An Interpretable Framework for Clustering Single-Cell RNA-Seq Datasets

Jesse M. Zhang[1], Jue Fan[2], H. Christina Fan[2], David Rosenfeld[2], and David N. Tse[1,*]

[1]Department of Electrical Engineering, Stanford, 94305 Stanford, California, USA
[2]BD Genomics, 94025 Menlo Park, California, USA
[*]dntse@stanford.edu

ABSTRACT

With the recent proliferation of single-cell RNA-Seq experiments, several methods have been developed for unsupervised analysis of the resulting datasets. These methods often rely on unintuitive hyperparameters and do not explicitly address the subjectivity associated with clustering. In this work, we present DendroSplit, an interpretable framework for analyzing single-cell RNA-Seq datasets that addresses both these issues. Under this framework, we cluster using feature selection to uncover multiple levels of biologically meaningful populations in the data. We analyze several landmark single-cell datasets, demonstrating both the method's efficacy and computational efficiency. We provide the full DendroSplit software package at `https://github.com/jessemzhang/dendrosplit`.

Background

In recent years, single-cell RNA-Seq has proven to be a powerful approach for studying biological samples in various settings[1]. Scientists have leveraged this technology to shed light on how cells differentiate[2–6], investigate known cell types[7–10], and discover new cell types and gene patterns[11–17]. These efforts have yielded a plethora of diverse datasets sharing characteristics such as missing entries (drop-out events) and high dimensionality. Additionally, technological breakthroughs such as droplet encapsulation, molecular barcoding, and cheap parallelization have produced datasets involving tens of thousands and even millions of individuals cells[17–22]. After obtaining such datasets, scientists are often interested in clustering the high-dimensional points corresponding to individual cells, ideally recovering known cell populations while discovering new and perhaps rare cell types. While the definition of a cell type is not precise[23], biologists agree that gene expression levels are highly relevant. With gene expression dictating protein expression (and hence cellular function), identifying the genes that distinguish a cell type is of paramount importance. Therefore from a computational perspective, there are two key problems in downstream analysis: 1) clustering and 2) feature selection, also known as differential expression.

General-purpose clustering algorithms such as K-means, DBSCAN[24], affinity propagation[25], and spectral clustering[26] have performed well for several single-cell datasets[27]. In order to achieve good performance, however, the datasets often need to be carefully preprocessed, and the algorithms require non-intuitive hyperparameter tuning. To address specific computational challenges of single-cell RNA-Seq datasets, researchers have developed a wide array of application-specific clustering algorithms[28–34] and packages for end-to-end analysis[21,35–39]. Regardless of which set of these tools one uses, finding the right approach for clustering a specific dataset requires careful design of the computational workflow, but often finding a good combination of clustering algorithm and hyperparameters is time-consuming and difficult. Additionally, none of these approaches explicitly addresses the inherent subjectiveness behind clustering, which stems from the potential existence of subtypes and sub-subtypes.

With an emphasis on interpretability and ease of exploratory analysis, we introduce DendroSplit, a framework for clustering single-cell RNA-Seq data. In addition to speed, the framework has the following advantages:

- Gene-based justification for all decisions made when generating clusters
- Interpretable hyperparameters
- Ability to cheaply produce multiple clusterings for the same dataset
- Ease of incorporation into existing single-cell RNA-Seq workflows

At a high level, the approach leverages a feature selection algorithm to generate biologically meaningful clusters. The end-to-end DendroSplit workflow is illustrated in Figure 1A. After preprocessing the $N \times M$ expression matrix X (where N

and M represent the number of cells and genes, respectively), we generate the $N \times N$ distance matrix D. We use hierarchical clustering to iteratively group cells based on their pairwise distances, obtaining a dendrogram, a tree-like data structure illustrating how grouping was performed. The split step starts at the root of the tree. Each node in the dendrogram represents a potential partitioning of a larger cluster into two smaller ones. If this "split" results in two adequately separated clusters (according to a metric we call the **separation score**), the split is deemed valid and the algorithm continues on the new clusters. Otherwise, the algorithm terminates for the subtree below the node. After the split step, DendroSplit performs a pairwise comparison of the resulting clusters, repeatedly merging clusters until all clusters are sufficiently separated. The merge step counteracts the greedy nature of hierarchical clustering, allowing DendroSplit to compare clusters that may have incorrectly ended up far away from one another in the dendrogram. The overall approach involves two intuitive hyperparameters: the separation score threshold for accepting a split, and the separation score threshold for accepting a merge.

We use the term "framework" to underline how specific design choices for certain components in the workflow such as the separation score will result in different clustering "methods." Our choice of separation score is motivated by a key assumption: **if two cell populations are of different types, then there should exist at least one gene that is differentially expressed between the two populations**. Given a candidate split in the dendrogram, we perform a Welch's t-test for each gene. The separation score is $-\log(p_{\text{min}})$ where p_{min} represents the smallest p-value achieved (Figure 1B), and we will be using this definition of separation score for all experiments presented in this work. We demonstrate that the deterministic method outlined in Figure 1 is applicable to a wide variety of single-cell datasets. We show how DendroSplit can help us investigate the most significant genes considered at each split or merge, providing insight for how clusters are generated. Finally, we show how DendroSplit can cheaply generate several clusterings for different hyperparameter values.

Some clustering approaches similar to DendroSplit exist in literature. For example, the most common method of generating clusters from a dendrogram involves simply cutting the dendrogram horizontally at some fixed height. This rigid approach often fails to generate meaningful clusters for more complex datasets. The Dynamic Tree Cut algorithm[40] adds significant flexibility and processes the dendrogram based on an adaptive cut. Though it does not explicitly use a dendrogram, the backSPIN algorithm[12] also uses cell-cell similarities to perform iterative splitting. Unlike DendroSplit, both of these algorithms require choosing unintuitive hyperparameter cutoffs based on nuanced criteria. The most similar clustering approach was used by[15] for analyzing their human brain single-cell dataset. Their approach fits into the DendroSplit framework, using a separation score based on random forests. This separation score, compared to the separation score mentioned above, has an element of randomness, is significantly more computationally expensive, and requires less intuitive hyperparameter choices.

Results

Ground-truth datasets

To test the effectiveness of the DendroSplit framework, we first test the approach on datasets where the ground truth is known.

Synthetic datasets

Figure 2 shows the performance of DendroSplit on four synthetic datasets[41]. Since the 2-dimensional data points have clear, intuitive clustering structure, pairwise Euclidean distance is a natural choice. Figure 2A shows the exploratory power of DendroSplit on a toy dataset of oddly-oriented clusters. Because DendroSplit saves the information gathered at each valid split, we can easily investigate how the clustering was performed. At a given split, we can identify the points that went into each partition and look at the partition-specific distributions of the feature that validated the split. Thus the true advantage of DendroSplit is in its ability to justify its behavior with interpretable results. Figure 2B shows that DendroSplit has the power to uncover several clusters especially when the distance metric (Euclidean) suits the type of data (2-D Gaussian balls). Figure 2C emphasizes that, like other methods, DendroSplit cannot automatically overcome poor choices in preprocessing and distance metric selection.

Single-cell RNA-Seq datasets

Figure 3 shows the performance of DendroSplit on four single-cell RNA-Seq datasets featuring high-quality labels.[34] refers to these datasets as "gold standards." We chose four datasets with varying amounts of cells, genes, and total clusters to understand how they affect the behavior of DendroSplit. We see that when N is on the order of 100s, the runtime is widely determined by M, the number of independent Welch's t-tests that must be performed at every split. Clusters obtained from the split and merge steps are compared to the true labels using the adjusted Rand index (see Methods), which ranges from 0 for poor matching to 1 for perfectly matched labels. Figure 3 shows that for the[2,5,8] datasets, most of the final ARI is achieved after the split step. Therefore most of the information captured by the clusters lies in one of the dendrogram's subtrees. Due to how the dendrogram is constructed, a cell may end up being split off into its own cluster and is temporarily labeled as a non-classified "Singleton." The merge step cleans up singletons and small clusters, resulting in a higher ARI. For the[9] dataset, the ARI increases dramatically after the merge step. This is due to the fact that after the split step, 1) 15 of the 124 cells end up as

singletons, and 2) splitting generates twice as many clusters as needed. In fact, for this dataset, dividing each true cluster into two equal-sized parts would result in an ARI of 0.74 when compared with the original labels. A more detailed visual analysis of the[9] dataset is given in Supplementary Figure 1. We defer the discussion on choosing split and merge thresholds to the Hyperparameter Sweeping section below. Under certain conditions, some cells may remain in their own clusters even after the merge step (see Methods). These cells are analogous to outliers.

Exploratory analysis

We further demonstrate the exploratory power of DendroSplit on a dataset of 430 cells, 5948 features from five primary human glioblastomas[7]. Without any further preprocessing, DendroSplit recovers five clusters corresponding to the five glioblastomas (Figure 4A). Furthermore, DendroSplit can justify its findings by showing us the gene that played the largest role in validating each split. Splits 4 and 5 in Figure 4B show distinctively how *SEC61G*, for example, distinguishes MGH26 cells from MGH29 and MGH31 cells. The analysis also gives insight on how the hierarchical clustering was performed. The cells from MGH26 were split in half during earlier stages of clustering, which is why they end up in separate superclusters at the root node. This is an artifact of the greedy nature of hierarchical clustering where clusters that should be close together may end up far apart. Merge 1 in Figure 4C shows DendroSplit fixing this. At the same time, we see that *PAN3* may be a valid marker for distinguishing these two subtypes within MGH26 cells. Further analysis and perhaps side information would be needed to decide whether or not these two subtypes are truly different. DendroSplit handles the subjectiveness associated with clustering by showing the factors that contribute to its decisions.

Performance on larger single-cell datasets

We use DendroSplit to re-analyze three large single-cell RNA-Seq datasets that utilize unique molecular identifiers (UMIs) for quantifying genes[12,17,20]. Unlike for previous single-cell RNA-Seq datasets, the labels for these datasets were assigned using diverse computational methods. Figure 5 first shows that performing a feature selection step prior to analysis with DendroSplit decreases the runtime dramatically. In fact, for the[12] dataset, filtering out genes using the procedure described by[20] reduces both M and the runtime by a factor of 16.6. Additionally, the filtering out of noisy features improves the quality of the distance metric, and we see that the ARI improves dramatically. We also report that using a much smaller split threshold of 15 results in 43 non-singleton clusters. When compared with[12]'s 47 subclasses, we achieve an ARI of 0.42. The gene filtering procedure is used for the all datasets presented in Figures 5 and 6.

For the three datasets analyzed in Figure 5, DendroSplit generates similar but not identical labels. Figure 6A shows that DendroSplit disagrees even more strongly on a dataset of 17426 cells, 908 features from fresh peripheral blood mononuclear cells[21] (PBMCs). Noting that the merge step does not increase the ARI significantly, we focus on the split step labels. Although 15 valid splits were recorded, we investigate only the 5 shown in Figure 6B. For the remaining splits, see Supplementary Figure 2. Split 1 was validated due to a lack of expression of several genes (*FCGR3A*, *LY86*, *FCN1*, and *IFI30*) in the red population, which we match to the authors' CD34+ cells. Split 2 shows the separation of the red cells from the green cells based on high expression of *NKG7* and *GNLY*, markers for natural killer (NK) cells. The green cluster in split 5 likely corresponds to cytotoxic T cells based on increased expression of *GZMH*. The red cluster in split 9 shows greater expression of *CD79A* and may therefore represent B cells. Finally, the red cluster in split 14 does not have an obvious match with any of[21]'s set of labels. DendroSplit shows us that the existence of this cluster is justified based on increased expression of several genes including *FCGR3A*, *CFD*, and *LST1*. A one-versus-rest differential expression analysis based on independent Welch's t-tests (see Supplementary Table 1) further shows that *PSAP* and *SERPINA1* are also overexpressed, indicating that the red cluster (cluster 6 after the merge step in Figure 6A) corresponds to some type of monocyte. We also repeat this analysis with the full 68579 cells, 20374 genes dataset, and the results are shown in Supplementary Figure 3.

Hyperparameter Sweeping

When choosing hyperparameters for DendroSplit, a relatively small split threshold such as 20 and a merge threshold that's half the split threshold often yields reasonable initial results. The DendroSplit approach can also rapidly generate several clusterings based on different split thresholds. Since DendroSplit saves the p-values and cell IDs considered at each split, we can obtain several split-step clustering results by exploiting the fact that the clusters generated with a smaller score threshold partition the clusters generated with a larger score threshold. The merge threshold can then be chosen by looking at pairwise separation scores between clusters. Figure 7 demonstrates the score threshold sweeping procedure for the[5] and[12] datasets. As observed in the experiments, larger datasets often require larger thresholds due to the t-statistic generally increasing with N.

Conclusion

Single-cell RNA-Seq has shown promise for numerous biological studies. With the technology rapidly improving, we can only expect increased cell throughput. The computational challenges associated with handling larger datasets include the

challenge of properly clustering a new dataset in an unsupervised yet interpretable way and without the need for expensive hyperparameter tuning.

In this work, we presented a novel framework for tackling the single-cell RNA-Seq clustering problem. We demonstrated that a dendrosplit-splitting approach based on a separation score was key for uncovering the multiple layers of biological information within a dataset. In addition to recovering results from a diverse set of single-cell studies, we showed that the framework could cheaply produce several clusterings of the same dataset. Most importantly, the algorithm could justify each of its decisions in an interpretable way. Thus, DendroSplit is suitable as a backend algorithm for interactive analysis and interpretation.

For the analyses in this work, we used a separation score based on a computationally cheap method of performing differential expression and a simple definition of cell type. Separation scores based on more complex methods of evaluating differential expression such as those presented by[31,42–48] may yield better results at the cost of greater computation. Additionally, just like for other clustering approaches, existing tools including those designed for outlier detection[13,49], drop-out imputation[50], and correcting other sources of technical noise[51–59] can be easily incorporated into the DendroSplit framework by applying the desired correction procedures before the clustering step.

Method

Both the software package and the code used to generate the results presented in this paper are available online. Both are written in Python 2.7 and are available at `https://github.com/jessemzhang/dendrosplit`.

Data preprocessing

For all single-cell datasets, we applied a logarithmic transformation $\log_{10}(X+1)$ to the raw expressions levels. We analyzed 9 datasets in this paper. For each dataset, genes that had 0 expression across all cells were removed. Additionally, all datasets consisting of over 1000 cells underwent feature selection based on the method proposed in[20]. The M genes were sorted into equal-sized bins depending on their mean expression values. Within a bin, genes were z-normalized based on their dispersions, where the dispersion for a gene is defined as the variance divided by the mean. Only genes corresponding to a z-score above a certain cutoff were retained. For the[12,17,21] datasets, we used DendroSplit's default setting of 5 bins with a z-cutoff of 1.5. For the[20] dataset, we first removed cells with less than 900 counts across all genes just like in[30], reducing the original 44808 cells to 11040. We then used[20]'s gene-filtering settings of 20 bins with a z-cutoff of 1.7. For the[21] dataset, reducing the number of genes resulted in several of the original 68579 cells having few and even 0 counts across all genes. We removed cells with less than 50 counts across all genes, resulting in 17426 cells, and again filtered out genes with 0 counts across all remaining cells. We also experimented with standardizing all log-transformed genes to have 0 mean and unit variance across all cells, but the increased computational overhead did not yield better results.

Distance metric

For all single-cell datasets, we used the correlation distance. The correlation distance between $\mathbf{x}_i$ and $\mathbf{x}_j$ corresponding to cells i and j is

$$d(\mathbf{x}_i, \mathbf{x}_j) = 1 - r(\mathbf{x}_i, \mathbf{x}_j)$$

where r is the Pearson correlation coefficient. Therefore d is bounded between 0 and 1. This distance metric has the advantage of being agnostic to both shift and scale, making it robust to certain biases we would expect to vary across datasets. As a caveat, the distance metric has the disadvantage of depending on the number of zeros, and therefore the distance between two cells before and after gene filtering may be different due to removal of entries equal to 0. For all experiments in this work, distance matrix computations were parallelized on 32 cores and computed using the `scikit-learn` Python package[60].

Hierarchical clustering

DendroSplit performs hierarchical clustering using the Scipy Python package[61]. One source of ambiguity for hierarchical clustering lies in the method for determining the distance between two clusters. We found that the "complete" method produces the best results, and this is the method used for all experiments reported above. For this method, the distance between two clusters is equal to the largest distance between a point from the first cluster and another point from the second cluster.

Separation score

The separation score effectively serves as a distance metric between two clusters, quantifying how different they are (see the supplementary material for further discussion). The cell-type assumption discussed in the Background section can also be phrased as: if two cell populations are of different types, then projection along one of the M axes should result in two

distinguishable point clouds. For all experiments performed in this work, we defined the separation score between the $N_1 \times M$ population $\mathbf{X}$ and the $N_2 \times M$ population $\mathbf{Y}$ as

$$s(\mathbf{X}, \mathbf{Y}) = -\log_{10}(\min_i p(\mathbf{X}_{.i}, \mathbf{Y}_{.i}))$$

where $p(\mathbf{X}_{.i}, \mathbf{Y}_{.i})$ represents the p-value achieved using a Welch's t-test for gene i. $\mathbf{X}_{.i}$ represents the ith column of $\mathbf{X}$ corresponding to the expression of gene i in population $\mathbf{X}$. Welch's t-test is similar to Student's t-test but is more reliable when the two populations have unequal variance and size[62]. Compared to other differential expression approaches, Welch's t-test is computationally cheap.

As an implementation note, if for a given split two or more genes have the exact same score, these genes are ranked by the magnitude of the t statistic. We note that because we are using Welch's t-test rather than Student's t-test, the degrees of freedom associated with each test is different, and hence outputting the largest t statistic is only approximately sound. This event may occur for larger datasets and for splits near the root of the dendrogram where p-values may be quite small, resulting in an underflow issue and a score of ∞.

Adjusted Rand index

The adjusted Rand index was used to quantify how well two sets of cluster labels matched. For a set of n elements, we let X and Y represent two partitions of the elements. X_i represents the set of elements in partition i according to X. The adjusted Rand index is defined as:

$$ARI = \frac{\sum_{ij}\binom{n_{ij}}{2} - [\sum_i \binom{a_i}{2} \sum_j \binom{b_j}{2}]/\binom{n}{2}}{\frac{1}{2}[\sum_i \binom{a_i}{2} + \sum_j \binom{b_j}{2}] - [\sum_i \binom{a_i}{2} \sum_j \binom{b_j}{2}]/\binom{n}{2}}$$

where n_{ij} is the number of elements in common between X_i and Y_j, $a_i = \sum_k n_{ik}$, and $b_j = \sum_k n_{kj}$.

Handling singletons

In addition to the split and merge thresholds, the two major hyperparameters discussed above, the DendroSplit framework can also be customized using three minor hyperparameters. These three hyperparameters are relevant for finding singletons (clusters containing one point), which are analogous to outliers.

Two of these hyperparameters are relevant for the split step. The first is the minimum cluster size. During a split, if one of the two candidate clusters contains less points than the minimum cluster size, that cluster is disbanded (each of its points are labeled as "Singleton") and the algorithm continues on the other candidate. The second hyperparameter is the disband percentile. If neither subtree meets the minimum cluster size requirement or if a candidate split does not achieve a high enough separation score, we look at the pairwise distances amongst samples in this final cluster. If all of them are greater than a certain percentile of a distances in the distance matrix, then all points in this final cluster are marked as singletons. For all experiments performed in this work, the minimum cluster size was set to 2 (the smallest value) and the disband percentile was set to 50.

Before merging clusters, each singleton obtained during the split step is assigned to the same cluster as its nearest neighbor. If the distance between a singleton and its nearest neighbor is greater than a certain percentile of all pairwise distances in the distance matrix, then the singleton remains unclassified. This percentile is the third minor hyperparameter and was set to 90 for all experiments performed in this work.

Availability of data and materials

The[2,5,7–9,12,17,20,21] datasets were obtained from public repositories. The[5,8] datasets were obtained from the GitHub repository for[30]. Both the software package and the code used to generate the results presented in this paper are available online at `https://github.com/jessemzhang/dendrosplit`. DendroSplit is licensed and distributed under a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International license.

Competing interests

JMZ is a past employee of BD Genomics. JF, HCF, and DR are past and current employees of BD Genomics.

Author's contributions

JMZ conceived the idea of splitting dendrograms and merging clusters using p-values, wrote the software package, performed analyses of data, interpreted results and wrote the manuscript. JF, HCF, DR, and DNT interpreted results, supervised the project and wrote the manuscript.

Acknowledgements

We thank Govinda Kamath and Vasilis Ntranos for feedback and useful discussions about analyzing single-cell RNA-Seq datasets.

Funding

JMZ and DNT are supported in part by the National Human Genome Research Institute of the National Institutes of Health under award number R01HG008164.

References

1. Yuan, G.-C. *et al.* Challenges and emerging directions in single-cell analysis. *Genome Biology* **18**, 84 (2017). URL https://doi.org/10.1186/s13059-017-1218-y.
2. Biase, F. H., Cao, X. & Zhong, S. Cell fate inclination within 2-cell and 4-cell mouse embryos revealed by single-cell rna sequencing. *Genome research* **24**, 1787–1796 (2014).
3. Trapnell, C. *et al.* The dynamics and regulators of cell fate decisions are revealed by pseudotemporal ordering of single cells. *Nature biotechnology* **32**, 381–386 (2014).
4. Goolam, M. *et al.* Heterogeneity in oct4 and sox2 targets biases cell fate in 4-cell mouse embryos. *Cell* **165**, 61 – 74 (2016). URL http://www.sciencedirect.com/science/article/pii/S0092867416300617.
5. Kolodziejczyk, A. A. *et al.* Single cell rna-sequencing of pluripotent states unlocks modular transcriptional variation. *Cell Stem Cell* **17**, 471 – 485 (2015). URL http://www.sciencedirect.com/science/article/pii/S193459091500418X.
6. Treutlein, B. *et al.* Reconstructing lineage hierarchies of the distal lung epithelium using single-cell rna-seq. *Nature* **509**, 371–375 (2014). URL http://dx.doi.org/10.1038/nature13173.
7. Patel, A. P. *et al.* Single-cell rna-seq highlights intratumoral heterogeneity in primary glioblastoma. *Science* **344**, 1396–1401 (2014). URL http://science.sciencemag.org/content/344/6190/1396. http://science.sciencemag.org/content/344/6190/1396.full.pdf.
8. Pollen, A. A. *et al.* Low-coverage single-cell mrna sequencing reveals cellular heterogeneity and activated signaling pathways in developing cerebral cortex. *Nat Biotech* **32**, 1053–1058 (2014). URL http://dx.doi.org/10.1038/nbt.2967.
9. Yan, L. *et al.* Single-cell rna-seq profiling of human preimplantation embryos and embryonic stem cells. *Nat Struct Mol Biol* **20**, 1131–1139 (2013). URL http://dx.doi.org/10.1038/nsmb.2660.
10. Buettner, F. *et al.* Computational analysis of cell-to-cell heterogeneity in single-cell rna-sequencing data reveals hidden subpopulations of cells. *Nat Biotech* **33**, 155–160 (2015). URL http://dx.doi.org/10.1038/nbt.3102.
11. Usoskin, D. *et al.* Unbiased classification of sensory neuron types by large-scale single-cell rna sequencing. *Nat Neurosci* **18**, 145–153 (2015). URL http://dx.doi.org/10.1038/nn.3881.
12. Zeisel, A. *et al.* Cell types in the mouse cortex and hippocampus revealed by single-cell rna-seq. *Science* **347**, 1138–1142 (2015). URL http://science.sciencemag.org/content/347/6226/1138. http://science.sciencemag.org/content/347/6226/1138.full.pdf.
13. Grun, D. *et al.* Single-cell messenger rna sequencing reveals rare intestinal cell types. *Nature* **525**, 251–255 (2015). URL http://dx.doi.org/10.1038/nature14966.
14. Ting, D. T. *et al.* Single-cell {RNA} sequencing identifies extracellular matrix gene expression by pancreatic circulating tumor cells. *Cell Reports* **8**, 1905 – 1918 (2014). URL http://www.sciencedirect.com/science/article/pii/S2211124714007050.
15. Lake, B. B. *et al.* Neuronal subtypes and diversity revealed by single-nucleus rna sequencing of the human brain. *Science* **352**, 1586–1590 (2016). URL http://science.sciencemag.org/content/352/6293/1586. http://science.sciencemag.org/content/352/6293/1586.full.pdf.
16. Deng, Q., Ramsköld, D., Reinius, B. & Sandberg, R. Single-cell rna-seq reveals dynamic, random monoallelic gene expression in mammalian cells. *Science* **343**, 193–196 (2014). URL http://science.sciencemag.org/content/343/6167/193. http://science.sciencemag.org/content/343/6167/193.full.pdf.

17. Birey, F. *et al.* Assembly of functionally integrated human forebrain spheroids. *Nature* **545**, 54–59 (2017). URL http://dx.doi.org/10.1038/nature22330.

18. Fan, H. C., Fu, G. K. & Fodor, S. P. A. Combinatorial labeling of single cells for gene expression cytometry. *Science* **347** (2015). URL http://science.sciencemag.org/content/347/6222/1258367. http://science.sciencemag.org/content/347/6222/1258367.full.pdf.

19. Klein, A. M. *et al.* Droplet barcoding for single-cell transcriptomics applied to embryonic stem cells. *Cell* **161**, 1187 – 1201 (2015). URL http://www.sciencedirect.com/science/article/pii/S0092867415005000.

20. Macosko, E. Z. *et al.* Highly parallel genome-wide expression profiling of individual cells using nanoliter droplets. *Cell* **161**, 1202–1214. URL http://dx.doi.org/10.1016/j.cell.2015.05.002.

21. Zheng, G. X. Y. *et al.* Massively parallel digital transcriptional profiling of single cells **8**, 14049 EP – (2017). URL http://dx.doi.org/10.1038/ncomms14049.

22. Cao, J. *et al.* Comprehensive single-cell transcriptional profiling of a multicellular organism. *Science* **357**, 661–667 (2017). URL http://science.sciencemag.org/content/357/6352/661. http://science.sciencemag.org/content/357/6352/661.full.pdf.

23. Trapnell, C. Defining cell types and states with single-cell genomics. *Genome Research* **25**, 1491–1498 (2015). URL http://www.ncbi.nlm.nih.gov/pmc/articles/PMC4579334/.

24. Ester, M., Kriegel, H.-P., Sander, J., Xu, X. *et al.* A density-based algorithm for discovering clusters in large spatial databases with noise. In *Kdd*, vol. 96, 226–231 (1996).

25. Dueck, D. & Frey, B. J. Non-metric affinity propagation for unsupervised image categorization. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, 1–8 (IEEE, 2007).

26. Ng, A. Y., Jordan, M. I. & Weiss, Y. On spectral clustering: Analysis and an algorithm. In *Advances in neural information processing systems*, 849–856 (2002).

27. Ntranos, V., Kamath, G. M., Zhang, J. M., Pachter, L. & David, N. T. Fast and accurate single-cell rna-seq analysis by clustering of transcript-compatibility counts. *Genome biology* **17**, 112 (2016).

28. Rostom, R., Svensson, V., Teichmann, S. A. & Kar, G. Computational approaches for interpreting scrna-seq data. *FEBS letters* (2017).

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

30. Wang, B., Zhu, J., Pierson, E., Ramazzotti, D. & Batzoglou, S. Visualization and analysis of single-cell rna-seq data by kernel-based similarity learning. *Nat Meth* **14**, 414–416 (2017). URL http://dx.doi.org/10.1038/nmeth.4207.

31. Guo, M., Wang, H., Potter, S. S., Whitsett, J. A. & Xu, Y. Sincera: a pipeline for single-cell rna-seq profiling analysis. *PLoS computational biology* **11**, e1004575 (2015).

32. Xu, C. & Su, Z. Identification of cell types from single-cell transcriptomes using a novel clustering method. *Bioinformatics* **31**, 1974–1980 (2015).

33. žurauskienė, J. & Yau, C. pcareduce: hierarchical clustering of single cell transcriptional profiles. *BMC Bioinformatics* **17**, 140 (2016). URL https://doi.org/10.1186/s12859-016-0984-y.

34. Kiselev, V. Y. *et al.* Sc3: consensus clustering of single-cell rna-seq data. *Nat Meth* **14**, 483–486 (2017). URL http://dx.doi.org/10.1038/nmeth.4236.

35. Satija, R., Farrell, J. A., Gennert, D., Schier, A. F. & Regev, A. Spatial reconstruction of single-cell gene expression data. *Nat Biotech* **33**, 495–502 (2015). URL http://dx.doi.org/10.1038/nbt.3192.

36. Wolf, F. A., Angerer, P. & Theis, F. J. Scanpy for analysis of large-scale single-cell gene expression data. *bioRxiv* 174029 (2017).

37. Lun, A. T., McCarthy, D. J. & Marioni, J. C. A step-by-step workflow for low-level analysis of single-cell rna-seq data with bioconductor. *F1000Research* **5** (2016).

38. Qiu, X. *et al.* Single-cell mrna quantification and differential analysis with census. *Nat Methods* **14**, 309–15 (2017).

39. McCarthy, D. J., Campbell, K. R., Lun, A. T. & Wills, Q. F. Scater: pre-processing, quality control, normalization and visualization of single-cell rna-seq data in r. *Bioinformatics* **33**, 1179–1186 (2017).

40. Langfelder, P., Zhang, B. & Horvath, S. Defining clusters from a hierarchical cluster tree: the dynamic tree cut package for r. *Bioinformatics* **24**, 719–720 (2007).

41. Franti, P. Clustering datasets (2015). URL http://cs.uef.fi/sipu/datasets/.

42. Love, M., Anders, S. & Huber, W. Differential analysis of count data–the deseq2 package. *Genome Biol* **15**, 550 (2014).

43. Robinson, M. D., McCarthy, D. J. & Smyth, G. K. edger: a bioconductor package for differential expression analysis of digital gene expression data. *Bioinformatics* **26**, 139–140 (2010).

44. Finak, G. *et al.* Mast: a flexible statistical framework for assessing transcriptional changes and characterizing heterogeneity in single-cell rna sequencing data. *Genome biology* **16**, 278 (2015).

45. Andrews, T. S. & Hemberg, M. Modelling dropouts allows for unbiased identification of marker genes in scrnaseq experiments. *bioRxiv* 065094 (2016).

46. Kharchenko, P. V., Silberstein, L. & Scadden, D. T. Bayesian approach to single-cell differential expression analysis. *Nature methods* **11**, 740–742 (2014).

47. Fan, J. *et al.* Characterizing transcriptional heterogeneity through pathway and gene set overdispersion analysis. *Nat Meth* **13**, 241–244 (2016). URL http://dx.doi.org/10.1038/nmeth.3734.

48. Korthauer, K. D. *et al.* A statistical approach for identifying differential distributions in single-cell rna-seq experiments. *Genome biology* **17**, 222 (2016).

49. Jiang, L., Chen, H., Pinello, L. & Yuan, G.-C. Giniclust: detecting rare cell types from single-cell gene expression data with gini index. *Genome biology* **17**, 144 (2016).

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

51. Vallejos, C. A., Marioni, J. C. & Richardson, S. Basics: Bayesian analysis of single-cell sequencing data. *PLoS computational biology* **11**, e1004333 (2015).

52. Johnson, W. E., Li, C. & Rabinovic, A. Adjusting batch effects in microarray expression data using empirical bayes methods. *Biostatistics* **8**, 118–127 (2007).

53. Benito, M. *et al.* Adjustment of systematic microarray data biases. *Bioinformatics* **20**, 105–114 (2004).

54. Leek, J. T., Johnson, W. E., Parker, H. S., Jaffe, A. E. & Storey, J. D. The sva package for removing batch effects and other unwanted variation in high-throughput experiments. *Bioinformatics* **28**, 882–883 (2012).

55. Leng, N. *et al.* Oefinder: a user interface to identify and visualize ordering effects in single-cell rna-seq data. *Bioinformatics* **32**, 1408–1410 (2016).

56. Brennecke, P. *et al.* Accounting for technical noise in single-cell rna-seq experiments. *Nature methods* **10**, 1093–1095 (2013).

57. Ilicic, T. *et al.* Classification of low quality cells from single-cell rna-seq data. *Genome biology* **17**, 29 (2016).

58. Risso, D., Ngai, J., Speed, T. P. & Dudoit, S. Normalization of rna-seq data using factor analysis of control genes or samples. *Nature biotechnology* **32**, 896–902 (2014).

59. Grün, D., Kester, L. & Van Oudenaarden, A. Validation of noise models for single-cell transcriptomics. *Nature methods* **11**, 637–640 (2014).

60. Pedregosa, F. *et al.* Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research* **12**, 2825–2830 (2011).

61. Jones, E., Oliphant, T., Peterson, P. *et al.* SciPy: Open source scientific tools for Python (2001–). URL http://www.scipy.org/. [Online; accessed ¡today¿].

62. Ruxton, G. D. The unequal variance t-test is an underused alternative to student's t-test and the mann–whitney u test. *Behavioral Ecology* **17**, 688–690 (2006).

Figures

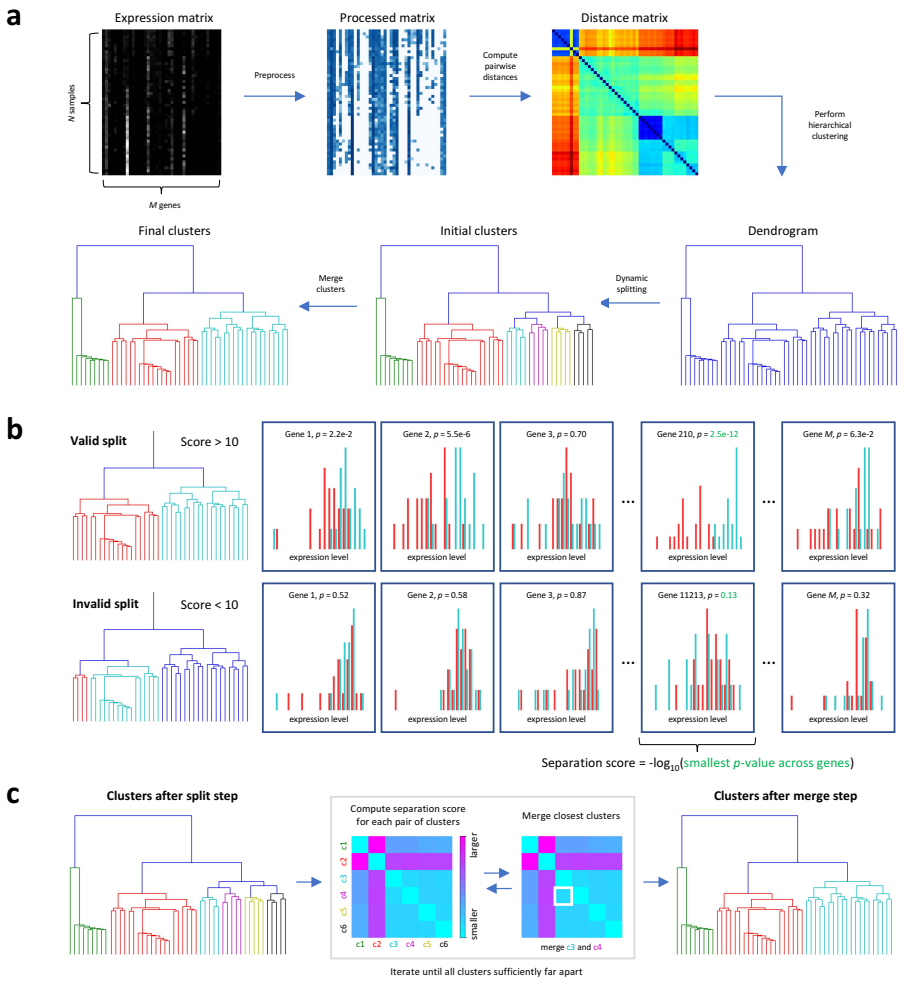

Figure 1. Overview of DendroSplit. (a) The workflow starts by preprocessing a $N \times M$ matrix of gene expressions before computing cell-cell pairwise distances, resulting in a $N \times N$ distance matrix. The distance matrix is fed into a hierarchical clustering algorithm to generate a dendrogram. The dynamic splitting step involves recursively splitting the tree into smaller subtrees corresponding to potential clusters. Finally, the subtrees are merged together during a cleanup step to produce final clusters. **(b)** A split corresponds to the partitioning of a larger cluster into two smaller clusters. A split is only deemed valid if the separation score, a metric for how well-separated two populations are, is above a predefined split threshold. Leveraging biological intuition, we rank how well each gene distinguishes the two subpopulations based on independent Welch's t-tests. We use the $-\log$ of the smallest p-value obtained as our separation score due to its interpretability and practical effectiveness. A split threshold of 10 would work for the example shown here. **(c)** During the merge step, the clusters obtained from the split step are compared to one another using pairwise separation scores. If the closest two clusters are not sufficiently far apart based on a predefined merge threshold, they are merged together and the process is repeated. When all clusters are sufficiently far apart, the algorithm terminates and the final labels are output.

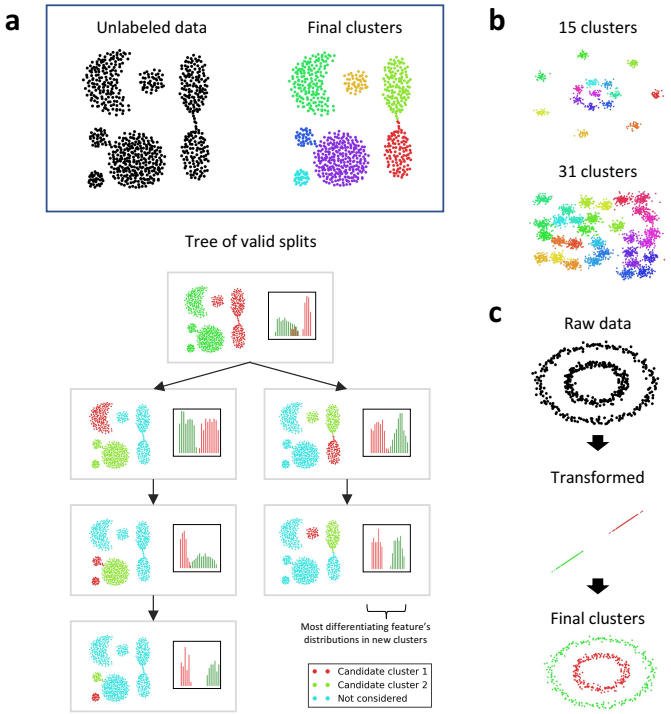

Figure 2. Synthetic datasets. (a) The DendroSplit approach is applied to a synthetic 2-dimensional dataset where pairwise distances are equal to the Euclidean distances between points. The dendrogram splitting process can be visualized using a tree, and each box in the tree represents a step in the algorithm where a larger cluster is partitioned into the red and green clusters. For each of the two features (dimensions), the split is evaluated based on the distributions of that feature within the candidate clusters. Teal points are "background" points and not considered for a given step. **(b)** DendroSplit is evaluated on two other 2-dimensional synthetic datasets and recovers the correct number of clusters both times. Euclidean distance is used. **(c)** We note that DendroSplit cannot overcome poor preprocessing and distance metric selection. Directly computing Euclidean distances for the points in the concentric circle dataset would yield poor performance, but using Euclidean distance after some preprocessing (e.g. mapping each point to its distance from the center) yields the correct results. For the examples shown here, the merge thresholds are 10, and the split thresholds are 40 for **(a)** and 30 for **(b,c)**.

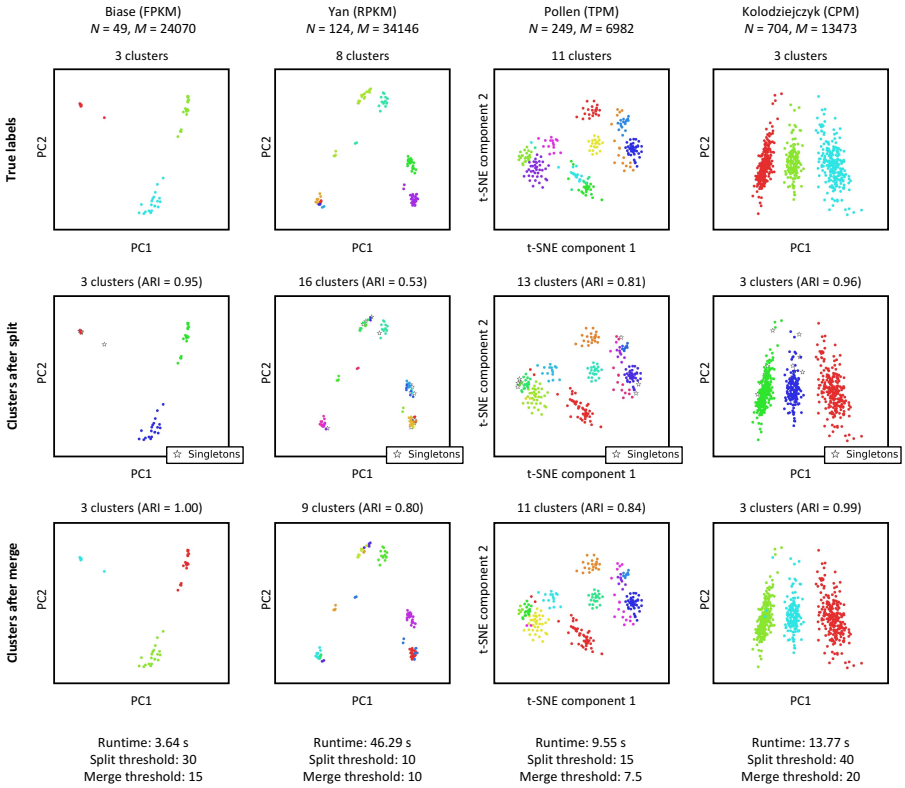

Figure 3. Gold standard datasets. DendroSplit is evaluated on four single-cell RNA-Seq datasets where the labels are highly likely to be correct[2,5,8,9,34]. In addition to visual inspection, cluster quality is evaluated using the adjusted Rand index (ARI) based on the true labels. We observe here that the split step tends to generate more clusters than expected, shrinking the ARI. Additionally, due to how the dendrogram is constructed, a cell may end up in its own cluster and is consequentially labeled as a "Singleton." The merge step treats both these cases. The cells are visualized using either the first two principal components (PC) or the first two t-distributed stochastic neighbor embedding[?] (t-SNE) components. The reported runtimes include computation of the pairwise distance matrices.

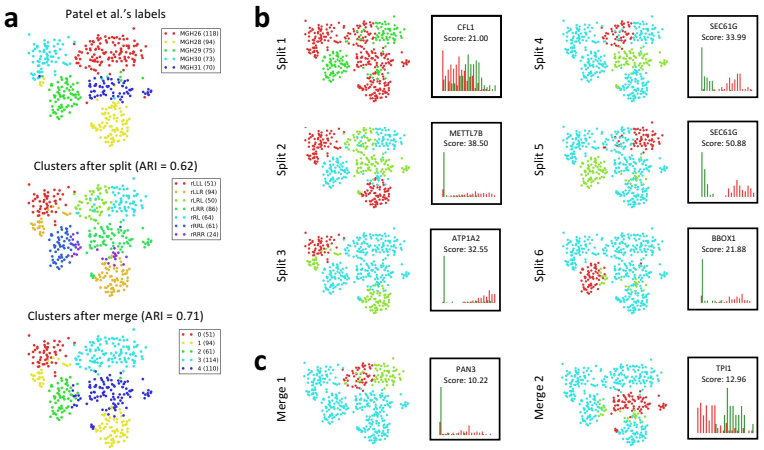

Figure 4. Exploratory analysis on Patel et al. dataset. (a) DendroSplit is evaluated on Patel et al.'s dataset of 430 cells, 5948 features (genes) from five primary human glioblastomas[7]. Gene expression is quantified using TPM. The split and merge thresholds are 20 and 15, respectively, and the analysis takes 9.64 seconds to run. The numbers in the legends represent the number of points in the corresponding clusters. For the split step, the names of the clusters are generated based on the position of the subtrees in the dendrogram. "r" represents the root node, and "rRL" represents the subtree found at the left child of the right child of the root. **(b)** We can evaluate how cells were partitioned at each step of the split procedure, and DendroSplit can also show us the within-cluster distributions of the gene that validates the split. **(c)** We can also evaluate how clusters obtained after the split step were combined during the merge procedure, and DendroSplit can show us the distributions of the most distinguishing gene between two merged clusters.

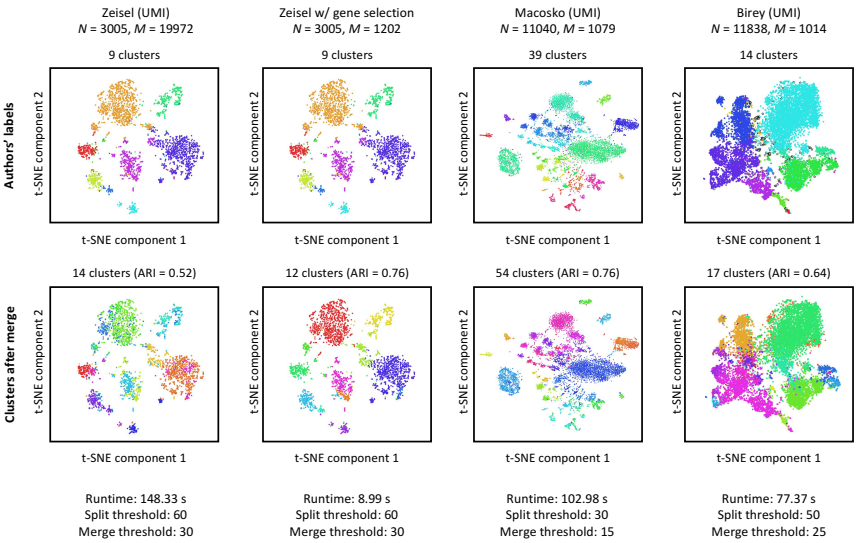

Figure 5. Larger datasets. DendroSplit is evaluated on three large single-cell datasets where the labels were assigned using computational methods[12,17,20]. Because M independent Welch's t-tests are performed at each potential split, the runtime of the algorithm scales linearly with M. As demonstrated with the Zeisel et al. dataset, decreasing M by a factor of 16.6 likewise decreases the runtime by the same factor. The datasets here are preprocessed by filtering out genes using the procedure described by[20]. This preprocessing step also improves the quality of the distance metric used, resulting in better performance.

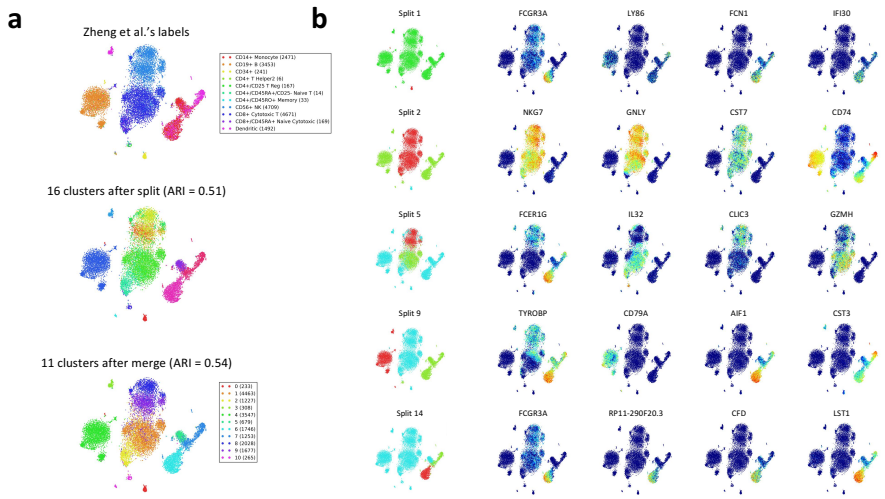

Figure 6. Exploratory analysis on PBMC dataset. (a) After gene selection and removal of cells with less than 50 counts across all genes, DendroSplit generates clusters for Zheng et al.'s remaining dataset of 17426 cells, 908 features (genes) from fresh peripheral blood mononuclear cells (PBMCs)[21]. Gene expression is quantified using UMI counts. The split and merge thresholds are 200 and 100, respectively, and the analysis takes 119.97 seconds to run. **(b)** 5 of the 15 recorded valid splits are shown along with the expression levels of the top 4 genes used for validating each split. The reported runtimes include computation of the pairwise distance matrices.

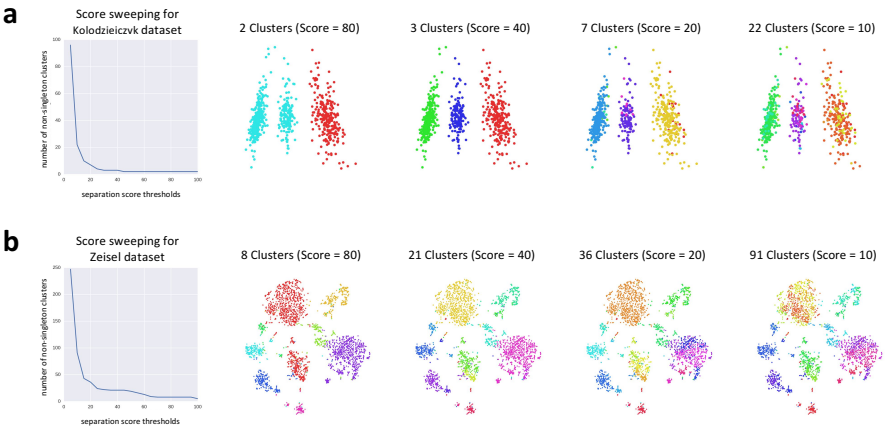

Figure 7. Split threshold sweeping. Since DendroSplit saves all relevant information at each valid split (e.g. p-values from Welch's t-tests and IDs of cells being compared), we can run the method with a small split threshold to gather information about several potential splits. From this information, we can generate the labels we would have obtained after the split step had we run the algorithm with a larger split threshold. **(a)** For the 704×13473 Kolodziejczyk et al.[5] dataset, running DendroSplit using a split threshold of 2 takes 37.63 seconds, and generating a set of new labels takes 0.032 seconds. **(b)** For the 3005×1202 Zeisel et al. dataset, running DendroSplit using a split threshold of 2 takes 13.48 seconds, and generating new labels takes 0.403 seconds. For both datasets, the DendroSplit runtimes include computation of the pairwise distance matrices.