Prior knowledge and sampling model informed learning with single cell RNA-Seq data

Sumit Mukherjee[1], Yue Zhang[2], Sreeram Kannan[1]*, Georg Seelig[1,2]*

[1]Department of Electrical Engineering, University of Washington, Seattle, WA, USA.

[2]Paul G. Allen School of Computer Science and Engineering, University of Washington, Seattle, WA, USA.

*emails: *ksreeram@uw.edu*, *gseelig@uw.edu*

Abstract

Single cell RNA-seq (scRNA-seq) experiments can provide a wealth of information about heterogeneous, multi-cellular systems. However, this information has to be inferred computationally from sequencing reads which constitute a sparse and noisy sub-sampling of the actual cellular transcriptomes. Here we present UNCURL, a unified framework for scRNA-seq data visualization, cell type identification and lineage estimation that explicitly accounts for the sequencing process. The main algorithmic novelty is a non-negative matrix factorization method that uses knowledge of the distribution resulting from the sequencing process to more accurately model the underlying cell state matrix. We also develop a systematic way for incorporating prior biological information such as bulk RNA expression profiles into the cell state matrix. We find that UNCURL dramatically improves performance over state-of-the-art methods both in the absence and presence of prior knowledge. Finally we demonstrate that using UNCURL as a data preprocessing tool significantly improves the performance of existing scRNA-seq analysis algorithms.

Introduction

High-throughput scRNA-seq technologies[1–4] can provide biological insights such as revealing cell type composition[5,6], cell lineage relationships[7–11] or even spatial relationships[12,13] between cells in heterogeneous multi-cellular systems. Enabling such insights are two key advantages of single cell transcriptomic datasets. First, having information about individual cells helps avoid aggregation and conflation of traits from disjoint groups of cells within a mixed sample[14]. Second, scRNA-seq provides very large sample size, both in terms of the number of cells and genes that can be assayed, compared to other methods with single-cell resolutions. However, advanced computational methods are required to extract latent biological information from the raw read-counts which provide only a heavily sampled version of the full cellular transcriptome[15,16].

Most commonly used computational tools for cell type identification[10,17], lineage estimation[7–11] and similar applications rely on an initial dimensionality reduction step using methods such as PCA[18] or tSNE[19]. However, these algorithms assume that the underlying data is drawn from a Gaussian or a t-distribution, an assumption that does not hold for scRNA-seq data[20]. The discrepancy between the assumed and actual distribution fundamentally limits the accuracy of the resulting predictions. Moreover, existing methods rely almost exclusively on unsupervised learning and do not incorporate useful and commonly available prior information such as bulk gene expression data or cell type specific marker genes to guide the analysis process. While there is a simple way to utilize prior knowledge with existing algorithms by using the known gene expression vectors for initialization, the variability in data type and quality severely restricts the utility of such initialization.

Here, we introduce UNCURL, a **un**ified **c**omp**u**tational framework for sc**R**NA-seq data processing and **l**earning that addresses these shortcomings. Moreover, unlike prior methods, UNCURL jointly tackles all major unsupervised learning tasks commonly used in the context of scRNA-seq data. An overview of the algorithmic workflow of UNCURL can be seen in **Figure 1 A**. The main technical contribution of UNCURL is a generalized non-negative matrix factorization (NMF) that explicitly accounts for the Poissonian or negative binomial sampling distribution. Our algorithm exploits the low-dimensional nature of the true biological state matrix, i.e. it assumes that each cell is in a convex combination of a few archetypal cell-states. Under this assumption, the true state matrix can be expressed as a product of an archetypal main state matrix, comprising of gene-expression in the archetypal states, and a matrix of mixing coefficients. UNCURL's downstream algorithms exploit these lower-dimensional matrices for unsupervised tasks such as visualization, clustering and lineage estimation. Working with the estimated (and factorized) true state matrix considerably improves performance compared to state-of-the-art methods for the same applications that operate directly on the sequencing data.

Additionally, UNCURL allows for the integration of prior information which leads to large improvements in accuracy. To enable semi-supervised learning, UNCURL's toolbox contains a method (qualitative normalization, qualNorm) for standardizing any prior biological information including bulk RNA-seq data, microarray data or even information about individual marker gene expression to a form compatible with scRNA-Seq data. We demonstrate that initialization using prior knowledge in an appropriately standardized manner dramatically improves performance compared to unsupervised learning.

Finally, UNCURL has a pre-processing mode, where it takes in the gene-expression matrix and any prior biological information and outputs the estimated state matrix. This estimated state matrix can be utilized as input (in lieu of the observed gene-expression matrix) by any existing unsupervised learning algorithm. With the rapid growth of algorithms designed for each of the specialized tasks of clustering, visualization, lineage reconstruction as well spatial estimation, UNCURL preprocessing can enable these specialized algorithms to benefit from the detailed modeling of the sequencing process, as well as considerable prior information and the regularization afforded by the convex mixture assumption in UNCURL. We demonstrate that UNCURL pre-processing significantly improves the performance of these downstream algorithms on these learning tasks.

Results

Estimated transcriptomic states

An implicit assumption shared by many scRNA-seq data analysis tools is that any biological sample contains a limited number of cell types and that any individual cell can be considered a “mixture” of these cells. Here, we make this “convex mixture model” explicit which allows us to apply NMF to the estimated cell state matrix. While NMF is well studied when the entries have Gaussian noise[21], in scRNA-seq, the sequencing process produces noise approximately following a Poisson or Negative Binomial distribution (potentially with zero-inflation[22]). While the sampling distribution is carefully modeled in differential expression studies[23], the most commonly used algorithms for visualization, cell-type identification as well as lineage tracing do not account for this model. Thus, while factoring the matrix, we need to account for the sampling distribution in order to estimate the true cell-state matrix as well as the mixing coefficients accurately from the observed gene expression matrix.

The sampled matrix factorization algorithm in UNCURL (**Figure 1 B**) takes the gene expression matrix as input and and alternatively estimates the two-matrices using the likelihood score under the known sampling model, a generalization of the popular Lee-Seung algorithm[21]. Each step is convex and can be solved using a regular gradient-descent based solver[24]. This factorization is exploited by all the downstream steps in UNCURL. Since alternating optimization algorithms are guaranteed to achieve only local minima, a good initialization is paramount in achieving good performance[25]. We initialize our algorithm using a Poisson version of the K-means++ algorithm (see Online Methods for details).

Dimensionality reduction with UNCURL

A typical first-step in scRNA datasets is to reduce the dimension of the data, from tens-of-thousands (i.e. the number of genes) to 2 or 3, in order to aid visualization. UNCURLs dimensionality reduction approach takes advantage of matrix factorization (as seen in **Figure 2 A**) by first projecting only the archetypal state matrix to the reduced dimension (using the multi-dimensional scaling or MDS algorithm). Because the number of archetypal states is typically several orders of magnitude smaller than the number of individual cells, the projection is more robust and computationally simpler. In a second step, low dimensional cell states are generated for all cells simply by taking the appropriate convex combination of low-dimensional representation of archetypal states. We hypothesize that the principled modeling of the data by the sampled matrix factorization should result in a better dimensionality reduction than existing methods.

To test the accuracy of our dimensionality reduction approach, we created a synthetic, standardized dataset using bulk data from mouse embryonic stem cells and differentiated fibroblasts[26]. We first simulated intermediate true transcriptomic states by generating hundred equally spaced points at convex combinations between these two ‘main states’. The cells are divided into 4 intermediate stages depending on the distance between the two extreme points. We simulate the observed “RNA-seq data” by Poisson sampling the true (synthetic) data as explained in Supplementary Methods.

In order to quantify the accuracy of different dimensionality reduction algorithms, we observe that good dimensionality algorithms should place similar cell-types together. Therefore, we define an error metric: the probability that a cell and its closest neighbor do not belong to the same cell type. We then use UNCURL to reduce the dimensionality assuming both Poisson and

Gaussian sampling distributions as seen in **Figure 2 B**. While both approaches lead to qualitatively good visualizations and lower error scores compared to off-the-shelf approaches such as tSNE and PCA, using the correct sampling distribution leads to the lowest error rate (mean error values over multiple runs are calculated for each algorithm). We furthermore note that UNCURL representations lie on a straight line, since there are only two archetypical states and UNCURL estimates all other states as convex combinations of such states.

Next we tested our dimensionality reduction approach on an actual RNA-seq dataset comprising of mouse embryonic stem cell and differentiated fibroblast cells collected two days apart and sequenced together[27]. As expected, the different cells lie on a continuum in all dimensionality reduction methods as seen in **Figure 2 C**. As pointed out earlier, UNCURL places all points in a line since there are two states, consistent with the biological interpretation that cells are ordered along the differentiation trajectory. Again, UNCURL has the lowest error score.

Having demonstrated UNCURL's effectiveness on a simple dataset comprising of two cell types, we next tested our approach on a more complex dataset collected from mouse brain and comprising of several different labeled cell types[5]. Considering the main non-pyramidal cell types leaves us with five distinct cell types namely oligodendrocytes, astrocytes, interneurons, microglia and endothelial cells. Unlike the previous example, these cell types are distinct mature cell types and we might expect clearly distinct clusters upon dimensionality reduction. Upon comparing the visualization for this dataset with different approaches we see that UNCURL has the best error-score. Both UNCURL and tSNE lead to clear separation of cell types in the low dimensional representation while PCA results in overlapping clusters (**Figure 2 D**).

Finally, we consider a dataset consisting of four cell types corresponding to different stages of olfactory neurogenesis[28]. This dataset has properties of both previous datasets in that there are more than two states but they are on a continuum. Consistent with the underlying data, all methods lead to overlapping low dimensional representations for the different cell types (**Figure 2 E**). While UNCURL cannot fully separate all cell types for this dataset, it correctly orders the clusters according to their degree of differentiation. Comparing the last two datasets, tSNE does well in the former but not in the latter. We observe that tSNE preserves local distances while deemphasizing farther distances, and this approach works well when the data has segregated clusters (former dataset), but fails when the data lie along a continuum (latter dataset). In comparison, UNCURL is designed with the convex mixtures assumption that makes it more universally applicable.

Prior knowledge improves UNCURL

While UNCURL is able to achieve very low error rates in the first two datasets, the error rate in the last dataset leaves room for improvement. This opens up a more general question: can one exploit prior knowledge of cell types to improve the state estimation in UNCURL? In principle, incorporating prior information about cell states should improve the performance but a major issue in using such information is the incompatibility between different data types (e.g. FISH images or microarray data with RNAseq data) and variability between experiments using the same technique (e.g. bulk RNA-seq batch effects). Because of this concerns, there presently exists no general framework to utilize information available in these different forms for the purposes of semi-supervision.

Here we develop such a framework called ’qualitative-normalization (qualNorm)’ (**Figure 3 A)**, which can be used to convert prior cell type-specific information into a form that is compatible with UNCURL and other algorithms. This information is expected to be in the form of gene expression data and can come from a variety of sources such as bulk RNA-seq, microarrays, or can even be qualitative prior knowledge about marker genes expressed in the form of a binary matrix. The basic premise of the qualNorm framework is the following: although the measured gene expression might vary between data sources due to biases, the qualitative information being conveyed should be preserved between assays and experiments.

Therefore, qualNorm proceeds through two main steps. First, the original data regardless of type and origin is converted into binary matrix form for a subset of high-confidence genes (i.e. a given gene is either “ON” or “OFF”). These high-confidence genes can be found, for example, through a differential expression analysis on the original data-type. This binary information cannot be directly imported into UNCURL; therefore in the second step, we convert these qualtitiative scores back into quantitative data using information in the observed scRNA-seq gene expression data.

For each gene of interest, qualNorm clusters the gene expression values in the observed scRNA-seq dataset into two clusters. These clusters correspond to the high and low expression clusters for the gene of interest. The cluster centers of the ‘high’ and ‘low’ clusters can then be seen as the expected ON/OFF value for this marker gene. Hence, our output matrix replaces the binary values with the corresponding high/low value for each gene. Thus the output of the qualNorm framework is a partial archetype matrix with some subset of genes and cell-types filled out with numerical values. This information is then used as an initialization to seed the sampled matrix factorization algorithm in UNCURL. A detailed illustration of this method can be seen in **Supplementary Figure 3.**

To demonstrate the utility of semi-supervision, we revisit the dimensionality reduction problem. Specifically, we focus on the data set of Hanchate et al., where all dimensionality reduction algorithms had relatively poor cluster separation. An upper bound on the performance with semi-supervision information is obtained when we feed the *aggregate means* of the true clusters (inferred from ground-truth labels) as the initialization. In order to test the validity of our qualNorm framework, we compare the performance with aggregate-mean initialization to the performance obtained when we process these aggregate means through the qualNorm framework. In **Figure 3 B**, the two algorithms are compared, and it is seen that semi-supervision even when qualitative has a significant impact on performance. Moreover, the visualization obtained using the qualitative means is strikingly similar to those obtained using aggregate means. This demonstrates the potential of our qualNorm framework, and we perform tests with more realistic supervision information in the next section on clustering.

Improving clustering with UNCURL

Clustering can be seen as a special case of state estimation with the additional constraint that cells have to belong to only one cell type and cannot be a mixture of different cell types. It is easy to see that solving the sampled matrix factorization problem with this additional constraint is equivalent to performing the Poisson or Negative Binomial equivalents of k-means algorithm.

Furthermore, since sampled matrix factorization already provides us with a set of archetypal states, these can now be used as the initial centers for our clustering algorithm.

To test the efficacy of this clustering approach with and without semi-supervision, we compare the average cluster purity obtained by our approach and several other commonly used clustering approaches (namely, k-means clustering, dimensionality reduction with PCA followed by k-means and dimensionality reduction with tSNE followed by k-means) on two different datasets as seen in **Figure 3 C**. When using semi-supervision, we compared three distinct modes namely bulk “semi-supervision” where the bulk RNA-Seq data is used directly as the initial estimate; “qualitative means” where the bulk RNA-Seq is subject to qualNorm framework and “aggregate means” where the means of scRNA-seq data of true clusters are used for initialization (this information is not available in real data and is used as an indicator of potential performance).

It can be seen that UNCURL outperforms other methods significantly already on the unsupervised version of the clustering problem. On the Zeisel dataset, UNCURL achieves 91% purity compared to the 75% purity for the second best algorithm, tSNE followed by Kmeans. QualNorm semi-supervision performs quite close to the aggregate-means bound for UNCURL, and can lead to near-perfect purity on both datasets. In contrast, initialization directly with bulk gene-expression values does not offer a consistent performance improvement, sometimes, even leading to worse performance. Finally, other algorithms are unable to fully utilize the semi-supervision information even when fed with aggregate-means data. This is because these algorithms are not tuned to the true sampling distributions; in comparison, UNCURL obtains exactly 100% purity on both datasets with this information.

To understand the impact of having only partial information, we consider the clustering problem described in the previous paragraph but vary the number of known cell types that are provided. We then generate the centers corresponding to the missing or unknown cell types using our version of the k-means++ algorithm. We observe that increasing the number of known cell types leads to monotonic improvement in accuracy over the unsupervised case (**Figure 3D**). These results highlight both the flexibility of our algorithm as well as the performance gain afforded by knowing a fraction of cell types. Furthermore we tested various subset sizes for the qualitative prior information and observed that even a small subset of genes is sufficient to get excellent performance when using qualNorm (**Supplementary Figure 5**).

Our approach for semi-supervision can also be extended to other similar tasks, such as inferring the spatial location of cells[12,13]. In the supplementary methods, we have demonstrated how a slightly modified version of our clustering algorithm along with qualNorm is able to reliable estimate the location of cells in the zebrafish embryo[12].

Lineage estimation

We developed a novel lineage estimation algorithm utilizing the detailed factorization information obtained with UNCURL as seen in **Figure 4 A**. The key idea behind UNCURL lineage estimation is to first exploit dimensionality reduction and construct a tree such that most cells lie close to it in that lower dimensional space. UNCURL approaches this problem in a bottom-up manner by first clustering the cells into K groups, with each cell being allotted to the nearest archetype (here K is the number of archetypes). Inside of each group, UNCURL fits a smooth curve in order to minimize the deviation between the curve and the the points in the group; this smooth curve serves an estimate for a particular lineage. Having obtained a smooth lineage for each branch, a global lineage tree is generated by connecting each branch to its closest neighbor.

To test the accuracy of lineage estimation using UNCURL, we compared against Monocle[7] and SLICER[9], two commonly used lineage estimation tools. We applied all three tools (with UNCURL in unsupervised mode) to a human embryonic stem cell differentiation dataset[7]. This dataset is known to comprise of three main cell types, namely embryonic stem cells, interstitial cells and differentiated myoblasts. We initiated all algorithms with the correct number of estimated states for the dataset and obtained estimated lineages, as seen in **Figures 4 B-D**. All three estimated lineages look qualitatively similar with a dense concentration of day 0 cells at the beginning of the trajectory. However, by looking at the markers of interstitial and myoblast cell types, we can qualitatively tell whether the estimated lineages match prior biological knowledge.

A further validation of the estimated lineages can be found by looking at the relative expression of the cell type specific markers of interstitial mesenchymal and myoblast cells, namely PDGFRA and MYOG[7]. Here we see that both UNCURL and Monocle have PDGFRA expressed at high levels at intermediate stages of the trajectory while MYOG is highly expressed only at the end of the trajectory. This is consistent with existing knowledge about this differentiation process. Moreover, upon estimating the gene expression patterns using the pseudotime ordered cells, we see qualitatively similar expression patterns compared to those estimated with the orderings inferred from Monocle (**Supplementary Figure 7**). This provides further support to the lineage estimated using UNCURL.

To quantify the accuracy of the estimated lineages, we tested on the synthetic dataset that we used for visualization which simulates mouse embryonic stem cell differentiation. We compared the performance of the three algorithms (UNCURL, Monocle and SLICER) on the synthetic data and found UNCURL to have the highest accuracy, measured by rank correlation with the true ordering (**Supplementary Figure 6** and **Supplementary Methods**). Moreover, even with the information about the number of expected ‘main branches’, Monocle was seen to estimate a noisy trajectory with many spurious branches. While SLICER did not have this problem, its ordering accuracy was seen to be slightly inferior to UNCURL.

We generated another dataset with a tree structured lineage containing three branches, to further probe UNCURL’s prediction accuracy for branched trajectories. This can be viewed as one cell type differentiating into two distinct lineages at the branching point. We then ran all three lineage-estimation algorithms on this dataset and visually inspected the resulting trajectories. Again UNCURL is seen to result in the most faithful reconstruction of the original trajectory (as seen in **Supplementary Figure 7**). Not only is UNCURL’s estimated trajectory less noisy than those estimated by the other algorithms, but very few cells are assigned to incorrect branches.

Estimated states improve performance of prior unsupervised learning algorithms

The first key algorithmic step used in UNCURL is its ability to account for the sampling distribution and estimate the true cellular transcript levels. UNCURL's downstream algorithms then exploit a factorized representation of this estimated state matrix to deliver superior performance. We hypothesized that the sampled matrix factorization of UNCURL is an important contributor to its performance, and therefore other algorithms should be able to benefit from this step. To test this hypothesis, we utilized the estimated state matrix output by UNCURL instead of the true gene expression matrix as an input to the other algorithms. As scRNA-seq continues to grow in popularity, the newer algorithms developed for inference can potentially exploit UNCURL-preprocessing to account for sampling distribution as well as prior biological knowledge.

We outline a general purpose workflow for using UNCURL as a data pre-processing tool for existing and future analysis tools in **Figure 5 A**. The unprocessed data which comprises of both SCS data and potentially raw prior information, is first passed through the state estimation pipeline of UNCURL to obtain a new estimated state matrix. This estimated state matrix is then compatible with any unsupervised learning algorithm that takes a gene-expression matrix as input, such as PCA, tSNE, or Monocle. A crucial added benefit of using UNCURL as a pre-processing tool is the ability to use prior information with otherwise unsupervised learning algorithms.

To test the utility of UNCURL as a pre-processing tool, we compared the result of unsupervised learning with and without pre-processing on several different datasets for different learning tasks. To evaluate improvement in clustering accuracy, we compared the cluster purity for common clustering algorithms before and after UNCURL pre-processing. Additionally we performed the same clustering after semi-supervised UNCURL pre-processing. As seen in **Figure 5 B**, pre-processing using UNCURL improves the accuracy of all clustering algorithms, both with and without semi-supervision. Furthermore, many algorithms show an additional improvement in accuracy when using semi-supervised pre-processing.

We then evaluated the improvement in dimensionality reduction possible through the use of UNCURL, by visually comparing the low dimensional representation of the dataset from Zeisel et al. [5] using PCA and tSNE, with unprocessed and processed data. It can be seen in **Figure 5 C**, that dimensionality reduction after pre-processing leads to better separation of the known cell types for both algorithms. While PCA shows a remarkable improvement in separation of cell types, tSNE (which was already quite good at separating cell types) also shows an incremental improvement in performance.

To test the improvement due to pre-processing on lineage estimation, we used our synthetic embryonic stem cell differentiation dataset for which we know the true ordering (**Figure 5 D**). We then compared the inferred ordering using Monocle with and without pre-processing. We observe that the inferred lineage using Monocle has a sharp improvement in both accuracy of ordering as well reduction in spurious branches when pre-processed using UNCURL.

Discussion

In this manuscript, we introduced a unified framework for data dimensionality reduction, clustering and lineage estimation with SCS datasets. Our framework, UNCURL, takes advantage of prior knowledge about the sampling distribution of SCS data and uses this information together with a convex mixture model assumption to estimate a true state matrix from observed SCS data. UNCURL further includes a computational toolbox, qualNorm, which can be used to incorporate prior biological knowledge from various sources into an improved estimate of the true state matrix.

By comparing against several benchmarking datasets, we demonstrated that UNCURL leads to superior separation of cell types in reduced dimensions as well as higher cluster purity for clustering tasks compared to prior tools. Moreover, we demonstrate that UNCURL estimates qualitatively similar trajectories on real datasets and is quantifiably better on synthetic data than existing lineage estimation algorithms. We further showed that semi-supervision using different types of prior information can lead to further improvement in accuracy of the learning tasks. We also highlight the utility of UNCURL as a data pre-processing tool by demonstrating the improvement in performance when it is used in conjunction with common unsupervised algorithms for clustering, dimensionality reduction and lineage estimation.

While UNCURL is demonstrated to be an efficient unified framework for several both unsupervised and semi-supervised learning tasks, it still has some limitations. While our method accounts for the sampling effect on the data, we do not take into account other sources of variability such as cell cycle effects and biological noise[29]. Moreover, presently the semi-supervision framework can only process prior information that can be binarized. While this still leads to improvement in accuracy, not all genes have binary states. Future work will be aimed at developing a learning framework that account for these other sources of variability and a more inclusive semi-supervision framework.

References

1. Klein, A. M. *et al.* Droplet barcoding for single-cell transcriptomics applied to embryonic stem cells. *Cell* (2015). doi:10.1016/j.cell.2015.04.044

2. Macosko, E. Z. *et al.* Highly parallel genome-wide expression profiling of individual cells using nanoliter droplets. *Cell* **161,** 1202–1214 (2015).

3. Zheng, G. X. Y. *et al.* Massively parallel digital transcriptional profiling of single cells. *bioRxiv* **8,** 65912 (2016).

4. Grün, D. & Van Oudenaarden, A. Design and Analysis of Single-Cell Sequencing Experiments. *Cell* **163,** 799–810 (2015).

5. Zeisel, a. *et al.* Cell types in the mouse cortex and hippocampus revealed by single-cell RNA-seq. *Science (80-.).* **347,** 1138–42 (2015).

6. Baron, M. *et al.* A Single-Cell Transcriptomic Map of the Human and Mouse Pancreas Reveals Inter- and Intra-cell Population Structure. *Cell Syst.* **3,** 346–360.e4 (2016).

7. Trapnell, C. *et al.* The dynamics and regulators of cell fate decisions are revealed by pseudotemporal ordering of single cells. *Nat. Biotechnol.* **32,** 381–6 (2014).

8. Shin, J. *et al.* Single-Cell RNA-Seq with Waterfall Reveals Molecular Cascades underlying Adult Neurogenesis. *Cell Stem Cell* **17,** 360–372 (2015).

9. Welch, J. D., Hartemink, A. J. & Prins, J. F. SLICER: inferring branched, nonlinear cellular trajectories from single cell RNA-seq data. *Genome Biol.* **17,** 106 (2016).

10. Jain, A. K. & Dubes, R. C. *Algorithms for Clustering Data*. (Prentice-Hall, Inc., 1988).

11. Setty, M. *et al.* Wishbone identifies bifurcating developmental trajectories from single-cell data. *Nat. Biotechnol.* **34,** 1–14 (2016).

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

13. Halpern, K. B. *et al.* Single-cell spatial reconstruction reveals global division of labour in the mammalian liver. *Nature* (2017). doi:10.1038/nature21065

14. Blyth, C. R. On Simpson â€™ s Paradox and the Sure- Thing Principle. *J. Am. Stat. Assoc.* **67,** 364–366 (2011).

15. Trapnell, C. Defining cell types and states with single-cell genomics. *Genome Res.* **25,** 1491–1498 (2015).

16. Wagner, A., Regev, A. & Yosef, N. Revealing the vectors of cellular identity with single-cell genomics. *Nat. Biotechnol.* **34,** 1145–1160 (2016).

17. Ding, C. & He, X. K-means clustering via principal component analysis. *Twentyfirst Int. Conf. Mach. Learn. ICML 04* **Cl,** 29 (2004).

18. Abdi, H. & Williams, L. Principal component analysis. *Wiley Interdiscip Rev Comput Stat.* **2,** 433–459 (2010).

19. Van Der Maaten, L. J. P. & Hinton, G. E. Visualizing high-dimensional data using t-sne. *J. Mach. Learn. Res.* **9,** 2579–2605 (2008).

20. Grün, D., Kester, L. & van Oudenaarden, A. Validation of noise models for single-cell transcriptomics. *Nat. Methods* **11,** 637–640 (2014).

21. Lee, D. & Seung, H. Algorithms for non-negative matrix factorization. *Adv. Neural Inf. Process. Syst.* 556–562 (2001). doi:10.1109/IJCNN.2008.4634046

22. Pierson, E. & Yau, C. ZIFA: Dimensionality reduction for zero-inflated single-cell gene expression analysis. *Genome Biol.* **16,** 241 (2015).

23. Anders, S. *et al.* Differential expression analysis for sequence count data. *Genome Biol.* **11,** R106 (2010).

24. Udell, M., Horn, C., Zadeh, R. & Boyd, S. Generalized Low Rank Models. *Arxiv* **9,** 1–68 (2015).

25. Jain, P., Netrapalli, P. & Sanghavi, S. Low-rank Matrix Completion using Alternating Minimization. 1–40 (2012). doi:10.1145/2488608.2488693

26. Zhang, Y. *et al.* An RNA-sequencing transcriptome and splicing database of glia, neurons, and vascular cells of the cerebral cortex. *J. Neurosci.* **34,** 11929–11947 (2014).

27. Islam, S. *et al.* Characterization of the single-cell transcriptional landscape by highly multiplex RNA-seq. *Genome Res.* **21,** 1160–1167 (2011).

28. Hanchate, N. K. *et al.* Single-cell transcriptomics reveals receptor transformations during olfactory neurogenesis. *Science (80-.).* **350,** 1251–1255 (2015).

29. Barron, M. & Li, J. Identifying and removing the cell-cycle effect from single-cell RNA-Sequencing data. *Bioarxiv* **6,** 33892 (2016).

Online Methods

Initialization with Poisson K-means++

K-means++[30] is a widely used seeding method for the k-means algorithm, which tries to identify k points in the data which have the highest mutual separation. However, the standard version of K-means++ is built on an implicit assumption of Gaussian noise, which justifies the Euclidean distance metric utilized. To use a similar approach to our problem, we define a notion of distance between points arising from Poisson sampled data. A distance measure d is called a semi-metric if it satisfies the following properties:

$$\begin{aligned} &d(x,x) = 0 \\ &d(x,y) \geq 0, equality\ only\ when\ x = y \\ &d(x,y) = d(y,x) \end{aligned}$$

When the data follows a Poisson distribution, the most intuitive distance measure would be the Poisson log-likelihood ($ll_p(y|x)$) with the assumption that one of the data points is the mean (say *x*) and the other is the point being considered (say y). However, this distance will not satisfy any of the aforementioned. To overcome this, we then design a normalized version of this distance measure which satisfies all of these properties:

$$\begin{aligned} d(x,y) = {} & ll_p(x|x) + ll_p(y|y) - \left(ll_p(x|y) + ll_p(y|x)\right) \\ & = (x-y)\log\left(\frac{x}{y}\right) \end{aligned}$$

This distance is based on the observation that value of $ll_p(x|x)$ is maximum when $x = y$. Thus, the $d(x,y)$ quantity measures the distance from the maximum value log-likelihood value for both x and y (for the sake of symmetry). This distance then replaces the Euclidean distance used in the standard implementation of K-means++ and is used to obtain initial seeds for our state estimation. A similar method isn't possible for the negative binomial distribution, since it is not a single parameter distribution.

QualNorm semi-supervision framework

Here we describe a method to convert qualitative cell type specific information to efficient initializations for various unsupervised learning algorithms. The inputs to the framework are the following: 1) a binary matrix of dimension $B \in \mathrm{R}^{n_0 \times k_0}$, where n_0 is the number of genes for which the information is provided and k_0 is the number of cell types for which the information is provided, 2) A single cell sequenced data matrix $X \in \mathrm{R}^{n \times d}$, where n is the number of genes and d is the number of cells and 3) the number of cell types expected in the data, k. In the case where bulk information is available about the cell types, the data has been binarized by thresholding around the central value for each differentially expressed gene (this step is left to the users' discretion).

We first seek to find the expected quantitative states for each gene in the subset of the genes/cell types for which we have prior information available. We then run the Poisson k-means algorithm with $k = 2$ for each gene and obtain the values of the medians $[m_1, m_2]$ of the two predicted clusters. Having done this, a new matrix of predicted means M is then compiled in the following way:

$$[M]_{ij} = \begin{cases} \max(m_1, m_2), & if\ [B]_{ij} = 1 \\ \min(m_1, m_2), & if\ [B]_{ij} = 0 \end{cases}$$

Having now obtained a matrix of predicted means M of dimension $\mathrm{R}^{n_0 \times k_0}$, we seek to obtain the information about the other cell types (in case $k_0 < k$) and genes (in case $n_0 < n$). We first obtain information about the missing cell types by using Poisson Kmeans++ to obtain $k - k_0$ additional means, starting from the current M. We then augment these means to M making its dimension $\mathrm{R}^{n_0 \times k}$. We then proceed to obtain information about the missing genes by performing one round of Poisson k-means clustering (see below) on the subset of genes for which qualitative information is known. Once the cells have been assigned to k clusters, we take the means of all genes for the cells in these clusters. These k means then provide us with a new M of dimension $\mathrm{R}^{n \times d}$. This matrix of predicted means is now used to initialize the various downstream algorithms.

State estimation with Sampled Matrix Factorization

For the task of lineage estimation, UNCURL works under the assumption that the true state of the cells lie in the convex hull spanned by the states of the archetypal cell types. For this problem, we assume that we are provided with a matrix of initial means $M \in \mathrm{R}^{n \times k}$ and a data matrix $X \in \mathrm{R}^{n \times d}$. The Sampled Matrix Factorization method assumes that the observed transcriptomic state of each cell is a discrete sampled version of the true state (the two sampling distributions explored in this paper are Poisson and Negative Binomial) i.e.

$$X \sim SamplingDistribution(X_{true}), \quad where\ X_{true} = M \times w$$

Here, X_{true} is the matrix of transcriptomic states of all cells (this is hidden from us) and $w \in \mathrm{R}^{k \times d}$ is the cell type fraction matrix which satisfies the property $\mathbf{1}^T w_i = 1$ and $w_i \geqslant 0$. These conditions ensure that each cell's original state lies in the convex hull of the cell states of the various cell types. Our goal now is to maximize the log-likelihood of the observed data matrix X, by finding the optimal M and w. While this problem is non-convex, the sub-problems of estimating either M or w with the other matrix fixed are convex problems. We thus adopt an EM like algorithm to estimate these model parameters. In the first step we estimate the mixture parameter while keeping the means fixed as follows:

$$w = \underset{w}{argmin}\, Prob(X|M, w, \Theta)$$

$$\text{subject to}$$

$$w \succcurlyeq 0$$

Here the cost function describes the log-likelihood of the observed data given the factor matrices M, w and any additional statistical parameters Θ. The parameter Θ is distribution dependent and is the empty set in case of the Poisson distribution, while it is the gene specific dispersion vector (calculated apriori) in case of the negative binomial distribution. Similar to this step, the following step fixes the new estimate of w and updates the estimate of the mean matrix M by solving the following optimization problem:

$$M = \underset{M}{argmin}\, Prob(X|M, w, \Theta)$$

subject to

$$M \succcurlyeq 0$$

The condition $M \succcurlyeq 0$ is a required condition because the true transcriptomic state cannot be negative. We repeat these two steps iteratively till convergence or till a maximum number of iterations.

Once converged, we normalize the columns of w to sum to 1 to ensure the condition $\mathbf{1}^T w_i = 1$ is satisfied. The condition $\mathbf{1}^T w_i = 1$ is not enforced during the optimization steps to ensure that cells with similar transcriptomic profiles but different cell sizes (thereby larger number of total transcripts) are allowed to converge to the optimal mixing weights. The post normalization step then ensures that cells with different cell sizes but with similar transcriptomic profiles end up having similar estimated states. We have implemented these steps using the optimization toolbox of Matlab and scipy in Python (for the Python implementation).

Dimensionality reduction using UNCURL

The objective of this section is to transform the data matrix $X \in \mathrm{R}^{n\times k}$ to a lower dimension data matrix $X^{LD} \in \mathrm{R}^{l\times d}$, where $l < n$. Dimensionality reduction with UNCURL follows directly from the state estimation procedure (described previously). In this step, we assume we are provided with an estimated mean matrix $M \in \mathrm{R}^{n\times k}$ and a cell type fraction matrix $w \in \mathrm{R}^{k\times d}$. We then calculate the Poisson distances between each of the means and compile them into a matrix $D \in \mathrm{R}^{k\times k}$ as follows:

$$[D]_{ij} = d\left(M_i, M_j\right)$$

Where $d(x, y)$ is the Poisson distance between points x and y. We then use Multi Dimensional Scaling (MDS)[31] to obtain a distance preserving lower dimensional representation of the mean matrix, $M^{LD} \in \mathrm{R}^{m\times k}$. Finally we obtain a lower dimensional representation of the data X^{LD} by performing the following operation:

$$X^{LD} = M^{LD} \times w$$

This method of dimensionality reduction forces the relative states of cells to stay unchanged. We argue that this leads to efficient separation of cell types even in the reduced dimensional representation.

Poisson clustering

UNCURL gives the user two choices for clustering, namely Poisson and Negative Binomial clustering. Poisson clustering is very similar to the classical k-means clustering with the difference being in the underlying distribution of the data. We assume we are provided the expected number of cell types k and the data matrix $X \in \mathrm{R}^{n\times d}$. The first step of the algorithm involves calculating the Poisson log-likelihood for each cell given a set of means $M \in \mathrm{R}^{n\times k}$ and then assigning each cell to the cell type for which it has the maximum log-likelihood value. The Poisson log-likelihood function is as follows:

$$ll_p(X_k|M_i) = \sum_j -[M]_{ij} + [X]_{jk}\log\left([M]_{ij}\right)$$

This is called the E step of the algorithm. Here, X_k is the observed data from the kth cell and M_i is the mean for the *ith* cell type. The result of this step is the identification of distinct sets of cells (cell types in this case). This is followed by the M step, where we calculate the optimal means for each cell type which maximizes the log-likelihood quantity. For the case of the Poisson distribution, this is simply the arithmetic mean of the data given by:

$$M_i = \frac{1}{|S_i|}\sum_{j \in S_i} X_j$$

Here, S_i is the set of cell indices for which the log-likelihood is highest for the ith cell type. The *M* step gives rise to a new estimate of means, which are then used to re-do the E step. This procedure is repeated till convergence or till a maximum number of iterations are performed.

Negative Binomial clustering

The Negative Binomial clustering performed by UNCURL follows the same general principles as the Poisson clustering while respecting the assumptions about the underling distribution of the data. Unlike the Poisson distribution, the Negative Binomial distribution is specified by two parameters r (number of failures before stopping) and p (success probability of each experiment). We initially assume that we are provided with matrices P and $R \in \mathrm{R}^{n\times k}$ containing the parameters for each gene in each cell type. The log-likelihood function is then specified in terms of these parameters as follows:

$$ll_{nb}(X_k|P_i, R_i) = \sum_j \log\binom{[X]_{jk} + [R]_{ji} - 1}{[X]_{jk}} + [R]_{ji}\log\left(1 - [P]_{ji}\right) + [X]_{jk}\log\left([P]_{ji}\right)$$

However, the M step in Negative Binomial is sufficiently different as there is no closed form solution to the optimal parameter estimation problem, unlike the Poisson case. Thus, the optimal parameters $[P]_{ji}$ and $[R]_{ji}$ for each gene j and each cell type i are estimated using an EM like algorithm. Another additional complication for this method is that negative binomial model parameters can only be estimated when the mean of data is smaller than the variance. To remedy this drawback, at each iteration we identify the genes that have higher mean than variance for each cell type. These genes are therefore closer to the Poisson distribution, which is a limiting case of the Negative Binomial distribution. So we estimate the M matrix for these genes instead of P and R. During the E step, we calculate the log-likelihood of negative binomial genes and Poisson genes separately and sum them to obtain the cumulative log-likelihood. Due to these extra steps involved, the Negative Binomial clustering is sufficiently slower than the Poisson clustering.

Lineage estimation

After the estimation of the true transcriptomic state of cells, one of the downstream learning tasks is to construct a lineage based on these new cell states. In order to do this, we perform dimensionality reduction (explained previously) to obtain a 2D representation. We then use the weight matrix $w \in \mathrm{R}^{k \times d}$ to identify its' dominant cell type by simply finding which weight element has the maximum value for a given cell. We then fit a smooth curve through the 2D representation of all cells belonging to one cell type (this can be any family of smooth curves). We then replace the points with the smoothed points and consolidate them in a set S_i, where i denotes the cell type index. This operation is now performed on all the cell types to obtain k disjoint sets of cells. We then compute the Minimum Spanning Tree[32] on each of these sets individually which enables us to trace progress within each cell type individually. Finally we connect cell types that are closest to each other in order to complete the lineage graph. This is done by connecting each set to its closest set and connecting the two closest points of the two sets to each other with a straight line.

Pseudotime calculation for cells

While the calculation of the cell lineage identified differentiation hierarchy of cells, a more quantified measurement of cell state is the pseudotime[33], which calculates the effective distance from the root cell. To calculate this value for each cell we have to first determine the root cell in a population of cells. Since the output of lineage calculation is a smooth tree, we can simply hypothesize that the root cell is going to be one of the leaf nodes of the tree. The leaf nodes are then calculated by first calculating the degrees of all the nodes and then selecting only the ones with $degree = 1$ as leaf nodes. Once the leaf nodes are obtained, we let the user choose the starting node among the leaf nodes in a manner similar to[9]. The pseudotime value of each cell is then their distance along the weighted lineage graph from the root cell.

Measuring seperability of clusters in reduced dimensions

To measure the separability of clusters in reduced dimensions given true labels, we define a nearest neighbor based error metric as follows:

$$ErrFunc(X^{LD}, L) = \frac{1}{kd}\sum_{i=1}^{d} I^{c}\left(L(i), L\left(N\left(X_i^{LD}\right)\right)\right)$$

Here $L(i)$ is the true label of *i* th cell and $L(N(X_i^{LD}))$ is the true label of it's nearest neighbor in the reduced dimensional representation. The function $I^c(x,y)$ is a binary function whose value is 0 if $x = y$ and 1 otherwise. This metric calculates the probability that two randomly chosen adjacent points in the reduced dimension representation belong to a different cell type.

Data Pre-processing

The datasets are subject to pre-processing in order to select genes of interest. For the Islam et. al. dataset, this was done by performing differential gene expression analysis using DESeq on the bulk dataset. For the Zeisel et. al. dataset, this was done by considering a list of around 3000 cell type specific genes that were provided in the original paper which were also present in the bulk dataset. For the Hanchate et. al. dataset, this was done by removing genes with very few reads in the same way as done in the original paper.

References

30. Arthur, D. & Vassilvitskii, S. K-Means++: the Advantages of Careful Seeding. *Proc. eighteenth Annu. ACM-SIAM Symp. Discret. algorithms* **8,** 1027–1025 (2007).
31. Kruskal, J. B. Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrika* **29,** 1–27 (1964).
32. Graham, R. L. & Hell, P. On the History of the Minimum Spanning Tree Problem. *IEEE Ann. Hist. Comput.* **7,** 43–57 (1985).
33. Trapnell, C. *et al.* The dynamics and regulators of cell fate decisions are revealed by pseudotemporal ordering of single cells. *Nat. Biotechnol.* **32,** (2014).

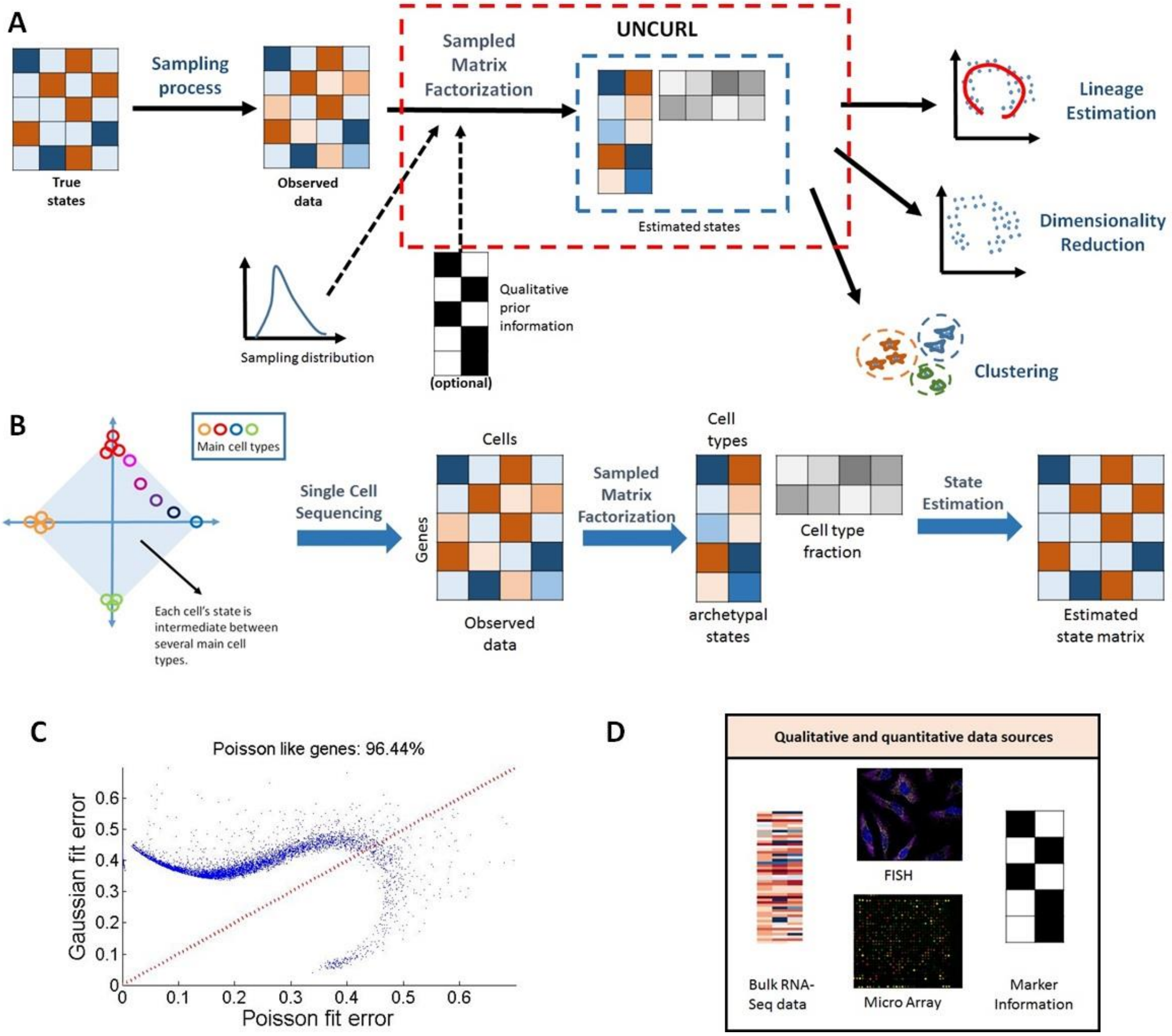

Figure 1: Learning with scRNA-Seq data using UNCURL. (A) The primary input for UNCURL is the highly sampled single cell sequenced data. The user is also expected to specify the appropriate sampling distribution for the data and optionally any prior information that is known about the specific dataset. UNCURL then converts the observed sampled data to an estimated

version of the true data using a novel technique called Sampled Matrix Factorization. This is then used in downstream unsupervised learning tasks. (B) Sampled Matrix Factorization using UNCURL. The true transcriptomic states of cells are assumed to lie along a continuum of states in a high dimension. These states are then sampled during the single cell sequencing process resulting in the transcript count matrix, which contain the observed states. UNCURL then reconstructs an estimated version of the true state from the observed states by a novel algorithm for 'Sampled Matrix Factorization', which can be viewed as an un-sampling process. (C) Comparison of fit error of all genes with data taken from [1] using Gaussian and Poisson distributions. 96.44% of genes have lower fit error for Poisson than Gaussian distribution. (D) Some of the different types of prior information supported by UNCURL, namely bulk RNA-Seq data, Micro array data, cell type specific marker information, FISH images etc.

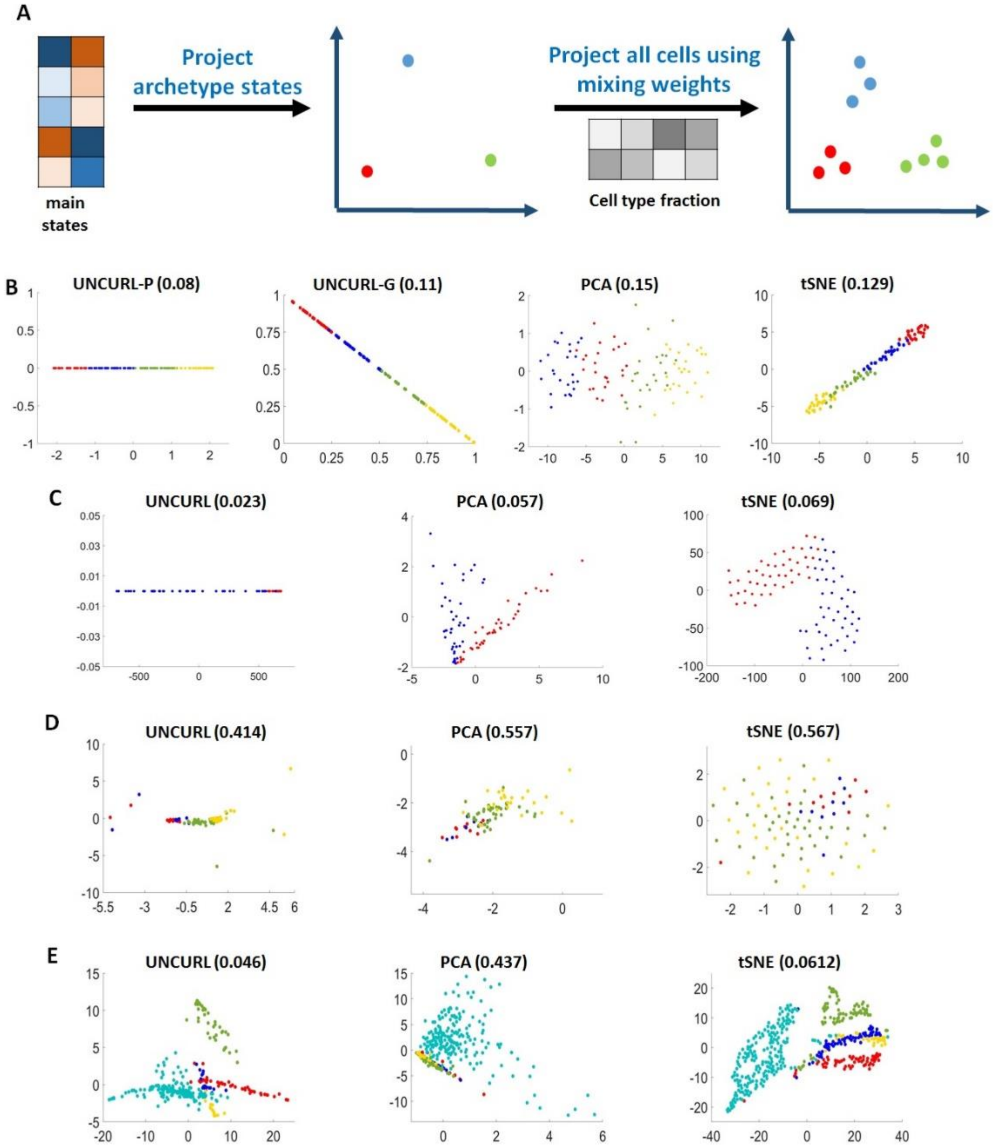

Figure 2: UNCURL leads to better low dimensional visualization of distinct cell types in single cell data. (A) Dimensionality reduction process of UNCURL. We first project the main means down to a lower dimension using the Multi-Dimensional Scaling algorithm (MDS), which

preserves the distances between the points in the reduced dimension. The cell type fraction matrix is then used to find the low dimensional representation of the cells. (B) Comparison of various algorithms on synthetic SCS data which models the sequencing process as a Poisson sampling process. We demonstrate that using UNCURL with the correct sampling distribution outperforms other common dimensionality reduction algorithms (as well as UNCURL with Gaussian as the sampling distribution). (C-E) Comparison of different dimensionality reduction methods on the various biological and synthetic SCS datasets namely (C) Mouse embryonic stem cell data from Islam et. al., (D) Olfactory development data from Hanchate et. al., (E) Mouse neuron data from Zeisel et. al.. The numbers in the bracket denote the error metric value which captures the probability that the closest neighbor of a cell belongs to a different labeled cell type.

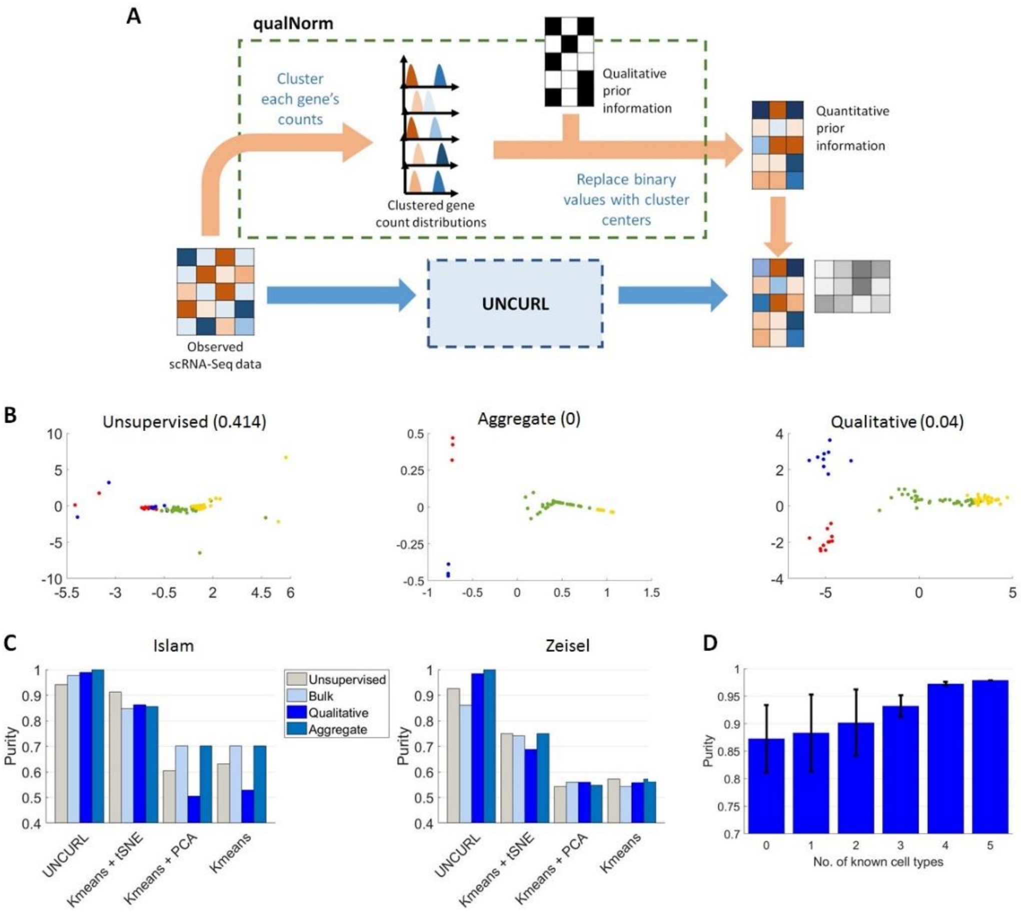

Figure 3: Distribution informed unsupervised and semi-supervised clustering leads to significant improvements in identifying cell types. (A) An illustration of the 'qualNorm' framework to convert qualitative prior information into good initialization points for unsupervised learning algorithms. The user provides cell type specific qualitative information as input, which is then converted to a binary matrix of gene expression for each cell type. We then identify the 'ON' and 'OFF' state values for each gene in the dataset and replace the binary matrix with these numerical values. This numeric matrix is then used to initialize unsupervised learning algorithms. (B) Semi-supervision using qualNorm can lead to improved visualization. Here we see the improvement

in visually separating cell types in reduced dimensions in the Olfactory dataset, when using aggregate means and qualitative means. We see that qualitative means lead to a largely improved separation between cell types and the visual representation is very similar to that obtained using aggregate means. (C) UNCURL has an average performance better than other competing clustering algorithms on two SCS datasets containing different no. of cell types for unsupervised learning tasks. When using semi-supervision, qualitative means inferred by the qualNorm lead to performance improvements that are comparable to using aggregate means (inferred using true labels). This is seen to be significantly better than using bulk datasets directly for semi-supervision. (D) Comparison of improvement in purity with prior information about different number of cell types. Here it can be seen in the case of data from Zeisel et. al., that the clustering purity monotonically increases as qualitative information about more cell types becomes available. Moreover, even information about a subset of cell types is enough to dramatically improve the clustering purity.

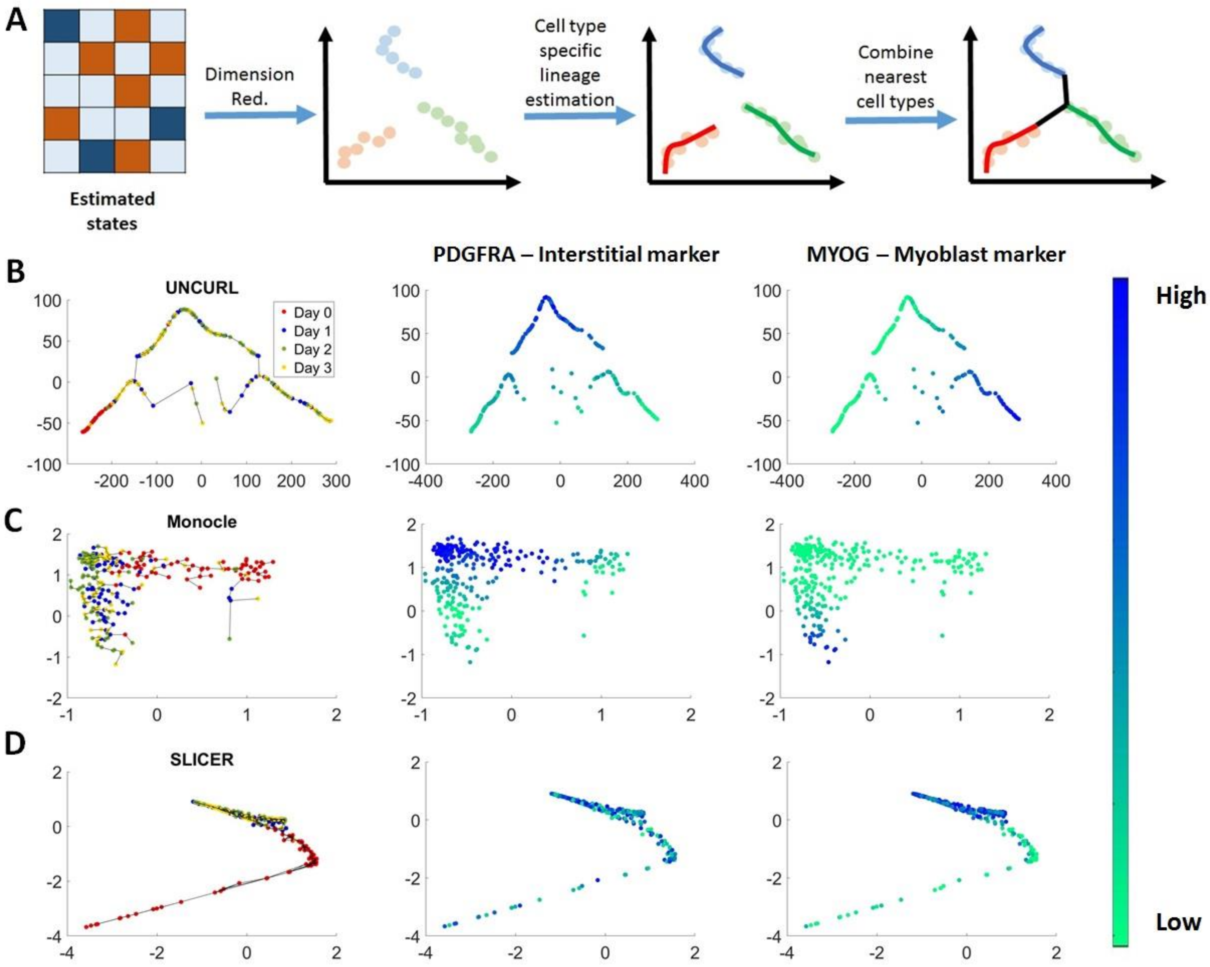

Figure 4: UNCURL leads to the estimation of smooth and accurate lineages that use meta information generated during the state estimation procedure. A) An illustration of UNCURL's lineage estimation process. The cells which are closest to each cell type (identified from convex mixture parameters) are divided into separate sets and visualized in a two dimensional space. Each set is then fitted with a smooth curve to obtain within-cell-type lineages. These smooth

curves are then joined with the closest curves, thereby completing the lineage tree construction. B-D) Comparison of lineage estimation with UNCURL to some other common lineage estimation methods namely Monocle and SLICER, on the dataset from [5]. The left-most panels consist of the estimated lineages by the different algorithms. The two other panels have relative expression levels of known marker genes visualized on the trajectories. UNCURL estimates a less noisy trajectory and separates out the different cell types well. Monocle identifies the correct ordering of cell types but the estimated trajectory is much more noisy. SLICER's estimated trajectory fails to separate out the interstitial cells from the myoblasts.

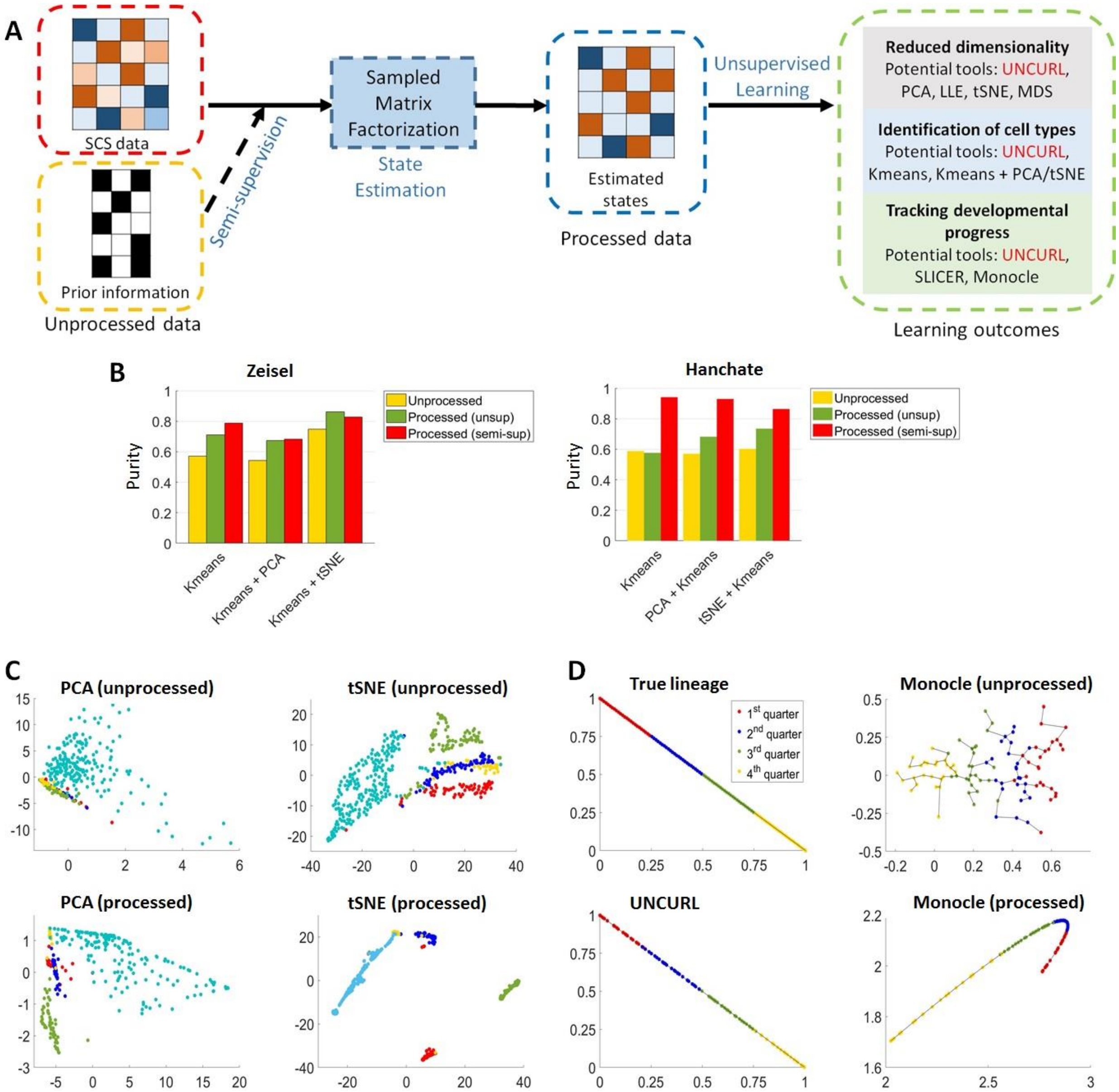

Figure 5: Estimated states of UNCURL are compatible with other unsupervised algorithms and can lead to an improvement in their performance. (A) An illustration of the workflow for using UNCURL as a data pre-processing tool. (B) Pre-processing using UNCURL is demonstrated to improve the accuracy of common clustering algorithms on the two different scRNA datasets. (C) Using estimated states instead of observed data leads to better separation of cell types in lower dimension using common methods such as PCA and tSNE (here demonstrated on the Neuronal dataset). (D) Monocle is seen to estimate smoother and more accurate lineages when using estimated data instead of observed data.