Semi-soft Clustering of Single Cell Data

Lingxue Zhu[1], Jing Lei[1], Bernie Devlin[2], and Kathryn Roeder[1]

[1]Department of Statistics and Data Science, Carnegie Mellon University
[2]Department of Psychiatry, University of Pittsburgh School of Medicine

Abstract

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 semi-soft clustering that can classify both pure and intermediate cell types from data on gene expression or protein abundance from individual cells. Called SOUP, for Semi-sOft clUstering with Pure cells, this novel algorithm reveals the clustering structure for both pure cells, which belong to one single cluster, as well as 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 the K cell types form in a similarity matrix, devised by pairwise comparison of the gene expression profiles of individual cells. Once pure cells are identified, they provide the key information from which the membership matrix can be computed. SOUP is applicable to general clustering problems as well, as long as the unrestrictive modeling assumptions hold. The performance of SOUP is documented via extensive simulation studies. Using SOUP to analyze two single cell data sets from brain shows it produce sensible and interpretable results.

1 Introduction

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 (Kowalczyk et al., 2009), 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 (Jones, 2009; Nadarajah et al., 2003). By the human mid-fetal period there are myriad cell types and the foundations of typical and atypical neurodevelopment are already established (Silbereis et al., 2016). 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 will develop such a method here.

Different types of cells will have different transcriptomes or gene expression profiles (Silbereis et al., 2016). Thus, they can be identified by these profiles (Darmanis et al., 2015), 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 (Tang et al., 2009; Ramsköld et al., 2012), 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 (Camp et al., 2015; Darmanis et al., 2015; Baron et al., 2016; Zeisel et al., 2015; Tasic et al., 2016).

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 Semi-sOft clUstering with Pure cells (SOUP). SOUP intelligently recovers the set of pure cells by exploiting the block structures in 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.

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 data sets, one from young adult mouse brain and the other scRNA-seq from early fetal development of the cortex. In both settings SOUP produces sensible and interpretable results, when compared to published analyses of these data.

2 Method

2.1 Background

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 semi-soft clustering, where we expect the existence of both (i) pure cells, each belonging to a single cluster and requiring a hard cluster assignment, as well as (ii) mixed cells (transitional cells) that are transitioning between two or more cell types, and hence should obtain soft assignments. Formally, with K distinct cell types, we assume

$$X = \Theta C^T + E\,, \tag{1}$$

where $\Theta \in {\mathbb{R}_+}^{n\times K}$ is a nonnegative membership matrix, $C \in \mathbb{R}^{p\times K}$ represents the cluster centers, and $E \in \mathbb{R}^{n\times p}$ is a zero-mean noise matrix. Each row of the membership matrix, $\Theta_i := (\theta_{i1}, \cdots, \theta_{iK})$, 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. Each column of the center matrix C represents the average expression levels of p genes in each cell type.

It is natural to assume independent noise across cells. Here, we further assume $\mathbb{E}(EE^T) = \sigma^2 I$, and work on the cell-cell similarity matrix,

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

where $Z = C^T C \in \mathbb{R}^{K\times K}$ represents the association among different cell types. SOUP requires Z to have larger entries along the diagonal, but not necessarily a diagonal matrix. Our algorithm aims at recovering the membership matrix Θ based on the factorization eq.(2).

Similar factorization problems have appeared in previous literature under different scenarios. The most popular one is the mixed-membership stochastic block model (MMSB) for network community

detection, which leads to $P = \Theta Z \Theta$, where P is the expectation of the network adjacency matrix, Z is the community connectivity matrix, and Θ is the membership matrix. The most related approach in this field is the algorithm proposed in Mao et al. (2017) based on matrix factorization, but this approach requires Z to be diagonal, which is unrealistic for single cells. Another similar formulation appears in topic modeling, where it is usually assumed $\mathbb{E}[X] = \Theta C^T$, with X being the word-document count matrix, C being the proportions of topics in each document, and Θ representing the topic-specific word distributions. Under this setting, it is required that *columns* of Θ sum to one, and the most related methods are the optimization-based algorithms based on the co-occurrence matrix, a normalized version of A (for example, Arora et al. (2012, 2013); Huang et al. (2016)). However, it is nontrivial to extend these algorithms to our scenario, where we require *rows* of Θ to sum to one.

Finally, recent work in Bing et al. (2017) considers 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 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 find it sensitive to noise, leading to poor performance (Supplementary Section S2.5). 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. We also incorporate a gene selection procedure prior to clustering, which further enhances the performance of SOUP.

2.2 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. In fact, it has been shown in Bing et al. (2017) that the existence of pure cells is essential for model (2) 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 2 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 steps are presented below.

Find pure cells Denote the set of pure cells in cluster k as

$$\mathcal{I}_k = \{1 \leq i \leq n : \theta_{ik} = 1 \text{ and } \theta_{il} = 0, \forall l \neq k\}, \tag{3}$$

and the set of all pure cells as

$$\mathcal{I} = \cup_{k=1}^{K} \mathcal{I}_k . \tag{4}$$

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

$$m_i = \max_{j \neq i} |A_{ij}|, \quad S_i = \{j \neq i : |A_{ij}| = m_i\}, \tag{5}$$

and we call S_i the *extreme neighbors* of cell i. It can be shown that a cell i is pure if and only if its extreme neighbors are the other pure cells in the same cluster, with $|A_{ij}| = m_j$ for all $j \in S_i$. Inspired by these observations, we define a purity score of each cell,

$$p_i = \frac{1}{|S_i|} \sum_{j \in S_i} \frac{|A_{ij}|}{m_j}, \tag{6}$$

then naturally $p_i \in [0, 1]$. Furthermore, the pure cells are the ones with the highest purity scores, that is, $\mathcal{I} = \{i : p_i = 1\}$ (Theorem S1).

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

$$\begin{aligned} \hat{S}_i &= \{j \neq i : \text{the top } \epsilon \text{ percent with the largest } |\hat{A}_{ij}|\}, \\ \hat{p}_i &= \frac{1}{|\hat{S}_i|} \sum_{j \in \hat{S}_i} \frac{|\hat{A}_{ij}|}{\hat{m}_j}, \text{ where } \hat{m}_i = \max_{j \neq i} |\hat{A}_{ij}|, \end{aligned} \tag{7}$$

and we estimate $\mathcal{I}$ with the top γ percent of cells:

$$\hat{\mathcal{I}} = \{i : \text{the top } \gamma \text{ percent with the largest } \hat{p}_i\}. \tag{8}$$

Finally, these pure cells are partitioned into K clusters, $\{\hat{\mathcal{I}}_k\}$, by standard clustering algorithms such as K-means.

Estimate membership Consider the top K eigenvectors of A, denote as $V \in \mathbb{R}^{n \times K}$, then it can be shown that 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, $\Theta_{\mathcal{I}\cdot}$. Then it is straightforward to recover the desired Q^* from $\Theta_{\mathcal{I}\cdot} = V_{\mathcal{I}\cdot}Q^*$, which further recovers the full membership matrix $\Theta = VQ^*$ (Theorem 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 Theorem 1, the optimization problem*

$$\min_{Q \in \mathbb{R}^{K \times K}} ||\Theta_{\mathcal{I}\cdot} - V_{\mathcal{I}\cdot}Q||_F^2 \tag{9}$$

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

In practice, we plug in the sample similarity matrix $\hat{A}$, and the complete algorithm is summarized in Supplementary S1.1.

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 $\epsilon \leq \min_k |\mathcal{I}_k|/n$, 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. Simulation results of sensitivity are presented in Supplementary Section S2.4.

Normalization In practice, cells can have different scaling due to sequencing depths and cell sizes, and proper normalization is required prior to using SOUP. Formally, we have

$$\mathbb{E}\left[X_{raw}\right] = \mathrm{Diag}((s_i))\Theta C^T ,$$

where s_i is the scaling factor of cell i. The factor s_i can be interpreted as efficiency, while C is the expected expression level of each type. Alternatively, s_i can represent the library size, with C being the *relative* type-specific expression profile. This factor can be estimated in various ways, sometimes with the help of spike-ins (Wang et al., 2017). Here, for RNA-seq data, we simply treat s_i as the library size, and normalize such that the total sum of counts, $\sum_g X_{ig}$, is 10^6 in each cell i, which is essentially the Transcript per Million (TPM) normalization.

Gene selection It is usually expected that not all genes are informative for clustering. For example, housekeeping genes are unlikely to differ across cell types, 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 Wang et al. (2017) based on the Gini index, and (ii) the Sparse PCA (SPCA) algorithm (Witten et al., 2009) (see Supplementary Section S3 for details).

3 Performance Evaluation

3.1 Simulations

There are no direct competitors of SOUP for semi-soft clustering in the single-cell literature, and here we use the following two candidates for comparison:

- Fuzzy C-Means (FC) (Bezdek, 1981), a generic soft clustering algorithm. Its tuning parameter m controls the cluster fuzziness, where $m = 1$ gives hard clustering, and $m = 2$ is the default choice. Here, we use two choices, $m \in \{1.5, 2\}$.
- DIMMSC (Sun et al., 2017), a probabilistic clustering algorithm for single cell data based on Dirichlet mixture models. It is designed for hard clustering, but internally estimates the

posterior probability of each cell belonging to different clusters, which can be treated as an estimator of Θ.

To mimic the gene expression patterns in real data, we use the peripheral blood mononuclear cells (PBMC) dataset (Zheng et al., 2017) to construct the cluster centers C. This dataset has been previously used to benchmark single-cell clustering algorithms in Yang et al. (2017), where the authors construct two scenarios, each containing three cell types. Here, we follow the same setting, and present the challenging scenario with three highly similar cell types: (i) CD8+/CD45RA+ naive cytotoxic T cells, (ii) CD4+/CD25+ regulatory T cells, and (iii) CD4+/CD45RA+/CD25- naive T cells. The complete results, including the other simpler scenario with three distinct cell types, are presented in Supplementary Section S2.

We use the 13,150 genes that are expressed in at least 10 cells, and compute average expression levels. Next, C is normalized to have unit column variance, so that it represents the *relative* expression profiles of each type. The data is then generated from a Poisson model, which is widely used and has been shown to be appropriate for scRNA-seq (Wang et al., 2017). The expression of gene g in cell i is simulated independently from

$$X_{ig} \sim \text{Poisson}(s_i \lambda_{ig}) \,, \text{where } \lambda_{ig} = (\Theta_i C^T)_g \tag{10}$$

and s_i is the expected library size, simulated from Negative Binomial with mean 10^4 and variance $2 * 10^4$, and $\Theta_i = (\theta_{i1}, ..., \theta_{iK})$ is the membership vector. We simulate 100 pure cells in each type, and another set of $\{100, 300, 500\}$ mixed cells with Θ_i's drawn from the uniform distribution. More scenarios can be found in Supplementary Section S2. Throughout this section, we use the true $K = 3$ as input for all algorithms, and the default parameters for SOUP ($\epsilon = 0.1$, $\gamma = 0.5$). We do not perform gene selections for this simulation.

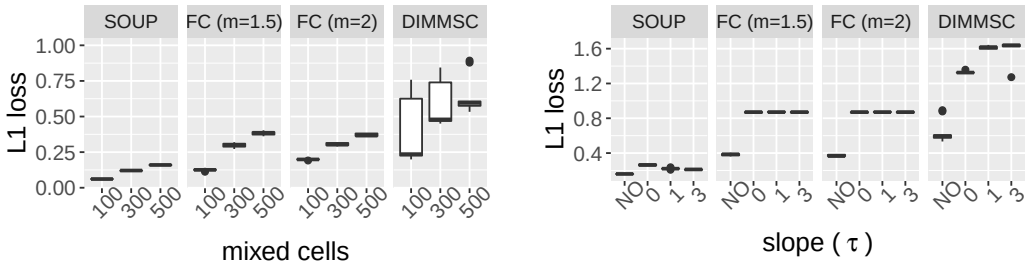
Soft membership estimation For comparable evaluation across different scenarios with different cell numbers, we present the average L_1 loss per cell, i.e., $\frac{1}{n}||\hat{\Theta} - \Theta||_1$, where $||\cdot||_1$ is the usual vector L_1 norm after vectorization, and SOUP achieves the best performance under all scenarios (Figure 1a). In particular, with 100, 300, and 500 mixed cells, the true pure proportions 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.

Robustness to dropouts One of the biggest challenges in single cell data is the existence of dropouts (Kolodziejczyk et al., 2015), where the mRNA for a gene fails to be amplified prior to sequencing, producing a "false" zero in the observed data. Here, we illustrate that SOUP is robust to such technical noise. Specifically, we introduce artificial dropouts by setting X_{ig} to be zero with probability π_{ig} for gene g in cell i. It is commonly believed that the dropout probability is higher for genes with lower expression levels (Kolodziejczyk et al., 2015), hence we use

$$\pi_{ig} = 1 - \text{logistic}\left(\tau \lambda_{ig} + \kappa\right) \,, \tag{11}$$

where the slope τ determines the dependency of dropout probability on the expression level λ_{ig}, and the intercept κ is set to be $\kappa = 0.85 - \tau * \text{mean}(\lambda_{\text{ig}})$, such that the average dropout probability

is roughly 30%. We evaluate the average L_1 loss when $\tau \in \{0, 1, 3\}$, where $\tau = 0$ is the special case with no dependency on λ_{ig}. For simplicity, we present only the results for 500 mixed cells, for which SOUP is robust and outperforms all other algorithms (Figure 1b).



(a) Without dropout

(b) With 30% dropout

Figure 1: Boxplot of the average L_1 losses of estimating Θ in 10 simulations. **(a)** Without dropout, where there are 100 pure cells per cluster, and the number of mixed cells is 100, 300, and 500. **(b)** With an average of 30% dropout and 500 mixed cells, where the dependency of dropout probability on expression levels is 0, 1, or 3. The losses without dropout are shown on the first columns for comparison, indicated by τ=NO.

3.2 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 with its majority type. In this final section, we benchmark SOUP as a hard clustering method on 7 labeled public single-cell datasets (Baron et al. (2016); Darmanis et al. (2015), details in Table S1). We compare SOUP to three popular single-cell clustering algorithms: (i) SC3 (Kiselev et al., 2017), (ii) CIDR (Lin et al., 2017), and (iii) Seurat (Satija et al., 2015). 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 Yang et al. (2017) and set the resolution parameter to be 0.9, and use the estimated number of principal components (nPC) from CIDR for Seurat. Even for hard clustering, SOUP is among the highest (Figure 2, showing Adjusted Rand Index (ARI)). Finally, we point out that when using the default choice of $\gamma = 0.5$, SOUP also achieves sensible performance, sometimes with even higher ARI (Table S1).

4 Results

4.1 Mouse brain cells

We first illustrate the application of SOUP to a mouse brain dataset (Zeisel et al., 2015). The data set contains 3,005 single cells of somatosensory cortex and hippocampal CA1 region from mice between postnatal 21-31 days. These cells are expected to be mostly mature. The authors developed the BackSPIN clustering algorithm that assigned these cells into 7 major types: pyramidal CA1,

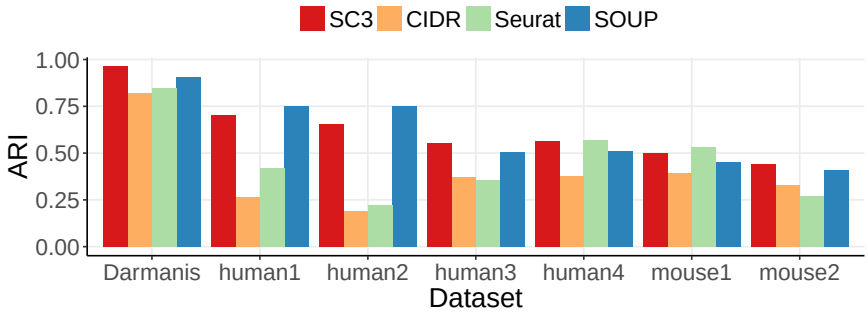


Figure 2: Adjusted Rand Index (ARI) on 7 labeled public datasets (Baron et al., 2016; Darmanis et al., 2015), using (i) SC3, (ii) CIDR, (iii) Seurat, and (iv) SOUP.

pyramidal SS, interneurons, astrocytes/ependymal, microglia, endothelial/mural, and oligodendrocytes. These labels are referred to as Zeisel labels in the following analysis, and we illustrate the performance of SOUP both when treated as hard clustering, as well as when using the estimated soft membership.

We start with 16,450 genes that are expressed in at least 10 cells, and use DESCEND and SPCA to select highly variable genes. Combining 2,785 genes selected by DESCEND and 178 genes selected by SPCA, we come to a total of 2,818 unique genes for SOUP. Among these selected genes, we see 41 marker genes that are previously used in Zeisel et al. (2015) to label the 7 major types as well as their subtypes (Table S2). We set $\gamma = 0.5$ for SOUP as usual. As for ϵ, because of the large number of cells, we use a smaller $\epsilon = 0.03$, corresponding to about 90 pure cells per type. Figure 3 visualizes the purity scores of cells on the first 3-dimensional principal space, and we see clear tight clusters formed by cells with high purity scores, and mixed cells with lower purity scores sitting in-between.

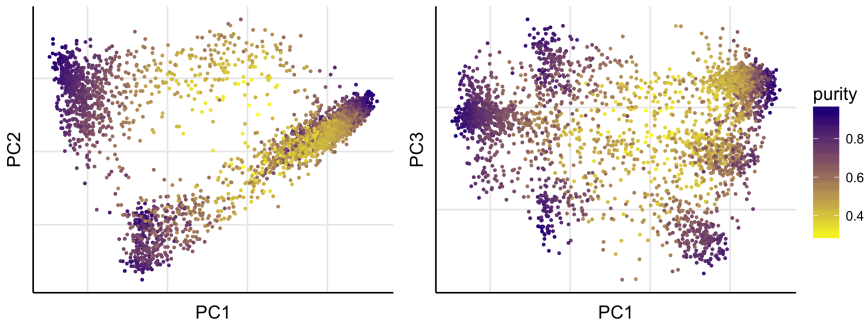


Figure 3: Purity scores of 3,005 mouse brain cells visualized in the leading principal subspaces.

We run SOUP with a sequence of $K = 2, 3, ..., 7$, and examine its results via hard clustering (Figure 4a). We observe high consistency between SOUP major assignments and the Zeisel labels, as well as interesting hierarchical structures. The major split when $K = 2$ picks up three neuronal types (pyramidal cells and interneurons). The oligodendrocytes stand out when $K = 3$, interneurons are revealed by $K = 4$, and $K = 7$ picks up all major cell types.

The ARI for SOUP hard assignment when $K = 7$ is 0.89, which is among the highest across the

state-of-the-art single cell clustering algorithms (Wang et al., 2017; Yang et al., 2017). Among all clusters of SOUP, cluster 2 contains a majority of pyramidal cells, but also a small Zeisel group of endothelial cells and oligodendrocytes, and cluster 7 is an oligodendrocyte cluster but also contains a Zeisel group of pyramidal cells and astrocytes (Figure 4b). We suspect three reasons for these results: (i) the sequencing depths are generally lower for astrocytes, microglia, and endothelial cells, making them difficult to separate (Figure S8a); (ii) some of the endothelial cells and oligodendrocytes do have similar expression patterns as pyramidal cells (Figure S9); and (iii) the existence of mixed cells in the dataset, for which hard assignments can be misleading. In particular, we examine the top 50% pure cells, and their major cell types are much more consistent with Zeisel labels, with ARI=0.97 (Figure S8b).

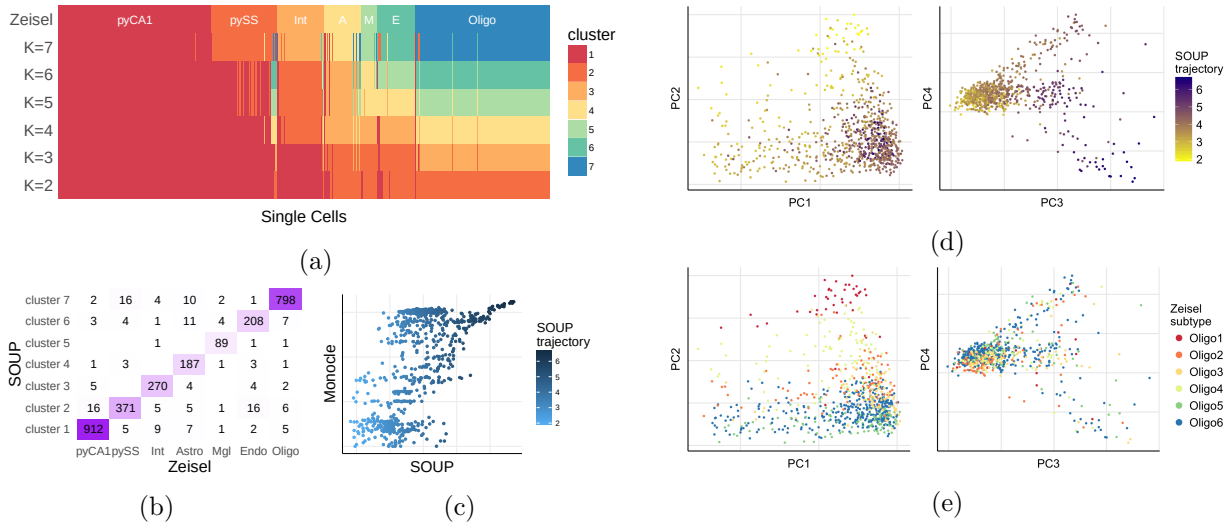


Figure 4: **(a)** SOUP cluster assignments of 3,005 mouse brain cells when $K = 2, 3, ..., 7$. Their Zeisel labels (Zeisel et al., 2015) are shown on the top. **(b)** Contingency table of Zeisel labels and SOUP clusters for 3,005 mouse brain cells. **(c)** Comparison of the developmental trajectory estimated by SOUP and Monocle for 3,005 mouse brain cells. **(d)-(e)** 798 Zeisel oligodendrocytes in SOUP major cluster 7, projected on the leading principal component subspaces. Cells are colored by (d) SOUP estimated developmental trajectory, and (e) Zeisel subtypes.

By applying BackSPIN on each of the major cell type, the 3,005 mouse brain cells were further divided into 47 subtypes in Zeisel et al. (2015) (Table S4), which we refer to as Zeisel subtypes. Here, we further examine whether SOUP can also reveal such fine-scaled clustering structure. Specifically, we re-run SOUP, including the gene selection procedure, within each of the seven SOUP major clusters identified above. For most major cell types, we observe largely consistent results between SOUP hard assignments and their Zeisel subtypes (details in Supplementary Section S4.1, Figure S10). The most substantial deviation between SOUP and Zeisel occurs in oligodendrocytes, where SOUP successfully distinguishes the Zeisel Oligo1 subtype, but the hard assignments differ in the remaining cells (Figure S10d). The six Zeisel oligodendrocyte subtypes are mapped to different states of maturation in Zeisel et al. (2015), and we argue that instead of hard clustering, a continuous trajectory can better characterize such underlying structure.

In particular, we illustrate that using SOUP estimated soft membership, a natural ordering of cells

can be obtained to represent such developmental process. For this purpose, we label the clusters $(1, ..., K)$ to be consistent with the developmental order, by setting cluster 1 (mostly Oligo1) as the starting point, and consecutively appending the next closest cluster according to the correlation among cluster centers. Finally, a developmental timepoint can be estimated for each cell i according to its SOUP estimated soft membership $(\hat{\theta}_{i1}, ..., \hat{\theta}_{iK})$:

$$t_i = \sum_{k=1}^{K} k \cdot \hat{\theta}_{ik} \,. \tag{12}$$

We focus on the 798 Zeisel oligodendrocytes that belong to SOUP major cluster 7, and visualize their estimated developmental trajectory on the leading principal components (PCs), computed with the 1,220 genes selected by DESCEND and SPCA (Figure 4d). For comparison, on the same low-dimensional spaces, we also visualize the six Zeisel subtypes (Figure 4e). We see that Zeisel subtypes mainly align with PC2 and overlook the remaining structures, while SOUP trajectory reveals sensible cell transitioning in all dimensions. We also compare SOUP with the results from the popular single-cell temporal ordering algorithm, Monocle (Trapnell et al., 2014) (Figure 4c), and observe that Monocle trajectory only picks up the structure along PC1 (Figure S11a).

4.2 Fetal brain cells

Next, we apply SOUP to a fetal brain single-cell RNA-seq dataset, with 220 developing fetal brain cells between 12-13 week postconception (Camp et al., 2015). Guided with marker genes, these single cells are labeled with 7 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. In fetal brains, it is expected that many cells are still transitioning between different types, leading to a significant proportion of mixed cells. These cells provide 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.

The published data have been normalized by Fragments Per Kilobase of transcript per Million (FPKM) and log-transformed. To apply SOUP, we first transform the data back to count scale, round to the closest integer, and then apply normalization as usual. We start with 12,694 genes that are non-zero in at least 2 cells. After gene selection, 430 genes are retained for SOUP clustering, including 300 selected by DESCEND and 158 selected by SPCA. The original cell types are labeled according to 18 marker genes in Camp et al. (2015), of which 12 are selected by our procedure (Table S2).

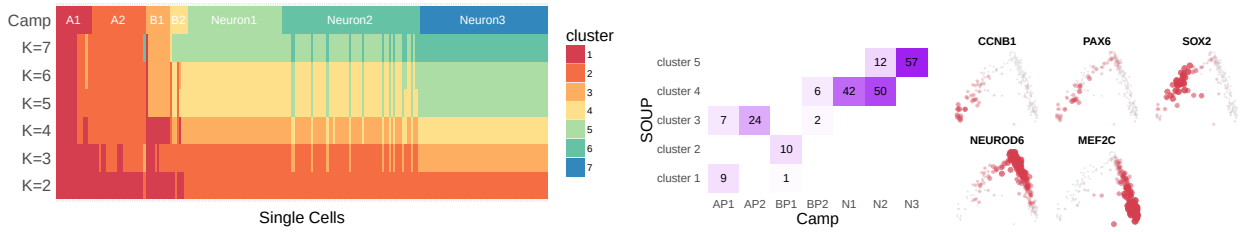
We run SOUP with $K = 2, 3, ..., 7$, and examine the major clusters of cells. Recall that SOUP provides the full soft-assignment membership matrix Θ, but here we treat it as a hard clustering method by only focusing on the dominating type of each cell. Figure 5a visualizes the major cell types compared to the Camp labels. SOUP results are largely consistent with the given labels, but they also suggest that the split between AP1 and AP2 in Camp labels is likely to be unnatural, and the split of three neuron subtypes is also ambiguous. These observations are consistent with the patterns in marker genes (Figure S12). The sequence of different K's also reveals the hierarchical structure in the data. For example, $K = 2$ separates progenitors versus neurons, and $K = 3$ gives three major groups: (i) early neural progenitors (mainly AP1), (ii) more matured progenitors and

early neurons, and (iii) the matured neurons. As soon as $K = 4$, the AP2 and BP2 cells pop up. Because BP2 cells are always clustered with early neurons, and there are always only two subtypes of neurons, we use $K = 5$ in the following analysis. The contingency table compared to Camp labels is shown in Figure 5b. Notice that with $K = 6$ and $K = 7$, the effective number of major clusters is 5 and 6, respectively, meaning that there is one column of Θ that is never the largest proportion for any cell, and this situation is usually an indication of a misspecified K.

Next, we examine the soft assignments of SOUP. Figure 5c visualizes the cells on the leading 2-dimensional principal space, colored and sized according to the estimated weights of each cell in each cluster $\hat{\theta}_{ik}$. For each cluster k, we label it by an anchor gene, which is one of the marker genes (Table S2) that has the largest anchor score, defined as

$$\text{anchor score}_{gk} = \frac{C_{gk} - \max\{C_{g,(-k)}\}}{sd(C_{g,(-k)})}, \tag{13}$$

where $C_{g,(-k)}$ represents the center values of gene g on the (K-1) clusters other than k.



(a) (b) (c)

Figure 5: **(a)** SOUP major cluster for 220 fetal brain cells when $K = 2, 3, ..., 7$, compared to Camp labels (Camp et al., 2015) on the top. Each entry is colored by the cluster assignment of each cell. **(b)** Contingency table of Camp labels and the major SOUP clusters. **(c)** SOUP estimated soft memberships. 220 fetal brain cells are visualized in the leading 2-dimensional principal space, and each cluster is labeled by an anchor gene. In each cluster, cells with proportion ≥ 0.01 are highlighted, and the sizes and color transparency represent their estimated proportions.

Finally, we estimate the developmental trajectory for the 220 fetal brain cells following eq.(12), using cluster 5 (matured neurons) as the ending point. We examine the expression levels of the 5 anchor genes along the SOUP developmental trajectory, and observe smooth patterns (Figure 6). Our SOUP trajectory is also highly consistent with the results from Monocle (Trapnell et al., 2014) when using the same set of 430 selected genes as for SOUP (Figure 6, Figure S13b).

5 Discussion

We develop SOUP, a novel semi-soft 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 soft clustering algorithms and, if treated as hard clustering, it also achieves comparable performance as state-of-the-art single cell clustering methods. It also gives an ordering of cells that reflects valuable information regarding

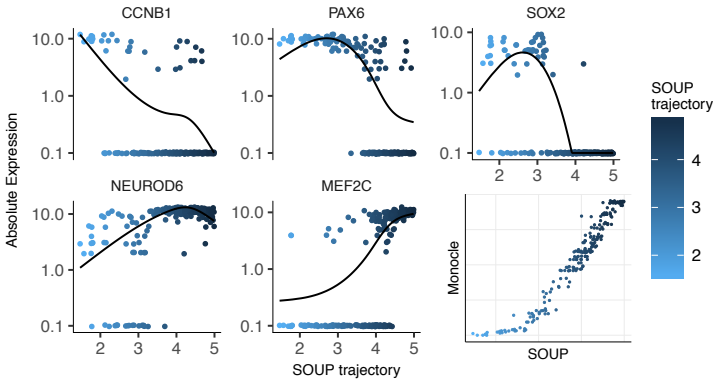


Figure 6: Expression levels of 5 anchor genes, visualized in log scale, where the 220 fetal brain cells are ordered by SOUP developmental trajectory. The smooth lines are fitted by natural cubic splines with 3 degrees of freedom. The comparison between SOUP and Monocle trajectories is visualized in the bottom right panel.

developmental periods. We present the results from two real datasets, one from human brain, another from mouse brain, and find SOUP reveals interesting patterns in the data, sometimes in agreement with prior published analyses, at other times differing in interpretation.

For most of our analyses, SOUP takes the number of clusters K as given. By applying SOUP to a sequence of different K's, we reveal interesting hierarchical structures in the data. Selecting the optimal K in an adaptive manner, however, is a challenging problem for many clustering methods, including SOUP. One simple solution is to apply standard selection techniques on the set of estimated pure cells, such as the elbow method for K-means, as well as using various criterion such as the Sihouette score. Yet, in practice, these approaches tend to substantially underestimate the cluster numbers. In Supplementary Section S3, we propose a cross validation procedure that obtains sensible results in simulations as well as public single cell datasets.

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 particularly sensible and robust when applied to single cell datasets. Moreover, it is computationally efficient, which makes it easily applicable to large single cell datasets. SOUP takes less than 15 minutes for 3,600 cells and 20,000 genes, benchmarked on a linux computer equipped with AMD Opteron(tm) Processor 6320 @ 2.8 GHz. Therefore, we believe SOUP can be versatile and applicable to a wider range of applications.

Acknowledgement

This work was supported by NIMH grants R37MH057881 and R01MH109900 and the Simons Foundation SFARI 124827. The R package of SOUP is available at https://github.com/lingxuez/SOUP.

References

Arora, S., Ge, R., Halpern, Y., Mimno, D., Moitra, A., Sontag, D., Wu, Y., & Zhu, M. (2013). A practical algorithm for topic modeling with provable guarantees. In *International Conference on Machine Learning*, (pp. 280–288).

Arora, S., Ge, R., & Moitra, A. (2012). Learning topic models–going beyond svd. In *Foundations of Computer Science (FOCS), 2012 IEEE 53rd Annual Symposium on*, (pp. 1–10). IEEE.

Baron, M., Veres, A., Wolock, S. L., Faust, A. L., Gaujoux, R., Vetere, A., Ryu, J. H., Wagner, B. K., Shen-Orr, S. S., Klein, A. M., Melton, D. A., & Yanai, I. (2016). A single-cell transcriptomic map of the human and mouse pancreas reveals inter- and intra-cell population structure. *Cell Syst*, *3*(4), 346–360.e4.

Bezdek, J. C. (1981). *Pattern recognition with fuzzy objective function algorithms*. Kluwer Academic Publishers.

Bing, X., Bunea, F., Ning, Y., & Wegkamp, M. (2017). Sparse latent factor models with pure variables for overlapping clustering. *arXiv:1704.06977*.

Caliński, T., & Harabasz, J. (1974). A dendrite method for cluster analysis. *Communications in Statistics-theory and Methods*, *3*(1), 1–27.

Camp, J. G., Badsha, F., Florio, M., Kanton, S., Gerber, T., Wilsch-Bräuninger, M., Lewitus, E., Sykes, A., Hevers, W., Lancaster, M., et al. (2015). Human cerebral organoids recapitulate gene expression programs of fetal neocortex development. *Proceedings of the National Academy of Sciences*, *112*(51), 15672–15677.

Darmanis, S., Sloan, S. A., Zhang, Y., Enge, M., Caneda, C., Shuer, L. M., Hayden Gephart, M. G., Barres, B. A., & Quake, S. R. (2015). A survey of human brain transcriptome diversity at the single cell level. *Proc Natl Acad Sci U S A*, *112*(23), 7285–90.

Fu, W., & Perry, P. O. (2017). Estimating the number of clusters using cross-validation. *arxiv:1702.02658*.

Huang, K., Fu, X., & Sidiropoulos, N. D. (2016). Anchor-free correlated topic modeling: Identifiability and algorithm. In *Advances in Neural Information Processing Systems 29*, (pp. 1786–1794). Curran Associates, Inc.

Jones, E. G. (2009). The origins of cortical interneurons: mouse versus monkey and human. *Cereb Cortex*, *19*(9), 1953–6.

Kiselev, V. Y., Kirschner, K., Schaub, M. T., Andrews, T., Yiu, A., Chandra, T., Natarajan, K. N., Reik, W., Barahona, M., Green, A. R., & Hemberg, M. (2017). Sc3: consensus clustering of single-cell rna-seq data. *Nat Methods*, *14*(5), 483–486.

Kolodziejczyk, A. A., Kim, J. K., Svensson, V., Marioni, J. C., & Teichmann, S. A. (2015). The technology and biology of single-cell rna sequencing. *Molecular cell*, *58*(4), 610–620.

Kowalczyk, T., Pontious, A., Englund, C., Daza, R. A. M., Bedogni, F., Hodge, R., Attardo, A., Bell, C., Huttner, W. B., & Hevner, R. F. (2009). Intermediate neuronal progenitors (basal

progenitors) produce pyramidal-projection neurons for all layers of cerebral cortex. *Cereb Cortex*, *19*(10), 2439–50.

Lin, P., Troup, M., & Ho, J. W. (2017). Cidr: Ultrafast and accurate clustering through imputation for single-cell rna-seq data. *Genome biology*, *18*(1), 59.

Mao, X., Sarkar, P., & Chakrabarti, D. (2017). On mixed memberships and symmetric nonnegative matrix factorizations. In *Proceedings of the 34th International Conference on Machine Learning*, vol. 70, (pp. 2324–2333).

Nadarajah, B., Alifragis, P., Wong, R. O. L., & Parnavelas, J. G. (2003). Neuronal migration in the developing cerebral cortex: observations based on real-time imaging. *Cereb Cortex*, *13*(6), 607–11.

Ramsköld, D., Luo, S., Wang, Y.-C., Li, R., Deng, Q., Faridani, O. R., Daniels, G. A., Khrebtukova, I., Loring, J. F., Laurent, L. C., Schroth, G. P., & Sandberg, R. (2012). Full-length mrna-seq from single-cell levels of rna and individual circulating tumor cells. *Nat Biotechnol*, *30*(8), 777–82.

Rousseeuw, P. J. (1987). Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, *20*, 53–65.

Satija, R., Farrell, J. A., Gennert, D., Schier, A. F., & Regev, A. (2015). Spatial reconstruction of single-cell gene expression data. *Nat Biotechnol*, *33*(5), 495–502.

Silbereis, J. C., Pochareddy, S., Zhu, Y., Li, M., & Sestan, N. (2016). The cellular and molecular landscapes of the developing human central nervous system. *Neuron*, *89*(2), 248–68.

Sun, Z., Wang, T., Deng, K., Wang, X.-F., Lafyatis, R., Ding, Y., Hu, M., & Chen, W. (2017). Dimm-sc: a dirichlet mixture model for clustering droplet-based single cell transcriptomic data. *Bioinformatics*, *34*(1), 139–146.

Tang, F., Barbacioru, C., Wang, Y., Nordman, E., Lee, C., Xu, N., Wang, X., Bodeau, J., Tuch, B. B., Siddiqui, A., Lao, K., & Surani, M. A. (2009). mrna-seq whole-transcriptome analysis of a single cell. *Nat Methods*, *6*(5), 377–82.

Tasic, B., Menon, V., Nguyen, T. N., Kim, T. K., Jarsky, T., Yao, Z., Levi, B., Gray, L. T., Sorensen, S. A., Dolbeare, T., Bertagnolli, D., Goldy, J., Shapovalova, N., Parry, S., Lee, C., Smith, K., Bernard, A., Madisen, L., Sunkin, S. M., Hawrylycz, M., Koch, C., & Zeng, H. (2016). Adult mouse cortical cell taxonomy revealed by single cell transcriptomics. *Nat Neurosci*, *19*(2), 335–46.

Trapnell, C., Cacchiarelli, D., Grimsby, J., Pokharel, P., Li, S., Morse, M., Lennon, N. J., Livak, K. J., Mikkelsen, T. S., & Rinn, J. L. (2014). The dynamics and regulators of cell fate decisions are revealed by pseudotemporal ordering of single cells. *Nat Biotechnol*, *32*(4), 381–386.

Wang, J., Huang, M., Torre, E., Dueck, H., Shaffer, S., Murray, J., Raj, A., Li, M., & Zhang, N. R. (2017). Gene expression distribution deconvolution in single cell rna sequencing. *bioRxiv 227033*.

Witten, D. M., Tibshirani, R., & Hastie, T. (2009). A penalized matrix decomposition, with applications to sparse principal components and canonical correlation analysis. *Biostatistics*, *10*(3), 515–534.

Yang, Y., Huh, R., Culpepper, H. W., Love, M. I., & Li, Y. (2017). Safe-clustering: Single-cell aggregated (from ensemble) clustering for single-cell rna-seq data. *bioRxiv*.

Zeisel, A., Muñoz-Manchado, A. B., Codeluppi, S., Lönnerberg, P., La Manno, G., Juréus, A., Marques, S., Munguba, H., He, L., Betsholtz, C., Rolny, C., Castelo-Branco, G., Hjerling-Leffler, J., & Linnarsson, S. (2015). Brain structure. cell types in the mouse cortex and hippocampus revealed by single-cell rna-seq. *Science*, *347*(6226), 1138–42.

Zheng, G. X. Y., Terry, J. M., Belgrader, P., Ryvkin, P., Bent, Z. W., Wilson, R., Ziraldo, S. B., Wheeler, T. D., McDermott, G. P., Zhu, J., Gregory, M. T., Shuga, J., Montesclaros, L., Underwood, J. G., Masquelier, D. A., Nishimura, S. Y., Schnall-Levin, M., Wyatt, P. W., Hindson, C. M., Bharadwaj, R., Wong, A., Ness, K. D., Beppu, L. W., Deeg, H. J., McFarland, C., Loeb, K. R., Valente, W. J., Ericson, N. G., Stevens, E. A., Radich, J. P., Mikkelsen, T. S., Hindson, B. J., & Bielas, J. H. (2017). Massively parallel digital transcriptional profiling of single cells. *Nat Commun*, *8*, 14049.

Supporting Information

Contents

S1 Details of SOUP **16**

S1.1 Algorithms . . . 16

S1.2 Gene selection . . . 18

S1.3 SOUP for count data . . . 18

S1.4 Theorems and proofs . . . 19

S2 Supplementary Simulations **21**

S2.1 Simulation settings . . . 21

S2.2 Soft membership estimation . . . 22

S2.3 Robustness to dropouts . . . 22

S2.4 Sensitivity to tuning parameters . . . 23

S2.5 Comparison to LOVE . . . 24

S3 Selecting K **26**

S3.1 Selecting K from pure cells . . . 26

S3.2 Selecting K with cross validation . . . 27

S3.3 Benchmarking on public datasets . . . 28

S4 Supplementary Results **29**

S4.1 Discussion of mouse brain data . . . 29

S4.2 Discussion of fetal brain data . . . 30

S5 Supporting Tables **34**

S1 Details of SOUP

S1.1 Algorithms

For the ease of readability, we summarize the two steps of SOUP in Algorithm 1 and Algorithm 2. Note that when solving for $\hat{Q}$ in Algorithm 2, theoretically one would consider the constrained

problem:

$$\min_{Q \in \mathbb{R}^{K \times K}} ||\hat{\Theta}_{\hat{\mathcal{I}}\cdot} - \hat{V}_{\hat{\mathcal{I}}\cdot} Q||_F^2 \,,$$
$$\text{s.t. } \hat{V}Q \text{ is an } n \times K \text{ membership matrix,}$$

where membership matrices are the ones with nonnegative entries and unit row sums. However, in practice, solving this constrained problem is computationally demanding, sometimes with empty feasible sets. Therefore, we adopt a heuristic approach that first solves the unconstrained optimization problem as in Algorithm 2, and normalizes the obtained $\hat{V}\hat{Q}$ afterwards to get a membership matrix.

Algorithm 1 findPure

Input: similarity matrix $\hat{A}$, quantile ϵ, quantile γ
Output: estimated set of pure cells $\hat{\mathcal{I}}$

1: For each cell i,

$$\hat{m}_i \leftarrow \max_{j \neq i} |\hat{A}_{ij}| \,,$$
$$\hat{S}_i \leftarrow \{j \neq i : |\hat{A}_{ij}| \geq \text{ the upper } \epsilon\text{-quantile of } |\hat{A}_{i,(-i)}|\} \,.$$

2: For each cell i, compute its purity score:

$$\hat{p}_i \leftarrow \frac{1}{|\hat{S}_i|} \sum_{j \in \hat{S}_i} \frac{|\hat{A}_{ij}|}{\hat{m}_j}.$$

3: $\hat{\mathcal{I}} \leftarrow \{i : \hat{p}_i \geq \text{ the uppser } \gamma\text{-quantile of } \{\hat{p}_j\}\}$.

Algorithm 2 SOUP clustering

Input: data X, number of clusters K, quantile ϵ, quantile γ
Output: estimated membership $\hat{\Theta}$

1: $\hat{A} \leftarrow XX^T$
2: $\hat{\mathcal{I}} \leftarrow$ *findPure*$(\hat{A}, K, \epsilon, \gamma)$, and apply K-means clustering on $X_{\hat{\mathcal{I}}\cdot}$ to get the partition $(\hat{\mathcal{I}}_1, ..., \hat{\mathcal{I}}_K)$.
3: $\hat{V} \leftarrow$ the top K eigenvectors of $\hat{A}$.
4: Let $\hat{\Theta}_{\hat{\mathcal{I}}\cdot}$ be the membership submatrix for pure cells, obtained by putting 1's and 0's in the corresponding columns according to $\{\hat{\mathcal{I}}_k\}$. Solve for

$$\hat{Q} = \arg \min_{Q \in \mathbb{R}^{K \times K}} ||\hat{\Theta}_{\hat{\mathcal{I}}\cdot} - \hat{V}_{\hat{\mathcal{I}}\cdot} Q||_F^2 \,,$$

where $||\cdot||_F$ is the Frobenius norm.
5: $\hat{\Theta} \leftarrow \hat{V}\hat{Q}$.

S1.2 Gene selection

It is usually expected that not all genes are informative for clustering. For example, housekeeping genes are unlikely to differ across cell types, hence provide limited information for clustering other than noise. Therefore, it is desirable to select a set of informative (i.e. highly variable) genes before applying SOUP. Here, we combine two selection approaches. The first is the DESCEND method proposed in Wang et al. (2017), where the authors developed a semi-parametric approach to estimate several distribution statistics for each gene, including the Gini index that indicates the excessive variability of gene expression levels across cells. The authors suggested to threshold the normalized difference between the observed and expected Gini index, and use the set of highly variable genes for clustering. Throughout this paper, we use the default threshold of 3. We refer the readers to the original paper (Wang et al., 2017) for more details.

The second approach is based on Sparse Principal Component Analysis (SPCA). In fact, it is common to first apply PCA, and select genes with the highest loadings in the top few principal components (PCs) for clustering (see Camp et al. (2015) for an example). SPCA provides a more rigorous algorithm to implement this idea, where it directly solves for the leading *sparse* PCs, and genes with nonzero entries are selected. In this paper, we use the efficient SPCA algorithm in Witten et al. (2009). The algorithm requires a tuning parameter, c, that controls the sparsity of the solution, where smaller c leads to sparser results, hence fewer selected genes. Throughout this paper, we always set $c = 0.05$, and use the top three sparse PCs.

In our experiments, we find that DESCEND and SPCA usually capture different structures in the data. SPCA usually picks up genes that differentiate major cell types, while DESCEND usually identifies genes that distinguish finer scaled clustering structures. Therefore, the best performance is achieved by combining both lists of genes.

S1.3 SOUP for count data

The SOUP algorithm is derived under a generic additive noise model (eq.(1)). Here, we point out that SOUP is applicable to more general scenarios as long as we construct a similarity matrix that has structures similar to eq.(2). In particular, under a Poisson model, we have

$$X_{ig} \sim \text{Poisson}((\Theta C^T)_{ig})$$

for cell i and gene g independently, we can compute the following similarity matrix

$$\hat{A}_{\text{poisson}} = XX^T - \text{Diag}((d_i)) \,,$$

where $d_i = \sum_g X_{ig}$. Then it can be shown that

$$\mathbb{E}\left[\hat{A}_{\text{poisson}}\right] = \Theta C^T C \Theta^T \,,$$

which has the same structure as eq.(2) with $\sigma^2 = 0$, hence SOUP still applies.

However, when applied to single cell datasets, we find it usually beneficial to log-transform the RNA-seq data and use the general algorithm for additive noise model (Algorithm 1 and Algorithm 2).

In addition, centering data with respect to genes when computing A is also helpful for identifying pure cells, because this usually makes Z more diagonally dominant, leading to a larger separation in purity scores between pure cells and mixed cells, as indicated in Theorem S1.

S1.4 Theorems and proofs

Theorem S1 (Pure cells)**.** *In model (2), under the same assumptions as Theorem 1, if we further require*

(d) $\Delta := \min_{k \neq l}(Z_{kk} \wedge Z_{ll} - |Z_{kl}|) > 0$,

then we have $\mathcal{I} = \{i : p_i = 1\}$, *where* p_i *is the purity score as defined in eq.(6). In addition, let*

$$\bar{\theta} = \max_{i \notin \mathcal{I}, 1 \leq k \leq K} \theta_{ik} \,,$$

then $0 < \bar{\theta} < 1$ *by definition, and we have*

$$p_i < 1 - \frac{(1-\bar{\theta})\Delta}{\max_k Z_{kk}} \,, \ \forall i \notin \mathcal{I} \,.$$

Proof of Theorem S1. Note that

$$A_{ij} = \sum_{k,l} \theta_{ik}\theta_{jl} Z_{kl} \,.$$

For any cell i, consider the two cases:

- If cell i is pure for type k, then for any $j \neq i$,

$$|A_{ij}| = \begin{cases} Z_{kk} & \text{if } j \text{ is also pure for type } k \\ |Z_{kl}| & \text{if } j \text{ is pure for type } l \neq k \\ |\sum_l \theta_{jl} Z_{kl}| & \text{if } j \text{ is mixed} \end{cases} \tag{S1}$$

  Note that $|Z_{kl}| \leq Z_{kk} - \Delta$, and we can also show that in the last case,

$$|A_{ij}| < \sum_l \theta_{jl}|Z_{kl}| \leq \theta_{jk} Z_{kk} + \sum_{l \neq k} \theta_{jl}(Z_{kk} - \Delta) = Z_{kk} - (1 - \theta_{jk})\Delta < Z_{kk} \,, \tag{S2}$$

  hence

$$m_i = Z_{kk} \,, \ S_i = \{j \neq i : j \text{ is also pure for type } k.\}$$

- If cell i is mixed, then for any $j \neq i$,

$$|A_{ij}| = \begin{cases} |\sum_k \theta_{ik} Z_{kl}| & \text{if } j \text{ is pure for type } l \\ |\sum_{k,l} \theta_{ik}\theta_{jl} Z_{kl}| & \text{if } j \text{ is also mixed} \end{cases} \,. \tag{S3}$$

For the first case, we already shown above that

$$|\sum_k \theta_{ik} Z_{kl}| < Z_{ll} - (1-\theta_{il})\Delta \leq Z_{ll} - (1-\bar{\theta})\Delta\,.$$

For the second case, we have

$$|\sum_{k,l} \theta_{ik}\theta_{jl} Z_{kl}| \leq \sum_l \theta_{jl} |\sum_k \theta_{ik} Z_{kl}| \leq \max_l |\sum_k \theta_{ik} Z_{kl}| \tag{S4}$$

Therefore, m_i is achieved at (i,j) for the pure cells j in some type l^*:

$$m_i = \max_l |\sum_k \theta_{ik} Z_{kl}|\,,\ S_i = \{j \neq i : j \text{ is pure for type } l^* \text{ where } l^* \in \arg\max_l |\sum_k \theta_{ik} Z_{kl}|\}\,.$$

Therefore, the conclusions follow because

- if i is pure for type k, then

  $$S_i = \mathcal{I}_k \backslash \{i\},\ \text{and } |A_{ij}| = m_j = Z_{kk}\,, \forall j \in S_i$$

  hence $p_i = 1$.

- if i is mixed, then S_i contains pure cells from cluster(s) $l^* \in \arg\max_l |\sum_k \theta_{ik} Z_{kl}|$, and for any $j \in S_i$,

  $$|A_{ij}| \leq Z_{l^*l^*} - (1-\bar{\theta})\Delta < Z_{l^*l^*} = m_j\,,$$

  hence $\frac{|A_{ij}|}{m_j} \leq 1 - \frac{(1-\bar{\theta})\Delta}{Z_{l^*l^*}}$, which further implies $p_i \leq 1 - \frac{(1-\bar{\theta})\Delta}{\max_k Z_{kk}}$.

□

Proof of Theorem 1. See Theorem 2 in Bing et al. (2017). Note that Assumption (b), the existence of pure cells, is necessary for identifiability. We refer the readers to Bing et al. (2017) for an example of an unidentifiable model where assumptions (a) and (c) hold, but no pure cells exist. □

Proof of Theorem 2. First, consider the noise free scenario, $\tilde{X} = \Theta C^T$, then we have

$$\mathbb{E}\left[\tilde{X}\tilde{X}^T\right] = \Theta Z \Theta^T := \tilde{A}\,. \tag{S5}$$

Consider the following symmetric decomposition problem,

$$\tilde{A} = GG^T\,, \tag{S6}$$

then $\Theta Z^{1/2}$ is one solution. Although the solution is not unique when Z is not diagonal, it has been shown in Mao et al. (2017) that under the same assumptions as in Theorem 1, for any solution G, there exists a $K \times K$ matrix O, such that

$$GO = \Theta Z^{1/2}\,. \tag{S7}$$

In particular, let $\tilde{A} = \tilde{V}\tilde{\Lambda}\tilde{V}^T$ be the SVD of $\tilde{A}$, then $\tilde{G} = \tilde{V}\tilde{\Lambda}^{1/2}$ is one solution to problem (S6), which implies

$$\Theta = \tilde{V}\tilde{Q}\,, \tag{S8}$$

where $\tilde{Q} = \tilde{\Lambda}^{1/2} O Z^{-1/2}$ is a $K \times K$ matrix. In order to find $\tilde{Q}$, recall that we are given the set of pure cells $\mathcal{I}$ and their partitions $\{\mathcal{I}_k\}$. Equivalently, we already know the corresponding rows of the membership matrix, $\Theta_{\mathcal{I}\cdot}$. Therefore, the desired $\tilde{Q}$ can be solved from

$$\Theta_{\mathcal{I}\cdot} = \tilde{V}_{\mathcal{I}\cdot} \tilde{Q} \,. \tag{S9}$$

In general, under model (2), we have

$$A = \tilde{A} + \sigma^2 I \,,$$

where $\tilde{A} = \Theta Z \Theta^T$ is the above noise-free similarity matrix. Let V be the top K eigenvectors for A, then $V = \tilde{V}$, hence the above arguments also hold for V. Finally, the solution to problem (9) is unique because its Hessian matrix, $(V_{\mathcal{I}\cdot})^T V_{\mathcal{I}\cdot}$, is positive definite due to assumption (b) in Theorem 1. Furthermore, the minimum is 0, achieved at $Q^* = \tilde{Q}$, and $\Theta = VQ^*$. □

S2 Supplementary Simulations

S2.1 Simulation settings

First, we describe the detailed simulation settings. As in the main text, we compare SOUP with Fuzzy C-means (FC) (Bezdek, 1981) with two choices of m, $m \in \{1.5, 2\}$, as well as DIMMSC (Sun et al., 2017). We use the true $K = 3$ as input for all algorithms, and the default parameters for SOUP ($\epsilon = 0.1, \gamma = 0.5$). Throughout this section, we do not perform gene selection. We apply normalization and log-transformation for both SOUP and FC, and give the raw counts to DIMMSC as input because it is built on a Multinomial count model.

We mimic the gene expression patterns in real data by computing the cluster centers from the peripheral blood mononuclear cells (PBMC) dataset (Zheng et al., 2017). We construct two scenarios following Sun et al. (2017): (i) a simple scenario with three distinct cell types (CD56+ NK cells, CD19+ B cells, and CD4+/CD25+ regulatory T cells); and (ii) a challenging scenario with cells from three similar cell types (CD8+/CD45RA+ naive cytotoxic T cells, CD4+/CD25+ regulatory T cells, and CD4+/CD45RA+/CD25- naive T cells). Among the 32,738 genes, we use the ones that are expressed in at least 10 cells, which lead to 13,969 genes in the simple scenario, and 13,150 genes in the challenging scenario. In the main text, only the challenging scenario is presented. Here, we present the full results in both settings.

We compute the cluster center $C \in \mathbb{R}^{p \times K}$ from the average expression levels within each cell type, and normalize each column to have unit variance, so that C represents the cluster-specific *relative* expression level. Then the expression of gene g in cell i is simulated from

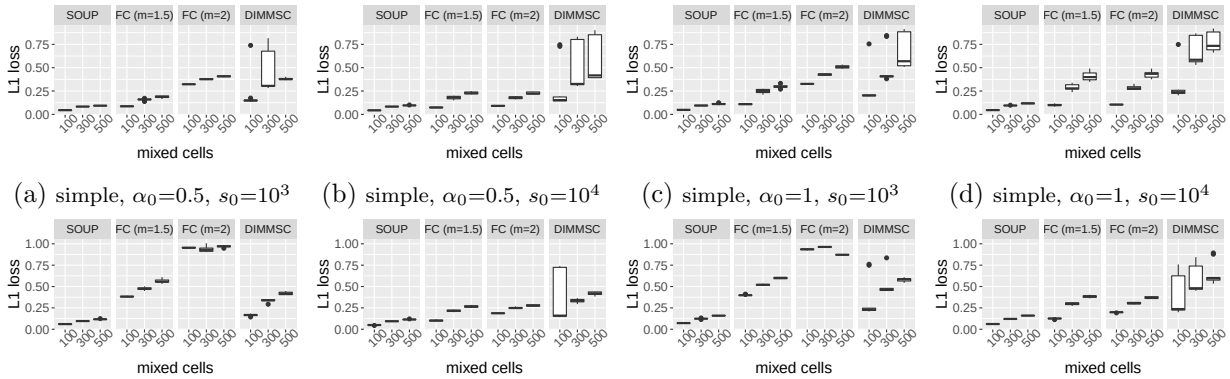
$$X_{ig} \sim \text{Poisson}(s_i \lambda_{ig}) \,, \text{where } \lambda_{ig} = (\theta_i C^T)_g \,, \tag{S10}$$

where s_i is the expected library size of cell i, and $\theta_i = (\theta_{i1}, ..., \theta_{iK})$ is its membership vector. The expected library size s_i is simulated from Negative Binomial distribution with mean s_0 and variance $2s_0$, where $s_0 \in \{10^3, 10^4\}$. As for the membership, we construct 100 pure cells per type, and another set of n_{mix} mixed cells whose memberships are generated from Dirichlet$((\alpha_0, \alpha_0, \alpha_0))$. The concentration parameter α_0 controls the "fuzziness" of mixed cells, where larger α_0 leads to fuzzier mixtures. In particular, we present the results with $\alpha_0 = 0.5$ where the memberships are highly

concentrated on a few types, as well as $\alpha_0 = 1$ that corresponds to the uniform distribution. We also evaluate the performance with different number of mixed cells, $n_{mix} \in \{100, 300, 500\}$.

S2.2 Soft membership estimation

We conduct 10 repetitions in each setting, and compare the L_1 losses of estimating Θ. To achieve comparable evaluation across different scenarios with different cell numbers, we present the average loss per cell, i.e., $\frac{1}{n}||\hat{\Theta} - \Theta||_1$, where $||\cdot||_1$ is the usual vector L_1 norm after vectorization. In Figure S1, we see that SOUP achieves the best performance in all settings, and under the challenging scenario, the advantage of SOUP is substantial when the library size is smaller (i.e., $s_0 = 10^3$). In addition, SOUP is robust to all factors, including the expected library size s_0, mixture fuzziness α_0, and the number of mixed cells. In particular, note that we always set $\gamma = 0.5$ for SOUP, which represents a prior guess of 50% pure cells, but with $n_{mix} \in \{100, 300, 500\}$, the true pure proportions are $\{0.75, 0.5, 0.375\}$, respectively. We see that SOUP stays stable even when γ substantially overestimates or underestimates the pure proportion.



(a) simple, α_0=0.5, s_0=10^3 (b) simple, α_0=0.5, s_0=10^4 (c) simple, α_0=1, s_0=10^3 (d) simple, α_0=1, s_0=10^4

(e) challenge, α_0=0.5, s_0=10^3 (f) challenge, α_0=0.5, s_0=10^4 (g) challenge, α_0=1, s_0=10^3 (h) challenge, α_0=1, s_0=10^4

Figure S1: Boxplots of average L_1 losses of SOUP estimated memberships in simulations with 10 repetitions, under the simple scenario (**(a)-(d)**) and the challenging scenario (**(e)-(h)**). In addition to 300 pure cells, there are $\{100, 300, 500\}$ mixed cells with memberships generated from Dirichlet$(\alpha_0, \alpha_0, \alpha_0)$). The library sizes of cells are simulated from a Negative Binomial with mean s_0 and variance $2s_0$.

S2.3 Robustness to dropouts

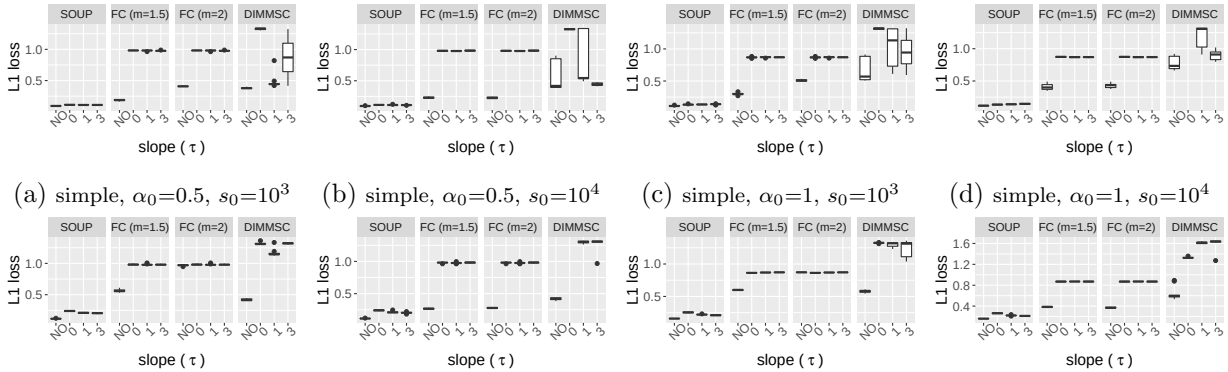
Next, we evaluate the robustness of different algorithms to dropouts. Artificial dropouts are introduced by randomly setting each entry to zero with probability π_{ig}, that is,

$$\tilde{X}_{ig} = X_{ig}\text{Bernoulli}(1 - \pi_{ig}) \tag{S11}$$

for gene g in cell i, where π_{ig} is the dropout probability. It is usually expected that dropout probability is higher for genes with lower expression levels (Kolodziejczyk et al., 2015), so we set

$$\pi_{ig} = 1 - \text{logistic}\left(\tau\lambda_{ig} + \kappa\right) , \tag{S12}$$

where the slope τ determines the dependency of dropout probability on the expression level λ_{ig}, and the intercept κ is chosen such that the average dropout probability is around 30%, with $\kappa = 0.85 - \tau * \bar{\lambda}$ where $\bar{\lambda} = \frac{1}{np}\sum_{i,g}\lambda_{ig}$. We present the results under three choices of dependencies, $\tau \in \{0, 1, 3\}$, where $\tau = 0$ is the special case when the dropout probability does not depend on λ_{ig}. Again, we evaluate the average L_1 loss of estimating Θ, and for simplicity, we present only the results with 500 mixed cells (Figure S2). We see that SOUP achieves robust performance with the existence of dropouts, especially under the simple scenario, and always outperforms other algorithms.



(a) simple, α_0=0.5, s_0=10^3 (b) simple, α_0=0.5, s_0=10^4 (c) simple, α_0=1, s_0=10^3 (d) simple, α_0=1, s_0=10^4

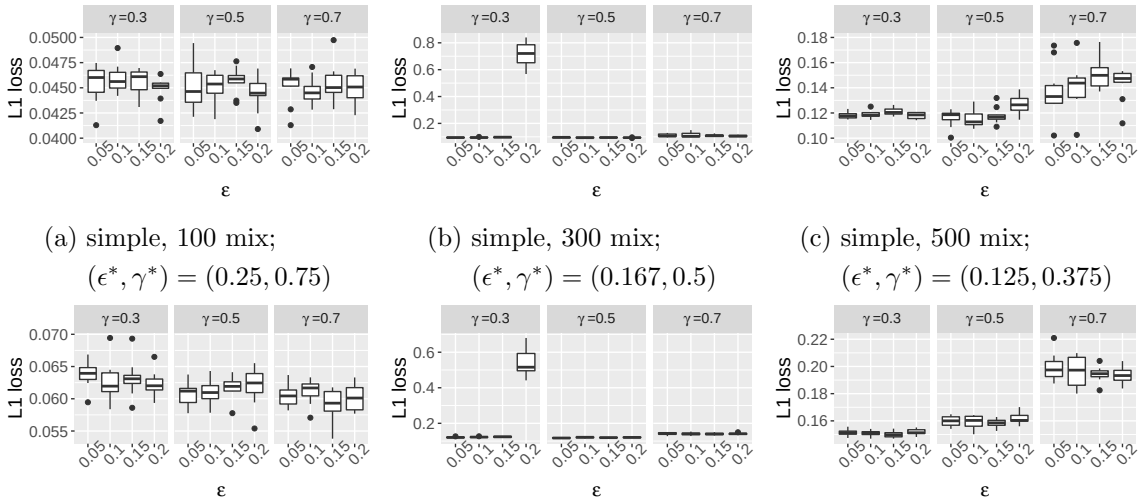
(e) challenge, α_0=0.5, s_0=10^3 (f) challenge, α_0=0.5, s_0=10^4 (g) challenge, α_0=1, s_0=10^3 (h) challenge, α_0=1, s_0=10^4

Figure S2: Boxplots of average L_1 losses of SOUP estimated memberships in 10 repetitions, with an average of 30% artificial dropouts, under the simple scenario (**(a)-(d)**) and the challenging scenario (**(e)-(h)**). In addition to 300 pure cells, there are 500 mixed cells whose memberships are generated from Dirichlet($\alpha_0, \alpha_0, \alpha_0$)). The library sizes of cells are simulated from a Negative Binomial with mean s_0 and variance $2s_0$. The dropout probability depends on the expression level via slope τ, where $\tau = 0$ represents no dependency. The losses without dropout are shown on the first columns for comparison, indicated by τ=NO.

S2.4 Sensitivity to tuning parameters

Here, we examine the sensitivity of SOUP to its two tuning parameters, ϵ and γ. Recall that ϵ represents the smallest proportion of per-type pure cells, $\min_k |\mathcal{I}_k|/n$, and γ represents the proportion of pure cells, $|\mathcal{I}|/n$. Following the previous simulation settings, we examine two scenarios: (i) without dropouts, and (ii) with an average of 30% artificial dropouts. For simplicity, we focus on the scenario of $\alpha_0 = 1$ and $s_0 = 10^4$, and use $\tau = 0$ for artificial dropouts. Note that there are 300 pure cells, therefore, as $n_{mix} \in \{100, 300, 500\}$, the optimal tuning parameters are $\epsilon^* \in \{0.25, 0.167, 0.125\}$ and $\gamma^* \in \{0.75, 0.5, 0.375\}$, respectively. We present the average L_1 losses of estimating Θ when using $\epsilon \in \{0.05, 0.1, 0.15, 0.2\}$ and $\gamma \in \{0.3, 0.5, 0.7\}$ under each scenario (Figure S3 and Figure S4). We see that SOUP is robust as long as $\epsilon < \epsilon^*$ and $\gamma \approx \gamma^*$. In particular, it is more important to choose a smaller ϵ so that the estimated extreme neighbors, $\hat{S}_i$, has a high precision in recovering S_i. Therefore, 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. On the other hand, the performance is more robust to the choice of γ, where the default choice $\gamma = 0.5$ is usually sensible .

In fact, it is usually beneficial to use a slightly more generous γ, and we see that as long as $\gamma \geq 0.5$, the performance is robust even when $\epsilon > \epsilon^*$.



(a) simple, 100 mix; $(\epsilon^*, \gamma^*) = (0.25, 0.75)$

(b) simple, 300 mix; $(\epsilon^*, \gamma^*) = (0.167, 0.5)$

(c) simple, 500 mix; $(\epsilon^*, \gamma^*) = (0.125, 0.375)$

(d) challenge, 100 mix; $(\epsilon^*, \gamma^*) = (0.25, 0.75)$

(e) challenge, 300 mix; $(\epsilon^*, \gamma^*) = (0.167, 0.5)$

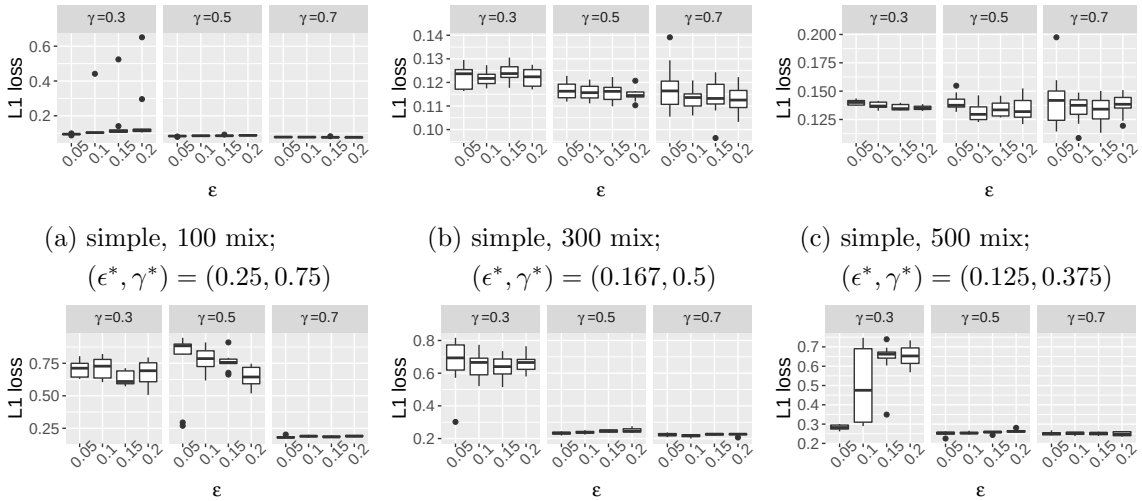
(f) challenge, 500 mix; $(\epsilon^*, \gamma^*) = (0.125, 0.375)$

Figure S3: Boxplots of average L_1 losses of SOUP estimated memberships in simulations using different tuning parameters, each repeated 10 times, under the simple scenario (**(a)-(c)**) and the challenging scenario (**(d)-(f)**). In addition to 300 pure cells, there are $\{100, 300, 500\}$ mixed cells with uniformly simulated memberships. SOUP is applied with different tuning parameters, $\epsilon \in \{0.05, 0.1, 0.15\}$ and $\gamma \in \{0.3, 0.5, 0.7\}$. The optimal parameters (ϵ^*, γ^*) under each scenario are also listed.

S2.5 Comparison to LOVE

One can potentially apply the LOVE algorithm (Bing et al., 2017) for *variable* clustering to X^T, by treating cells as "variables", and use the estimated allocation matrix as $\hat{\Theta}$. Here, we use the general routine for non-diagonal covariance matrix (corresponding to a non-diagonal Z in eq.(2)), and examine its performance in both simulations and real data.

We illustrate the performance of LOVE under the PBMC challenging scenario with $\alpha_0 = 1$ and $s_0 = 10^4$, with 10 repetitions under each setting. The LOVE algorithm estimates K, the number of clusters, as part of the procedure of finding pure cells, and requires a tuning parameter δ to offset the noise level in the data. Here, we follow the instructions in Bing et al. (2017) to search for the optimal $\delta = c\sqrt{\log p/n}$ using cross validation over a grid of $c \in \{0.005, 0.01, ..., 0.195, 0.2\}$. We see that without dropouts, LOVE successfully estimates $\hat{K}_{\text{LOVE}} = 3$ in most of the 10 repetitions, under all choices of mixed cells $n_{\text{mix}} \in \{100, 300, 500\}$ (Figure S5a). We further evaluate the L_1 losses of the LOVE estimated $\hat{\Theta}$ when $\hat{K}_{\text{LOVE}} = 3$ (Figure S5b). However, after artificial dropouts are introduced, we find LOVE to be sensitive to such noise. Here we present the results with flat dropouts (i.e., $\tau = 0$ in eq.(S12)), and LOVE substantially overestimates the number of clusters, with $\hat{K}_{\text{LOVE}} > 10$ in most repetitions (Figure S5a). Because of the substantially overestimated K,



(a) simple, 100 mix; $(\epsilon^*, \gamma^*) = (0.25, 0.75)$

(b) simple, 300 mix; $(\epsilon^*, \gamma^*) = (0.167, 0.5)$

(c) simple, 500 mix; $(\epsilon^*, \gamma^*) = (0.125, 0.375)$

(d) challenge, 100 mix; $(\epsilon^*, \gamma^*) = (0.25, 0.75)$

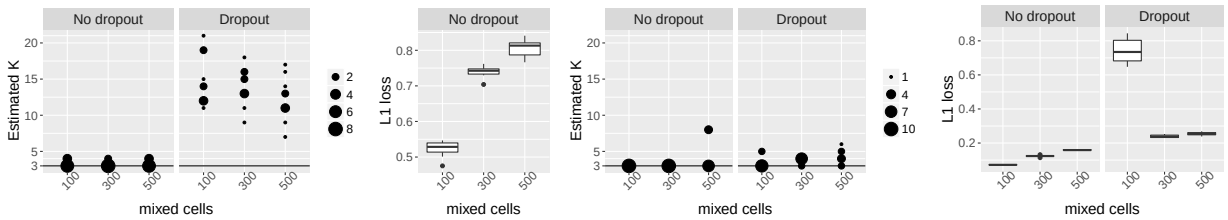
(e) challenge, 300 mix; $(\epsilon^*, \gamma^*) = (0.167, 0.5)$

(f) challenge, 500 mix; $(\epsilon^*, \gamma^*) = (0.125, 0.375)$

Figure S4: Boxplots of average L_1 losses of SOUP estimated memberships in simulations with an average of 30% artificial dropouts, under the PBMC simple scenario (**(a)-(c)**) and PBMC challenging scenario (**(d)-(f)**), each repeated 10 times. In addition to 300 pure cells, there are $\{100, 300, 500\}$ mixed cells with uniform memberships. SOUP is applied with different parameters, $\epsilon \in \{0.05, 0.1, 0.15\}$ and $\gamma \in \{0.3, 0.5, 0.7\}$. The optimal parameters (ϵ^*, γ^*) under each scenario are also listed.

the evaluation of $\hat{\Theta}$ is uninformative and we omit the results. When applied to the 7 public single cell datasets, we observe similar performance where LOVE largely overestimates K (Table S3).

On the contrary, SOUP achieves lower L_1 losses in estimating Θ when $K = 3$ (Figure S5d). For a fair comparison, we point out that it is also possible to estimate K for SOUP in a data-driven manner. In particular, we propose a cross validation procedure (details presented in Section S3), which usually selects the true $\hat{K}_{\text{cv}} = 3$ in simulations when data do not contain dropouts, and $\hat{K}_{\text{cv}} \in [3, 5]$ in most repetitions when artificial dropouts are introduced (Figure S5c). It also gives more sensible estimations of K in the 7 public datasets (Table S3).



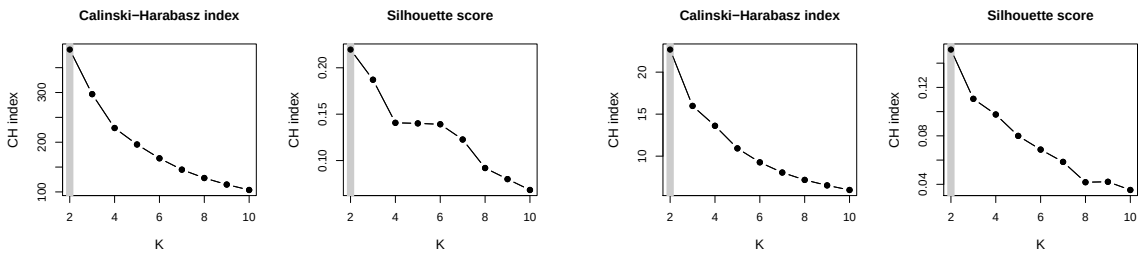
(a) LOVE $\hat{K}_{\text{LOVE}}$ (b) LOVE $\hat{\Theta}$ (c) SOUP $\hat{K}_{cv}$ (d) SOUP $\hat{\Theta}$

Figure S5: Comparison of LOVE and SOUP under the challenging simulation scenario with $\alpha_0 = 1$ and $s_0 = 10^4$, in 10 repetitions. We examine the scenario with and without artificial dropouts, where the dropout probability follows eq.(S12) with $\tau = 0$. **(a)** The estimated $\hat{K}_{\text{LOVE}}$ from LOVE. **(b)** Average L_1 losses of LOVE estimated $\hat{\Theta}$ when $\hat{K}_{\text{LOVE}} = 3$. The scenario with dropouts is omitted because of the overestimated K's. **(c)** The optimal $\hat{K}_{\text{cv}}$ selected by 5-fold cross validation of SOUP. **(d)** Average L_1 losses of SOUP estimated $\hat{\Theta}$ with input $K = 3$.

S3 Selecting K

S3.1 Selecting K from pure cells

One of the main challenges of SOUP is how to select the number of clusters, K. A simple solution is to focus only on the pure cells, treat it as a hard clustering problem, and apply standard selection techniques. For example, one can apply K-Means to the pure cells with a sequence of different K's, and choose the optimal K according to certain metrics. Here, we examine the performance of two widely used metrics: (i) Calinski-Harabasz (CH) index (Caliński & Harabasz, 1974), which is used to identify the number of clusters in CIDR (Lin et al., 2017), and (ii) Silhoutte score (Rousseeuw, 1987), another popular metric. Both metrics select K that achieves the highest score, where CH index measures the normalized ratio between between-cluster and within-cluster variations, and Silhouette score measures how points are similar to its own cluster compared to other clusters. However, when applied to the single cell datasets, both metrics reveal only the major clusters, leading to $K = 2$ in the two brain datasets (Figure S6). We will show in Section S3.3 that this is true in other public single-cell datasets as well (Table S3).



(a) Mouse brain

(b) Fetal brain

Figure S6: Calinski-Harabasz index and Silhoutte score for selecting the number of clusters in two brain single cell datasets, where $\hat{K} = 2$ is always selected. The scores are computed using K-means hard-clustering results, applied on the pure cells identified by SOUP. **(a)** 3,005 mouse brain cells. **(b)** 220 fetal brain cells.

S3.2 Selecting K with cross validation

The cross validation idea has been recently applied to hard clustering to select the cluster number (Fu & Perry, 2017). Here, we propose a similar procedure for our soft clustering problem. Specifically, we randomly split the samples into 10 folds, and treat one fold as the testing set at a time. For each choice of K, SOUP is applied to the training samples X_{TRN}, and the estimated cluster centers $\hat{C}_{TRN}$ are obtained using least squares. After then, we estimate the membership matrix $\hat{\Theta}_{TST}$ for the testing samples X_{TST} by first solving the unconstrained optimization problem:

$$\min_{\Theta_{TST}} ||X_{TST} - \Theta_{TST}\hat{C}_{TRN}^T||_F^2 \,,$$

and then normalize to get the proper membership matrix $\hat{\Theta}_{TST}$. The prediction errors of the testing samples are calculated as

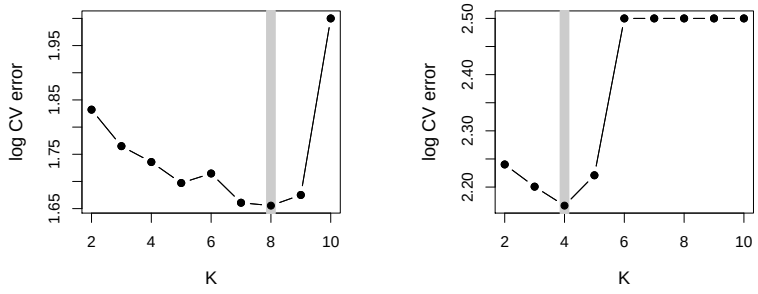
$$CV_{\text{error}} = \frac{1}{n_{TST} \cdot p}||X_{TST} - \hat{X}_{TST}||_F^2, \text{ where } \hat{X}_{TST} = \hat{\Theta}_{TST}\hat{C}_{TRN}^T \,.$$

Finally, the cross validation (CV) error is averaged across folds. To account for the stochastic nature of the splitting, we repeat this procedure for 10 times, and the K that achieves the smallest average CV error is selected.

We test this procedure in simulations under the challenging scenario (Section S2.1), with $\alpha_0 = 1$ and $s_0 = 10^4$. Note that we do not perform gene selection in simulations, and SOUP is applied to the full set of over 13,000 genes, which is much more computationally demanding than in general applications. To reduce the computational load, we illustrate the performance with 5 folds in 5 repetitions (Figure S5c). Our cross validation procedure, applied over $K \in \{2, ..., 10\}$, usually selects the true $\hat{K}_{cv} = 3$ when the simulated data do not have dropouts. The only exception is when there are 500 mixed cells, where $K = 3$ and 8 are always the two choices with the lowest CV errors, and in some repetitions $\hat{K}_{cv} = 8$ is selected. With artificial dropouts, the selected K is still close to the truth, with $\hat{K}_{cv} \in [3, 5]$ in most repetitions.

We further apply this cross validation procedure to the two brain single cell datasets, and obtain sensible results, with $\hat{K}_{cv} = 8$ for mouse brain data (Zeisel et al., 2015) and $\hat{K}_{cv} = 4$ for fetal

brain data (Camp et al., 2015) (Figure S7). Note that for the mouse brain data, $K = 7$, the presumed true number of clusters, achieves the second smallest CV error. For the fetal brain data, $K = 4$ successfully distinguishes two subtypes of neuronal progenitors as well as early and matured neurons (Figure 5a).



(a) Mouse brain (b) Fetal brain

Figure S7: 10-fold cross validation errors of SOUP, averaged across 10 repetitions, for selecting the number of clusters in two single cell brain datasets, plotted in log scale. For visualization purpose, the values are capped at 2 for mouse brain data, and 2.5 for fetal brain data. **(a)** 3,005 mouse brain cells, where $\hat{K}_{cv} = 8$ is selected. **(b)** 220 fetal brain cells, where $\hat{K}_{cv} = 4$ is selected.

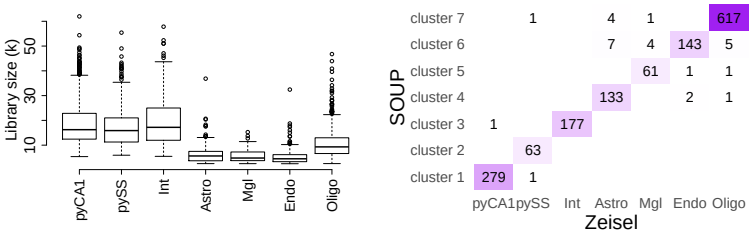
S3.3 Benchmarking on public datasets

Finally, we benchmark different selection procedures in the 7 public datasets with known labels that were used in Section 3, and we focus on the set of selected informative genes via DESCEND and SPCA (Table S1). As in the main text, we use $\gamma = 0.8$ for SOUP to pick 80% of the cells as pure. We compare the selected K among $\{2, 3, ..., 20\}$ using (i) optimal CH index or Silhouette score computed with K-means hard-clustering results of the pure cells, and (ii) minimal 10-fold CV error. We then apply SOUP clustering using the selected K, and compute the Adjusted Rand Index (ARI), using the reference labels as gold standard (Table S3). Again, we see that standard selection metrics substantially underestimates the number of clusters, revealing only major split of cell types with poor ARI. On the contrary, the cross validation procedure gives more sensible estimates that are close to the reference, and lead to similar ARI as if the true K is given. In fact, in the six pancreatic datasets from Baron et al. (2016), using $\hat{K}_{cv}$ usually leads to even higher ARI than using the reference K (Table S1, Table S3). This is probably because there are several reference cell types with less than 5 cells, which are difficult to separate as distinct clusters.

S4 Supplementary Results

S4.1 Discussion of mouse brain data

Major types In the main text, we suspect three reasons for SOUP being slightly different from the Zeisel labels in Zeisel et al. (2015). Figure S8a illustrates that astrocytes, microglia, and endothelial cells usually have smaller library sizes, when compared to other cell types, hence can be more difficult to separate. In addition, Figure S8b visualizes the SOUP hard assignments of the top 50% pure cells with the highest purity scores, which is highly consistent with Zeisel labels, suggesting the inconsistency between SOUP and Zeisel labels mostly occurs in mixed cells.



(a) library sizes (b) pure cells

Figure S8: Library sizes of 3,005 mouse brain cells, and SOUP assignments of the 50% pure cells.. **(a)** Library sizes of 3,005 mouse brain cells on 16,450 genes, grouped by their Zeisel labels in Zeisel et al. (2015). **(b)** Contingency table of the top 50% pure cells with the highest purity scores, identified by SOUP.

Next, we examine major cluster 2 and cluster 7 of SOUP. We focus on the 19 marker genes for pyramidal SS, endothelial cells, astrocytes, and oligodendrocytes (Table S2), and examine their expression levels in the 1,211 cells with inconsistent SOUP and Zeisel labels, including (i) Zeisel pyramidal SS and endothelial cells in SOUP cluster 2, and (ii) Zeisel oligodendrocytes, pyramidal SS, and astrocytes in SOUP cluster 7 (Figure S9). We see that the 16 Zeisel endothelial cells in SOUP cluster 2 express several pyramidal markers, especially MEF2C, so are many of the oligodendrocytes in SOUP cluster 7. In addition, the Zeisel pyramidal SS cells and astrocytes in SOUP cluster 7 express many of the oligodendrocyte markers, including MBP and OLIG1. These observations suggest that SOUP indeed reveals sensible structures in the data.

Subtypes The 3,005 mouse brain cells are further divided into 47 subtypes by BackSPIN in Zeisel et al. (2015) (Table S4). To examine these subtypes, we re-run SOUP, including the gene selection procedure, within each of the 7 SOUP major clusters individually. Each of the SOUP major cluster is mapped to one of the Zeisel major types with the largest overlap, as shown in Figure 4b. The cells that have inconsistent major types between SOUP and Zeisel labels are referred to as "other". In some major types, there are some cells that do not belong to any Zeisel subtypes in Zeisel et al. (2015), and are labeled as "none". The input K for SOUP is set to be one more than the number of Zeisel subtypes, to account for the "other" type. For comparison, we also list the optimal K selected by cross validation of SOUP, which is usually close to the Zeisel subtype numbers (Table S4). Figure S10 visualizes the comparison between SOUP

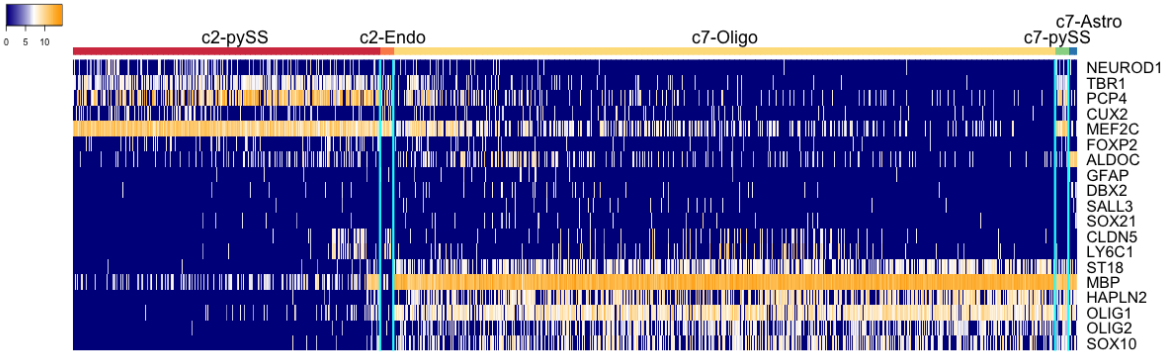


Figure S9: Expression levels of 19 marker genes in 1,211 mouse brain cells in SOUP cluster 2 and cluster 7, with inconsistent SOUP and Zeisel labels. Gene expressions have been normalized and log-transformed. One column per cell, where the color bar on the top indicates groups, indicating from left to right: 371 pyramidal SS cells in cluster 2, 16 endothelial cells in cluster 2, and 798 oligodendrocytes in cluster 7, 16 pyramidal SS cells in cluster 7, and 10 astrocytes in cluster 7.

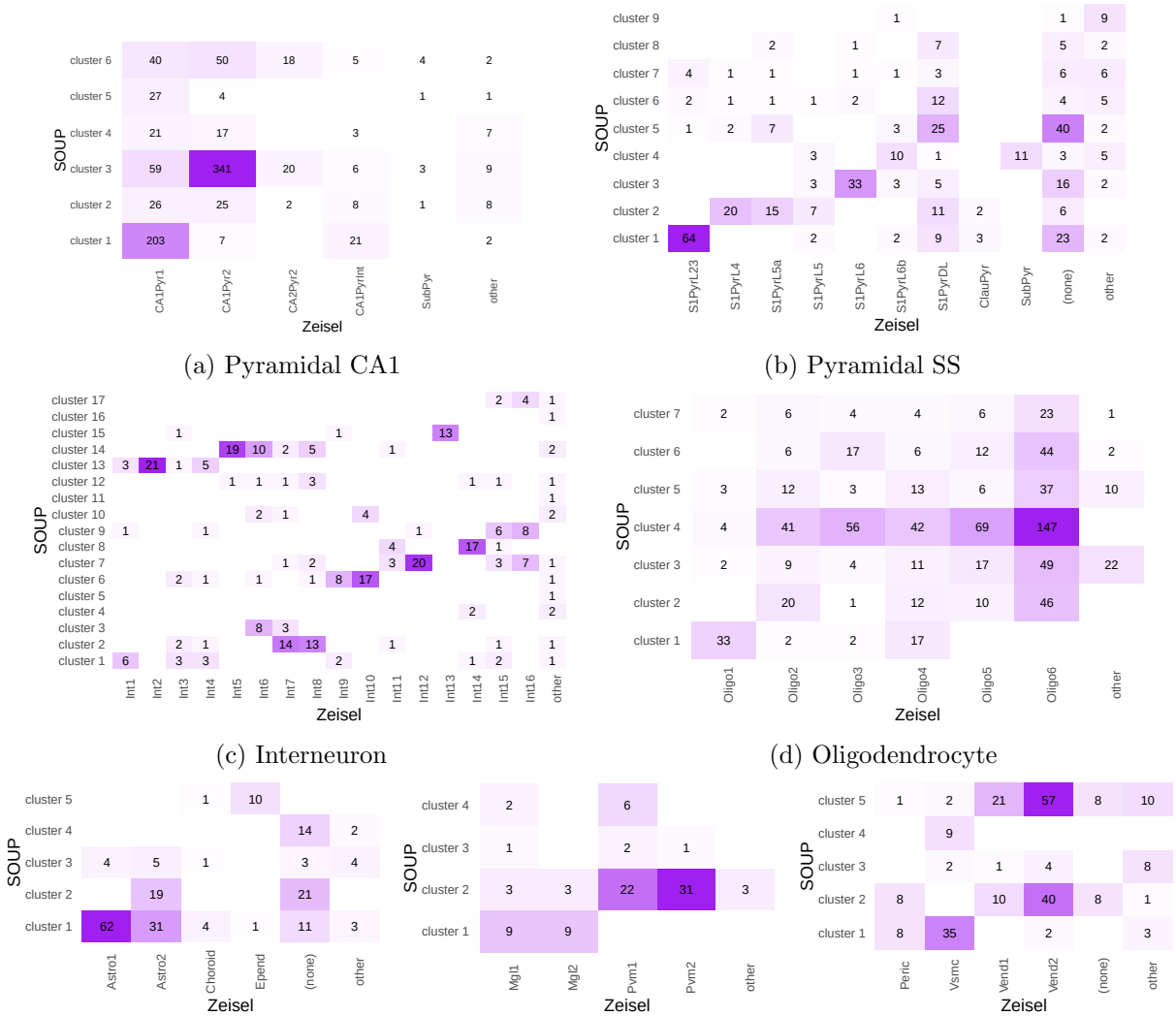
hard clusters and Zeisel subtypes, where SOUP clusters are ordered by picking cluster 1 as the starting point, and consecutively finding the next closest cluster with the highest center correlation. We see that SOUP successfully reveals many subtypes of pyramidal SS cells (Figure S10b) and interneurons (Figure S10c), although it tends to collapse some of the Zeisel subtypes. It separates the two microglia subtypes from perivascular macrophage (pvm) subtypes (Figure S10f), the two endothelial subtypes from pericytes and vascular smooth muscle cells (Vsmc) (Figure S10g), and the two astrocyte subtypes from choroid and ependymal cells (Figure S10e). For pyramidal CA1 cells, SOUP separates the majority of the two large Zeisel subtypes, but also creates further splits within CA1Pyr1 (Figure S10a).

As discussed in the main text, oligodendrocytes exert different results between Zeisel subtypes and SOUP (Figure S10d). We already illustrate that the SOUP trajectory (Figure 4d) better characterizes the maturation procedure of oligodendrocytes, when compared to the Zeisel subtype assignments (Figure 4e). For a complete comparison, here we visualize the 798 Zeisel oligodendrocytes in the same principal component subspaces, and examine the estimated trajectory of Monocle (Trapnell et al., 2014) when using the preferred DDRTree method on the same 1,220 genes as for SOUP (Figure S11a), as well as the SOUP hard assignments (Figure S11b). As mentioned in the main text, Monocle trajectory only reveals the information along PC1, and SOUP hard assignments reveal sensible structures in all dimensions.

S4.2 Discussion of fetal brain data

We see in the main text the SOUP identified two instead of three neuron subtypes. To validate this finding, we again examine the expression levels of the 12 marker genes (Table S2) in the 220 fetal brain cells, where the differentiation among the three Camp neuron subtypes is ambiguous (Figure S12).

Next, we examine the SOUP estimated developmental trajectory. We evaluate the change of



(a) Pyramidal CA1

(b) Pyramidal SS

(c) Interneuron

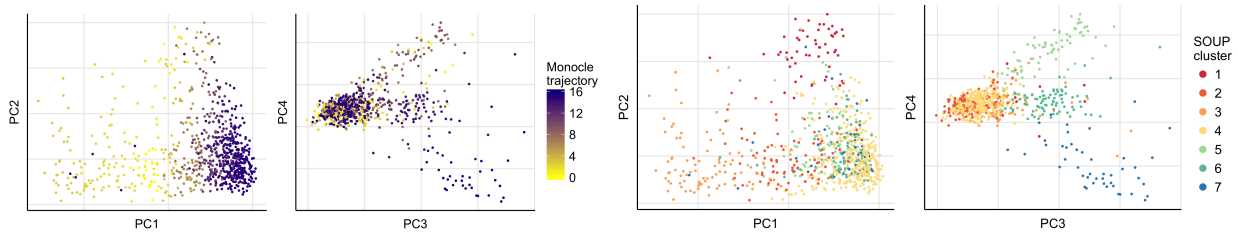
(d) Oligodendrocyte

(e) Astrocyte

(f) Microglia

(g) Endothelial

Figure S10: SOUP fine-scaled clustering results of mouse brain cell subtypes, applied within each of the seven major clusters, compared to the Zeisel subtypes identified in Zeisel et al. (2015). Each SOUP major cluster is mapped to one of the Zeisel major types with the largest overlap. Cells that belong to other Zeisel major types, i.e., inconsistent between SOUP and Zeisel major clusters, are labeled as "other". Some cells are not assigned to any Zeisel subtypes, and are referred to as "none". The subiculum pyramidal (SubPyr) cells are assigned under the major Pyramidal CA1 type in Zeisel et al. (2015), but we find them to be mainly clustered together with Pyramidal SS cells by SOUP. Therefore, the SubPyr subtype is also visualized for Pyramidal SS cells.



(a) Monocle trajectory

(b) SOUP cluster

Figure S11: 798 Zeisel oligodendrocytes in SOUP major cluster 7, projected on the leading principal component subspaces. Cells are colored by **(a)** Monocle estimated developmental trajectory, and **(b)** SOUP fine-scaled hard assignments.

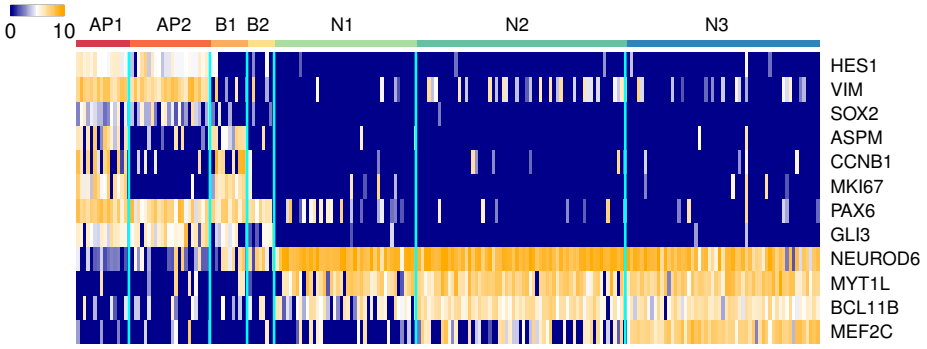
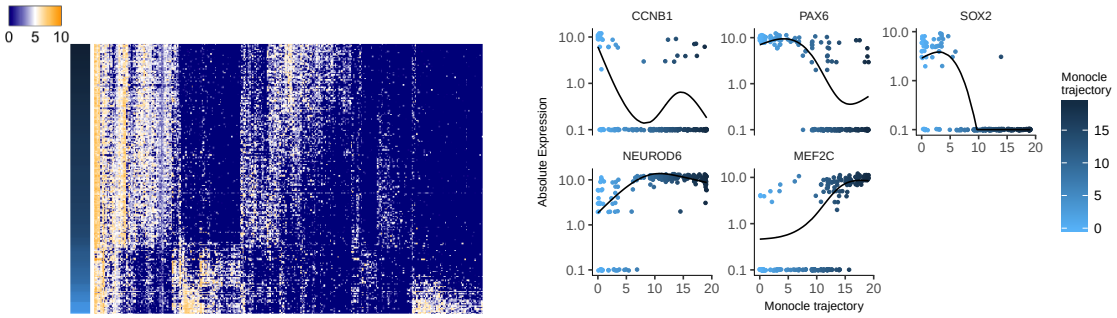


Figure S12: Expression levels of 12 marker genes in 220 fetal brain cells, where gene expression has been normalized and log-transformed. One column per cell, where cells are ordered according to Camp labels in Camp et al. (2015), from left to right: AP1, AP2, BP1, BP2, N1, N2, and N3.

expression profiles of the 315 PC genes identified in Camp et al. (2015), along the SOUP trajectory (Figure S13a). We observe a smooth transition along different periods that is consistent with the developmental order, suggesting the SOUP estimation is sensible. Finally, we compare our results to the trajectory estimated by Monocle (Trapnell et al., 2014) using the preferred DDRTree method, where it is applied on the same set of 430 selected genes as for SOUP. Figure S13b presents the gene expression trajectories of the 5 anchor genes, when the 220 fetal brain cells are ordered according to the Monocle estimated trajectory.



(a) SOUP trajectory

(b) Monocle trajectory

Figure S13: Expression levels in 220 fetal brain cells along SOUP and Monocle estimated developmental trajectories. **(a)** Expression levels of the 315 PC genes identified in Camp et al. (2015), visualized in log scale, where 220 fetal brain cells on the rows are ordered according to SOUP developmental trajectory, indicated by the color bar on the left. **(b)** Expression levels of 5 anchor genes in 220 fetal brain cells, visualized in log scale, where cells are ordered by Monocle developmental trajectory.

S5 Supporting Tables

Table S1: Major characteristics of 7 public datasets, including human brain cells from Darmanis et al. (2015), and pancreatic cells from four human donors and two mouse strains from Baron et al. (2016). For SOUP, we list the number of selected genes by SPCA and DESCEND, the ARI using two different $\gamma \in \{0.5, 0.8\}$, as well as the total runtime including gene selection and clustering, where DESCEND is paralleled with 5 cores, and the computation is conducted on a linux computer equipped with AMD Opteron(tm) Processor 6320 @ 2.8 GHz.

	Dataset characteristics			SOUP			
	K	# cells	# genes	# selected genes	runtime (sec)	ARI (γ=0.5)	ARI (γ=0.8)
Darmanis	8	420	22,085	1,106	40	0.8890	0.9071
Baron human1	14	1,937	20,125	1,632	227	0.7513	0.7519
Baron human2	14	1,724	20,125	1,571	188	0.7946	0.7502
Baron human3	14	3,605	20,125	1,923	848	0.4071	0.5030
Baron human4	14	1,303	20,125	1,516	112	0.7756	0.5092
Baron mouse1	13	822	14,878	895	50	0.5782	0.4541
Baron mouse2	13	1,064	14,878	1,324	78	0.6280	0.4093

Table S2: The 12 Marker genes for fetal brain (Camp et al., 2015) and 41 Marker genes for mouse brain (Zeisel et al., 2015) that are selected by DESCEND and SPCA.

Dataset	Cell type(s)	Marker genes
Fetal brain	AP	HES1, VIM, SOX2
	AP & BP	ASPM, CCNB1, MKI67, PAX6, GLI3
	Neuron	NEUROD6, MYT1L, BCL11B, MEF2C
Mouse brain	Pyramidal CA1	SPINK8, POU3F1, LMO1
	Pyramidal SS	NEUROD1, TBR1, PCP4, CUX2, MEF2C, FOXP2
	Interneurons	PNOC, GAD1, GAD2, TOX3, SOX6, NPAS1, LHX6, PAX6, DLX2, DLX1, ELAVL2
	Astrocytes	ALDOC, GFAP, DBX2, SALL3, SOX21
	Microglia	MAF, FBLIM1, NLRP3, SPI1, AIF1, CX3CR1, MRC1, LYVE1
	Endothelial	CLDN5, LY6C1
	Oligodendrocytes	ST18, MBP, HAPLN2, OLIG1, OLIG2, SOX10

Table S3: Selected K and corresponding SOUP ARI on 7 public datasets Baron et al. (2016); Darmanis et al. (2015). We select the optimal K among $\{2, ..., 20\}$, using (i) optimal Calinski-Harabasz (CH) index or Silhouette score, computed from K-means hard-clustering results of the pure cells, where pure cells are identified by SOUP with $\gamma = 0.8$; and (ii) minimal 10-fold cross validation (CV) error of SOUP. We further compare the ARI of SOUP hard assignments using different selected K's. Finally, we present the estimated K from LOVE, where the tuning parameter is selected by cross validation over grid $c \in \{0.005, 0.01, ..., 0.2\}$, and the number of cross validation is set to the default choice of 50. All algorithms are applied to the selected genes by DESCEND and SPCA.

Dataset	Selected K				SOUP ARI with selected K			LOVE
	Reference	CH index	Silhouette	CV	CH index	Silhouette	CV	Est. $\hat{K}$
Darmanis	8	3	2	5	0.5073	0.1844	0.8278	35
Baron human1	14	2	4	9	0.3433	0.3908	0.8121	99
Baron human2	14	2	3	8	0.3329	0.3652	0.9009	108
Baron human3	14	2	2	12	0.3944	0.3944	0.5938	171
Baron human4	14	2	2	9	0.3439	0.3439	0.6871	74
Baron mouse1	13	2	2	7	0.5411	0.5411	0.8533	58
Baron mouse2	13	2	2	9	0.2823	0.2823	0.6067	49

Table S4: 47 Zeisel subtypes of mouse brain cells identified in Zeisel et al. (2015), as well as the optimal $\hat{K}_{cv}$ estimated by 10-fold cross validation of SOUP, searched over $K \in \{2, ..., 20\}$.

Major type	Zeisel K	SOUP $\hat{K}_{cv}$	Zeisel subtypes
Pyramidal CA1	5	6	CA1Pyr1, CA1Pyr2, CA2Pyr2, CA1PyrInt, Subiculum (SubPyr)
Pyramidal SS	8	7	S1PyrL23, S1PyrL4, S1PyrL5a, S1PyrL5, S1PyrL6, S1PyrL6b, S1PyrDL, Claustrum (ClauPyr)
Interneuron	16	7	Interneuron1, ..., Interneuron16
Astrocyte	4	6	Astrocyte1, Astrocyte2, Choroid, Ependymal
Microglia	4	4	Microglia1, Microglia2, Perivascular macrophage 1 (Pvm1), Pvm2
Endothelial	4	3	Endothelial1, Endothelial2, Pericyte, Vascular smooth muscle cell (Vsmc)
Oligodendrocyte	6	7	Oligo1, ..., Oligo6