977. Preprint, posted April 23, 2017.
24. Kiselev VY, et al. (2017) SC3: Consensus clustering of single-cell RNA-seq data. *Nat Methods* 14:483–486.
25. Lin P, Troup M, Ho JW (2017) CIDR: Ultrafast and accurate clustering through imputation for single-cell RNA-seq data. *Genome Biol* 18:59.
26. Satija R, Farrell JA, Gennert D, Schier AF, Regev A (2015) Spatial reconstruction of single-cell gene expression data. *Nat Biotechnol* 33:495–502.
27. Setty M, et al. (2016) Wishbone identifies bifurcating developmental trajectories from single-cell data. *Nat Biotechnol* 34:637–645.
28. Bezdek JC (1981) *Pattern Recognition with Fuzzy Objective Function Algorithms* (Kluwer Academic Publishers, Norwell, MA).
29. Sun Z, et al. (2017) DIMM-SC: A Dirichlet mixture model for clustering droplet-based single cell transcriptomic data. *Bioinformatics* 34:139–146.
30. Zappia L, Phipson B, Oshlack A (2017) Splatter: Simulation of single-cell RNA sequencing data. *Genome Biol* 18:174.
31. Kolodziejczyk AA, Kim JK, Svensson V, Marioni JC, Teichmann SA (2015) The technology and biology of single-cell RNA sequencing. *Mol Cell* 58:610–620.
32. Yang Y, et al. (September 8, 2018) SAFE-clustering: Single-cell aggregated (from ensemble) clustering for single-cell RNA-seq data. *Bioinformatics*, 10.1093/bioinformatics/bty793.
33. Zhong S, et al. (2018) A single-cell RNA-seq survey of the developmental landscape of the human prefrontal cortex. *Nature* 555:524–528.
34. Zhu X, Bergles DE, Nishiyama A (2008) NG2 cells generate both oligodendrocytes and gray matter astrocytes. *Development* 135:145–157.
35. Wang J, et al. (2018) Gene expression distribution deconvolution in single-cell RNA sequencing. *Proc Natl Acad Sci USA* 115:E6437–E6446.
36. Witten DM, Tibshirani R, Hastie T (2009) A penalized matrix decomposition, with applications to sparse principal components and canonical correlation analysis. *Biostatistics* 10:515–534.