UNCURL-App: Interactive Database-Driven Analysis of Single Cell RNA Sequencing Data

Yue Zhang[1†], Shunfu Mao[2†], Sumit Mukherjee[3], Sreeram Kannan[2] and Georg Seelig[1,2*]

Abstract

Analysis of single cell RNA sequencing (scRNA-Seq) datasets is a complex and time-consuming process, requiring both biological knowledge and technical skill. In order to simplify and systematize this process, we introduce UNCURL-App, an online GUI-based interactive scRNA-Seq analysis tool. UNCURL-App introduces two key innovations: First, prior knowledge in the form of cell type, anatomy, and Gene Ontology databases is integrated directly with the rest of the analysis process, allowing users to automatically map cell clusters to known cell types based on gene expression. Second, tools for interactive re-analysis allow the user to iteratively create, merge, or delete clusters in order to arrive at an optimal mapping between clusters and cell types.

Availability: The website is at `https://uncurl.cs.washington.edu/`. Source code is available at `https://github.com/yjzhang/uncurl_app`

Keywords: single cell; single cell RNA-seq; visualization; interactive data analysis; database

Background

Single cell RNA sequencing (scRNA-seq) has become an essential and ubiquitous tool for exploring the diversity of cell types in multicellular organisms. Progress in experimental technology development has driven rapid growth in the number of scRNA-seq datasets [1, 2] with a search in 2020 for "single-cell RNA-seq" on NCBI GEO returning tens of thousands of results. Over little more than a decade, scRNA-seq experiments progressed from first proof-of-principle demonstrations using a handful of cells [3] to the construction of "cell atlases" that enumerate all of the cell types present in an organ or organism [4–9]. However, while experimental approaches have become higher throughput and more widely available, it remains challenging to map experimentally determined single cell transcriptomes to biologically meaningful cell types. Given the very large throughput in cell number and the high complexity of many of the systems under investigation, reliable data analysis has become the main bottleneck of the scRNA-seq workflow.

The process of assigning sequenced cells to cell types is a multi-step process that requires the user to make decisions based on their judgment, because the ground truth about abundance and identity of cell types in an experiment of interest is typically not available. In practice, sequencing data is often first "pre-processed," i.e. corrected for variability introduced by experimentally sampling the actual cellular transcriptome or "batch-corrected" if data from multiple experiments need to be integrated. Then, data are visualized in two dimensions and cells are clustered. Differential expression analysis identifies genes that are characteristic of each cluster, and these differentially expressed genes are used to assign clusters to cell types based on known gene-cell type associations. It is almost always necessary to iterate over this process and repeatedly remove, merge or split clusters to arrive at a satisfactory mapping of clusters to cell types consistent with known biology. These tasks require users who have both technical proficiency and knowledge of the underlying biology.

A wide range of computational tools have been developed to guide and assist each step in the analysis workflow from preprocessing [10–13], to clustering [14–18], data integration through batch effect correction [19, 20] and cell type annotation [21–23]. There are also a number of integrated analysis frameworks that combine several of these tasks into one package [24, 25]. However, these tools are typically restricted to command-line usage and require programming knowledge, hindering the accessibility of scRNA-seq analysis. These tools are also limited in their interactivity; even web-based tools such as scQuery [23] typically do not allow cluster assignments or cell type labels to be changed by the user. Moreover, in particular the last step of assigning labels to clusters remains heavily dependent on a user's prior knowledge. Thus, even with

[*] Correspondence: gseelig@uw.edu
[1] Paul G. Allen School of Computer Science and Engineering, University of Washington, Seattle, USA
Full list of author information is available at the end of the article
[†] Equal contributor

all of these computational tools, the process of analyzing a new scRNA-seq data set remains somewhat idiosyncratic.

To aid in the task of analyzing scRNA-seq data, we here introduce UNCURL-App. UNCURL-App combines data preprocessing, dimensionality reduction, clustering, differential expression, and interactive data analysis within an online graphical user interface. UNCURL-App introduces two key innovations: First, cell type databases are integrated directly with the rest of the analysis process, accelerating mapping of clusters to cell types. Second, UNCURL-App includes tools for interactive re-analysis that allow the user to create, merge, or delete clusters, thus making it possible to iteratively refine clusters using knowledge about gene expression, putative cell type annotations, and other information accessible through UNCURL-App. Because the entire workflow can be performed in a browser and because external knowledge is made available during the analysis process, we expect UNCURL-app not only to accelerate scRNA-seq analysis but also to further extend the user base for this technology.

Results

Workflow & Interface

From the user's perspective, the first step in the UNCURL-App pipeline is to upload the data as a gene-cell read count matrix (Supplemental Figures 1 & 2). Next, the app automatically performs preprocessing and clustering using UNCURL, dimensionality reduction, and differential expression. Then, the user is redirected to an interactive web site, from which they can view the results of the previous steps, query for cell types, or perform interactive re-analysis (Figure 1).

There are three main visualization components: the dimensionality-reduced scatterplot of cells, the barplot showing the most differentially expressed genes, and the cell type query results. A labeled screenshot of the main UNCURL-App view is shown in Figure 2. The scatterplot, on the top left of the screen, shows a dimensionality-reduced view of the cells in the dataset, where each point represents a cell. This view can be colored by cluster, gene expression for selected gene(s), or custom label sets based on uploaded files or user-defined criteria. For example, a user may select all cells belonging to a given cluster that also have positive expression of a certain gene, or select all cells that both belong to a certain cluster and have a certain label in an uploaded file (Supplemental Figure 3). The user may also select cells by drawing a box or shape on the plot itself.

On the top right of the screen, the barplot shows the top differentially expressed genes for the selected cluster or label. The barplot is automatically updated whenever the user clicks on a cell on the scatterplot, showing the top genes for the cluster that the cell belongs to.

The bottom left of the screen shows the database query view. From this view, the user may query cell type databases using the top differentially expressed genes. Databases include Enrichr [26], CellMarker [27], and Gene Ontology [28, 29], as well as our new cell type database, CellMeSH (see Methods). Submitting a query will return a list of cell types with a confidence score, overlapping genes, and references for each gene-cell type pair.

Data preprocessing, clustering, and differential expression

The first step in the analysis pipeline builds on UNCURL, a tool for preprocessing and clustering scRNA-seq data using probabilistic matrix factorization [12]. UNCURL has been shown to have state-of-the-art performance in clustering large-scale scRNA-seq datasets, and performs exceptionally well on sparse datasets. It assumes that the observed read count matrix is distributed with either a Poisson, Log-Normal, or Gaussian distribution, with the parameters of the distribution coming from a hidden state matrix. This hidden matrix is the product of two non-negative matrices of rank k: M, the archetype matrix, of shape $genes \times k$, and W, the weights matrix, of shape $k \times cells$, where each column sums to 1. These two matrices are the outputs of UNCURL. The rank k can be manually set as an input parameter, or automatically determined using the gap score [30]. By default, k is set to 10, which tends to produce good results in practice, but can be changed interactively by merging or splitting clusters as discussed in more detail below.

The result of UNCURL is then used for dimensionality reduction and clustering. Dimensionality reduction is done using standard methods, such as tSNE [31] or UMAP [32]. This produces a two-dimensional scatterplot of cells. By default, clustering is done using argmax on the W matrix returned by UNCURL (as described in [12]). Each column in W represents the weights for each archetype in one cell, so the archetype with the maximum weight is the most likely cluster assignment for that cell. Clustering can also be done using the Louvain [33] or Leiden [34] community detection algorithms, which also use the W matrix as input. However, only clustering using UNCURL is compatible with iterative cluster refinement as detailed below.

In order to identify the most differentially expressed genes in each cluster, UNCURL-App uses one of two methods: the t-test, or the ratio of means. These metrics can either be calculated for one cluster against all other clusters, or against a single cluster. The t-test has been shown to be one of the best performing methods for identifying DE genes in scRNA-seq datasets, and is also much faster than more complex methods [35].

Interactive data analysis

UNCURL-App has the capacity to merge, split, or delete clusters of cells in an interactive fashion. After the initial analysis process is completed, there are often refinements to the clustering that users would like to make, no matter the quality of the initial clustering. For example, the user may want to split a large cluster, merge multiple similar clusters, or delete a group of poor quality cells or potential doublets. This cluster refinement might be based on the shape of the scatterplot, differential expression results, cell type queries, or some other metrics.

The user-driven changes in clustering are incorporated into UNCURL by using them to generate new initializations and then re-running the optimization process, as shown in Figure 3. This process fundamentally relies on the UNCURL algorithm [12], and was inspired by the UTOPIAN software for interactive non-negative matrix factorization [36], but in UNCURL-App, cells take the place of documents. Say that we have matrices M and W, with shapes $g \times K$ and $K \times c$. In order to split a selected cluster, we first run k-means with $k = 2$ on the cells assigned to the selected cluster. This generates new matrices $M_{cluster}$ and $W_{cluster}$, of shape $g \times 2$ and $2 \times c$ representing the means and cell cluster assignments. The column and row corresponding to the selected cluster are deleted from M and W, and $M_{cluster}$ and $W_{cluster}$ are appended to M and W, creating M_{new} and W_{new}, with shapes $g \times (K+1)$ and $(K+1) \times c$. Then, UNCURL is re-run with M_{new} and W_{new} as the initializations, which affects other clusters as well. The process is analogous for merging clusters and assigning cells to new clusters: we create new initializations for M and W using the selected clusters or cells, and then re-run the optimization process.

After generating M_{new} and W_{new}, a new visualization, clustering, and differential expression results are calculated using W_{new}. Running re-clustering also automatically updates the differential expression results.

Examples

In order to validate the UNCURL-App workflow, we used the app to analyze three different scRNA-seq datasets, as described below. For these datasets, we performed clustering and cell type annotation using UNCURL-App with default settings. Cell type labels were generated by querying the top 50 genes by 1-vs-rest ratio with the CellMeSH database (see Methods).

The running times of the non-interactive steps are shown in Table 1. The running time for UNCURL scales linearly with the number of cells, while the running time for tSNE scales with order $n \log n$, where n is the number of cells. With larger numbers of cells, running tSNE is the most time-consuming step. This can be obviated by using UMAP as the dimensionality reduction method.

Example: Tabula Muris lung cells

As a first example, we consider a subset of the Tabula Muris dataset from [7] containing only cell types found in the lung. This dataset contains 5449 cells and 14 annotated cell types. The labels in the original study were generated by first running graph-based clustering and then manually examining the marker genes for each cluster.

After uploading the dataset and processing it with default settings, we see the clustering and initial cell type assignments in Figure 4a. The clustering was based on UNCURL, and the scatterplot visualization was generated using tSNE. Based on the scatterplot, it was apparent that cluster 4 appeared to consist of at least two groups of cells that should not be grouped together. In addition, the top cell types from a CellMeSH query on the top genes in this cluster included both B and T cells (Figure 4b), suggesting that this cluster might be a mixture of at least these two cell types. Based on these observations, we decided to split this cluster using our interactive data analysis tools, resulting in the clusters given in Figure 4c. The post-split cell type assignments (Figure 4d) appeared to be more consistent with known biology than the original assignments.

Based on Figure 4f, there is generally good concordance between the generated clusters and original clusters, as well as between assigned labels and the original labels. Of the labels that were different, in most cases UNCURL-App assigned cell types that were closely related to the original ground truth label (for example, pneumocytes and columnar cells are subsets of epithelial cells, and neutrophils are a subset of leukocytes). The stromal cells are split into multiple clusters in UNCURL-App, which could represent heterogeneity in the original sample that was not captured by the original labels. No prior information about the cell types present in the dataset was used at any point in this process.

Example: 10X PBMCs

Next, we turned to a dataset comprised of 8000 human peripheral blood mononuclear cells (PBMCs) from [37]. This dataset was created by randomly sampling 1000 cells from each of 8 scRNA-seq datasets comprised of cells that were flow-sorted based on known cell-type markers. Thus, the ground truth cell type labels represent pure samples, as opposed to the computational assignments used as ground truth in the other example datasets.

UNCURL-App was run with default settings to generate 10 initial clusters (Figure 5a). Looking at the resulting clusters and putative cell type assignments (5b), it appeared that clusters 2 and 6, labeled Neutrophils and Monocytes, were very similar, and could just represent a single group of cells. A pairwise differential expression analysis (Figure 5c) further illustrates that only related genes, S100A8 and S100A9, appear to be significantly differentially expressed between these clusters. Plotting the expression levels of these genes (Figure 5d), it seems that the small group of cells to the left of the main cluster has much higher expression of these genes, suggesting that this group might constitute a separate cluster. Thus, we first merged clusters 2 and 6, and then split off that small group of cells. These operations resulted in the clustering shown in Figure 5f.

As with the previous dataset, there now is good correspondence with the ground truth clusters and labels (Figure 5h, i). Cells of the same ground truth type are generally assigned to the same cluster, and the cluster labels returned by CellMeSH generally correspond to the ground truth labels. CD34+ cells are generally recognized as hematopoietic stem cells [38], so the CellMeSH label here seems to be accurate. One major difference is that CellMeSH labeled all four T cell subtypes as "T-Lymphocytes", even though they were clustered into distinct clusters. To investigate further, we looked at the full list of CellMeSH labels for these clusters, not just the top one. These results are shown in Figure 5g, with the cell types most similar to the ground truth highlighted in green. For example, Cluster 0 corresponds to naive T-cells, which are selected as CD4+. Cluster 5 corresponds to naive cytotoxic T-cells, which are CD8+, and the "CD8+ T-Lymphocytes" label is the third highest label, below "T-Lymphocytes" and "Lymphocytes" (Figure 5g). Cluster 6 corresponds to memory T-cells, which can be either CD8+ or CD4+; the second and third labels are "CD8+ T-Lymphocytes" and "CD4-Positive T-Lymphocytes". Cluster 7 corresponds largely to regulatory T-cells, which are CD4+, and the second and third highest CellMeSH labels are "CD4-Positive T-Lymphocytes" and "T-Lymphocytes, Regulatory". This shows a good correspondence between the true and assigned labels at a more fine-grained level.

Example: SPLiT-seq spinal cord

For a final test we turned to a larger dataset comprised of 22,614 mouse spinal cord nuclei from 2 and 11-day old mice sequenced using SPLiT-seq [9]. This dataset has 44 annotated cell types, which is substantially more than the previous two datasets. However, many of these annotated cell types are closely related (for example, there are 15 types of excitatory neurons), so for the "ground truth" comparisons in this section, we combine many of the annotated cell types into larger clusters of similar cells. Even after this process, many of the cell types are similar, with many subtypes of neurons.

We first ran UNCURL-App with default settings to generate an initial clustering with 10 clusters, several of which exhibit substantial heterogeneity (Figure 6a). For example, cluster 8 (labeled as "Endothelial Cells") represents at least four different groups of non-neuronal cells. Thus, we split them into four different clusters (Figure 6b). It is clear that splitting the clusters worked to separate what appeared to be distinct cell types. In addition, the clusters that CellMeSH labels as "Neurons" or "Interneurons" (3, 6, 9, 0) all appear to be rather heterogeneous. Results after splitting some of the neuronal clusters are shown in Supplemental Figures 4-7.

As with the previous datasets, there is generally good concordance between the cell types from the original paper and the clusters generated by UNCURL-App, as shown in Figure 6d. Also similarly to previous datasets, the CellMeSH annotations were generally coarser grained than the original hand-annotated labels, with all of the neuron clusters being labeled as "Interneurons" or just "Neurons". For the non-neuronal results, interpreting the labels identified by CellMeSH is more challenging (Supplemental Figure 5). Oligodendrocytes, astrocytes, and endothelial cells were correctly identified. For cluster 8, the ground-truth label was "VLMC", or "vascular and leptomeningeal cells". This is a highly specific category that does not appear in the CellMeSH ontology but was used as a cell label in Ref. [39]. Still, while coarse, the first three labels suggested by CellMeSH (Stromal Cells, Fibroblasts, Mesenchymal Stem Cells) seem consistent with cells derived from the meninges, the membrane enveloping the brain and spinal cord. In cluster 10, the ground-truth label "Ependymal" was not correctly identified by CellMeSH, and the returned results did not seem to relate to ependymal cells. This points to a paucity of annotated publications with gene markers for this cell type. For cluster 11, all of the top CellMeSH results were immune cells, a group which the published label, "microglia", belongs to. "Microglia" was one of the top 10 cell types returned.

Discussion

Comparison with existing tools

Unlike other general-purpose toolkits for scRNA-seq analysis such as scanpy [24], Seurat [25], and Monocle 2 [40], UNCURL-App is a web-based GUI tool that does not require command line usage. This allows a much wider range of potential users, such as biologists who are not programmers. One comparable web-based tool is scQuery [23]. Both scQuery and UNCURL-App perform clustering and dimensionality reduction on uploaded single-cell datasets, and can identify cell types. With regards to the user interface, whereas UNCURL-App is a single-page application that presents all of its information on a single screen, scQuery has multiple views for different tasks. Unlike in scQuery, where cell type annotations are ultimately derived from scRNA-seq data from GEO, cell type annotations in UNCURL-App are based on the published scientific literature. UNCURL-App is also capable of interactively merging, splitting, and deleting clusters of cells, unlike scQuery.

There are a number of tools that classify cells given gene markers for known cell types, such as [21, 41]. We view these tools as complementary to UNCURL-App and CellMeSH. These tools require some knowledge of the cell types present in the dataset, as well as a way to manually find gene markers for these cell types, whereas such prior knowledge is unnecessary in the UNCURL-App/CellMeSH pipeline. In addition, CellMeSH can be used to improve the workflow for these tools by automatically selecting gene markers, obviating the need for manually finding them.

There also exist tools that perform single cell similarity search on reference datasets, such as CellAtlasSearch and scMatch [22, 42]. Rather than using marker genes, these methods compare the entire gene expression profile of every single cell to a reference database, using locality-sensitive hashing in the case of CellAtasSearch [22] or Pearson or Spearman correlation in the case of scMatch [42]. These tools do not include functionality for clustering or low-dimensional visualization. The advantage of UNCURL-App comes with its integration of clustering, differential expression, interactive re-analysis, and cell type querying into one easy-to-use platform.

Conclusion

UNCURL-App provides a useful way to perform interactive scRNA-seq data analysis, including cell type annotation. In the future, we hope to augment UNCURL-App with new analysis capabilities, such as cell lineage and gene network analysis. We also hope to connect UNCURL-App to additional sources of information for cell type and functional annotation. This could come in the form of connections to new databases, or expansions to the CellMeSH database. Our ultimate goal is to increase UNCURL-App's utility as a general tool for scRNA-seq analysis.

Methods

Cell type annotation

UNCURL-App has a number of interfaces to external databases, which are used to assist with identifying cell types present in the dataset, as well as helping to better understand underlying biological processes. First, UNCURL-App contains an interface to the Enrichr tool for gene set analysis [26, 43]. This tool contains interfaces to a variety of gene set databases that can be used to help identify cell function. We also provide an interface to Gene Ontology [28, 29], which is queried using the goatools package [44]. In addition, we have two databases specifically for cell type identification, CellMarker and CellMeSH.

CellMarker [27] is a hand-curated database of cell types, annotated with marker genes based on a literature search. This dataset consists of 673 cell types, where each cell type is associated with an average of 72 and a median of 9 marker genes. To search this database given a list of query genes, we use the hypergeometric test for the overlap between the query gene set and the marker genes.

$$1 - \sum_{k=0}^{k_c-1} \frac{\binom{K_c}{k}\binom{N-K_c}{n-k}}{\binom{N}{n}} \tag{1}$$

where N is the total number of genes, n is the number of genes in the query set, K_c is the number of genes for the cell type, and k_c is the number of genes that overlap between the query and the cell type. This is the probability that, given that the query gene set is randomly sampled, the overlap is greater than or equal to the actual overlap. To find the top cell types for a query gene set, this p-value is calculated for all cell types and ranked in ascending order.

CellMeSH is a new database that maps cell types to their associated genes. It was created by combining two existing literature indices: the MEDLINE citation index [45], which contains publication abstracts with associated metadata, and the gene2pubmed database [46], which contains a mapping of genes to publications. The key metadata from MEDLINE are the associated Medical Subject Headings, or MeSH terms [47], a subset of which represent cell types. For each cell type from MeSH, we found all publications where they occur, and all genes that occur in the same publications, thus creating an association between cell types and genes. This database contains 292 cell types with at least one associated gene. Searching this database can be done using a hypergeometric test. A query returns an ordered list of cell types sorted by relevance.

Implementation

UNCURL-App and the associated backend tools and databases are written in Python. The primary package is the uncurl-app package, which uses the Flask library as the server backend. Visualization is done in javascript using the plotly library [48]. The backend, which interfaces with the dimensionality reduction and differential expression methods, is provided by the uncurl-analysis package, and the databases are provided by the cellmarker and cellmesh packages.

Deployment

UNCURL-App has been tested to run on Ubuntu 16.04 and above, and can be deployed on a local or cloud server using Docker. We have created an example UNCURL-App deployment at `https://uncurl.cs.washington.edu/`. This deployment limits its upload size to 100MB.

Competing interests

The authors declare that they have no competing interests.

Author's contributions

Conceptualized the tool: YZ, SM1, SM2, SK, GS. Implemented the application: YZ. Implemented the database: SM1, YZ. Wrote and edited the paper: YZ, SM1, SM2, SK, GS.

Acknowledgements

We thanks Li Liu, Tao Peng, and Matthew Hirano for testing UNCURL-App and suggesting new features. This work was supported by NIH R01HG009136 and R01HG009892 to G.S.

Author details

[1]Paul G. Allen School of Computer Science and Engineering, University of Washington, Seattle, USA. [2]Department of Electrical and Computer Engineering, University of Washington, Seattle, USA. [3]AI for Good Research Lab, Microsoft, Redmond, USA.

References

1. Chen, X., Teichmann, S.A., Meyer, K.B.: From Tissues to Cell Types and Back: Single-Cell Gene Expression Analysis of Tissue Architecture. Annual Review of Biomedical Data Science **1**(1), 29–51 (2018). doi:10.1146/annurev-biodatasci-080917-013452. Accessed 2019-02-27
2. Svensson, V., Beltrame, E.d.V., Pachter, L.: A curated database reveals trends in single-cell transcriptomics. bioRxiv, 742304 (2019). doi:10.1101/742304. Accessed 2020-01-24
3. 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.: mRNA-Seq whole-transcriptome analysis of a single cell. Nature Methods **6**(5), 377–382 (2009). doi:10.1038/nmeth.1315. Number: 5 Publisher: Nature Publishing Group
4. Han, X., Wang, R., Zhou, Y., Fei, L., Sun, H., Lai, S., Saadatpour, A., Zhou, Z., Chen, H., Ye, F., Huang, D., Xu, Y., Huang, W., Jiang, M., Jiang, X., Mao, J., Chen, Y., Lu, C., Xie, J., Fang, Q., Wang, Y., Yue, R., Li, T., Huang, H., Orkin, S.H., Yuan, G.-C., Chen, M., Guo, G.: Mapping the Mouse Cell Atlas by Microwell-Seq. Cell **172**(5), 1091–110717 (2018). doi:10.1016/j.cell.2018.02.001. Accessed 2018-02-22
5. Cao, J., Packer, J.S., Ramani, V., Cusanovich, D.A., Huynh, C., Daza, R., Qiu, X., Lee, C., Furlan, S.N., Steemers, F.J., Adey, A., Waterston, R.H., Trapnell, C., Shendure, J.: Comprehensive single-cell transcriptional profiling of a multicellular organism. Science **357**(6352), 661–667 (2017). doi:10.1126/science.aam8940. Publisher: American Association for the Advancement of Science Section: Research Article. Accessed 2020-03-09
6. Regev, A., Teichmann, S.A., Lander, E.S., Amit, I., Benoist, C., Birney, E., Bodenmiller, B., Campbell, P., Carninci, P., Clatworthy, M., Clevers, H., Deplancke, B., Dunham, I., Eberwine, J., Eils, R., Enard, W., Farmer, A., Fugger, L., Göttgens, B., Hacohen, N., Haniffa, M., Hemberg, M., Kim, S., Klenerman, P., Kriegstein, A., Lein, E., Linnarsson, S., Lundberg, E., Lundeberg, J., Majumder, P., Marioni, J.C., Merad, M., Mhlanga, M., Nawijn, M., Netea, M., Nolan, G., Pe'er, D., Phillipakis, A., Ponting, C.P., Quake, S., Reik, W., Rozenblatt-Rosen, O., Sanes, J., Satija, R., Schumacher, T.N., Shalek, A., Shapiro, E., Sharma, P., Shin, J.W., Stegle, O., Stratton, M., Stubbington, M.J.T., Theis, F.J., Uhlen, M., van Oudenaarden, A., Wagner, A., Watt, F., Weissman, J., Wold, B., Xavier, R., Yosef, N., Human Cell Atlas Meeting Participants: The Human Cell Atlas. eLife **6**, 27041 (2017). doi:10.7554/eLife.27041. Accessed 2020-01-24
7. Consortium, T.T.M.: Single-cell transcriptomics of 20 mouse organs creates a Tabula Muris. Nature **562**(7727), 367 (2018). doi:10.1038/s41586-018-0590-4. Accessed 2019-04-20
8. 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.: Adult mouse cortical cell taxonomy revealed by single cell transcriptomics. Nature Neuroscience **19**(2), 4216 (2016). doi:10.1038/nn.4216. Accessed 2017-11-03
9. Rosenberg, A.B., Roco, C.M., Muscat, R.A., Kuchina, A., Sample, P., Yao, Z., Gray, L., Peeler, D.J., Mukherjee, S., Chen, W., Pun, S.H., Sellers, D.L., Tasic, B., Seelig, G.: Single-cell profiling of the developing mouse brain and spinal cord with split-pool barcoding. Science, 8999 (2018). doi:10.1126/science.aam8999. Accessed 2018-05-11
10. Pierson, E., Yau, C.: ZIFA: Dimensionality reduction for zero-inflated single-cell gene expression analysis. Genome Biology **16**, 241 (2015). doi:10.1186/s13059-015-0805-z. Accessed 2017-08-02
11. Wang, B., Zhu, J., Pierson, E., Ramazzotti, D., Batzoglou, S.: Visualization and analysis of single-cell RNA-seq data by kernel-based similarity learning. Nature Methods **14**(4), 414 (2017). doi:10.1038/nmeth.4207. Accessed 2018-01-10
12. Mukherjee, S., Zhang, Y., Fan, J., Seelig, G., Kannan, S.: Scalable preprocessing for sparse scRNA-seq data exploiting prior knowledge. Bioinformatics **34**(13), 124–132 (2018). doi:10.1093/bioinformatics/bty293. Accessed 2018-06-29
13. Dijk, D.v., Sharma, R., Nainys, J., Yim, K., Kathail, P., Carr, A.J., Burdziak, C., Moon, K.R., Chaffer, C.L., Pattabiraman, D., Bierie, B., Mazutis, L., Wolf, G., Krishnaswamy, S., Pe'er, D.: Recovering Gene Interactions from Single-Cell Data Using Data Diffusion. Cell **174**(3), 716–72927 (2018). doi:10.1016/j.cell.2018.05.061. Accessed 2019-03-15
14. 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.: SC3: consensus clustering of single-cell RNA-seq data. Nature Methods **14**(5), 483–486 (2017). doi:10.1038/nmeth.4236. Accessed 2017-08-18
15. Lin, P., Troup, M., Ho, J.W.K.: CIDR: Ultrafast and accurate clustering through imputation for single-cell RNA-seq data. Genome Biology **18**, 59 (2017). doi:10.1186/s13059-017-1188-0. Accessed 2017-08-17
16. Zhang, J.M., Fan, J., Fan, H.C., Rosenfeld, D., Tse, D.N.: An interpretable framework for clustering single-cell RNA-Seq datasets. BMC Bioinformatics **19** (2018). doi:10.1186/s12859-018-2092-7. Accessed 2018-03-29
17. Sun, Z., Chen, L., Xin, H., Jiang, Y., Huang, Q., Cillo, A.R., Tabib, T., Kolls, J.K., Bruno, T.C., Lafyatis, R., Vignali, D.A.A., Chen, K.,

Ding, Y., Hu, M., Chen, W.: A Bayesian mixture model for clustering droplet-based single-cell transcriptomic data from population studies. Nature Communications **10**(1), 1649 (2019). doi:10.1038/s41467-019-09639-3. Accessed 2019-04-18

18. Diaz-Mejia, J.J., Meng, E.C., Pico, A.R., MacParland, S.A., Ketela, T., Pugh, T.J., Bader, G.D., Morris, J.H.: Evaluation of methods to assign cell type labels to cell clusters from single-cell RNA-sequencing data. bioRxiv, 562082 (2019). doi:10.1101/562082. Accessed 2019-03-05
19. Kiselev, V.Y., Yiu, A., Hemberg, M.: scmap: projection of single-cell RNA-seq data across data sets. Nature Methods (2018). doi:10.1038/nmeth.4644. Accessed 2018-04-05
20. Haghverdi, L., Lun, A.T.L., Morgan, M.D., Marioni, J.C.: Batch effects in single-cell RNA-sequencing data are corrected by matching mutual nearest neighbors. Nature Biotechnology **36**(5), 421–427 (2018). doi:10.1038/nbt.4091. Accessed 2018-12-19
21. Pliner, H.A., Shendure, J., Trapnell, C.: Supervised classification enables rapid annotation of cell atlases. Nature Methods **16**(10), 983–986 (2019). doi:10.1038/s41592-019-0535-3. Accessed 2020-01-23
22. Srivastava, D., Iyer, A., Kumar, V., Sengupta, D.: CellAtlasSearch: a scalable search engine for single cells. Nucleic Acids Research (2018). doi:10.1093/nar/gky421. Accessed 2018-06-03
23. Alavi, A., Ruffalo, M., Parvangada, A., Huang, Z., Bar-Joseph, Z.: A web server for comparative analysis of single-cell RNA-seq data. Nature Communications **9**(1), 4768 (2018). doi:10.1038/s41467-018-07165-2. Accessed 2018-12-13
24. Wolf, F.A., Angerer, P., Theis, F.J.: SCANPY: large-scale single-cell gene expression data analysis. Genome Biology **19**, 15 (2018). doi:10.1186/s13059-017-1382-0. Accessed 2018-02-09
25. Satija, R., Farrell, J.A., Gennert, D., Schier, A.F., Regev, A.: Spatial reconstruction of single-cell gene expression data. Nature Biotechnology **33**(5), 495 (2015). doi:10.1038/nbt.3192. Accessed 2018-01-04
26. Kuleshov, M.V., Jones, M.R., Rouillard, A.D., Fernandez, N.F., Duan, Q., Wang, Z., Koplev, S., Jenkins, S.L., Jagodnik, K.M., Lachmann, A., McDermott, M.G., Monteiro, C.D., Gundersen, G.W., Ma'ayan, A.: Enrichr: a comprehensive gene set enrichment analysis web server 2016 update. Nucleic Acids Research **44**(W1), 90–97 (2016). doi:10.1093/nar/gkw377. Accessed 2018-02-23
27. Zhang, X., Lan, Y., Xu, J., Quan, F., Zhao, E., Deng, C., Luo, T., Xu, L., Liao, G., Yan, M., Ping, Y., Li, F., Shi, A., Bai, J., Zhao, T., Li, X., Xiao, Y.: CellMarker: a manually curated resource of cell markers in human and mouse. Nucleic Acids Research (2018). doi:10.1093/nar/gky900. Accessed 2018-10-11
28. Ashburner, M., Ball, C.A., Blake, J.A., Botstein, D., Butler, H., Cherry, J.M., Davis, A.P., Dolinski, K., Dwight, S.S., Eppig, J.T., Harris, M.A., Hill, D.P., Issel-Tarver, L., Kasarskis, A., Lewis, S., Matese, J.C., Richardson, J.E., Ringwald, M., Rubin, G.M., Sherlock, G.: Gene Ontology: tool for the unification of biology. Nature genetics **25**(1), 25–29 (2000). doi:10.1038/75556. Accessed 2019-10-13
29. The Gene Ontology Resource: 20 years and still GOing strong. Nucleic Acids Research **47**(D1), 330–338 (2019). doi:10.1093/nar/gky1055. Accessed 2019-10-13
30. Tibshirani, R., Walther, G., Hastie, T.: Estimating the number of clusters in a data set via the gap statistic. Journal of the Royal Statistical Society: Series B (Statistical Methodology) **63**(2), 411–423 (2001). doi:10.1111/1467-9868.00293. Accessed 2019-02-25
31. Maaten, L.v.d., Hinton, G.: Visualizing Data using t-SNE. Journal of Machine Learning Research **9**(Nov), 2579–2605 (2008). Accessed 2018-01-04
32. McInnes, L., Healy, J.: UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction. arXiv:1802.03426 [cs, stat] (2018). arXiv: 1802.03426. Accessed 2018-03-30
33. Blondel, V.D., Guillaume, J.-L., Lambiotte, R., Lefebvre, E.: Fast unfolding of communities in large networks. Journal of Statistical Mechanics: Theory and Experiment **2008**(10), 10008 (2008). doi:10.1088/1742-5468/2008/10/P10008. Accessed 2018-03-30
34. Korsunsky, I., Fan, J., Slowikowski, K., Zhang, F., Wei, K., Baglaenko, Y., Brenner, M., Loh, P.-R., Raychaudhuri, S.: Fast, sensitive, and accurate integration of single cell data with Harmony. bioRxiv, 461954 (2018). doi:10.1101/461954. Accessed 2019-01-15
35. Soneson, C., Robinson, M.D.: Bias, robustness and scalability in single-cell differential expression analysis. Nature Methods **15**(4), 255–261 (2018). doi:10.1038/nmeth.4612. Accessed 2018-05-24
36. Choo, J., Lee, C., Reddy, C.K., Park, H.: UTOPIAN: User-Driven Topic Modeling Based on Interactive Nonnegative Matrix Factorization. IEEE Transactions on Visualization and Computer Graphics **19**(12), 1992–2001 (2013). doi:10.1109/TVCG.2013.212
37. 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.: Massively parallel digital transcriptional profiling of single cells. Nature Communications **8**, 14049 (2017). doi:10.1038/ncomms14049. Accessed 2017-06-04
38. Siena, S., Bregni, M., Brando, B., Ravagnani, F., Bonadonna, G., Gianni, A.M.: Circulation of CD34+ hematopoietic stem cells in the peripheral blood of high-dose cyclophosphamide-treated patients: enhancement by intravenous recombinant human granulocyte-macrophage colony-stimulating factor. Blood **74**(6), 1905–1914 (1989)
39. Marques, S., Zeisel, A., Codeluppi, S., Bruggen, D.v., Falcão, A.M., Xiao, L., Li, H., Häring, M., Hochgerner, H., Romanov, R.A., Gyllborg, D., Muñoz-Manchado, A.B., Manno, G.L., Lönnerberg, P., Floriddia, E.M., Rezayee, F., Ernfors, P., Arenas, E., Hjerling-Leffler, J., Harkany, T., Richardson, W.D., Linnarsson, S., Castelo-Branco, G.: Oligodendrocyte heterogeneity in the mouse juvenile and adult central nervous system. Science **352**(6291), 1326–1329 (2016). doi:10.1126/science.aaf6463. Publisher: American Association for the Advancement of Science Section: Report. Accessed 2020-03-06
40. Qiu, X., Mao, Q., Tang, Y., Wang, L., Chawla, R., Pliner, H.A., Trapnell, C.: Reversed graph embedding resolves complex single-cell trajectories. Nature Methods **14**(10), 979 (2017). doi:10.1038/nmeth.4402. Accessed 2018-01-29
41. Zhang, A.W., O'Flanagan, C., Chavez, E., Lim, J.L., McPherson, A., Wiens, M., Walters, P., Chan, T., Hewitson, B., Lai, D., Mottok, A., Sarkozy, C., Chong, L., Aoki, T., Wang, X., Weng, A.P., McAlpine, J.N., Aparicio, S., Steidl, C., Campbell, K.R., Shah, S.P.: Probabilistic cell type assignment of single-cell transcriptomic data reveals spatiotemporal microenvironment dynamics in human cancers. bioRxiv, 521914 (2019). doi:10.1101/521914. Accessed 2019-06-04
42. Hou, R., Denisenko, E., Forrest, A.R.R.: scMatch: a single-cell gene expression profile annotation tool using reference datasets. Bioinformatics (2019). doi:10.1093/bioinformatics/btz292. Accessed 2019-05-08
43. Chen, E.Y., Tan, C.M., Kou, Y., Duan, Q., Wang, Z., Meirelles, G.V., Clark, N.R., Ma'ayan, A.: Enrichr: interactive and collaborative HTML5 gene list enrichment analysis tool. BMC Bioinformatics **14**, 128 (2013). doi:10.1186/1471-2105-14-128. Accessed 2018-02-23
44. Klopfenstein, D.V., Zhang, L., Pedersen, B.S., Ramírez, F., Vesztrocy, A.W., Naldi, A., Mungall, C.J., Yunes, J.M., Botvinnik, O., Weigel, M., Dampier, W., Dessimoz, C., Flick, P., Tang, H.: GOATOOLS: A Python library for Gene Ontology analyses. Scientific Reports **8**(1), 1–17 (2018). doi:10.1038/s41598-018-28948-z. Accessed 2019-10-13
45. MEDLINE®: Description of the Database (2019). `https://www.nlm.nih.gov/bsd/medline.html` Accessed 2019-06-07
46. Maglott, D., Ostell, J., Pruitt, K.D., Tatusova, T.: Entrez Gene: gene-centered information at NCBI. Nucleic Acids Research **35**(Database issue), 26–31 (2007). doi:10.1093/nar/gkl993. Accessed 2019-06-07
47. Medical Subject Headings (2019). `https://www.nlm.nih.gov/mesh/meshhome.html` Accessed

2019-06-07

48. Modern Analytics Apps for the Enterprise (2019). https://plot.ly Accessed 2019-02-27

Tables

Figures

Additional Files

Supplemental Figures

1. Data upload interface
2. UNCURL-App preprocessing options
3. Custom cell selection interface
4. Split-seq spinal cord scatterplot after splitting out neuronal clusters
5. Top CellMeSH cell types for split-seq spinal cord data
6. Heatmap comparing UNCURL-App clusters with published labels (coarse-grained)
7. Heatmap comparing UNCURL-App clusters with published labels (fine-grained)

Table 1 Runtime of UNCURL-App. These times were based on an Amazon Web Services (AWS) t2.medium instance, with two processors and 4GB of memory. Times can be reduced based on the settings; using UMAP for dimensionality reduction will greatly reduce the time taken for that step. All times are in seconds.

Dataset	# of Cells	UNCURL	Dimensionality Reduction (tSNE)	Differential Expression	Total
Tabula Muris Lung	5449	127	96	11	234
10X PBMC	8000	104	129	9	242
Split-seq Spinal Cord	22614	268	550	16	834

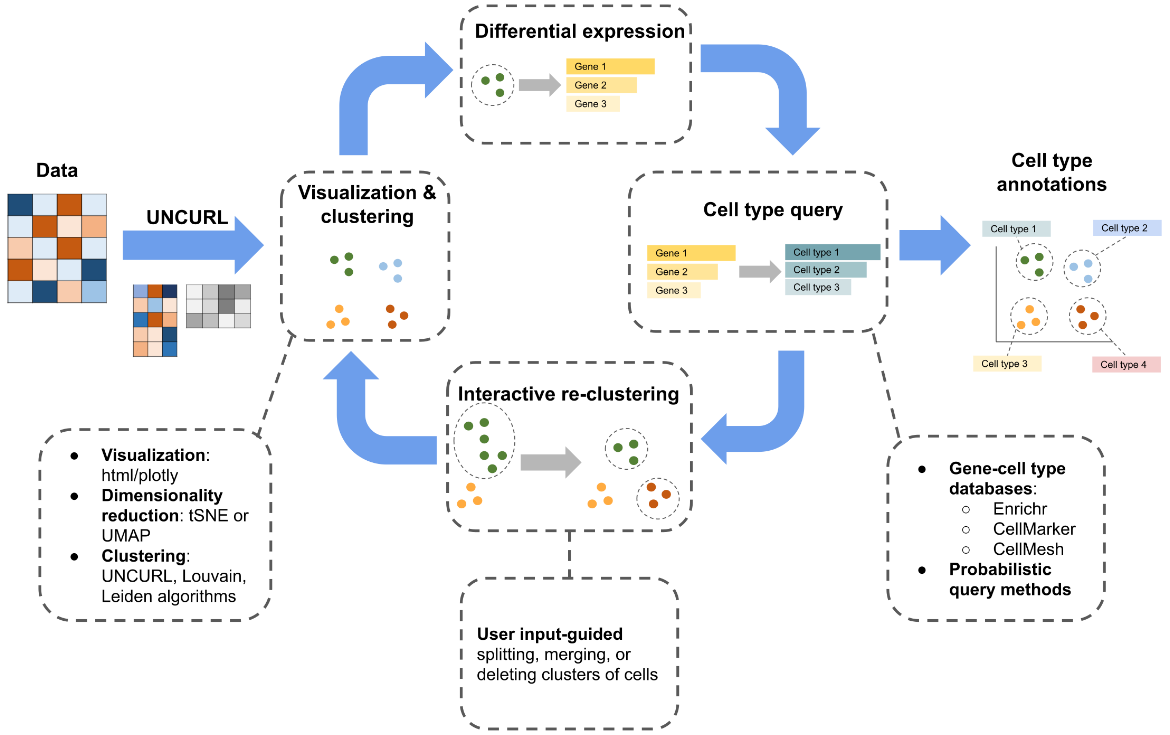
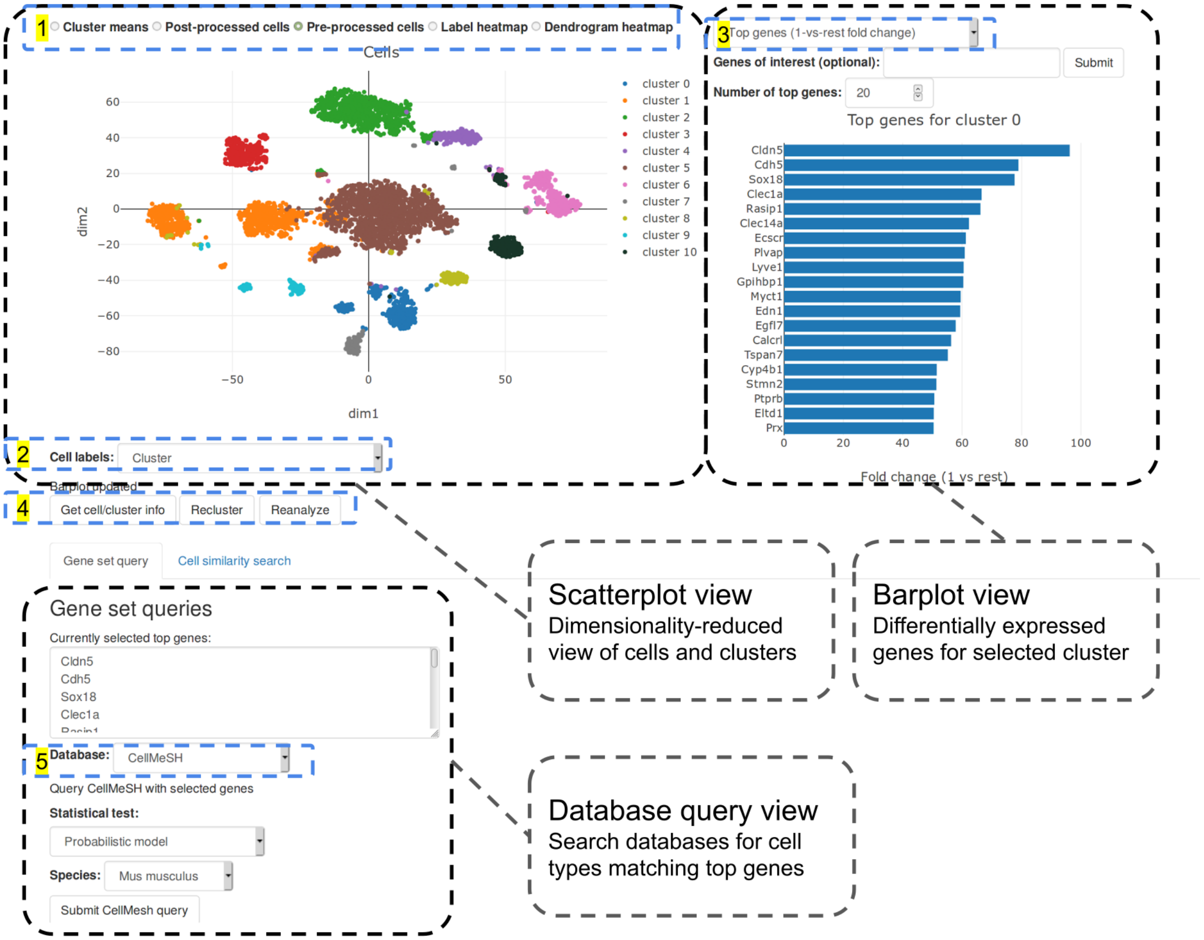

Differential expression
Gene 1
Gene 2
Gene 3
Data
UNCURL
Visualization & clustering
Cell type query
Gene 1
Gene 2
Gene 3
Cell type 1
Cell type 2
Cell type 3
Cell type annotations
Cell type 1
Cell type 2
Cell type 3
Cell type 4
Interactive re-clustering
Visualization: html/plotly
Dimensionality reduction: tSNE or UMAP
Clustering: UNCURL, Louvain, Leiden algorithms
User input-guided splitting, merging, or deleting clusters of cells
Gene-cell type databases:
Enrichr
CellMarker
CellMesh
Probabilistic query methods

Figure 1 Overview Major components and workflow of UNCURL-App. Given the user-uploaded gene-cell matrix: 1) UNCURL is run on the data, producing a clustering; 2) dimensionality reduction is used for visualization, and 3) differentially expressed genes are identified. Then, the user may do cell type annotation using database queries on the top genes, and perform interactive re-clustering.



○ Cluster means ○ Post-processed cells ● Pre-processed cells ○ Label heatmap ○ Dendrogram heatmap

Top genes (1-vs-rest fold change)

Genes of interest (optional):

Number of top genes: 20

Cell labels: Cluster

Barplot updated

Currently selected top genes:

Cldn5
Cdh5
Sox18
Clec1a
Rasip1

Database: CellMeSH

Query CellMeSH with selected genes

Statistical test: Probabilistic model

Species: Mus musculus

Figure 2 User Interface Main user interface of UNCURL-App. The top left contains the scatterplot, showing a views of the cells and clusters. The top right is a barplot containing the top genes per cluster. Clicking on a cluster in the scatterplot updates the top genes. On the bottom is the database query view, which is used to identify cell types present in the dataset. **Options: (1)** These are options for what type of visualization to show in the scatterplot."Cluster means" shows one point for each cluster. "Post-processed cells" shows a dimensionality-reduced view of cells that takes the W matrix returned by UNCURL as input. "Pre-processed cells" (default) shows a similar view, but using the original data as input for dimensionality reduction. "Label heatmap" shows a comparison of two cell label sets. "Dendrogram heatmap" shows a clusters vs genes dendrogram. **(2)** Options for how to color the cells/points in the scatterplot. Can be based on cluster, gene expression, or a custom label set. **(3)** Options for how to identify the top genes. Can be 1-vs-rest or pairwise, p-value or ratio. **(4)** "Get cell/cluster info": get read/gene count for the selected cell and cluster. "Recluster": options to merge, split, delete, or create a new cluster. "Reanalyze": options to re-run the whole analysis. **(5)** There are a number of databases that can be used for cell type identification, including CellMarker, CellMeSH, and Gene Ontology.

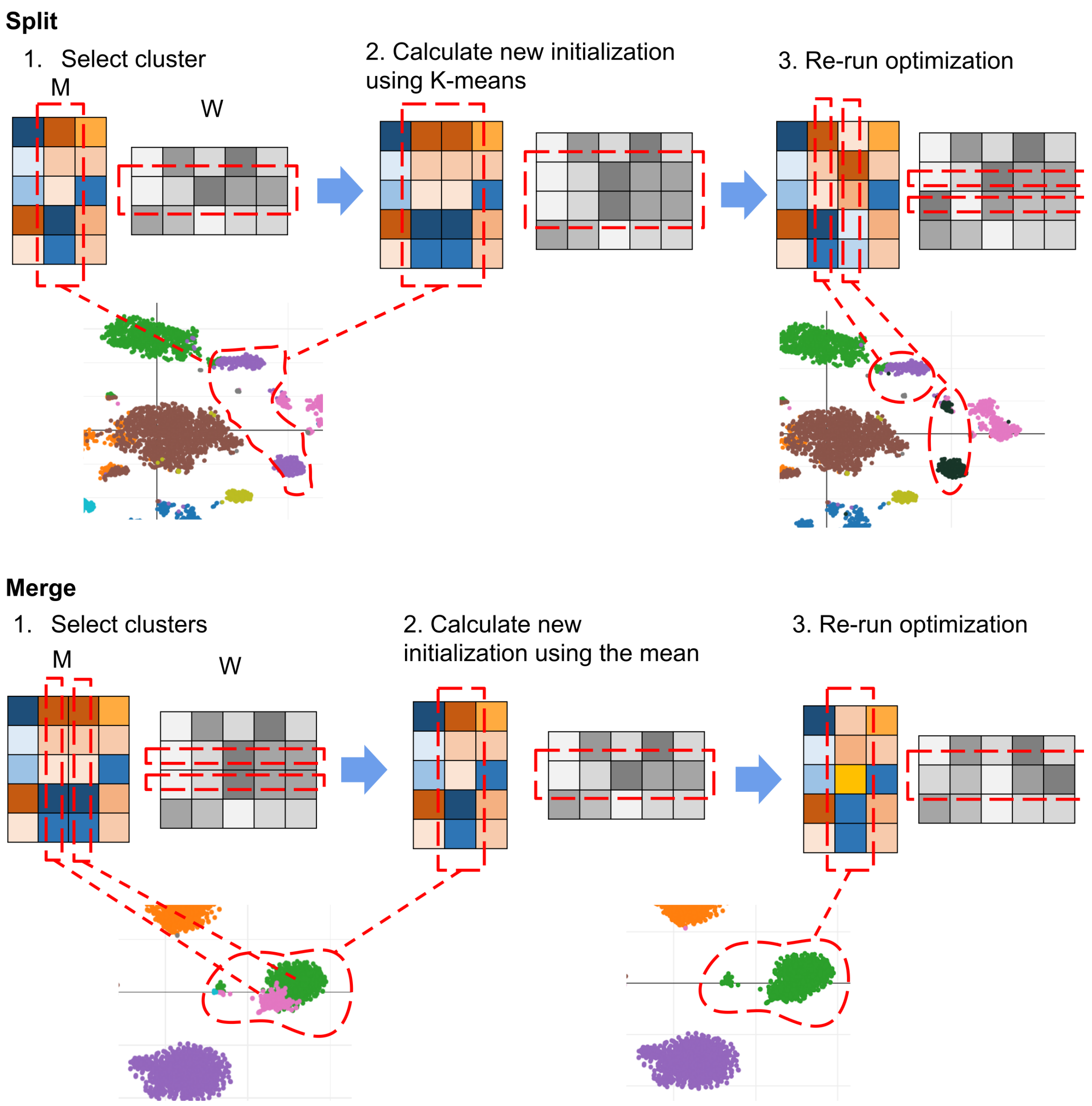
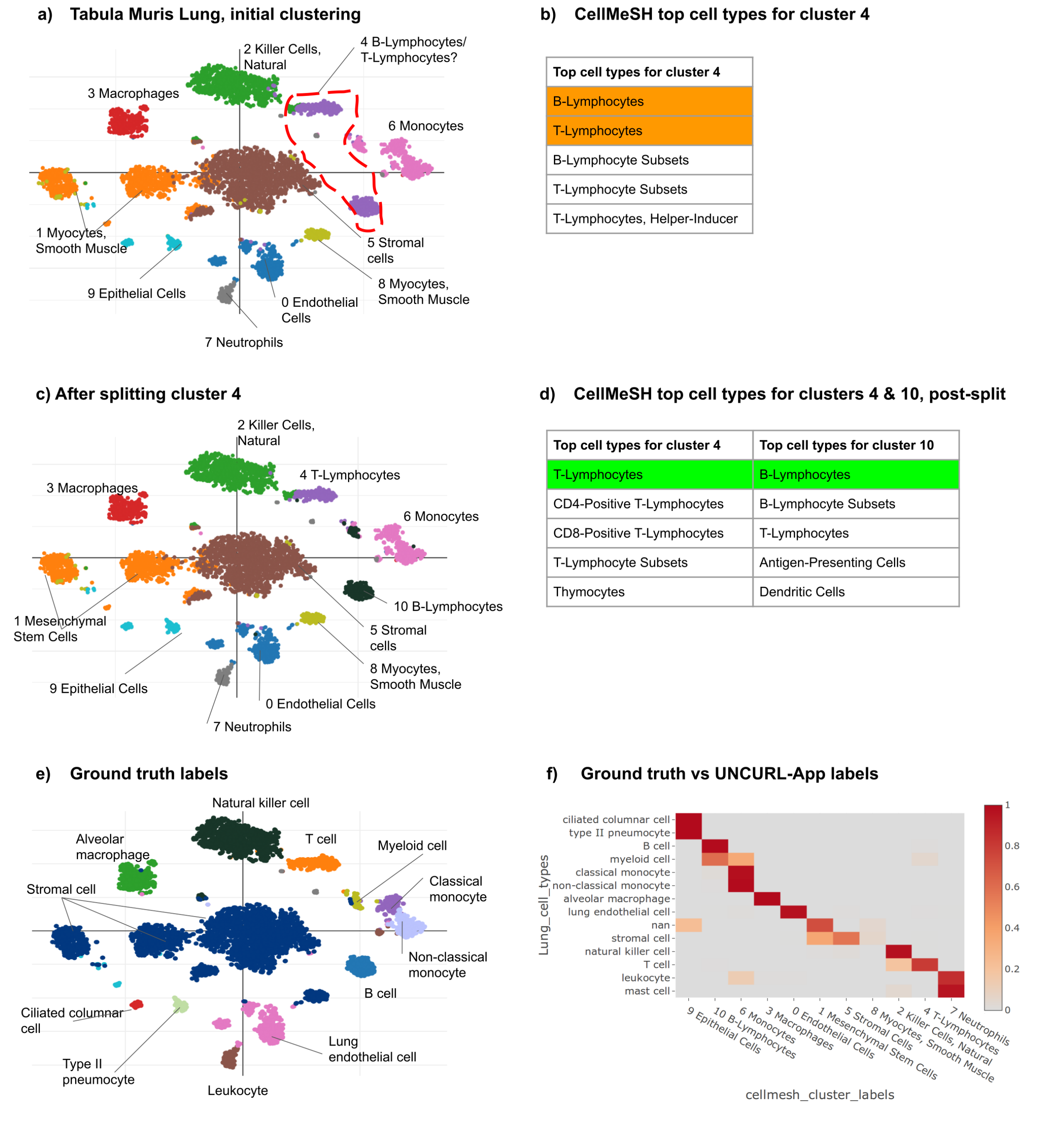


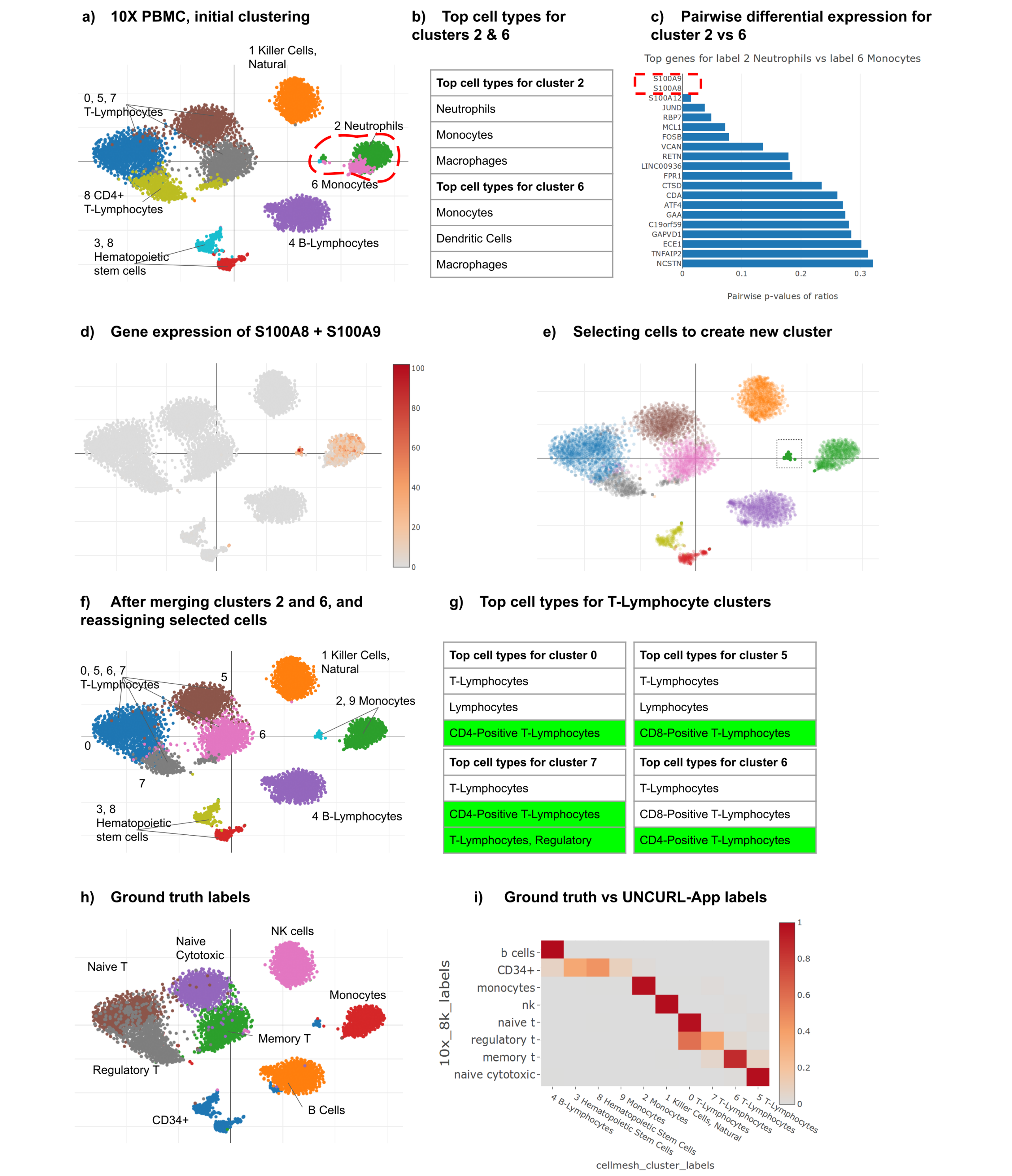
Figure 3 Splitting/Merging clusters Process of splitting and merging clusters works in UNCURL-App. Both processes begin with the M and W matrices returned by UNCURL, and the cluster(s) to merge or split. In order to split a cluster, a new initialization for M and W is created by running k-means on the cluster of cells to be split. Then, the UNCURL optimization process is re-run to create new matrices. The process for merging is analogous. A new initialization is created using the mean of the selected cells, and then the UNCURL optimization process is re-run to create a new M and W.



Top cell types for cluster 4
B-Lymphocytes
T-Lymphocytes
B-Lymphocyte Subsets
T-Lymphocyte Subsets
T-Lymphocytes, Helper-Inducer

Top cell types for cluster 4	Top cell types for cluster 10
T-Lymphocytes	B-Lymphocytes
CD4-Positive T-Lymphocytes	B-Lymphocyte Subsets
CD8-Positive T-Lymphocytes	T-Lymphocytes
T-Lymphocyte Subsets	Antigen-Presenting Cells
Thymocytes	Dendritic Cells

Figure 4 Tabula Muris Lung Examples of running UNCURL-App on the Tabula Muris lung dataset from [7] **(a) (c) (e)** These show the same tSNE scatterplot, with clusters from the initial clustering and labels from CellMeSH, the clusters and labels after splitting cluster 4, and the ground truth labels from the original paper, respectively. **(b) (d)** These show the most relevant cell types returned by CellMeSH for the given clusters. Cell types matching ground truth labels are highlighted in green. **(f)** This is a heatmap showing the relationship between the ground truth clusters and the clusters generated by UNCURL-App, after the split. The colors indicate the proportion of the ground truth cluster (left) that overlap with the UNCURL-App cluster (bottom).



Top cell types for cluster 2
Neutrophils
Monocytes
Macrophages
Top cell types for cluster 6
Monocytes
Dendritic Cells
Macrophages

Top cell types for cluster 0	Top cell types for cluster 5
T-Lymphocytes	T-Lymphocytes
Lymphocytes	Lymphocytes
CD4-Positive T-Lymphocytes	CD8-Positive T-Lymphocytes
Top cell types for cluster 7	**Top cell types for cluster 6**
T-Lymphocytes	T-Lymphocytes
CD4-Positive T-Lymphocytes	CD8-Positive T-Lymphocytes
T-Lymphocytes, Regulatory	CD4-Positive T-Lymphocytes

Figure 5 10X PBMC Examples of running UNCURL-App on the 10X PBMC dataset from [37] **(a) (f) (h)** These show the same tSNE scatterplot, with clusters from the initial clustering and labels from CellMeSH, the clusters and labels after merging clusters 2 and 6 and creating cluster 9, and the ground truth labels from the original dataset, respectively. **(b) (g)** These show the most relevant cell types returned by CellMeSH for the given clusters. Cell types matching ground truth labels are highlighted in green. **(d)** This shows the sum of the gene expressions of the two genes S100A8 and S100A9. **(e)** This shows the process by which the user can select a group of cells to create a new cluster, using the "Box Select" tool from plotly. **(i)** This is a heatmap showing the relationship between the ground truth clusters and the clusters generated by UNCURL-App, after the split. The colors indicate the proportion of the ground truth cluster (left) that overlap with the UNCURL-App cluster (bottom).

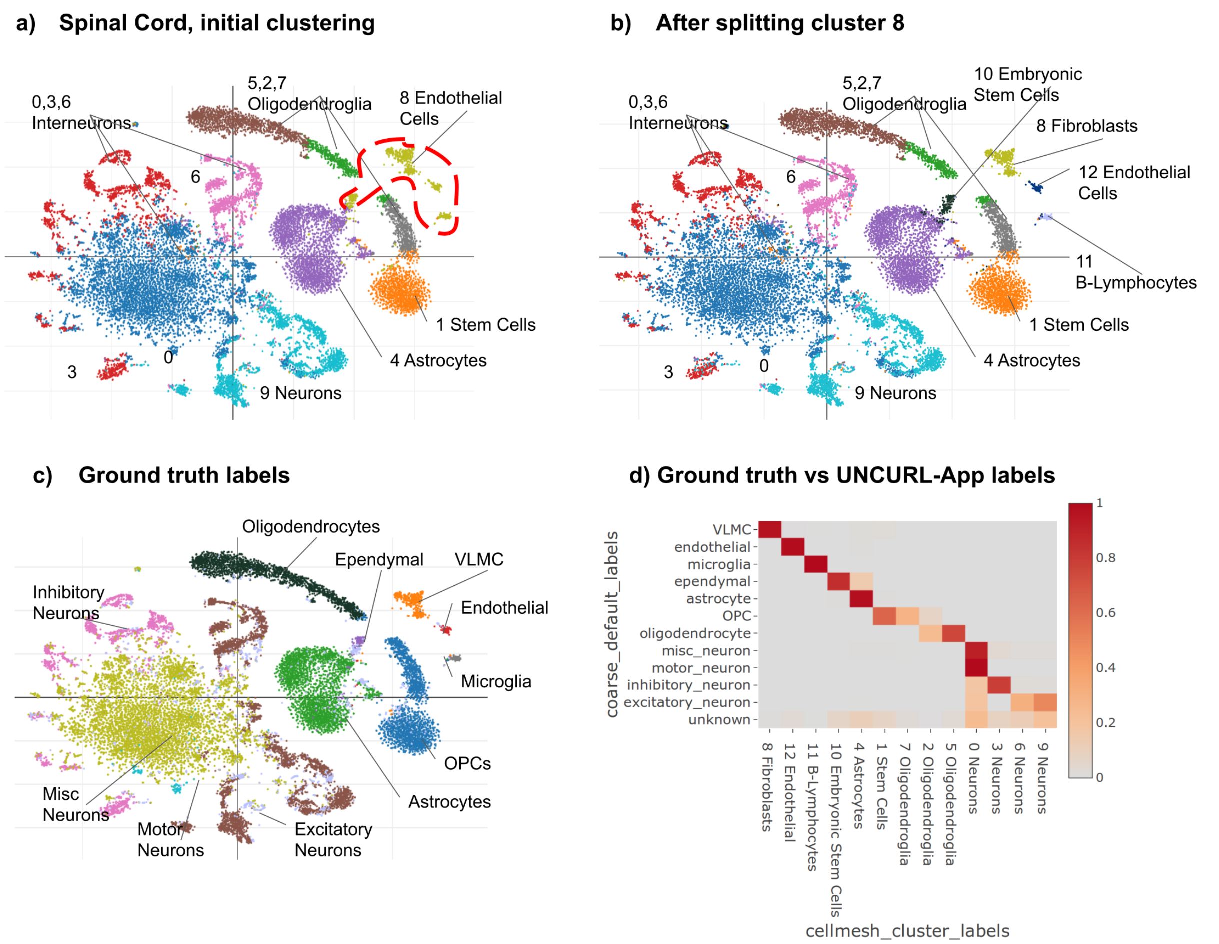


Figure 6 Spinal Cord Examples of running UNCURL-App on the split-seq spinal cord dataset from [9] **(a)** tSNE scatterplot, with 10 clusters from the initial clustering and labels from CellMeSH. Cluster 8 which consists of multiple disconnected groups of cells is highlighted. **(b)**Clusters and labels from CellMeSH after splitting cluster 8. **(c)** Clusters and labels after refinement with ground truth labels from the original dataset. **(d)** This is a heatmap showing the relationship between the ground truth clusters and the clusters generated by UNCURL-App, after the split. The colors indicate the proportion of the ground truth cluster (left) that overlap with the UNCURL-App cluster (bottom).