UCSF
UC San Francisco Electronic Theses and Dissertations

Title

CHOIR improves significance-based detection of cell types and states from single-cell data

Permalink

https://escholarship.org/uc/item/87n2w8tm

Author

Sant, Cathrine

Publication Date

2024

Peer reviewed|Thesis/dissertation

eScholarship.org Powered by the California Digital Library
University of California

CHOIR improves significance-based detection of cell types and states from single-cell data

by
Cathrine Sant

DISSERTATION
Submitted in partial satisfaction of the requirements for degree of
DOCTOR OF PHILOSOPHY

in

Neuroscience

in the

GRADUATE DIVISION
of the
UNIVERSITY OF CALIFORNIA, SAN FRANCISCO

Approved:

DocuSigned by:
Saul Villeda
711626DA5837464...

Saul Villeda
Chair

DocuSigned by:
Lennart Mucke
Signed by:D2534B6...

Lennart Mucke

DocuSigned by:
M. Ryan Corces
DocuSigned by:35...6A...

M. Ryan Corces

DocuSigned by:
Hani Goodarzi
DocuSigned by:...487...

Hani Goodarzi

DocuSigned by:
Marina Sirota
925B61AB9C41499...

Marina Sirota

Committee Members

Acknowledgements

I have been so fortunate to complete the work described in this dissertation under the guidance of two incredible mentors, Lennart Mucke and Ryan Corces, who have each had a profound impact on me both as a scientist and as a person. I leave every meeting with refreshed optimism and energy because of their encouragement and insight. In addition, I would like to thank my thesis committee, Saul Villeda, Hani Goodarzi, and Marina Sirota, whose encouragement and advice has been so valuable to me. I am so grateful for the research community that I have found at UCSF and at the Gladstone Institutes. The work in this dissertation would not have been possible without the advice, help, and support of the members of the Mucke and Corces labs. I want to thank Megumi Mori for making collaboration such a joyful process, for her patience, and for her incisive perspectives. I am so grateful to her and to all those who have made my time as a graduate student such a true delight, both within and outside UCSF—Jaya Blanchard, Emily Twedell, Shreya Menon, Zachary Gardell, and Sara Knowles—as a sounding board for new ideas, an accomplice in conference small talk, or simply an open ear.

I am deeply indebted to Lea Grinberg, who first introduced me to research when I joined her lab as an undergraduate student. Lea's dedication and mentorship cemented my interest in pursuing a PhD and in continuing to study neurological disease. My early training was fostered by the immense kindness and advice of Amber Nolan and Elisa de Paula França Resende—I am so grateful for their trust and guidance. I want to particularly thank Alexander Ehrenberg, who has for many years set the standard for thoughtful scientific inquiry, impeccable work ethic, and generous friendship to which I aspire. I also want to thank the teachers who inspired my love of writing, Heather Peterson, Fehmeen Picetti, Eric Kruger, and John Florio.

My parents, Susan and Christian Petersen, have never once wavered in their support for me, and I credit them with all of the perseverance and enterprise that I possess. I am so indescribably grateful to be their daughter. My husband, Kyle Sant, is a constant, brilliant light in my life, and I am so lucky to have

felt the inestimable benefit of his optimism, his calm, and his resolve through the entirety of my time as a graduate student. Lastly, I must thank our dog, Miko, who made sure that no late evening I spent working was spent alone.

Contributions

The content contained in this dissertation is reproduced in its entirety from the following article, currently in revision:

> *Sant, C., Mucke, L. & Corces, M. R. CHOIR improves significance-based detection of cell types and states from single-cell data. bioRxiv (2024). https://doi.org/10.1101/2024.01.18.576317*

C.S., L.M., and M.R.C. conceived the project. C.S. led the design of the CHOIR software with input from L.M. and M.R.C. C.S. led the single-cell analyses presented in this paper. C.S., L.M., and M.R.C. wrote the manuscript.

CHOIR improves significance-based detection of cell types and states from single-cell data

Cathrine Sant

Abstract

Clustering is a critical step in the analysis of single-cell data, as it enables the discovery and characterization of putative cell types and states. However, most popular clustering tools do not subject clustering results to statistical inference testing, leading to risks of overclustering or underclustering data and often resulting in ineffective identification of cell types with widely differing prevalence. To address these challenges, we present CHOIR (cluster hierarchy optimization by iterative random forests), which applies a framework of random forest classifiers and permutation tests across a hierarchical clustering tree to statistically determine which clusters represent distinct populations. We demonstrate the enhanced performance of CHOIR through extensive benchmarking against 15 existing clustering methods across 230 simulated and 5 real single-cell RNA-seq, ATAC-seq, spatial transcriptomic, and multi-omic datasets. CHOIR can be applied to any single-cell data type and provides a flexible, scalable, and robust solution to the important challenge of identifying biologically relevant cell groupings within heterogeneous single-cell data.

Table of Contents

Chapter 1: Introduction 1

Chapter 2: Results 4

Overview of the CHOIR algorithm 4

Benchmarking results 7

CHOIR accurately identifies simulated cell groups 8

CHOIR avoids overclustering in simulated datasets consisting of a single population 9

CHOIR avoids underclustering in pools of distinct cell lines 10

Orthogonal validation of CHOIR using multi-modal datasets 11

Chapter 3: Discussion 16

Chapter 4: Methods 21

Details of the CHOIR algorithm 21

Random forest classifiers used in a permutation test framework 21

Generating the initial hierarchical clustering tree 22

Pruning the hierarchical clustering tree 24

Simulated data 25

Benchmarking 26

Pre-processing 26

Clustering 27

Performance Metrics 28

Data availability 30

Code availability 30

Acknowledgements 30

Contributions 30

Competing interests 30

References 31

Main Figures 37

Extended Data Figures 44

Supplementary Materials 50

List of Figures

Chapter 2:

Main Figures

Fig. 2.1: CHOIR is a hierarchical clustering algorithm that uses permutation testing for cluster identification by statistical inference 37

Fig. 2.2: CHOIR outperforms 15 existing clustering methods across 100 simulated datasets 38

Fig. 2.3: CHOIR prevents underclustering in an scRNA-seq dataset consisting of pooled cell lines 39

Fig. 2.4: Multi-omic data enables orthogonal validation of identified clusters 41

Fig. 2.5: CHOIR identifies anatomically localized clusters in a spatially resolved snRNA-seq dataset 42

Extended Data Figures

Extended Data Fig. 2.1: CHOIR effectively clusters multi-omic data and identifies cluster-specific features in both modalities 44

Extended Data Fig. 2.2: CHOIR supports clustering of atlas-scale datasets 46

Extended Data Fig. 2.3: The computational time required by CHOIR is affected by multiple factors and is on par with other methods 47

Extended Data Fig. 2.4: CHOIR cluster 28 represents a population of neurons that is localized in the developing thalamus and expresses high levels of Gbx2 transcripts 48

Supplementary Figures

Supplementary Fig. 2.1: CHOIR provides compatibility for multiple data types, formats, and analysis outputs 60

Supplementary Fig. 2.2: CHOIR is compatible with single- and multi-omic modalities 61

Supplementary Fig. 2.3: A three-tiered benchmarking approach was used to assess the performance of CHOIR relative to existing clustering methods 63

Supplementary Fig. 2.5: Detailed benchmarking results from simulated datasets with one ground truth group and 1000 cells 65

Supplementary Fig. 2.6: Detailed benchmarking results from simulated datasets with one ground truth group and 1500 cells 66

Supplementary Fig. 2.7: Detailed benchmarking results from simulated datasets with one ground truth group and 2000 cells 67

Supplementary Fig. 2.8: Detailed benchmarking results from simulated datasets with one ground truth group and 2500 cells 68

Supplementary Fig. 2.10: Detailed benchmarking results from simulated datasets with 5 ground truth groups and 10000 cells 70

Supplementary Fig. 2.11: Detailed benchmarking results from simulated datasets with 5 ground truth groups and 15000 cells 71

Supplementary Fig. 2.12: Detailed benchmarking results from simulated datasets with 5 ground truth groups and 20000 cells 72

Supplementary Fig. 2.13: Detailed benchmarking results from simulated datasets with 5 ground truth groups and 25000 cells 73

Supplementary Fig. 2.14: Detailed benchmarking results from simulated datasets with 10 ground truth groups and 5000 cells 74

Supplementary Fig. 2.15: Detailed benchmarking results from simulated datasets with 10 ground truth groups and 10000 cells 75

Supplementary Fig. 2.16: Detailed benchmarking results from simulated datasets with 10 ground truth groups and 15000 cells 76

Supplementary Fig. 2.17: Detailed benchmarking results from simulated datasets with 10 ground truth groups and 20000 cells 77

Supplementary Fig. 2.18: Detailed benchmarking results from simulated datasets with 10 ground truth groups and 25000 cells 78

Supplementary Fig. 2.19: Detailed benchmarking results from simulated datasets with 20 ground truth groups and 5000 cells 79
Supplementary Fig. 2.20: Detailed benchmarking results from simulated datasets with 20 ground truth groups and 10000 cells 80
Supplementary Fig. 2.21: Detailed benchmarking results from simulated datasets with 20 ground truth groups and 15000 cells 81
Supplementary Fig. 2.22: Detailed benchmarking results from simulated datasets with 20 ground truth groups and 20000 cells 82
Supplementary Fig. 2.23: Detailed benchmarking results from simulated datasets with 20 ground truth groups and 25000 cells 83
Supplementary Fig. 2.24: Benchmarking results from simulated datasets generated using scDesign3 based on equally subsampled PBMC populations 84
Supplementary Fig. 2.25: Benchmarking results from simulated datasets generated using scDesign3 based on unequally subsampled PBMC populations 85
Supplementary Fig. 2.26: Benchmarking results from simulated datasets with incremental downsampling of smallest ground truth group 86
Supplementary Fig. 2.27: Benchmarking results from simulated datasets with incremental reductions of the level of differential expression 88
Supplementary Fig. 2.28: Benchmarking results from simulated datasets with incremental reductions of the library size 90
Supplementary Fig. 2.29: Benchmarking results from simulated datasets with two batches of equal size 92
Supplementary Fig. 2.30: Benchmarking results from simulated datasets with two batches of unequal sizes 94
Supplementary Fig. 2.31: Benchmarking results from simulated cell differentiation trajectory datasets 96

Supplementary Fig. 2.32: Clusters identified by CHOIR do not require further rounds of subclustering 97

Supplementary Fig. 2.33: Additional evidence that CHOIR prevents underclustering in an scRNA-seq dataset consisting of pooled cell lines 98

Supplementary Fig. 2.34: Comparison of results obtained with different clustering methods on the Hao et al. 2021 CITE-seq dataset 99

Supplementary Fig. 2.35: CHOIR identifies three functionally distinct cell types of the embryonic heart 100

Supplementary Fig. 2.36: CHOIR cluster 36 represents the pool of medial ganglionic eminence interneuron progenitors characterized by high expression of Lhx6 101

Supplementary Fig. 2.37: CHOIR cluster 31 represents a neuronal progenitor pool of the spatially restricted E14 dorsal pallium characterized by high expression of Dbx1 and Neurog1 103

Supplementary Fig. 2.38: CHOIR cluster 39 represents a population of neurons localized within the developing cerebellum characterized by high expression of Fgf3 105

Supplementary Fig. 2.39: Non-default parameters for GiniClust3 and Seurat that identified certain anatomically restricted cell types resulted in severe overclustering of other cell types 107

Supplementary Fig. 2.40: Consensus clustering using scTriangulate does not improve overall clustering results 108

List of Notes and Tables

Chapter 2:

Supplementary Notes

Supplementary Note 2.1: How to use CHOIR on atlas-scale data 50

Supplementary Note 2.2: Description of each parameter used by CHOIR 52

Supplementary Tables

Supplementary Tables 2.1–2.7 .... https://files.corces.gladstone.org/Publications/2025_Sant_CHOIR

Source Data 2.1 .... https://files.corces.gladstone.org/Publications/2025_Sant_CHOIR

Chapter 1

Introduction

Cellular heterogeneity is a fundamental characteristic of complex biological systems, which are functionally dependent on the concerted action of many distinct, specialized cell populations. Single-cell technologies make this cellular heterogeneity accessible, enabling the recognition and characterization of biologically distinct cell types and states. Therefore, partitioning cells into empirical groups (or clusters) representing biologically distinguishable cell types or cell states is an important objective in the analysis of single-cell data. Whether the clusters obtained from such analyses are reliable proxies for biologically distinct cell populations materially affects the ability of subsequent analysis steps, such as differential expression analyses, to yield meaningful biological insights. Underclustering, in which multiple distinct populations are grouped into the same cluster, may obscure subtle but potentially impactful differences between cell subtypes or cell states, whereas overclustering, in which cluster divisions are driven by noise rather than true biological signal, may lead to the misguided investigation of spurious new cell types or states.

A variety of clustering approaches have been developed for use on single-cell data. However, in many biological systems studied with single-cell approaches, the prevalence of specific putative cell types can differ by orders of magnitude[1]. This wide range presents a challenge for many clustering methods, including Louvain[2] and Leiden[3] clustering, which often perform best when cluster sizes are relatively similar[4]. Although some tools have been specifically developed to identify rare cell types[5,6], the simultaneous and reliable identification of both rare and abundant cell populations remains a major challenge.

In practice, clustering of single-cell data has become a very manual process that is impacted by subjective decisions based, in part, on biological intuition, which varies from user to user. More

specifically, the granularity of clustering results is often greatly influenced by parameter values selected by the user, such as the "resolution" parameter in Louvain clustering, which, in effect, determines the number of clusters and enables the algorithm to find smaller clusters of tightly connected nodes at higher resolution values. Users often manually assess multiple clustering results based on toggling of these parameters before selecting one that matches their intuition. Moreover, it has become commonplace to use this initial clustering result as a starting point for manual curation by merging or dividing ("subclustering") certain selected clusters. However, gauging the extent to which clusters should be merged or subdivided is difficult, as many clustering methods will partition data even if they are derived from a single cell population and no biologically meaningful groups are present[4]. In the context of this subjective approach, limited safeguards exist against overclustering or underclustering of the data. Because the manual assessment of clustering results is neither scalable nor reproducible, methods have been developed to circumvent this process. Methods to estimate the number of "true" clusters in the data, as well as cluster stability methods[7,8], can help assess the reproducibility of clusters across subsamples or parameters, but often overestimate the number of clusters present[9] and do not account for statistical uncertainty. Similarly, reference-based clustering methods have been developed to automate the annotation of new datasets, leveraging the cell type markers identified in previous datasets. However, these supervised approaches are limited to—and inherently biased by—the annotations used as input, which are largely based on the same flawed principle of leveraging biological intuition to determine proper clustering. Additionally, these reference-based approaches only represent the cell states present in the reference datasets and, thus, cannot represent the full spectrum of uncharted cell states that might exist across model systems or occur as the result of perturbations or disease. Because of these limitations, such methods may fail to identify novel cell types or states.

Another potential limitation of existing single-cell clustering methods is that most have been optimized specifically for single-cell RNA sequencing (scRNA-seq) data. However, the rise of additional single-cell data types, including multi-omic approaches, enables concerted insights into cellular phenotypes across multiple modalities, such as mRNA, chromatin accessibility, and surface protein

abundance. To fully leverage this expanding landscape of technologies, computational tools must be adaptable across multiple data types and be able to perform joint multi-modal analyses.

Thus, there is a need for single-cell clustering methods that can reliably detect rare as well as common cell types across single- and multi-omic platforms, while neither overclustering nor underclustering the data. In addition, the ideal clustering method would reduce the chance of spurious discoveries by incorporating statistical inference and minimizing the reliance on user-supplied parameters with substantial effects on clustering output. Imposing such algorithmic guardrails could improve one's confidence in the biological relevance of single-cell results and save time and effort that would otherwise be spent applying intuition to manually curate clustering results. To address these needs, we have developed CHOIR (cluster hierarchy optimization by iterative random forests), a user-friendly R package for clustering single-cell data of any modality. CHOIR integrates seamlessly with popular single-cell sequencing software packages such as Seurat[10], SingleCellExperiment[11], ArchR[12], and Signac[13]. It uses a hierarchical permutation test approach based on random forest classifier predictions to identify clusters representing distinct cell types or states with increased levels of accuracy and ease, compared to a broad range of available clustering algorithms and to ground truths established or estimated by other methodologies.

Chapter 2

Results

Overview of the CHOIR algorithm

CHOIR is based on the premise that, if clusters contain biologically different cell types or states, a classifier that considers features present in cells from each cluster should be able to distinguish the clusters with a higher level of accuracy than classifiers trained on randomly permuted cluster labels (**Fig. 2.1**). The use of a permutation testing approach allows CHOIR to introduce statistical significance thresholds into the clustering process. Operating as a self-contained clustering algorithm, CHOIR iteratively applies this permutation test approach across the nodes of a hierarchical clustering tree to determine the appropriate extent of each branch of the clustering tree such that the final clusters represent statistically distinct populations across a range of cluster sizes. Random forest classifiers are used because of their flexible applicability to data derived from various distributions. Consequently, CHOIR is applicable to single-cell datasets across diverse technologies, including RNA-seq, ATAC-seq, and proteomics.

CHOIR takes as input a Seurat[10], SingleCellExperiment[11], or ArchR[12] object containing one or more feature matrices (**Supplementary Fig. 2.1**). Input data should already have undergone quality control and normalization steps. CHOIR proceeds in two main steps (**Fig. 2.1**). First, a hierarchical clustering tree is generated spanning the network structure of the data using a "top-down" approach that proceeds from an initial "partition," in which all cells are in the same cluster, to a partition in which all cells are demonstrably overclustered. To generate this hierarchical clustering tree, the data is iteratively clustered using modularity-based algorithms[2,3] with increasing resolution parameters. The silhouette score, which measures the similarity of cells to their own clusters compared to other clusters[14], is assessed at each new level of the emerging tree. When the silhouette score is maximized, the cells from each "parent cluster" at this level are

subsetted for re-analysis. For each parent cluster subset, a new set of highly variable features is identified, a new dimensionality reduction is generated, and these cells are then subclustered to generate a hierarchical clustering subtree. For example, when applied to a multi-omic dataset with both RNA-seq and ATAC-seq features from 51,409 human retina cells[15], CHOIR identified three parent clusters (**Extended Data Fig. 2.1, Supplementary Table 2.1–2.2**). Each of the resulting subtrees continued to be subdivided until the farthest pair of nearest neighboring clusters was found to be overclustered by the permutation test approach, as explained further below.

Second, this overclustered hierarchical clustering tree is "pruned" using a bottom-up approach to identify a final set of clusters. Starting at the bottom level of the clustering tree, CHOIR identifies sets of clusters that belong to the same composite cluster at the next level of the clustering tree. For each such cluster pair, balanced random forest classifiers are applied to learn the cluster labels and permuted cluster labels. For these permutation test comparisons, the null hypothesis is that the cells belonging to each of the two clusters are no more distinguishable than a chance separation of the cells into two random groups. If this null hypothesis is rejected, the clusters will retain separate identities. If the null hypothesis is not rejected, the clusters will be merged and assume a single identity (**Fig. 2.1**). This process is repeated up the branches of the hierarchical clustering tree until a final cluster solution is reached, wherein each cluster can be accurately distinguished from all other clusters (**Fig. 2.1**). The combined top-down tree construction and bottom-up branch pruning approach used by CHOIR ensures that the most similar groups of cells are compared first, reducing the chance of conflating noise with biological signal. In contrast, approaches that only assess groups of cells from a top-down perspective—pervasive in existing tools—risk masking signal from heterogeneous groups of cells, which can lead to inaccurate, underclustered results. Similarly, approaches that only assess groups of cells from a bottom-up perspective require the user to pre-determine where the bottom of the tree is, introducing the potential for underclustering. Relatedly, many existing clustering methods use the same matrix to both generate clusters and evaluate the fidelity of the resulting clusters, a statistical phenomenon known as "double-dipping"[16]. To address this issue, CHOIR includes an option to implement count splitting, such that independent matrices are used to generate and evaluate

CHOIR clusters (**Supplementary Fig. 2.2**). However, unless stated otherwise, the analyses presented in this study did not leverage this count splitting approach. As none of the other methods tested implement count splitting, comparing CHOIR with count splitting to other methods without count splitting would involve comparisons based on different input matrices, which would not be valid. Lastly, as mentioned above, nearly all existing clustering methods require users to extensively tweak parameter settings in order to obtain optimal results. In contrast, the primary results we present here across a diversity of biological contexts, data modalities, and dataset sizes were derived exclusively using the default parameters of CHOIR.

When clusters are identified in single-cell data, those clusters are typically interpreted as potentially useful cell groupings for subsequent analyses. Because the relatedness of clusters could also provide informative metrics, CHOIR preserves a record of all of the pairwise comparisons conducted before reaching the final set of clusters. This information can then be used to demonstrate the degree of relatedness of clusters (**Extended Data Fig. 2.1**). In addition, CHOIR records feature importances found by the random forest classifier for each comparison, which represent the gene products or other features that best differentiate each cluster and can be used to aid cluster annotation. For instance, in the multi-omic human retina dataset[15], the feature importances measured in the comparison of CHOIR cluster 1, which represents rod cells, and CHOIR cluster 4, which represents cone cells, revealed features with differential gene expression and chromatin accessibility between these cell types (**Extended Data Fig. 2.1**).

The compatibility of CHOIR across single- and multi-omic modalities allows for such joint identification of important RNA and ATAC features and enables the comparison of clustering results across modalities. Although applying CHOIR to different modalities yielded different sets of final clusters, the majority of the cell types identified by CHOIR remained unchanged (**Supplementary Fig. 2.2, Supplementary Table 2.2**). Comparison to existing methods for clustering snATAC-seq or multi-omic data showed that CHOIR successfully prevented the overclustering of common homogeneous cell types such as rod cells based on arbitrary patterns of noise that was observed with the Louvain clustering implemented by ArchR, Signac, and Seurat (**Supplementary Fig. 2.2, Supplementary Table 2.2**).

CHOIR is also applicable to atlas-scale datasets with millions of cells through parallelization. For such large datasets, CHOIR can be applied in a three-step procedure to maximize computational efficiency (detailed in **Supplementary Note 2.1**). To illustrate this capacity, we applied CHOIR to an snRNA-seq dataset from human brain comprising 3,293,958 nuclei[17] (**Supplementary Table 2.1, 2.3**). Briefly, we first identified parent clusters (**Extended Data Fig. 2.2**), which were then analyzed in parallel to identify the final set of CHOIR clusters. These parallel runs were then merged to identify 1,302 clusters (**Extended Data Fig. 2.2**). The clusters identified by CHOIR represent biologically meaningful cell groupings (**Extended Data Fig. 2.2**) and largely aligned with the 3,313 published subclusters or groups of those subclusters (**Extended Data Fig. 2.2**). As expected, the clusters identified by CHOIR were more conservative than the published subclusters, which were identified based on the minimum criterion of a single statistically enriched gene.

Benchmarking results

Evaluating the performance of single-cell clustering algorithms presents challenges due to the lack of genuine "ground truth" information for most real single-cell sequencing datasets. Although simulated data provide a known ground truth and allow precise control of size and complexity, they cannot reproduce all aspects of real single-cell datasets. Furthermore, using published cell type annotations as a stand-in for ground truth cell identities risks biasing benchmarking results in favor of methods similar to those used in the original analysis. We therefore used complementary strategies in a three-tiered benchmarking approach to comprehensively compare the performance of CHOIR with that of 15 other unsupervised clustering methods[5,6,9,10,18-28] (**Supplementary Table 2.4**) in the analysis of both simulated and real data (**Supplementary Fig. 2.3** and **Supplementary Table 2.1**). We benchmarked the performance of each method across an array of parameter combinations to assess both the best possible performance, as obtained from the single parameter combination that yielded most robust results, and the default performance, as obtained by running the default parameters (see **Supplementary Note 2.2** for a description of the parameters of CHOIR and how they impact performance). Overall, our results demonstrate that CHOIR

outperforms available clustering methods across a range of species, cell numbers, tissue types, and sequencing modalities.

CHOIR accurately identifies simulated cell groups

Using the R package Splatter[29], we generated 100 simulated datasets representing 1, 5, 10, or 20 distinct ground-truth cell populations, ranging from 500 to 25,000 cells (**Fig. 2a**). When assessing performance across these simulated datasets, CHOIR performed as well as or better than the 15 clustering methods used for comparison, as measured by the Adjusted Rand Index (ARI)[30] and deviation from the ground-truth number of clusters (**Fig. 2.2, Supplementary Figs. 2.4–2.23, Source Data 2.1**). To ensure that these results were not dependent on the software used for data simulation, we also generated 10 datasets with the simulation method scDesign3[31] from real subsampled PBMC cell populations[32,33] (**Supplementary Figs. 2.24–2.25, Source Data 2.1**). On these datasets, CHOIR again performed as well as or better than all other methods tested.

To assess how various aspects of snRNA-seq datasets affect CHOIR's performance, we used five of the simulated datasets produced with Splatter as the baseline to generate 105 additional simulated datasets in which we incrementally reduced the prevalence of rare cell populations, the degree of differential expression, or the library size. Across these datasets, we assessed the point of failure at which each parameter combination tested for CHOIR and the other 15 clustering methods no longer accurately distinguished the ground truth groups (**Supplementary Figs. 2.26–2.28, Source Data 2.1**). Additionally, we generated 10 simulated datasets with multiple batches, with either balanced or imbalanced batch sizes (**Supplementary Figs. 2.29–2.30, Source Data 2.1**). For all of these tests, CHOIR performed as well as or better than all of the other methods, except sc-SHC, which is particularly suited for clustering these simulated datasets because both Splatter and sc-SHC use parametric distributions to model different cell populations.

The methodology used by Splatter allows for the generation of concrete ground truth labels based on different underlying distributions. However, the resulting simulated data does not fully capture the

continuous nature of some biological contexts. To bridge the gap between simulated and real data, we generated 5 datasets using Splatter's simulation of cell differentiation trajectories (**Supplementary Fig. 2.31, Source Data 2.1**). All of the tools tested proved ineffective in clustering these simulated trajectory datasets, highlighting an inherent incompatibility between the categorical assignments of clustering methods and the spectra of continuous differences.

Across the range of size and complexity represented by these simulated datasets, the computational time required by CHOIR scaled approximately linearly with the number of cells in each dataset (**Extended Data Fig. 2.3, Source Data 2.1**). The computational time required by CHOIR is also directly tied to the number of iterations run for each permutation test; the default value of 100 iterations was selected in order to balance statistical power with efficiency (**Extended Data Fig. 2.3, Source Data 2.1**). In general, the computational time required by CHOIR is governed primarily by the size and complexity of the dataset (**Extended Data Fig. 2.3**).

CHOIR avoids overclustering in simulated datasets consisting of a single population

Overclustering poses the risk of false discoveries of spurious new cell types or states. Within the set of 25 simulated scRNA-seq datasets representing a single ground truth cell population, CHOIR was one of only three methods (along with SC3 and sc-SHC) that consistently avoided overclustering (**Fig. 2.2, Supplementary Figs. 2.4–2.8, Source Data 2.1**). Indeed, several of the clustering methods tested (CIDR, scCAN, SHARP, SIMLR, and Spectrum) assume a minimum of two clusters and never returned a single cluster. Even methods that permit the identification of a single cluster often misinterpret noise as structure, leading to overclustering. This weakness limits the utility of these methods for analyses where the presence or absence of discrete cell populations may be unknown, such as in subclustering analyses, analyses of cell lines, or analyses of homogeneous cell populations obtained by fluorescence-activated cell sorting (FACS). Although SC3 and sc-SHC consistently prevented overclustering across the simulated datasets consisting of a single ground truth population, these methods resulted in instances of underclustering across the more complex simulated datasets, in contrast to CHOIR (**Fig. 2.2, Supplementary Figs. 2.9, 2.14, 2.19–2.21,**

2.23, Source Data 2.1). The five existing methods with the best performance across these simulated datasets, Cytocipher, GiniClust3, SCCAF, sc-SHC, and Seurat, were used for comparison in the subsequent benchmarking assessments of CHOIR on real datasets. It is worth noting that many of the methods that performed poorly on the simulated datasets also required prohibitively long run times to benchmark on larger real datasets.

We also examined the performance of CHOIR by simulating the commonly used workflow in which clusters are manually selected for additional rounds of subclustering. Only CHOIR and sc-SHC consistently prevented overclustering when clusters identified in an initial round of clustering were subsetted and reclustered (**Supplementary Fig. 2.32, Source Data 2.1**). Thus, the clusters identified by CHOIR can be considered "terminal" clusters that do not require further rounds of subclustering.

CHOIR avoids underclustering in pools of distinct cell lines

To assess whether CHOIR also guards against underclustering, we evaluated the performance of CHOIR on a publicly available scRNA-seq dataset composed of pooled cancer cell lines[34] (**Supplementary Table 2.1**). Although there might be distinct cell types or states within each cell line, distinct cell lines can be assumed to represent the minimal distinct identities in such a dataset. Therefore, multiple cell lines should never be grouped together into a single cluster. Across all parameter settings tested for the six clustering methods, only CHOIR was able to distinguish all 190 cell lines (**Fig. 2.3, Supplementary Fig. 2.33, Supplementary Table 2.5**). CHOIR consistently prevented underclustering cells from distinct cell lines, as measured by the entropy of cluster accuracy (**Fig. 2.3,** see **Methods**). All parameter settings, including the defaults, of Cytocipher, GiniClust3, SCCAF, sc-SHC, and Seurat resulted in the inclusion of multiple cell lines in a single cluster (**Fig. 2.3, Supplementary Fig. 2.33**), providing clear examples of underclustering.

As one of the primary goals of CHOIR is the identification of rare cell populations, we conducted an additional analysis in which we separately downsampled each cell line to just 50 cells and evaluated whether, at this extremely low prevalence (around 0.1% of the total number of cells), the six clustering

methods could distinguish the downsampled cell line from other cell lines. In this assessment, CHOIR performed markedly better than all other methods tested (**Fig. 2.3, Supplementary Table 2.5**).

Within some cell lines, CHOIR distinguished multiple clusters, raising the question of potential overclustering. Because divergent proliferation states often emerge in cultured cancer cell lines, we calculated a proliferation module score for each cluster identified by CHOIR. In 74.3% of the cell lines in which CHOIR identified multiple clusters, those clusters had significantly different proliferation module scores (**Fig. 2.3**). For two such cell lines (A375 melanoma cells and T47D breast cancer cells), we additionally projected the clusters identified by CHOIR onto larger, independent datasets from the same lines[35,36] (**Supplementary Table 2.1**) to further confirm whether these clusters represent biologically distinct cell populations. In both cases, the nearest neighbors of the projected cells of each CHOIR cluster in the independent datasets also significantly differed in the proliferation module score (**Fig. 2.3**). For the remaining minority of cell lines in which no significant differences were found in the proliferation module score, we suspected that the clusters identified by CHOIR reflected other changes in cell states. To illustrate this, we examined the three subclusters identified within the CCF-STTG1 human astrocytoma cell line. One cluster expressed high levels of pluripotency marker genes such as *SOX2*[37], whereas another cluster resembled muscle cells, expressing high levels of *TPM2*[38] (**Fig. 2.3**). A third cluster within this line did not express high levels of either *SOX2* or *TPM2* and may reflect differentiation along a different path, as evidenced by expression of *IL1RAPL1*, a membrane protein expressed in neurons[39] (**Fig. 2.3**).These results confirm that the subclusters identified by CHOIR within a single cell line indeed represent distinct cell states and also highlight the ability of CHOIR to identify such states from samples comprising relatively few cells in total.

Orthogonal validation of CHOIR using multi-modal datasets

Multi-modal datasets provide a useful opportunity to leverage orthogonal data from the same cells to assess whether clusters are biologically distinct. For this purpose, we performed clustering on one modality and assessed the resulting clusters for differential features in the other modality. We collected two

publicly available datasets with different types of multi-modal data (**Supplementary Table 2.1**): (1) CITE-seq data from human PBMCs encompassing both scRNA-seq and antibody-based surface marker proteomics[40] and (2) sci-Space data from developing mouse embryos comprising snRNA-seq data and spatial coordinates[41]. For each dataset, we benchmarked the performance of CHOIR against the five best-performing methods prioritized above (Cytocipher, GiniClust3, SCCAF, sc-SHC, and Seurat).

In the multi-omic CITE-seq dataset containing snRNA-seq and surface marker protein data from the same 159,827 human PBMCs[40], each clustering method was applied only to the RNA modality. The surface marker protein data, comprising 228 oligonucleotide-conjugated antibodies, was then used to assess the minimum number of differentially expressed proteins among the 50 nearest cluster pairs identified by each of the clustering algorithms. We then interpreted observation of a pair of clusters with no differentially expressed proteins between them to be indicative of overclustering, as the ideal method would identify the highest number of clusters while still maintaining some differentially expressed proteins between every cluster pair. However, we note that the total number of features used for clustering from the RNA modality was much larger than the total number of features used for differential expression analysis from the protein modality. We therefore refer to the RNA-based distinction of clusters that did not differ by protein detection as "likely overclustering." Of the six clustering methods tested, only CHOIR maximized the number of clusters without likely overclustering the data, as evidenced by the fact that each cluster differed from its neighboring clusters in the expression of multiple proteins (**Fig. 2.4, Supplementary Fig. 2.34, Supplementary Table 2.6**).

Although all parameter settings tested for Cytocipher avoided likely overclustering, as indicated by differentially expressed proteins between the clusters, all of these parameter settings resulted in fewer clusters than those obtained with CHOIR, indicating possible underclustering by Cytocipher. Supporting this notion, CHOIR identified a population of FOXP3-expressing regulatory T cells (Tregs), a subset of T lymphocytes involved in suppressing immune responses[42] (**Fig. 2.4, Supplementary Fig. 2.34, Supplementary Table 2.6**); a population of *CCR7*-expressing naïve T cells, a subset of T lymphocytes that are antigen-inexperienced[43] (**Fig. 2.4, Supplementary Fig. 2.34, Supplementary Table 2.6**); and a

population of *CLEC9A*-expressing conventional type 1 dendritic cells, a small subset of antigen-presenting cells that are key to generating cytotoxic T cell responses[44] (**Fig. 2.4, Supplementary Fig. 2.34, Supplementary Table 2.6**). None of these populations were identified by any of the parameter settings tested in Cytocipher. Although some parameter settings for GiniClust3, SCCAF, and Seurat yielded more clusters than CHOIR, in all such cases pairwise comparisons of the clusters revealed no differentially expressed proteins, implying likely overclustering (**Fig. 2.4, Supplementary Table 2.6**). No parameter settings tested for sc-SHC identified as many clusters as were identified by CHOIR, and the comparison of some clusters obtained with sc-SHC showed no differentially expressed proteins, implying both under- and overclustering. Furthermore, although Harmony batch correction was applied during CHOIR, Cytocipher, SCCAF, and Seurat clustering, only CHOIR and Cytocipher avoided the spurious segregation of clusters of the same cell type based on batch effects, whereas GiniClust3, Seurat and SCCAF were susceptible to this confound (**Supplementary Fig. 2.34**).

In the sci-Space dataset[41], which consisted of sagittal sections of whole E14 mouse embryos, the clustering methods were again applied only to the RNA modality, without reference to the spatial coordinate information. Using the default parameter settings of CHOIR, 42 clusters were identified (**Fig. 2.5, Supplementary Table 2.7**). Given that the E14 developmental period is characterized by the formation of many organ systems, we would expect to identify both anatomically dispersed cell types (e.g., connective tissue cells) and a number of spatially constrained organ-specific clusters (e.g., heart, lung, or liver cells) (**Fig. 2.5**). Indeed, some cell types, for example, keratinized epithelial cells, were found to be distributed throughout the body (**Fig. 2.5**), whereas other cell types were found to be restricted to circumscribed anatomical locations, for example, hepatocytes in the liver (**Fig. 2.5**) and mesenchymal cells in the lung (**Fig. 2.5**). We used such differences in distribution as an orthogonal validation of the biological relevance of clusters identified. For each tissue section, we evaluated the percentage of all clusters that had a highly localized spatial distribution, defined as a mean radial area that was less than 5% of the approximate whole embryo section area. Across all parameter settings tested for each method, CHOIR obtained the highest mean percentage of spatially localized clusters (**Fig. 2.5**). The spatial restriction of these clusters to

anatomically defined regions suggests that these clusters represent biologically distinct cell populations, a conclusion that was further validated by the expression of known developmental marker genes, as described below.

CHOIR identified several clusters with highly specific anatomical localization that were missed by other clustering methods. For example, CHOIR was the only method that distinguished three distinct cell types of the heart: cardiac interstitial cells, cardiomyocytes, and heart endothelial cells (**Fig. 2.5, Supplementary Fig. 2.35**). The cardiac interstitial cell cluster, the cardiomyocyte cluster, and the heart endothelial cell cluster identified by CHOIR were all anatomically localized to the heart across all sections (**Fig. 2.5**). However, these cell clusters were appropriately differentiated based on the expression of marker genes for connective tissue (*Col1a1*, *Col14a1*[45]), endothelial cells (*Cdh5*, *Nos3*, *Pecam1*, *Vwf*[46,47]), or cardiomyocytes (*Myh6*, *Myh7*[48]), respectively, and were clearly distinct when viewed in the refined dimensionality reduction of the corresponding parent cluster (**Fig. 2.5, Supplementary Fig. 2.35**).

CHOIR was also the only method whose default parameters distinguished certain anatomically restricted neuronal subpopulations. In total, CHOIR identified 22 nervous system clusters, ranging in size from 37 to 6,875 cells (**Fig. 2.5**). Of these, cluster 28 represents a population of neurons that are localized within the developing thalamus and are characterized by high *Gbx2* expression[49] (**Fig. 2.5, Extended Data Fig. 2.4**). Cluster 36 represents the pool of medial ganglionic eminence interneuron progenitors characterized by high expression of *Lhx6*[50] (**Fig. 2.5, Supplementary Fig. 2.36**). Additionally, CHOIR identified a neuronal progenitor pool of the spatially restricted E14 dorsal pallium characterized by expression of *Dbx1* and *Neurog1*[51-53] (**Supplementary Fig. 2.37**), and a population of cerebellar neurons with characteristic *Fgf3* expression[54] (**Supplementary Fig. 2.38**).

Across all parameter settings tested for each method, CHOIR was the only method that identified all of the above-mentioned heart and brain cell types, in addition to many other such examples (**Supplementary Figs. 2.35–2.38, Supplementary Table 2.7**). Some non-default parameters for GiniClust3 and Seurat also identified several of these cell types; however, those parameter settings resulted in severe overclustering of other cell types (**Supplementary Fig. 2.39**). Consensus clustering using

scTriangulate[55] (**Supplementary Fig. 2.40**) did not improve upon CHOIR's overall performance, as measured by the mean percentage of spatially localized clusters (**Supplementary Fig. 2.40**). While consensus clustering did occasionally enable identification of cell groups not identified by CHOIR, there were many more examples in which consensus clustering resulted in obfuscation of cell groups that CHOIR identified and other tools missed (**Supplementary Fig. 2.40**).

Thus, using a tiered benchmarking approach that leveraged both simulated and real datasets, we found that CHOIR accurately identified ground truth populations, successfully avoiding overclustering of simulated datasets composed of a single ground truth population as well as underclustering of pooled cancer cell lines. Using multi-omic datasets, we further demonstrated that the clusters identified by CHOIR are clearly distinct in an orthogonal modality, indicating that CHOIR is able to detect biologically relevant clusters that are difficult or impossible to reliably identify by other clustering methods.

Chapter 3

Discussion

Defining reliable clusters that represent distinct cell types and states is critical to gaining biological insights from single-cell analyses. However, the subjective nature of commonly used approaches to clustering does not sufficiently protect against overclustering or underclustering of the data, which can lead to misleading conclusions about fundamental and disease-relevant biological processes. In addition, many existing clustering approaches preferentially identify clusters of similar sizes, often resulting in overclustering of common homogeneous cell types and underclustering of rare cell populations. CHOIR addresses these challenges by incorporating statistical inference into the clustering process, reducing the chance of spurious discoveries. By tying cluster identification to a significance threshold, CHOIR ensures that the clusters identified all meet a standardized criterion. Common homogeneous cell populations are not spuriously split due to noise, and rare populations are identified as distinct even when the corresponding number of cells is low.

The primary tradeoff when using a clustering tool such as CHOIR is the increased processing time required. For large datasets, CHOIR could be orders of magnitude slower than a single iteration of standard Louvain or Leiden clustering. However, although CHOIR has a longer run time than standard Louvain or Leiden clustering, it saves many hours of manual optimization of cluster definition and only needs to be performed once per dataset, as the clusters identified by CHOIR do not require further rounds of subclustering. Moreover, the default parameters of CHOIR perform very well across an array of data types and biological contexts, eliminating the need for user input or the testing of many parameter combinations. Indeed, the default parameter settings of CHOIR were always among the best performing parameter combinations tested. To further enable the annotation of called clusters, CHOIR defines marker features for

each cluster in the course of the hierarchical clustering procedure. We provide guidelines for using CHOIR to analyze datasets greater than 500,000 cells in **Supplementary Note 2.1**.

Existing hierarchical clustering approaches exclusively utilize either a top-down or bottom-up approach. Top-down approaches start with all cells in a single group and proceed to attempt to iteratively divide that group until further division is deemed unnecessary. While this is intuitive, issues may occur when the clusters being compared at a certain level, often near the top of the tree, are composed of multiple highly distinct cell groups, because this comparison may fail statistical testing due to the inherent variability within each cluster. If subdivided further, multiple distinct cell groupings would be identified, thereby making top-down approaches susceptible to underclustering. Bottom-up approaches avoid this pitfall by ensuring that the most similar groups of cells are compared first, reducing the chance of conflating noise with heterogeneous biological signal. However, it is challenging to know what the true "bottom" of the tree is without first constructing the tree from the top down. CHOIR leverages the strengths of both approaches and is implemented in two main steps: a top-down construction of a hierarchical clustering tree, followed by a bottom-up pruning of the branches of that clustering tree. These two steps are used to prevent underclustering and overclustering, respectively. To our knowledge, CHOIR is the only available method to take this two-step approach.

Although Cytocipher and SCCAF also implement a bottom-up approach to prevent instances of overclustering, they rely on the user to determine the input Louvain or Leiden resolution, subjectively setting a threshold at which the dataset is deemed to be overclustered. Furthermore, these methods do not assess whether underclustering has occurred and are, thus, vulnerable to it. In contrast, sc-SHC takes only a top-down approach. This method obviates the need for the user to determine the input clusters but is still susceptible to underclustering due to the top-down approach itself.

The two-step approach used by CHOIR also makes it compatible with count splitting[16], wherein two independent matrices are generated, one for the creation of the clustering tree and another to assess and prune the branches. Using CHOIR with count splitting can provide an additional guard against overclustering by avoiding any "double-dipping" into the same data. However, count splitting inherently

makes clustering more conservative, which may lead to instances of underclustering, particularly in datasets with low read depths.

In addition to combining top-down and bottom-up steps, CHOIR implements an additional strategy to prevent underclustering: as the initial hierarchical clustering tree is constructed, new dimensionality reductions and sets of highly variable features are generated for a preliminary set of high-level clusters that typically represent the major cell classes present in the dataset. For random forest classifier comparisons applied to clusters that stem from the same parent cluster, the subsetted dimensionality reduction and feature set are used. This allows CHOIR to identify more nuanced differences between cell subtypes or states that are not easily detected using existing clustering methods, as highlighted by the identification of proliferation states within cancer cell lines by CHOIR (**Fig. 2.3**), as well as the identification of functionally distinct cell types in the heart (**Fig. 2.5**) and cell types specific to different brain regions (**Fig. 2.5**) within the challenging, more continuous data structure observed in the dataset from developing mouse embryos.

To achieve accurate clustering results, these nuanced differences between cell subtypes or states must not be conflated with batch-driven noise—a large potential source of overclustering. Other methods such as Cytocipher and SCCAF purport to prevent overclustering, but neither algorithm has batch correction checks built into the clustering method itself. Therefore, although the initial clusters may be determined using a batch-corrected dimensionality reduction, the cell-by-gene count matrix itself would have to be batch-corrected to prevent batch effects from biasing the ultimate cluster labels that pass the significance and accuracy thresholds imposed, and this is not routinely done in single-cell analyses. CHOIR offers an effective solution to this challenge. For datasets that require batch correction, CHOIR uses an internal Harmony-based batch correction approach[56]. Harmony is applied to generate the dimensionality reductions used, and batch labels are additionally used to subset the input to the random forest classifiers, so that for each cluster comparison only cells from the same batch are compared. In instances where the two clusters being compared do not each have sufficient representation from the same batch or batches, these clusters are assumed to be batch-confounded instances of the same cell population and are merged. For example, if cluster 1 predominantly consists of cells belonging to batch A, but cluster 2 predominantly consists of cells

belonging to batch B, these two clusters would be merged. Thus, batch information is used at every stage of the clustering algorithm in order to avoid overclustering due to batch confounds. It should be noted that this optional batch correction approach assumes that cell groups of interest, such as unique cell types or states, are not batch-confounded.

Beyond confounds due to batch effects, factors such as read depth and cell count will always affect the granularity with which cell types and states can be accurately distinguished. Even high-quality clustering results are simply proxies for the highly continuous landscape of real biological features. Although CHOIR performs well on datasets across a range of developmental continuity, there may be instances where trajectory analyses should be combined with clustering approaches in order to represent the nuances of developmental lineages. As the size and diversity of available single-cell datasets increase, so too will the absolute number of bona fide cellular states. Although CHOIR does not define "singleton clusters" composed of a single cell, it is typically able to distinguish rare clusters with as few as 20–30 cells and clusters of cells that make up less than 0.05% percent of a dataset, highlighting an enabling strength of this new algorithm.

In light of the ever-widening field of single-cell sequencing technologies, we extended the utility of CHOIR beyond scRNA-seq data by implementing computational methods free of assumptions about the distribution of the data. The random forest classifier and permutation test framework used by CHOIR were specifically chosen because they are free of such assumptions. CHOIR is therefore amenable to feature matrices of any modality and is applicable to multi-modal data with an arbitrary number of features and modalities. Consequently, CHOIR is uniquely poised to address both current needs and future developments in single-cell analysis.

Notably, across an array of dataset sizes, modalities, and biological contexts, the default parameters implemented in CHOIR were consistently either the optimal parameters or yielded similar results as the optimal parameter settings, a feature that enhances performance and mitigates subjectivity in cluster identification. Indeed, the examples presented in this manuscript exclusively used the default parameters of CHOIR (see **Supplementary Note 2.2** for details). In light of the strengths described above, the

computational approach we developed should facilitate cell type and cell state identification across a diversity of modalities and biological contexts.

Chapter 4

Methods

Details of the CHOIR algorithm

Random forest classifiers used in a permutation test framework

CHOIR operates around a central algorithmic cog: a permutation test using random forest classifiers to learn and predict the cluster labels of cells in a pair of clusters. The cells belonging to the pair of clusters are split equally into a training set and test set. If the two clusters do not contain the same number of cells, the larger cluster is downsampled to create a balanced representation of the two clusters in the training and test set. CHOIR uses balanced random forest classifiers rather than imbalanced random forest classifiers because balanced random forest classifiers are not prone to overestimating accuracy in situations of severe class imbalance. The random forest classifier is then trained using a cell by feature matrix. By default, CHOIR will use only the highly variable features. For multi-omic datasets, highly variable features from all modalities are used jointly as the input for the random forest classifier. Subsequently, the trained random forest classifier is used to predict the cluster assignments of the cells in the test set. This process yields a prediction accuracy score between zero and one, where one represents perfect prediction. Higher prediction scores indicate that the two clusters are more different, and thus more readily distinguishable. In parallel, CHOIR shuffles the cluster labels and repeats the same process using both true and permuted labels on bootstrapped samples over a number of iterations (100 by default), resulting in a permutation test that compares the true prediction accuracy for the clusters to the prediction accuracy for a chance division of the cells into two random groups. This permutation testing yields a p-value that determines whether the two clusters are merged or kept separate. The significance threshold used can be adjusted according to an alpha parameter. The default significance level used by CHOIR is 0.05 with Bonferroni multiple comparison

correction. The implications of adjustments to each parameter setting used by CHOIR are detailed in **Supplementary Note 2.2**.

For datasets that require batch correction (optional), CHOIR uses a batch-aware approach to run the permutation tests. When batch correction is enabled, only cells from the same batch are compared in a single random forest classifier iteration. The batches present in the data alternate across all of the iterations of the permutation test. In addition, if the two clusters being compared have very few cells belonging to the same batch, they are merged to avoid batch confounds. This batch correction approach assumes that any experimental groups of interest such as unique cell types or states are not batch-confounded.

Because the iterations of the permutation test are parallelized, the efficiency of CHOIR greatly improves with additional computational cores. In addition, CHOIR applies three filters that reduce the total number of comparisons performed. First, CHOIR imposes a minimum accuracy threshold for the random forest classifier predictions, below which clusters will be automatically merged. This accuracy threshold defaults to random chance (0.5); higher values can be set in order to identify a more conservative set of clusters. Second, by default, CHOIR only runs the permutation test comparison if the two clusters share at least one nearest neighbor. Clusters with no nearest neighbors remain separate. The minimum number of nearest neighbors required can be adjusted by the user. Third, CHOIR records the distance (in the generated dimensionality reduction space) between each cluster and its nearest distinguishable neighbor. Multiplied by a set value, this distance is used to generate a distance threshold above which this cluster is not merged with any other clusters. By default, this threshold is set at a two-fold increase in distance. CHOIR also uses downsampling to increase efficiency for larger datasets. By default, downsampling occurs at each random forest classifier comparison. The downsampling rate is determined based on the overall dataset size or can be set by the user.

Generating the initial hierarchical clustering tree

The first step in the CHOIR algorithm is the top-down generation of a hierarchical clustering tree. In the first layer of this clustering tree, all cells belong to a single cluster. The first section of the tree is

referred to as the "root tree." Using all cells, CHOIR identifies a set of features with highly variable levels of expression from the provided cell-by-feature matrix. CHOIR can accept a variety of inputs. For scRNA-seq data, CHOIR has been most thoroughly tested with log normalized counts. For snATAC-seq data, we have found that using a matrix of cell-by-gene scores performs better than sparser formats. By default, CHOIR calculates variable features using VST for Seurat and SingleCellExperiment objects, or extracted variables from iterative LSI for ArchR objects. Also by default, CHOIR uses 25,000 variable features for ATAC-seq data and 2,000 variable features for other types of data. CHOIR then computes a dimensionality reduction for all cells, using one of three methods: principal component analysis (PCA), latent semantic indexing (LSI), or iterative LSI. PCA is used as the default, except for ATAC-seq data. For datasets that require batch correction, CHOIR generates a Harmony-corrected dimensionality reduction[56]. For multi-omic datasets, a joint dimensionality reduction is created across all of the provided modalities. Users may supply additional parameters to specify how the dimensionality reduction and batch correction steps are computed, or supply a pre-generated dimensionality reduction to CHOIR. We note that batch correction should not be used when a single batch is expected to contain cell types or states not present in any other batch, for example, when disease and control samples are batch confounded. In such a scenario, CHOIR would assume that the batch confounded cell groups would need to be merged into other cell groups. CHOIR then computes a nearest neighbor adjacency matrix. To generate the layers of the clustering tree, CHOIR uses either Louvain[2] or Leiden[3] clustering with increasing resolution parameters. CHOIR incrementally increases the resolution parameter at each level of the clustering tree. Because increases in the resolution parameter for Louvain and Leiden clustering do not lead to strictly hierarchical divisions of clusters, CHOIR uses the tool MRtree[57] to reshape clustering trees into a strictly hierarchical clustering tree.

At each level of the emerging root tree, the silhouette score is assessed. By default, an approximate silhouette score is used to avoid generating full distance matrices. Subdivision of the root tree continues until the silhouette score has reached a maximum score that is not exceeded in the subsequent levels of the tree. The root tree is then terminated at the level with the maximum silhouette score. This level is referred to as the "parent clusters." Each parent cluster is then subsetted, a new set of highly variable features is

identified, and a new dimensionality reduction and nearest neighbor adjacency matrix are generated. These new elements are used as the basis for a "subtree" originating from the parent cluster. Each of the resulting subtrees continues to be subdivided until the farthest pair of nearest neighboring clusters is found to be overclustered by the permutation test approach. At this level, the clustering tree is terminated, as it is likely that most, if not all, of the clusters are overclustered.

Pruning the hierarchical clustering tree

To prune back the branches of the hierarchical clustering tree, CHOIR iterates through each branch point of the clustering tree using a bottom-up approach. At each branch point, pairs of clusters are compared with the permutation test approach. For these permutation tests, the input matrices used for the random forest classifiers depend on the parent cluster origins of the two clusters being compared. If both clusters are from the same subtree, the highly variable features generated for that subtree are used as input to the permutation test framework. Otherwise, the highly variable features from the root tree are used.

When all levels of the clustering tree have been assessed, the resulting clusters can be assumed to have no instances of overclustering. If any of the resulting clusters are identical to clusters in the bottom tier of the tree (where pruning began), these clusters are subdivided and undergo additional rounds of pruning, ensuring that no instances of underclustering remain. Thus, CHOIR assesses both overclustering and underclustering to identify the final clusters.

For users who already have a set of clusters generated by a different tool and wish to assess whether these clusters are underclustered or overclustered, CHOIR can accept externally generated clusters, a dimensionality reduction, and a nearest neighbor adjacency matrix. Users may provide an externally generated clustering tree, or CHOIR can generate a hierarchical clustering tree using the cluster centroid distances in the provided dimensionality reduction space with the CHOIR "inferTree" function. With these inputs, CHOIR will then proceed to prune the clustering tree in the same manner as described above.

Count splitting

One source of potential overclustering is the common practice known as "double-dipping," in which the same data is used to both partition cells into clusters and assess whether those partitions are sufficiently different. Count splitting, in which the feature counts for each cell are split across two or more matrices using a Poisson or negative binomial distribution, has been proposed as a way to avoid double-dipping[16]. CHOIR allows users to implement the R package countsplit[16] to compute two count split matrices that can then be used as input for the clustering steps. Specifically, one matrix is used to calculate highly variable features, dimensionality reductions, nearest neighbor adjacency matrices, and the initial clustering tree. The other matrix is exclusively used as input for the random forest classifiers. Alternatively, users can input any two independent matrices containing the same cells. For instance, using one modality for cluster generation and a different modality for pruning would similarly prevent double-dipping. Because count splitting inherently decreases the effective library size available for each of these tasks, it is likely to perform best in datasets with high read depths.

Simulated data

Synthetic datasets were generated using the R package Splatter (1.20.0)[29], which simulates scRNA-seq data from either single or multiple cell populations based on a gamma-Poisson distribution. We generated a set of 100 simulated datasets, spanning 1 to 20 ground truth groups and 250 to 20,000 cells, with 10,000 features each (**Fig. 2.2**). Each group and cell number combination was repeated 5 times, using randomized parameter values. Subsequently, we generated 115 additional simulated datasets by individually modifying specific Splatter parameters used in the generation of five of the original 100 simulated datasets (Simulated Datasets 46–50). For each of these five datasets, we separately downsampled the smallest ground truth group stepwise to 100 random cells, reduced the degree of differential expression of two of the ground truth groups, decreased the library size, or split the data into two balanced or imbalanced batches. We also generated five datasets using Splatter's trajectory mode. In addition, we

generated 10 simulated datasets using scDesign3 with either balanced or imbalanced ground truth group sizes, based upon real subsampled PBMC populations.

Benchmarking

Pre-processing

For each dataset, a standardized pre-processing pipeline was applied using Snakemake (7.32.4)[58]. Depending on the file format of the input files (bam, fastq, or mtx) available for each dataset, Cell Ranger (7.2.0) and Cell Ranger ARC (2.0.1) functions were applied to generate a count matrix. For data available as bam or fastq files, analyses of human and mouse data were performed using the hg38 genome and mm10 genome, respectively, using refgenie[59,60] for genome resource management.

For sn/scRNA-seq data, a series of quality control steps was applied: (1) Cell Ranger or emptyDrops (DropletUtils 1.16.0)[61] was used to identify droplets containing only ambient RNA, (2) the number of reads per cell, number of features per cell, and percent mitochondrial reads were used to filter out low-quality cells or nuclei that were more than three median absolute deviations from the median[62], and (3) doublets were detected with DoubletFinder (2.0.3)[63] using default parameters. Due to the size of the human brain atlas dataset, this dataset was not independently pre-processed. Instead, count matrices following quality control steps applied by the authors were used. Data were supplied to clustering methods as raw counts or log normalized counts. Log normalization was performed using the Seurat function NormalizeData, wherein gene read counts are divided by total reads per cell, multiplied by a factor of 10,000, and transformed using log1p. All plots in the manuscript that show "Log normalized expression" use this normalization. For CHOIR and Seurat, datasets larger than 25,000 cells were provided as BPCells (0.1.0) counts matrices for maximal efficiency.

For snATAC-seq data, a series of quality control steps was applied using ArchR (1.0.3)[12]. Cells with transcription start site (TSS) enrichment scores <2 or fragment counts <2500 were excluded. ArchR was also used to detect and filter out doublets. For the Wang et al. 2022[15] multi-omic ATAC-seq and RNA-

seq dataset, clustering was conducted on the shared set of cells that passed quality control thresholds in both pre-processing pipelines.

Clustering

Each benchmarking dataset was clustered using CHOIR alongside a set of existing methods (**Supplementary Table 2.4**) according to the data type, across a range of reasonable parameter measures informed by available method documentation, including the default parameter values for each method. For the simulated datasets, 15 existing clustering methods were used for comparison. Note that, although SC3 is still widely used, it is not actively maintained or recommended for use by its developers. The newer version, SC3s[64], would be the more appropriate tool to benchmark against; however, SC3s requires the user to specify the number of clusters to identify, and therefore did not meet our criteria for method inclusion. Of the 15 methods tested, the five methods with the best performance, Cytocipher, GiniClust3, SCCAF, sc-SHC, and Seurat, were benchmarked using the real single-cell datasets. For datasets that require batch correction, CHOIR uses an internal Harmony-based batch correction approach[56]. Harmony batch correction was applied to the other methods as well, with the exception of sc-SHC, which does not use a dimensionality reduction for the input to clustering and has an internal batch-correction method, and GiniClust3, which is not compatible with batch correction. For the sci-Space dataset, scTriangulate consensus clustering was run twice: once with the default parameter settings for CHOIR, Cytocipher, and Seurat as input, and again with the default parameter settings for CHOIR, Cytocipher, GiniClust3, SCCAF, sc-SHC, and Seurat as input, using the 2,000 variable features identified by CHOIR. All clustering methods were run using 16 cores (if parallelized). Computational time and memory usage, including the maximum virtual memory utilization (Gb) and integral memory usage (Gbytes seconds), were recorded (**Supplementary Tables 2.2–2.7, Source Data 2.1**), and a maximum runtime of 96 hours was imposed. CHOIR numbers clusters in order from largest to smallest. A standard color palette for the cluster order was used throughout the analyses shown. For methods in which the resulting clusters were not numbered from largest to smallest, the color palette was still applied in order from largest to smallest, to aid visual comparison.

Performance Metrics

For simulated datasets, we were primarily concerned with the similarity between the generated clusters and the ground truth groups, which was measured using the ARI[30] for simulated datasets with more than one ground truth group. For the cancer cell line dataset, we assumed that the cell line identities represented the minimal distinct identities present in the dataset. We quantitatively assessed the degree of underclustering for each method by using the entropy of cluster accuracy[65] to measure the diversity of cell lines present in each cluster. For the entropy of cluster accuracy, higher values indicate that multiple cell lines are grouped in a single cluster, and lower values indicate a prevention of underclustering. In addition, we assessed whether each cell line was accurately distinguished without instances of underclustering by identifying the primary cluster containing the most cells from a given cell line. The cell line was considered to be accurately distinguished if the majority of the cells in this primary cluster were from that same cell line, and if it was not the primary cluster for any other cell line. We then extended this analysis by taking each cell line separately and downsampling it to 50 random cells (the number of cells in each cell line in the original dataset ranged from 56 to 1,990 cells) and rerunning all of the parameter settings for each clustering method that succeeded in accurately distinguishing that cell line in the original dataset. We then measured again whether this cell line was accurately distinguished when downsampled. To investigate cases where CHOIR identified multiple clusters within one cell line, we calculated a proliferation module score using the Seurat function "AddModuleScore" on the genes in GO term CELL_PROLIFERATION_GO_0008283.v2023.2 and calculated the distribution of this module score across the cells of each cluster identified by CHOIR. We then assessed whether these distributions were significantly different with a two-sided Mann-Whitney test.

For the multi-omic CITE-seq dataset, we computed clusters on the RNA-seq modality and evaluated separation along the orthogonal surface marker protein modality. For these clusters, we identified the closest 50 cluster pairs, as determined by the centroid distance in the reduced PCA dimensions generated using the default parameters of CHOIR. Using edgeR (3.36.0)[66] within the Libra (1.0.0)[67] package interface,

we then conducted pairwise differential expression analyses of the surface marker protein data, using pseudobulked values for each sample for each of the cluster pairs. The minimum number of differentially expressed genes with an FDR-adjusted p-value <0.05 was assessed, as well as the number of clusters obtained. These metrics were used to identify methods that maximize the number of clusters while preserving pairwise differences among clusters. Because the surface marker protein data was not used to assign the cluster labels, it represents an orthogonal validation of the cluster differences.

For the sci-Space dataset, clusters were also computed using the snRNA-seq data, without any use of the spatial coordinate data. The cluster labels were then imposed upon the respective spatial coordinates of each section to assess the spatial distribution of clusters. For each section, the mean percentage of all clusters that had a highly localized spatial distribution across the section, defined as a mean radial area <5% of the approximate whole embryo section area, was computed for all parameter settings tested for each clustering method. A rectangular area was computed as an approximation of the sci-Space sampling area for each section. Expression levels of corresponding marker genes for each cluster were visualized on Uniform Manifold Approximation and Projection (UMAP) embeddings and were also mapped onto spatial coordinates. MAGIC (2.0.3)[68] was used to impute expression levels for visualizations only.

Data availability

All datasets were acquired from the accession numbers provided in the original publications (**Supplementary Table 2.1**). We additionally made other analysis files available on our publication page at https://github.com/corceslab/2025_Sant_CHOIR.

Code availability

CHOIR is available as an R package at https://github.com/corceslab/CHOIR, and documentation for CHOIR is hosted at https://www.CHOIRclustering.com. The code to reproduce the analyses from this paper is available at https://github.com/corceslab/2025_Sant_CHOIR. The simulated datasets generated here are available at https://files.corces.gladstone.org/Publications/2025_Sant_CHOIR.

Acknowledgements

We thank members of the Corces and Mucke laboratories for helpful comments, and A. Dalton, S. Sanchez, and R. Mott for administrative assistance. This work was supported by National Institutes of Health grants P01 AG073082 (L.M. and M.R.C.), UM1 HG012076 (M.R.C.), and U01 AG072573 (M.R.C.), and the Dolby Family Fund (L.M. and M.R.C.). C.S. was supported by a National Science Foundation Graduate Research Fellowship.

Contributions

C.S., L.M., and M.R.C. conceived the project. C.S. led the design of the CHOIR software with input from L.M. and M.R.C. C.S. led the single-cell analyses presented in this paper. C.S., L.M., and M.R.C. wrote the manuscript.

Competing interests

The authors declare no competing interests.

References

1 Tabula Sapiens, C. *et al.* The Tabula Sapiens: A multiple-organ, single-cell transcriptomic atlas of humans. *Science* **376**, eabl4896 (2022).

2 Blondel, V. D., Guillaume, J. L., Lambiotte, R. & Lefebvre, E. Fast unfolding of communities in large networks. *J. Stat. Mech.* **P10008**, 1–12 (2008).

3 Traag, V. A., Waltman, L. & van Eck, N. J. From Louvain to Leiden: guaranteeing well-connected communities. *Sci. Rep.* **9**, 1–12 (2019).

4 Kiselev, V. Y., Andrews, T. S. & Hemberg, M. Challenges in unsupervised clustering of single-cell RNA-seq data. *Nat. Rev. Genet.* **20**, 273–282 (2019).

5 Herman, J. S., Sagar & Grun, D. FateID infers cell fate bias in multipotent progenitors from single-cell RNA-seq data. *Nat. Methods* **15**, 379–386 (2018).

6 Dong, R. & Yuan, G. C. GiniClust3: a fast and memory-efficient tool for rare cell type identification. *BMC Bioinformatics* **21**, 1–7 (2020).

7 Peyvandipour, A., Shafi, A., Saberian, N. & Draghici, S. Identification of cell types from single cell data using stable clustering. *Sci. Rep.* **10**, 1–12 (2020).

8 Tang, M. *et al.* Evaluating single-cell cluster stability using the Jaccard similarity index. *Bioinformatics* **37**, 2212–2214 (2021).

9 Grabski, I. N., Street, K. & Irizarry, R. A. Significance analysis for clustering with single-cell RNA-sequencing data. *Nat. Methods* **20**, 1196–1202 (2023).

10 Hao, Y. *et al.* Dictionary learning for integrative, multimodal and scalable single-cell analysis. *Nat. Biotechnol.*, 1–22 (2023).

11 Amezquita, R. A. *et al.* Orchestrating single-cell analysis with Bioconductor. *Nat. Methods* **17**, 137–145 (2020).

12 Granja, J. M. *et al.* ArchR is a scalable software package for integrative single-cell chromatin accessibility analysis. *Nat. Genet.* **53**, 403–411 (2021).

13 Stuart, T., Srivastava, A., Madad, S., Lareau, C. A. & Satija, R. Single-cell chromatin state analysis with Signac. *Nat. Methods* **18**, 1333–1341 (2021).

14 Rousseeuw, P. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *J. Comput. Appl. Math.*, 53–65 (1987).

15 Wang, S. K. *et al.* Single-cell multiome of the human retina and deep learning nominate causal variants in complex eye diseases. *Cell Genom.* **2**, 1–39 (2022).

16 Neufeld, A., Gao, L. L., Popp, J., Battle, A. & Witten, D. Inference after latent variable estimation for single-cell RNA sequencing data. *Biostatistics* **25**, 270–287 (2024).

17 Siletti, K. *et al.* Transcriptomic diversity of cell types across the adult human brain. *Science* **382**, 1–11 (2023).

18 Kiselev, V. Y. *et al.* SC3: consensus clustering of single-cell RNA-seq data. *Nat. Methods* **14**, 483–486 (2017).

19 Lin, P., Troup, M. & Ho, J. W. CIDR: Ultrafast and accurate clustering through imputation for single-cell RNA-seq data. *Genome Biol.* **18**, 1–11 (2017).

20 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. Methods* **14**, 414–416 (2017).

21 Sinha, D., Kumar, A., Kumar, H., Bandyopadhyay, S. & Sengupta, D. dropClust: efficient clustering of ultra-large scRNA-seq data. *Nucleic Acids Res.* **46**, 1–9 (2018).

22 Hu, M. W. *et al.* PanoView: An iterative clustering method for single-cell RNA sequencing data. *PLoS Comput. Biol.* **15**, 1–17 (2019).

23 Yang, Y. *et al.* SAFE-clustering: single-cell aggregated (from Ensemble) clustering for single-cell RNA-seq data. *Bioinformatics* **35**, 1269–1277 (2019).

24 John, C. R., Watson, D., Barnes, M. R., Pitzalis, C. & Lewis, M. J. Spectrum: fast density-aware spectral clustering for single and multi-omic data. *Bioinformatics* **36**, 1159–1166 (2020).

25 Miao, Z. *et al.* Putative cell type discovery from single-cell gene expression data. *Nat. Methods* **17**, 621–628 (2020).

26 Wan, S., Kim, J. & Won, K. J. SHARP: hyperfast and accurate processing of single-cell RNA-seq data via ensemble random projection. *Genome Res.* **30**, 205–213 (2020).

27 Tran, B., Tran, D., Nguyen, H., Ro, S. & Nguyen, T. scCAN: single-cell clustering using autoencoder and network fusion. *Sci. Rep.* **12**, 1–10 (2022).

28 Balderson, B., Piper, M., Thor, S. & Boden, M. Cytocipher determines significantly different populations of cells in single-cell RNA-seq data. *Bioinformatics* **39**, 1–16 (2023).

29 Zappia, L., Phipson, B. & Oshlack, A. Splatter: simulation of single-cell RNA sequencing data. *Genome Biol.* **18**, 1–15 (2017).

30 Hubert, L. & Arabie, P. Comparing partitions. *J. Classif.* **2**, 193–218 (1985).

31 Song, D. *et al.* scDesign3 generates realistic in silico data for multimodal single-cell and spatial omics. *Nat. Biotechnol.* **42**, 247–252 (2024).

32 Duò, A., Robinson, M. D. & Soneson, C. A systematic performance evaluation of clustering methods for single-cell RNA-seq data. *F1000Res.* **7**, 1–23 (2018).

33 Zheng, G. X. *et al.* Massively parallel digital transcriptional profiling of single cells. *Nat. Commun.* **8** (2017).

34 Kinker, G. S. *et al.* Pan-cancer single-cell RNA-seq identifies recurring programs of cellular heterogeneity. *Nat. Genet.* **52**, 1208–1218 (2020).

35 Yang, C., Tian, C., Hoffman, T. E., Jacobsen, N. K. & Spencer, S. L. Melanoma subpopulations that rapidly escape MAPK pathway inhibition incur DNA damage and rely on stress signalling. *Nat. Commun.* **12**, 1–14 (2021).

36 Dave, A. *et al.* The Breast Cancer Single-Cell Atlas: Defining cellular heterogeneity within model cell lines and primary tumors to inform disease subtype, stemness, and treatment options. *Cell. Oncol.* **46**, 603–628 (2023).

37 Masui, S. *et al.* Pluripotency governed by Sox2 via regulation of Oct3/4 expression in mouse embryonic stem cells. *Nat. Cell Biol.* **9**, 625–635 (2007).

38 Jin, Y. *et al.* Comprehensive analysis of tropomyosin isoforms in skeletal muscles by top-down proteomics. *J. Muscle Res. Cell Motil.* **37**, 41–52 (2016).

39 Montani, C., Gritti, L., Beretta, S., Verpelli, C. & Sala, C. The synaptic and neuronal functions of the X-linked intellectual disability protein Interleukin-1 receptor accessory protein Like 1 (IL1RAPL1). *Dev. Neurobiol.* **79**, 85–95 (2019).

40 Hao, Y. *et al.* Integrated analysis of multimodal single-cell data. *Cell* **184**, 3573–3587 (2021).

41 Srivatsan, S. R. *et al.* Embryo-scale, single-cell spatial transcriptomics. *Science* **373**, 111–117 (2021).

42 Vignali, D. A., Collison, L. W. & Workman, C. J. How regulatory T cells work. *Nat. Rev. Immunol.* **8**, 523–532 (2008).

43 Pennock, N. D. *et al.* T cell responses: naive to memory and everything in between. *Adv. Physiol. Educ.* **37**, 273–283 (2013).

44 Böttcher, J. P. & Reis e Sousa, C. The role of type 1 conventional dendritic cells in cancer immunity. *Trends Cancer* **4**, 784–792 (2018).

45 Tao, G. *et al.* Collagen XIV is important for growth and structural integrity of the myocardium. *J. Mol. Cell. Cardiol.* **53**, 626–638 (2012).

46 Fish, J. E. & Marsden, P. A. Endothelial nitric oxide synthase: insight into cell-specific gene regulation in the vascular endothelium. *Cell. Mol. Life Sci.* **63**, 144–162 (2006).

47 Goncharov, N. V. *et al.* Markers of endothelial cells in normal and pathological conditions. *Biochem. (Mosc.) Suppl. Ser. A Membr. Cell Biol.* **14**, 167–183 (2020).

48 Guo, Y. & Pu, W. T. Cardiomyocyte maturation: New phase in development. *Circ. Res.* **126**, 1086–1106 (2020).

49 Mallika, C., Guo, Q. & Li, J. Y. Gbx2 is essential for maintaining thalamic neuron identity and repressing habenular characters in the developing thalamus. *Dev. Biol.* **407**, 26–39 (2015).

50 Liodis, P. *et al.* Lhx6 activity is required for the normal migration and specification of cortical interneuron subtypes. *J. Neurosci.* **27**, 3078–3089 (2007).

51 Rueda-Alaña, E., Martinez-Garay, I., Encinas, J. M., Molnar, Z. & Garcia-Moreno, F. Dbx1-derived pyramidal neurons are generated locally in the developing murine neocortex. *Front. Neurosci.* **12**, 1–10 (2018).

52 Fode, C. *et al.* A role for neural determination genes in specifying the dorsoventral identity of telencephalic neurons. *Genes Dev.* **14**, 67–80 (2000).

53 Thompson, C. L. *et al.* A high-resolution spatiotemporal atlas of gene expression of the developing mouse brain. *Neuron* **83**, 309–323 (2014).

54 Yaguchi, Y. *et al.* Fibroblast Growth Factor (FGF) gene expression in the developing cerebellum suggests multiple roles for FGF signaling during cerebellar morphogenesis and development. *Dev. Dyn.* **238**, 2058–2072 (2009).

55 Li, G. *et al.* Decision level integration of unimodal and multimodal single cell data with scTriangulate. *Nat. Commun.* **14**, 1–16 (2023).

56 Korsunsky, I. *et al.* Fast, sensitive and accurate integration of single-cell data with Harmony. *Nat. Methods* **16**, 1289–1296 (2019).

57 Peng, M. *et al.* Cell type hierarchy reconstruction via reconciliation of multi-resolution cluster tree. *Nucleic Acids Res.* **49**, e91 (2021).

58 Mölder, F. *et al.* Sustainable data analysis with Snakemake. *F1000Res.* **10**, 1–29 (2021).

59 Stolarczyk, M., Reuter, V. P., Smith, J. P., Magee, N. E. & Sheffield, N. C. Refgenie: a reference genome resource manager. *GigaScience* **9**, 1–6 (2020).

60 Stolarczyk, M., Xue, B. & Sheffield, N. C. Identity and compatibility of reference genome resources. *NAR Genom. Bioinform.* **3**, 1–6 (2021).

61 Lun, A. T. L. *et al.* EmptyDrops: distinguishing cells from empty droplets in droplet-based single-cell RNA sequencing data. *Genome Biol.* **20**, 1–9 (2019).

62 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).

63 McGinnis, C. S., Murrow, L. M. & Gartner, Z. J. DoubletFinder: Doublet detection in single-cell RNA sequencing data using artificial nearest neighbors. *Cell Syst.* **8**, 329–337 (2019).

64 Quah, F. X. & Hemberg, M. SC3s: efficient scaling of single cell consensus clustering to millions of cells. *BMC Bioinformatics* **23**, 1–5 (2022).

65 Tian, L. *et al.* Benchmarking single cell RNA-sequencing analysis pipelines using mixture control experiments. *Nat. Methods* **16**, 479–487 (2019).

66 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).

67 Squair, J. W. *et al.* Confronting false discoveries in single-cell differential expression. *Nat. Commun.* **12**, 1–15 (2021).

68 van Dijk, D. *et al.* Recovering gene interactions from single-cell data using data diffusion. *Cell* **174**, 716–729 (2018).

Main Figures

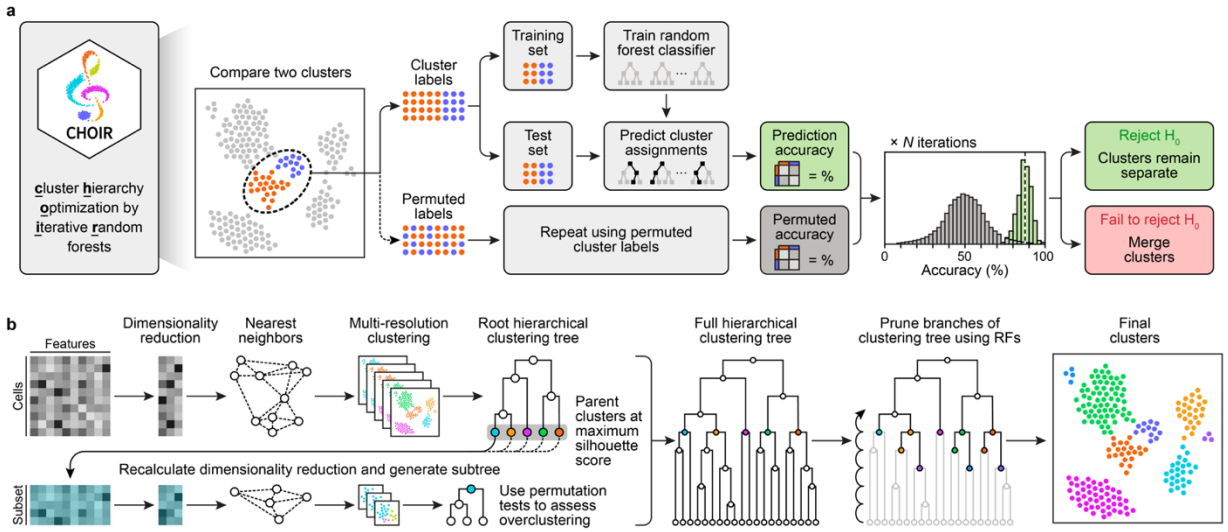


Fig. 2.1: CHOIR is a hierarchical clustering algorithm that uses permutation testing for cluster identification by statistical inference

a. Schematic demonstrating how CHOIR identifies clusters that should be merged by applying a permutation testing approach to assess the accuracy of random forest classifiers (RFs) in predicting cluster assignments from a normalized feature matrix.

b. Schematic demonstrating how CHOIR constructs and iteratively prunes a hierarchical clustering tree using statistical inference to prevent underclustering and overclustering.

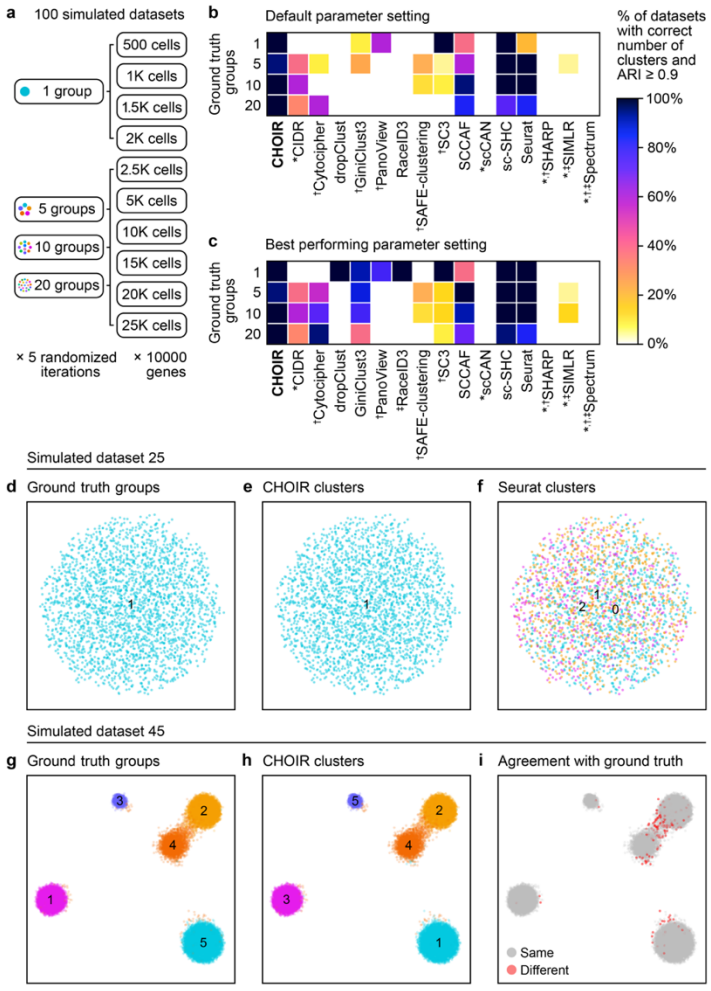


Fig. 2.2: CHOIR outperforms 15 existing clustering methods across 100 simulated datasets
a. Schematic summarizing the Simulated Datasets 1–100 created using the R package Splatter.
b–c. Heatmaps showing the percentage of simulated datasets in which the correct number of clusters was identified (all datasets) and with an Adjusted Rand Index >0.9 (datasets with >1 ground truth groups) for the default parameter setting (**b**) or the best performing parameter setting (**c**) for each method. Symbols indicate clustering methods that *assumed ≥2 clusters, †had some parameter settings that failed to run, or ‡had some parameter settings that did not complete within the maximum allotted runtime of 96 hours. For the CHOIR, CIDR, SAFE-clustering, scCAN, SHARP, and Spectrum methods, the best-performing parameter setting in (**c**) was the same as the default parameter setting in (**b**).
d–f. UMAP embeddings for simulated dataset 25, consisting of 2,500 cells and a single ground truth group of cells colored by the ground truth grouping (**d**) or the clusters achieved using the default parameters of CHOIR (**e**) or Seurat (**f**).
g–i. UMAP embeddings for simulated dataset 45, consisting of 20,000 cells and five ground truth groups of cells colored by the ground truth groupings (**g**), the clusters achieved using the default parameters of CHOIR (**h**), and the agreement between these CHOIR clusters and the ground truth labels (**i**).

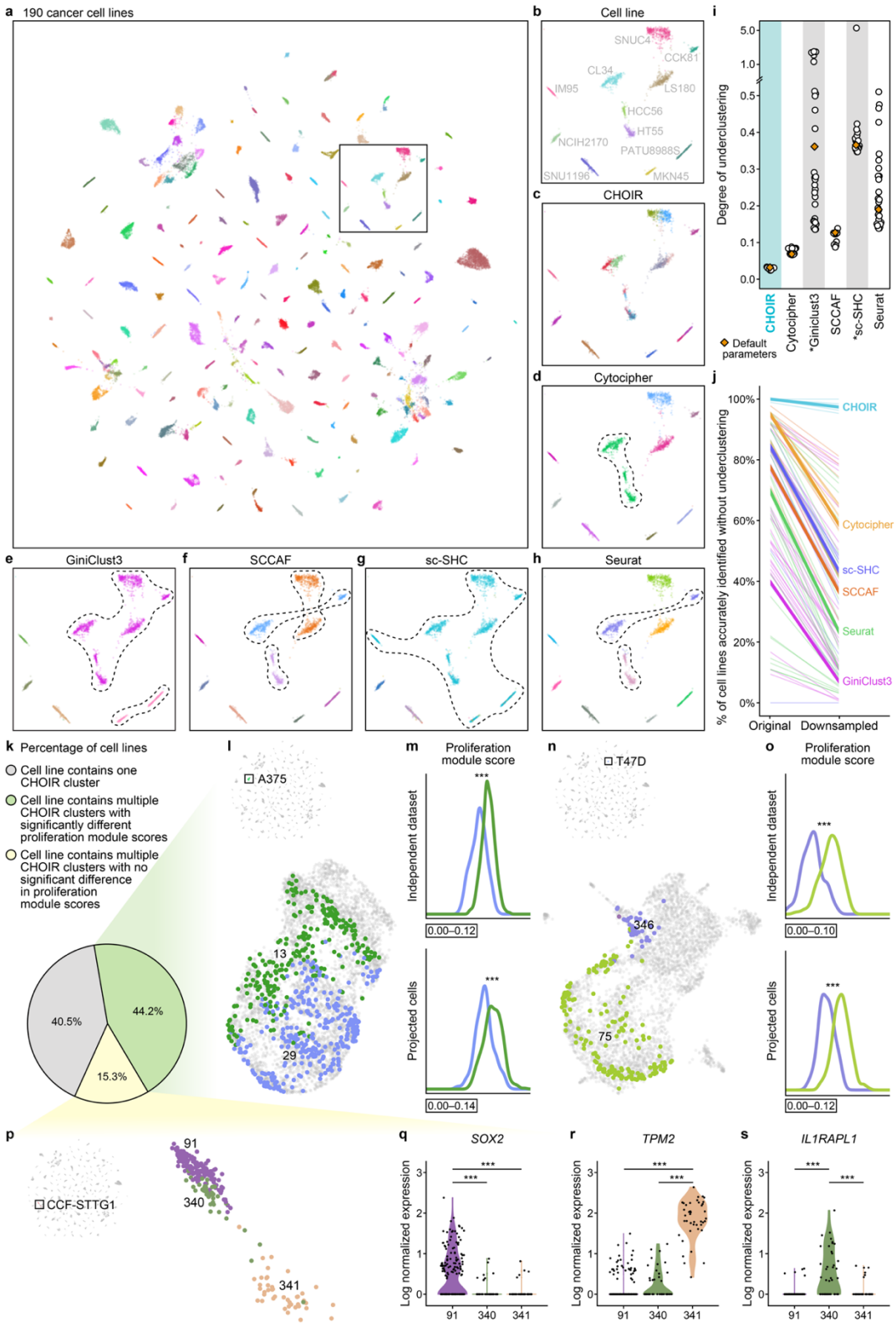


Fig. 2.3: CHOIR prevents underclustering in an scRNA-seq dataset consisting of pooled cell lines
a. UMAP embedding of a pooled cancer cell line scRNA-seq dataset from Kinker et al. 2020 consisting of 48,879 cells, colored according to the 190 cancer cell lines. (Figure caption continued on the next page.)

(Figure caption continued from the previous page.) **b–h.** UMAP embedding showing the cells within the highlighted square in (**a**), colored by cell line (**b**) and the clusters identified by CHOIR (**c**), Cytocipher (**d**), GiniClust3 (**e**), SCCAF (**f**), sc-SHC (**g**), and Seurat (**h**), using the default parameters for each method. Dashed lines indicate instances where the clustering method failed to distinguish individual cell lines, resulting in multiple cell lines grouped within a single cluster.
i. The entropy of cluster accuracy for all method and parameter combinations tested, representing the degree of underclustering. *Some parameters tested for GiniClust3 and sc-SHC resulted in a single cluster, for which the entropy of cluster accuracy could not be computed.
j. Line plot showing the effect of cell line downsampling on the cluster accuracy of each method. The performance of the default parameter setting for each method is shown as a dark line and other parameter settings tested are shown as a light line.
k. Pie chart showing the percentage of the 190 cell lines for which the default parameters of CHOIR resulted in a single cluster encompassing the cell line, multiple clusters within the cell line that had a significantly different proliferation module score, or multiple clusters within the cell line that did not have a significantly different proliferation module score. Differences in proliferation module scores were assessed using a Mann-Whitney U test with FDR=0.05.
l. UMAP embedding showing CHOIR cluster 13 and cluster 29 identified within the A375 cell line projected onto the dimensionality reduction space of the Yang et al. 2021 independent scRNA-seq dataset consisting of 4,794 A375 cells.
m. Distribution of the proliferation module score across the nearest neighbor A375 cells from the Yang et al. 2021 independent dataset (top) and the projected cells from Kinker et al. 2020 (bottom). For each plot, the minimum and maximum proliferation module scores are shown at the bottom. Differences in proliferation module scores were assessed using a Mann-Whitney U test, ***$P<0.001$.
n. UMAP embedding showing CHOIR cluster 75 and cluster 346 identified within the T47D cell line projected onto the dimensionality reduction space of the Dave et al. 2023 independent scRNA-seq dataset consisting of 4,582 T47D cells.
o. Distribution of the proliferation module score across the nearest neighbor T47D cells from the Dave et al. 2023 independent dataset (top) and the projected cells from Kinker et al. 2020 (bottom). For each plot, the minimum and maximum proliferation module scores are shown at the bottom. Differences in proliferation module scores were assessed using a Mann-Whitney U test, ***$P<0.001$.
p. Subsetted UMAP embedding showing CHOIR clusters 91, 140, and 141 identified within the CCF-STTG1 cell line.
q–s. Violin plots showing the expression level of *SOX2,* a pluripotency marker gene (**q**); *TPM2*, a gene typically expressed in muscle fibers (**r**); and *IL1RAPL1*, a gene expressed in neurons (**s**). Differential expression was assessed using Seurat, ***$P<0.001$.

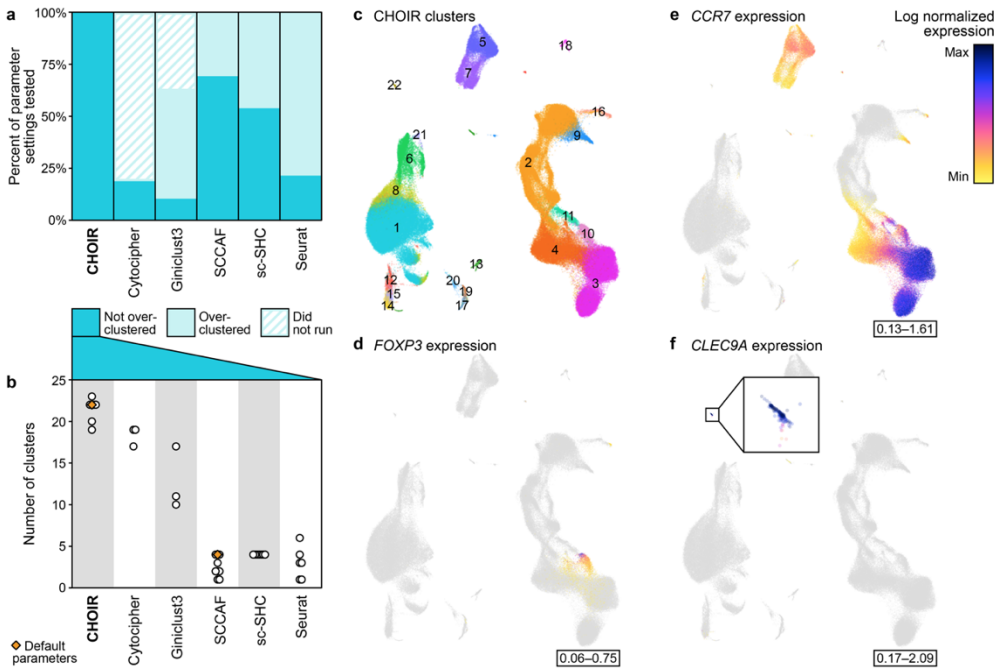


Fig. 2.4: Multi-omic data enables orthogonal validation of identified clusters
a. Stacked bar plot showing the percentage of parameter settings for each method tested on the Hao et al. 2021 human PBMC CITE-seq dataset that did (not overclustered) or did not (likely overclustered) result in at least one differentially expressed protein in all 50 closest pairwise cluster comparisons, or that failed to run.
b. Dot plot showing the number of clusters that was identified by the subset of all method and parameter combinations tested that did not result in likely overclustering (**a**).
c. UMAP embedding of cells from the Hao et al. 2021 human PBMC CITE-seq dataset colored according to the 22 clusters identified by the default parameter settings for CHOIR.
d–f. UMAP embedding as in panel (**c**), but colored according to the expression level of the regulatory T cell marker *FOXP3* (**d**), the expression level of naïve T cell marker *CCR7* (**e**), or the expression level of conventional dendritic cell type 1 marker *CLEC9A* (**f**). Because of the small size of cluster 22, a zoom-in window was used to display *CLEC9A* expression in panel (**f**). For each plot, the minimum and maximum normalized expression values are shown at the bottom. Grey color indicates no expression detected.

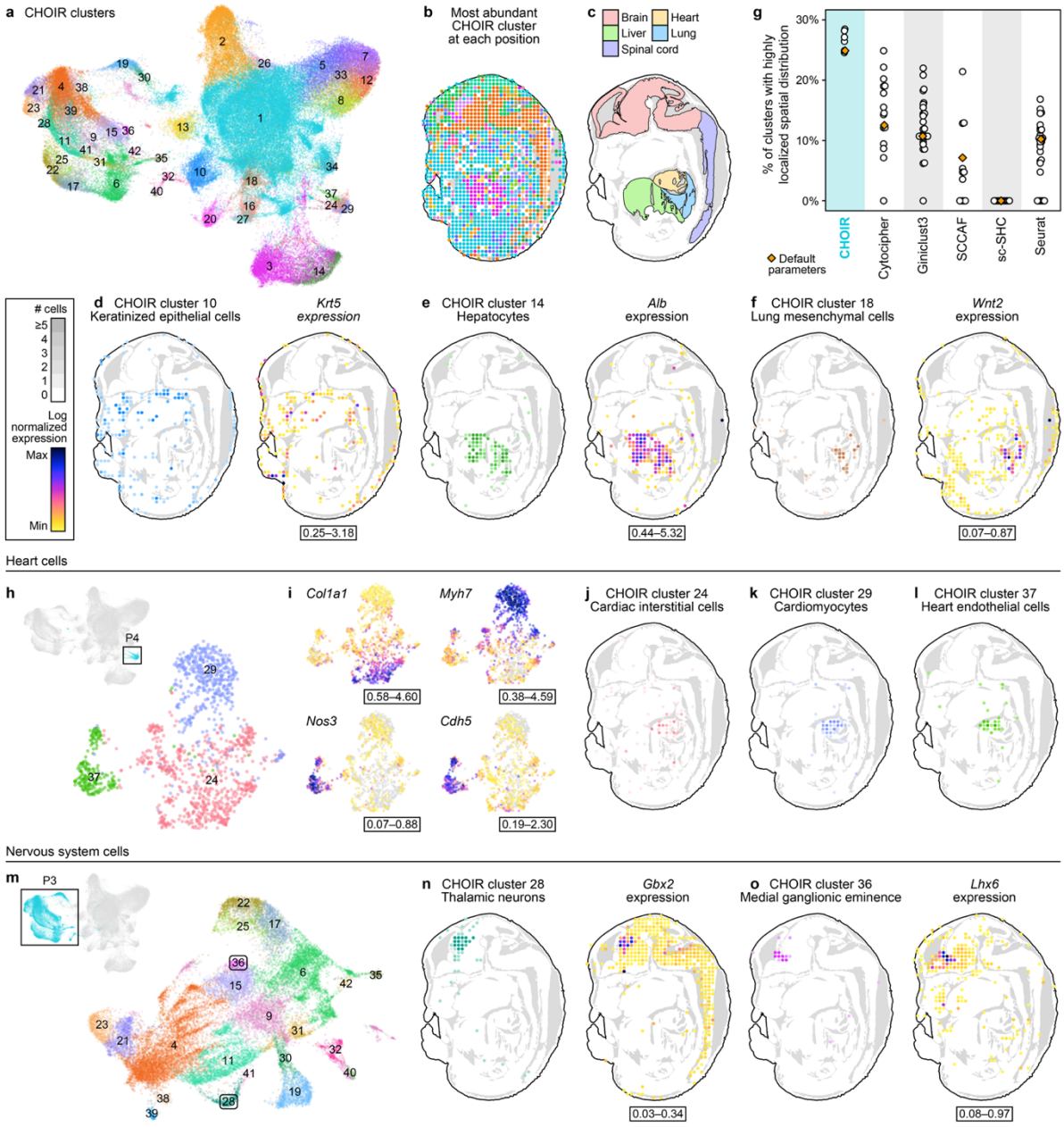
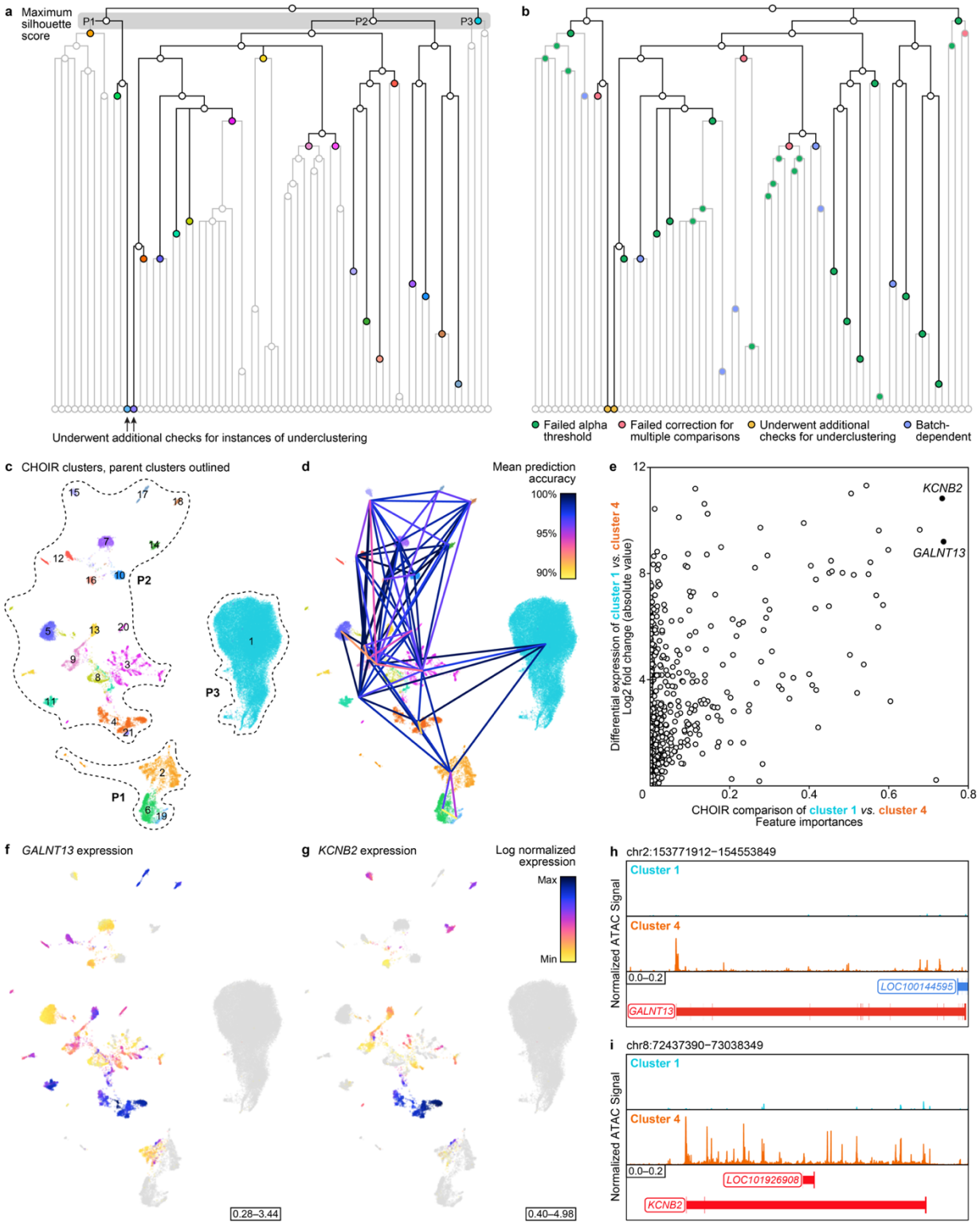


Fig. 2.5: CHOIR identifies anatomically localized clusters in a spatially resolved snRNA-seq dataset
a. UMAP embedding colored according to the 42 clusters identified by applying the default parameters of CHOIR to the snRNA-seq features of the Srivatsan et al. 2021 whole mouse embryo sci-Space dataset. Cluster colors persist throughout this figure.
b. Spatial coordinate map of the mouse embryo with each spatial point colored by the CHOIR cluster with the highest number of cells at each spatial coordinate position. Slide 14 is shown throughout this figure because it had the highest number of recovered nuclei and positions.
c. Anatomical boundaries, adapted from annotations used in Srivatsan et al. 2021. (Figure caption continued on the next page.)

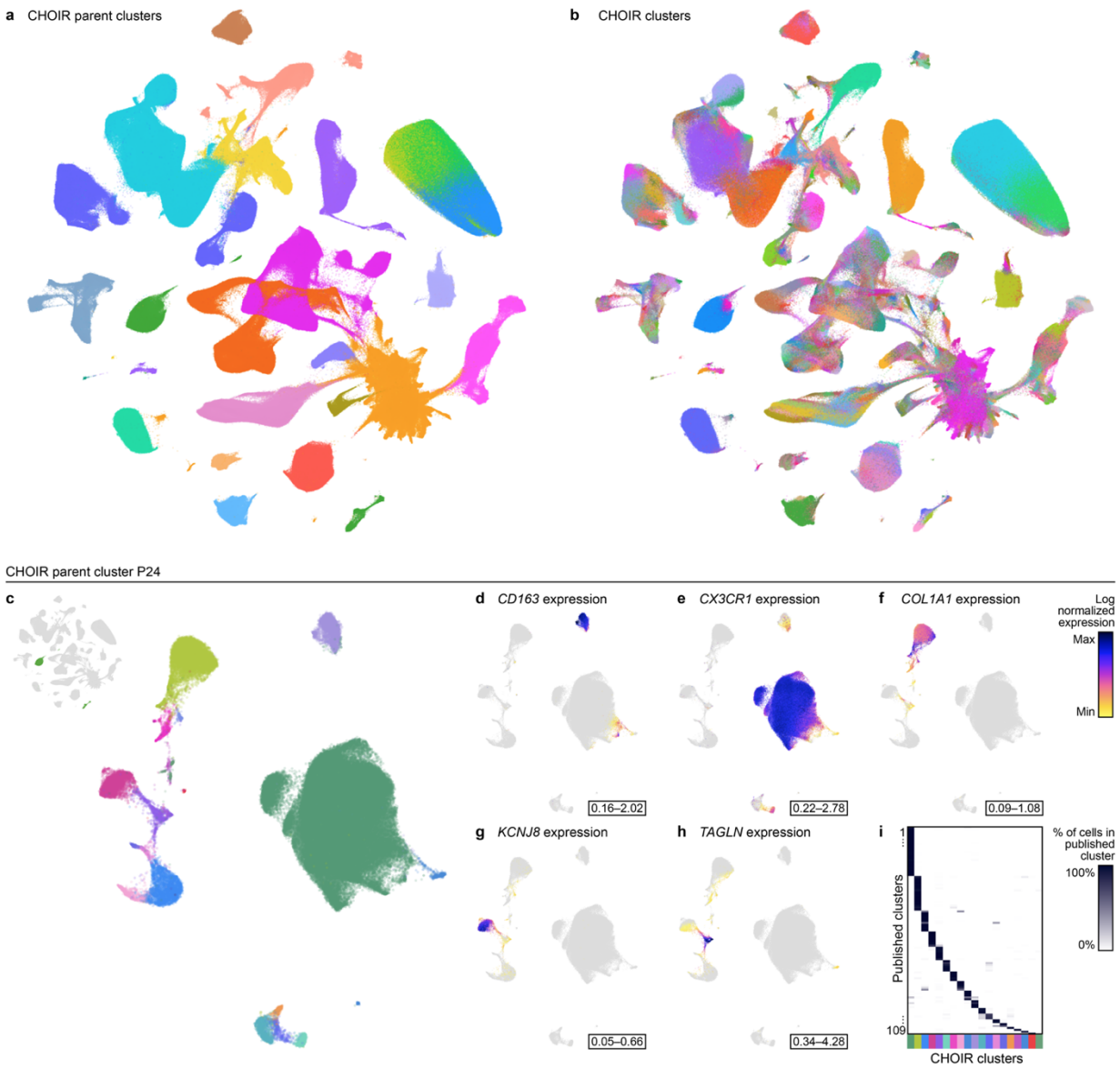
(Figure caption continued from the previous page.) **d–f.** Cluster distributions (left) and corresponding marker gene expression levels (right) for CHOIR cluster 10: keratinized epithelium cells (**d**); cluster 14, hepatocytes (**e**); and cluster 18, lung mesenchyme cells (**f**), all mapped onto the corresponding spatial coordinates. For plots showing CHOIR clusters, the darker the shade of any given color, the more cells of the respective cluster were detected at the indicated location.
g. Dot plot showing the mean percentage of all clusters per section that had a highly localized spatial distribution for all method and parameter combinations tested (see **Methods**). For Cytocipher, some parameter settings failed to run.
h–i. UMAP embedding computed on the subsetted dimensionality reduction of parent cluster P4, identified using default parameters of CHOIR and colored by CHOIR cluster (**h**) and the expression levels of connective tissue marker gene *Col1a1*, cardiomyocyte marker gene *Myh7*, or endothelial cell marker genes *Nos3* and *Cdh5* (**i**). See inset legend next to panel (**d**).
j–l. Cluster distributions for CHOIR cluster 24, *Col1a1*-expressing cardiac interstitial cells (**j**), cluster 29, *Myh7*-expressing cardiomyocytes (**k**), and cluster 37, *Nos3*/*Cdh5*-expressing heart endothelial cells (**l**) mapped onto spatial coordinates. The darker the shade of any given color, the more cells of the respective cluster were detected at the indicated location. See inset legend next to panel (**d**).
m. UMAP embedding computed on the subsetted dimensionality reduction of parent cluster P3 consisting of nervous system cells colored by the clusters identified by CHOIR using the default parameters.
n–o. Cluster distributions (left) and corresponding marker gene expression levels (right) for CHOIR cluster 28, *Gbx2*-expressing thalamic neurons (**n**), and cluster 36, *Lhx6*-expressing medial ganglionic eminence interneuron progenitors (**o**). For plots showing CHOIR clusters, the darker the shade of any given color, the more cells of the respective cluster were detected at the indicated location. See inset legend next to panel (**d**). For all gene expression plots in this figure, the minimum and maximum normalized expression values are shown at the bottom of each plot. Grey color indicates no expression detected.

Extended Data Figures

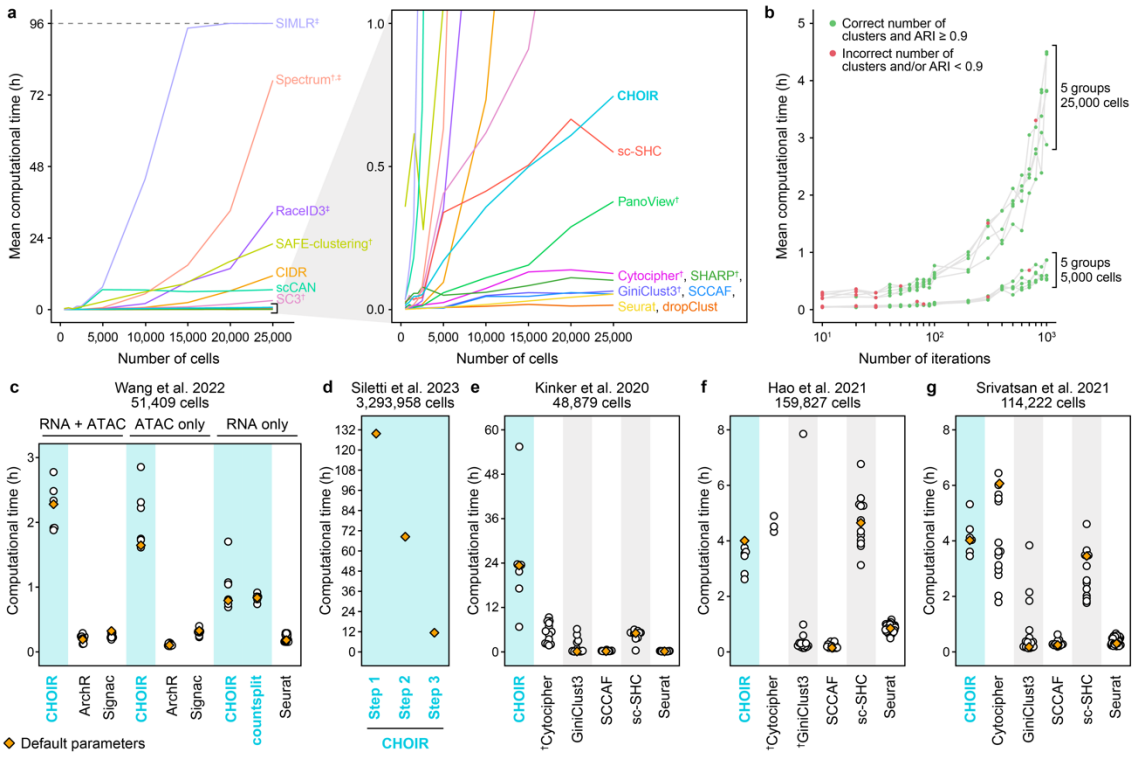


Extended Data Fig. 2.1: CHOIR effectively clusters multi-omic data and identifies cluster-specific features in both modalities (Figure caption continued on the next page.)

(Figure caption continued from the previous page.) **a.** Example of a hierarchical clustering tree generated by applying the default parameters of CHOIR to the RNA-seq and ATAC-seq features of the Wang et al. 2022 multi-omic ATAC-seq and RNA-seq dataset from human retinal cells. Final clusters identified are shown in color, pruned branches are shown in light grey. P1–P3 refer to parent clusters identified by maximizing the silhouette score, highlighted by the grey box.
b. Clustering tree as in panel (**a**), but colored according to the criteria used by CHOIR to prune each node of the tree until reaching the final set of clusters.
c. UMAP embedding colored according to the 21 clusters identified by CHOIR. Parent clusters identified by maximizing the silhouette score are outlined in dashed lines.
d. Mean prediction accuracy scores indicated by color of lines connecting pairs of final clusters identified by CHOIR. Each line connects the two clusters compared and is colored by the mean prediction accuracy. Not all final cluster pairs were directly compared by permutation testing because some pairs did not meet distance or adjacency criteria.
e. Dot plot comparing the feature importances extracted from the random forest comparisons computed by CHOIR for cluster 1 (rod cells) versus cluster 4 (cone cells) with the log fold change of gene expression between cluster 1 and cluster 4 identified using Seurat. Each dot represents an individual gene and all genes are shown.
f–g. UMAP embeddings colored according to the expression level of the two RNA-seq features with the highest feature importances in the comparison of CHOIR cluster 1 versus cluster 4: *GALNT13* (**f**) and *KCNB2* (**g**) were both enriched in cone cells. For each plot, the minimum and maximum normalized expression values are shown at the bottom. Grey color indicates no expression detected.
h–i. Genome track visualizations of the two corresponding ATAC-seq gene score features, which were among the ten ATAC-seq features with the highest feature importance in the comparison of CHOIR cluster 1 versus cluster 4: the *GALNT13* locus (chr2:153,771,912−154,553,849) (**h**) and the *KCNB2* locus (chr8:72,437,390−73,038,349) (**i**).



Extended Data Fig. 2.2: CHOIR supports clustering of atlas-scale datasets
a–b. UMAP embedding colored according to the parent clusters (**a**) and final clusters (**b**) identified by applying the default parameters of CHOIR to the Siletti et al. 2023 human brain atlas snRNA-seq dataset consisting of 3,293,958 cells.
c–h. UMAP embedding computed on the subsetted dimensionality reduction of parent cluster P24 identified using default parameters of CHOIR and colored by CHOIR cluster (**c**) and the expression levels of macrophage marker gene *CD163* (**d**), microglial marker gene *CX3CR1* (**e**), fibroblast marker gene *COL1A1* (**f**), pericyte marker gene *KCNJ8* (**g**), or vascular smooth muscle cell marker gene *TAGLN* (**h**). For each plot, the minimum and maximum normalized expression values are shown at the bottom. Grey color indicates no expression detected.
i. Confusion matrix heatmap showing the concordance between CHOIR clusters and published cluster annotations within parent cluster P24. Darker color indicates that the published cluster (y-axis) overlaps strongly with a single CHOIR cluster (x-axis).

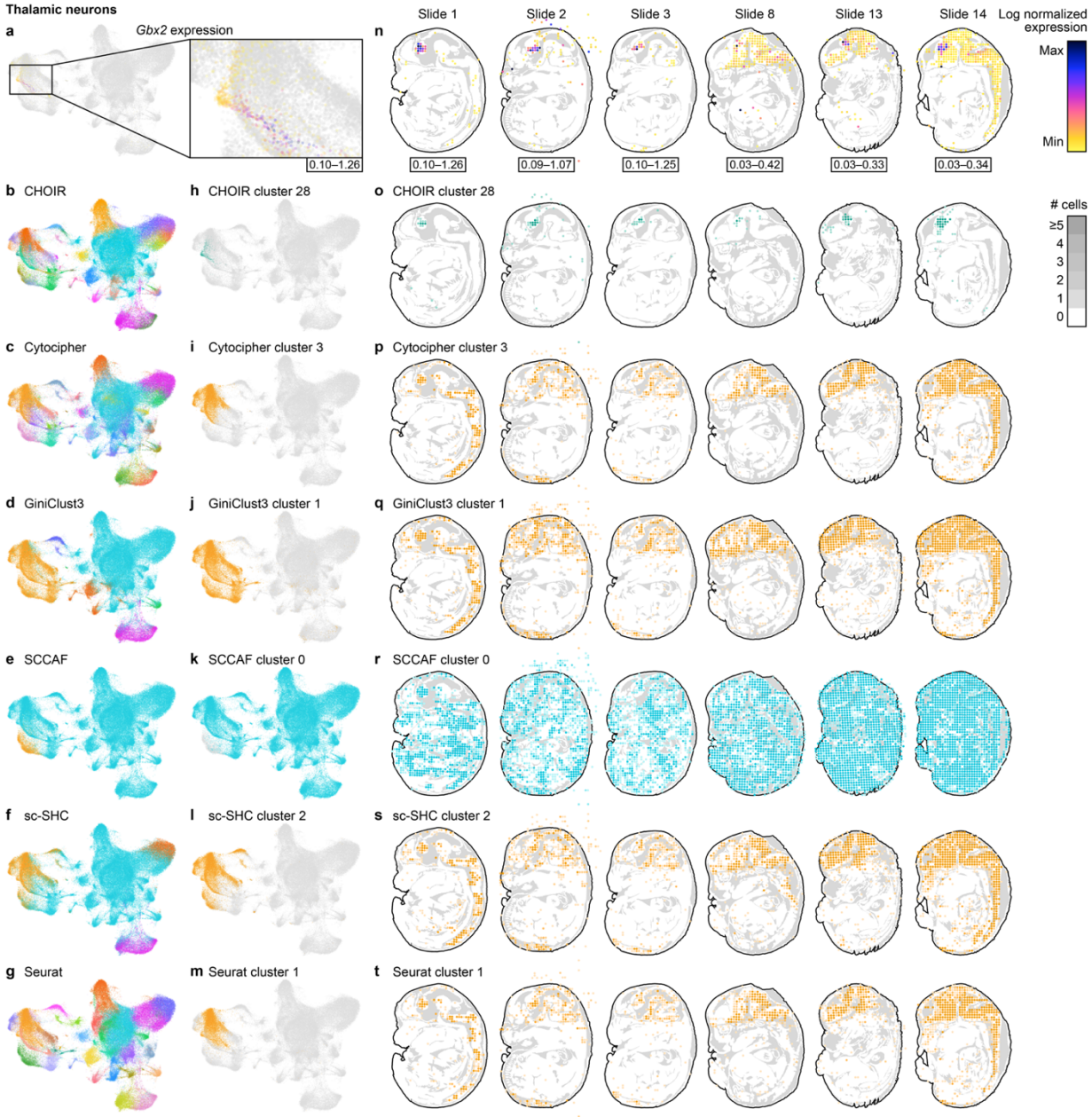


Extended Data Fig. 2.3: The computational time required by CHOIR is affected by multiple factors and is on par with other methods

a. Line plots showing computational time required for the best-performing parameter setting for each of the 15 clustering methods applied to simulated data, averaged across all simulated datasets of each size. Methods with a maximum runtime under one hour are highlighted in the box to the right. Symbols indicate clustering methods that †failed to run or ‡did not complete within the maximum allotted runtime of 96 hours for at least one dataset. For CHOIR, the best-performing parameter setting was the default setting.

b. Line plots showing the computational time required by CHOIR when using increasing numbers of iterations for the permutation tests across five simulated datasets of 5,000 cells and five simulated datasets of 25,000 cells. Each dot is colored by the clustering accuracy for each parameter setting and dataset.

c–f. Computational time of each parameter setting tested for each method for the Wang et al. 2022 multi-omic, ATAC-seq, and RNA-seq dataset (**c**), Siletti et al. 2023 human brain atlas snRNA-seq dataset (**d**), Kinker et al. 2020 cancer cell line dataset (**e**), Hao et al. 2021 CITE-seq dataset (**f**), and Srivatsan et al. 2021 sci-Space dataset (**g**). Symbols indicate clustering methods in which one or more parameter settings †failed to run.



Extended Data Fig. 2.4: CHOIR cluster 28 represents a population of neurons that is localized in the developing thalamus and expresses high levels of Gbx2 transcripts

a–g. UMAP embedding of the Srivatsan et al. 2021 whole mouse embryo sci-Space dataset, colored according to the expression level of thalamic marker gene *Gbx2* (**a**), or the clusters identified by the default parameters of CHOIR (**b**), Cytocipher (**c**), GiniClust3 (**d**), SCCAF (**e**), sc-SHC (**f**), or Seurat (**g**). In panel (**a**), a zoom-in of *Gbx2* expression is shown to the right and the minimum and maximum normalized expression values are shown at the bottom of the zoom-in. Grey color indicates no expression detected.

h–m. UMAP embedding of the Srivatsan et al. 2021 whole mouse embryo sci-Space dataset, colored by the cluster shown in panels (**a–g**) that harbors the *Gbx2*-expressing thalamic neurons for each of the methods shown to the left. (Figure caption continued on the next page.)

(Figure caption continued from the previous page.) **n.** Distribution of thalamic marker gene *Gbx2* in all sections with >25 cells belonging to CHOIR cluster 28 (thalamic neurons). For each plot, the minimum and maximum normalized expression values are shown at the bottom. Grey color indicates no expression detected.
o–t. Distribution of the method-specific cluster shown to the left in panels (**h–m**) across all sections shown in panel (**n**). The darker the shade of the cluster color, the more cells of the respective cluster were detected at the indicated location.

Supplementary Materials

Supplementary Note 2.1: How to use CHOIR on atlas-scale data

Using CHOIR "straight out of the box" works best on datasets with under 500,000 cells. While some larger datasets up to 1 million cells may also run smoothly, other datasets of this size or larger may encounter computational limits. In general, our assumption as developers is that end-users who are attempting to analyze datasets larger than 500,000 cells have access to high-performance computational resources that enable parallelization. At this scale of dataset, annotation of cell groups can often take weeks of manual effort without the assistance of a tool like CHOIR. While CHOIR will likely require at least 24 hours to run on a dataset of 500,000 cells, it will store all of the information an end-user would need to both annotate clusters and check the decisions made by CHOIR. Specifically, for each cluster, including those that were merged along the way, CHOIR stores the features found to be most important for the cluster (e.g., marker genes) and the reasoning behind why CHOIR merged or did not merge that cluster. These records can be found in `CHOIR→records→feature_importance_records` and `CHOIR→records→comparison_records`.

To use CHOIR on atlas-scale data, we recommend a three-step approach that partitions the dataset into smaller relevant chunks (step 1), runs CHOIR in parallel on those smaller chunks (step 2), and then stitches the results back together to provide a final set of clusters (step 3). A detailed tutorial for how to accomplish this is available on the CHOIR website (https://www.CHOIRclustering.com) and we encourage users to reference that documentation, as it will be maintained over time.

Step 1: Generate parent clusters

On the full dataset, run the CHOIR function `buildParentTree`. This function runs the initial dimensionality reduction, optional batch correction, and subsequent steps to generate the "parent" clustering tree. This clustering tree extends from a single cluster encompassing all cells to the cluster level at which the silhouette score is maximized. If any of these resulting "parent clusters" are above 450,000

cells, a minimal number of further subdivisions are run on that parent cluster until the resulting subclusters are all under 450,000 cells.

Step 2: Run CHOIR on each subsetted parent cluster

Subset each parent cluster and save it as a separate object. Run the standard CHOIR method on each of these objects. Extract the final clusters, clustering trees, and comparison records from each object using the CHOIR function `getRecords`.

Step 3: Combine subtrees and standardize significance thresholds

Run the CHOIR function `combineTrees` on the complete set of records extracted in step 2. In doing so, the significance threshold will be standardized across all clustering trees, yielding a final set of clusters.

Supplementary Note 2.2: Description of each parameter used by CHOIR

The following text details each input parameter used by CHOIR and how changes to key parameters (indicated by *) affect the performance.

`alpha`* – A numerical value indicating the significance level used for permutation test comparisons of cluster distinguishability. Defaults to 0.05. Decreasing the alpha value will yield more conservative clusters (fewer clusters) and will often decrease the computational time required, because fewer cluster comparisons may be needed.

`ArchR_matrix` – For `ArchR` objects, a character string or vector indicating which matrix or matrices to use in the provided object. The default value, `NULL`, will use the "GeneScoreMatrix" for ATAC-seq data or the "GeneExpressionMatrix" for RNA-seq data. For multi-omic datasets, provide a vector with a value corresponding to each modality.

`ArchR_depthcol` – For `ArchR` objects, a character string or vector indicating which column to use for correlation with sequencing depth. The default value, `NULL`, will use the "nFrags" column for ATAC-seq data or the "Gex_nUMI" for RNA-seq data. For multi-omic datasets, provide a vector with a value corresponding to each provided value of `use_assay` or `ArchR_matrix` in the same order.

`atac` – A Boolean value or vector indicating whether the provided data is ATAC-seq data. For multi-omic datasets, provide a vector with a value corresponding to each provided value of `use_assay` or `ArchR_matrix`. Defaults to `FALSE`.

`batch_correction_method` – A character string indicating which batch correction method to use. Permitted values are "Harmony" and "none". Defaults to "none". Batch correction should only be used when the different batches are not expected to also have unique cell types or cell states. Using batch correction would ensure that clusters do not originate from a single batch, thereby making the final cluster calls more conservative.

batch_correction_params – A list of additional parameters to be passed to the selected batch correction method for each iteration. Only applicable when `batch_correction_method` is "Harmony".

batch_labels – A character string that, if applying batch correction, specifies the name of the column in the input object metadata containing the batch labels. Defaults to `NULL`.

cluster_params – A list of additional parameters to be passed to `Seurat` function `FindClusters` for clustering at each level of the tree. By default, when the `Seurat FindClusters` parameter `group.singletons` is set to `TRUE`, function `buildTree` relabels clusters such that each singleton constitutes its own cluster.

collect_all_metrics – A Boolean value indicating whether to collect and save additional metrics from the random forest classifiers, including feature importances for every comparison. Defaults to `FALSE`. Setting this parameter to `TRUE` will slightly increase the computational time required. This parameter has no effect on the final cluster calls.

countsplit* – A Boolean value indicating whether or not to use count split input data (see `countsplit` package), such that one matrix of counts is used for clustering tree generation and a separate matrix is used for all random forest classifier permutation testing. Defaults to `FALSE`. Enabling count splitting is likely to result in more conservative final cluster calls and is likely to perform best in datasets with high read depths.

countsplit_suffix – A character vector indicating the suffixes that distinguish the two count split matrices to be used. Suffixes are appended onto the input string/vector for parameter `use_slot` for `Seurat` objects, `use_assay` for `SingleCellExperiment` objects, or `ArchR_matrix` for `ArchR` objects. When count splitting is enabled, the default value `NULL` uses suffixes "_1" and "_2".

distance_approx* – A Boolean value indicating whether or not to use approximate distance calculations. Defaults to `TRUE`, which will use centroid-based distances. Setting distance approximation to

`FALSE` will substantially increase the computational time and memory required, particularly for large datasets. Using approximated distances (`TRUE`) rather than absolute distances (`FALSE`) is unlikely to have a meaningful effect on the distance thresholds imposed by CHOIR.

`distance_awareness`* – A numerical value representing the distance threshold above which a cluster will not merge with another cluster and significance testing will not be used. Specifically, this value is a multiplier applied to the distance between a cluster and its closest distinguishable neighbor based on random forest comparison. Defaults to 2, which sets this threshold at a two-fold increase in distance over the closest distinguishable neighbor. This threshold allows CHOIR to avoid running the full permutation test comparison for clusters that are highly likely to be distinct, saving computational time. To omit all distance calculations and perform permutation testing on all comparisons, set this parameter to `FALSE`. Setting this parameter to `FALSE` or increasing the input value will increase the number of permutation test comparisons run and, thus, the computational time. In rare cases, very small distant clusters may be erroneously merged when distance thresholds are not used. The intent of this parameter is only to avoid running permutation test comparisons between clusters that are so different that they should not be merged. We do not recommend decreasing this parameter value below 1.5, as lower values may result in instances of overclustering.

`downsampling_rate`* – A numerical value indicating the proportion of cells to be sampled per cluster to train/test each random forest classifier. For efficiency, the default value, "auto", sets the downsampling rate according to the dataset size. Decreasing this parameter may decrease the computational time required, but may also make the final cluster calls more conservative. If input is provided to both `downsampling_rate` and `sample_max` parameters, the minimum resulting cell number is calculated and used for each comparison.

`exclude_features`* – A character vector indicating features that should be excluded from input to the random forest classifier. Defaults to `NULL`, which means that no features will be excluded. This parameter can be used, for example, to exclude features correlated with cell quality, such as mitochondrial genes.

Failure to exclude problematic features could result in clusters driven by cell quality, while over-exclusion of features could reduce the ability of CHOIR to distinguish cell populations that differ by those features.

`feature_set`* – A string indicating whether to train random forest classifiers on "all" features or only variable ("var") features. Defaults to "var". Computational time and memory required may increase if more features are used. Using all features instead of variable features may result in more conservative cluster calls.

`key` – The name under which CHOIR-related data for this run is stored in the object. Defaults to "CHOIR".

`max_clusters`* – Indicates the extent to which the hierarchical clustering tree will be expanded. Defaults to "auto", which will expand the tree until cases of underclustering have been eliminated in all branches. Alternatively, supply a numerical value indicating the maximum number of clusters to which to the tree should be expanded. Using the default value for this parameter is highly recommended to avoid instances of underclustering. Setting a numerical value in this parameter hampers the ability of CHOIR to ensure that underclustering has not occurred.

`max_repeat_errors`* – A numerical value indicating the maximum number of such repeat errors that will be taken into account. This parameter is used to account for situations in which random forest classifier errors are concentrated among a few cells that are repeatedly misassigned. If set to 0, such repeat errors will not be evaluated. Defaults to 20. These situations are relatively infrequent, but setting this parameter to lower values (especially 0) may result in underclustering due to a small number of intermediate cells. Setting this parameter to higher values may lead to instances of overclustering and is not recommended.

`min_accuracy`* – A numerical value indicating the minimum accuracy required of the random forest classifier, below which clusters will be automatically merged. Defaults to 0.5, representing the random chance probability of assigning correct cluster labels; therefore, decreasing the minimum accuracy is not recommended. Increasing the minimum accuracy will lead to more conservative cluster assignments and will often decrease the computational time required, because fewer cluster comparisons may be needed.

`min_cluster_depth` – A numerical value indicating the maximum cluster size at the bottom of the clustering tree, prior to pruning branches. Defaults to 2,000. Increasing this parameter can cause a computational bottleneck when generating the initial clustering tree for some large datasets; therefore, the default value is recommended. However, changing this value is unlikely to have meaningful effects on the final cluster results.

`min_connections`* – A numerical value indicating the minimum number of nearest neighbors between two clusters for those clusters to be considered adjacent. Non-adjacent clusters will not be merged. Defaults to 1. This threshold allows CHOIR to avoid running the full permutation test comparison for clusters that are highly likely to be distinct, saving computational time. Therefore, setting this parameter to 0 will increase the number of permutation test comparisons run and, thus, the computational time. The intent of this parameter is only to avoid running permutation test comparisons between clusters that are so different that they should not be merged. Therefore, we do not recommend increasing this parameter value beyond 10, as higher values may result in instances of overclustering.

`n_cores` – A numerical value indicating the number of cores to use for parallelization. By default, CHOIR will use the number of available cores minus 2. CHOIR is parallelized at the computation of permutation test iterations. Therefore, any number of cores up to the number of iterations will theoretically decrease the computational time required. In practice, 8–16 cores are recommended for datasets up to 500,000 cells.

`n_iterations`* – A numerical value indicating the number of iterations run for each permutation test comparison. Increasing the number of iterations will approximately linearly increase the computational time required but provide a more accurate estimation of the significance of the permutation test. Decreasing the number of iterations runs the risk of leading to underclustering due to lack of statistical power. The default value, 100 iterations, was selected because it avoids underclustering, while minimizing computational time and the diminishing returns from running CHOIR with additional iterations.

`n_trees`* – A numerical value indicating the number of trees in each random forest. Defaults to 50. Increasing the number of trees is likely to increase the computational time required. Though not entirely

predictable, increasing the number of trees up to a point may enable more nuanced distinctions, but is likely to provide diminishing returns.

`n_var_features`* – A numerical value indicating how many variable features to identify. Defaults to 2,000 features for most data inputs, or 25,000 features for ATAC-seq data. Increasing the number of features may increase the computational time and memory required. If the provided value is either substantially higher or lower, instances of underclustering may occur.

`neighbor_params` – A list of additional parameters to be passed to `Seurat` function `FindNeighbors` (or, in the case of multi-modal data for `Seurat` or `SingleCellExperiment` objects, `Seurat` function `FindMultiModalNeighbors`).

`normalization_method` – A character string or vector indicating which normalization method to use. In general, input data should be supplied to CHOIR after normalization, except when the user wishes to use `Seurat SCTransform` normalization. Permitted values are "none" or "SCTransform". Defaults to "none". Because CHOIR has not been tested thoroughly with `SCTransform` normalization, we do not recommend this approach at this time. For multi-omic datasets, provide a vector with a value corresponding to each provided value of `use_assay` or `ArchR_matrix` in the same order.

`object` – CHOIR accepts data input objects of class `Seurat`, `SingleCellExperiment`, or `ArchRProject`. For multi-omic data, we recommend using `ArchRProject` objects.

`p_adjust`* – A string indicating which multiple comparison adjustment method to use. Permitted values are "bonferroni", "fdr", and "none". Defaults to "bonferroni". Other correction methods may be less conservative, identifying more clusters, as CHOIR applies filters that reduce the total number of tests performed.

`random_seed` – A numerical value indicating the random seed to be used. Defaults to 1. CHOIR uses randomization throughout the generation and pruning of the clustering tree. Therefore, changing the random seed may yield slight differences in the final cluster assignments.

reduction – An optional matrix of dimensionality reduction cell embeddings provided by the user for subsequent clustering steps. By default, this parameter is set to `NULL`, and the dimensionality reduction(s) will be calculated using the method specified by the `reduction_method` parameter.

reduction_method – A character string or vector indicating which dimensionality reduction method to use. Permitted values are "PCA" for principal component analysis, "LSI" for latent semantic indexing, and "IterativeLSI" for iterative latent semantic indexing. These three methods implement the `Seurat` function `RunPCA`, the `Signac` function `RunSVD`, and the `ArchR` function `addIterativeLSI`, respectively. The default value, `NULL`, will select a method based on the input data type, specifically "IterativeLSI" for `ArchR` objects, "LSI" for `Seurat` or `SingleCellExperiment` objects when parameter `atac` is `TRUE`, and "PCA" in all other cases. For multi-omic datasets, provide a vector with a value corresponding to each provided value of `use_assay` or `ArchR_matrix` in the same order.

reduction_params – A list of additional parameters to be passed to the selected dimensionality reduction method. By default, CHOIR will use the default parameter settings of the dimensionality reduction method indicated by the input to parameter `reduction_method`. Input to this parameter is passed to each downstream dimensionality reduction method and will overwrite or augment those defaults. Altering the performance of the dimensionality reduction in CHOIR will affect downstream clustering results, but not in ways that are predictable.

sample_max* – A numerical value indicating the maximum number of cells to be sampled per cluster to train/test each random forest classifier. Defaults to `Inf` (infinity), which does not cap the number of cells used, so all cells will be used in all comparisons. Decreasing this parameter may decrease the computational time required but may result in instances of underclustering. If input is provided to both the `downsampling_rate` and `sample_max` parameters, the minimum resulting cell number is calculated and used for each comparison.

subtree_reductions – A Boolean value indicating whether to generate a new dimensionality reduction and set of highly variable features for each subtree. Defaults to `TRUE`, which enables CHOIR to compare similar clusters using a more nuanced set of features. Setting this parameter to `FALSE` may decrease computational time but is likely to result in instances of underclustering.

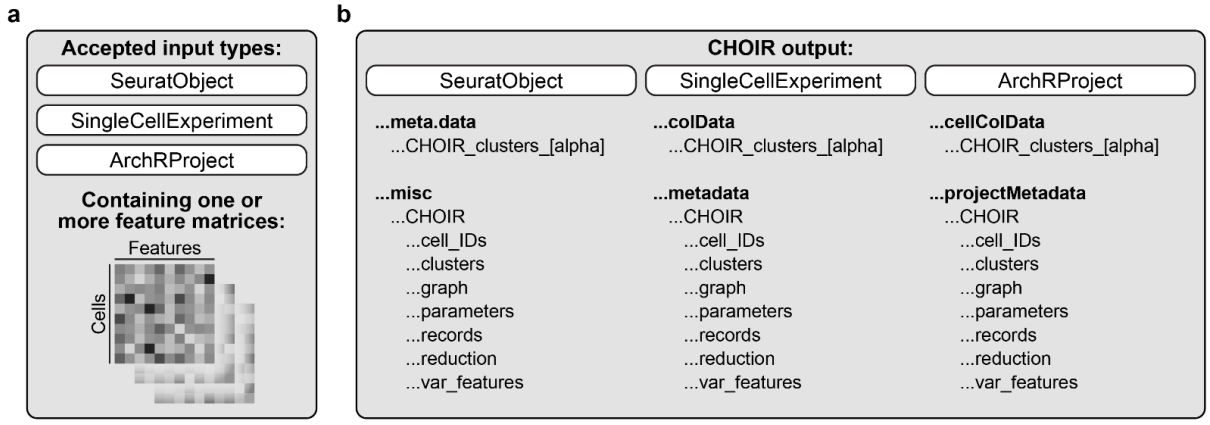
use_assay – For `Seurat` or `SingleCellExperiment` objects, a character string or vector indicating the assay(s) to use in the provided object. The default value, `NULL`, will choose the current active assay for `Seurat` objects and the "log_counts" assay for `SingleCellExperiment` objects.

use_slot – For `Seurat` objects, a character string or vector indicating the slot(s) to use in the provided object. The default value, `NULL`, will choose a slot based on the selected assay. If an assay other than "RNA", "sketch", "SCT", or "integrated" is provided, you must specify a value for `use_slot`.

use_variance* – A Boolean value indicating whether to use the variance of the random forest accuracy scores as part of the permutation test threshold. Defaults to `TRUE`. Setting this parameter to `FALSE` will make CHOIR considerably less conservative, identifying more clusters, particularly on large datasets.

var_features – An optional character vector of names of variable features to be used for subsequent clustering steps. By default, this parameter is set to `NULL`, and variable features will be calculated as part of running CHOIR. Input to this parameter is required when a dimensionality reduction is supplied to parameter `reduction`.

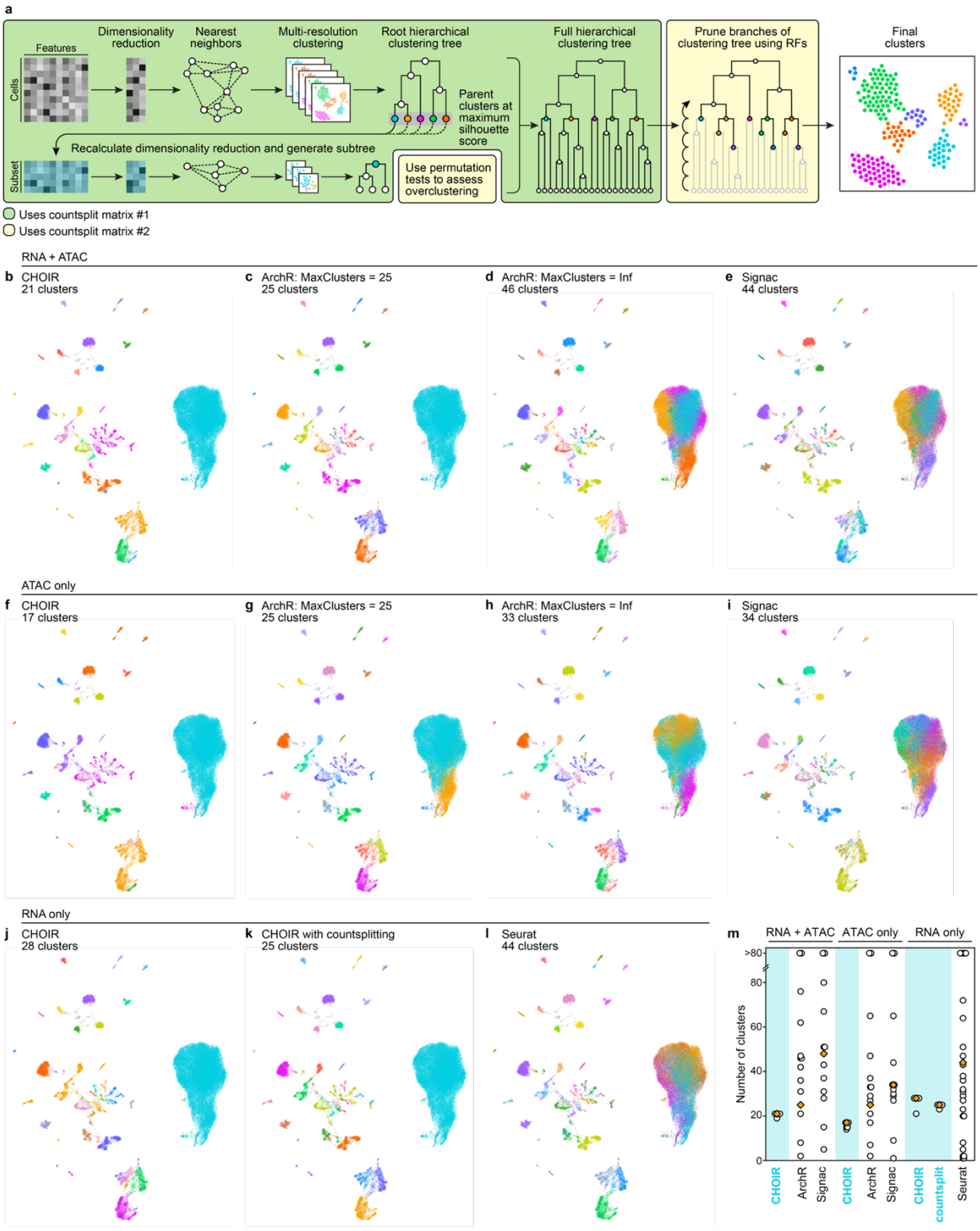
verbose – A Boolean value indicating whether to use verbose output during the execution of CHOIR. Defaults to `TRUE`, but can be set to `FALSE` for a cleaner output.



Supplementary Fig. 2.1: CHOIR provides compatibility for multiple data types, formats, and analysis outputs

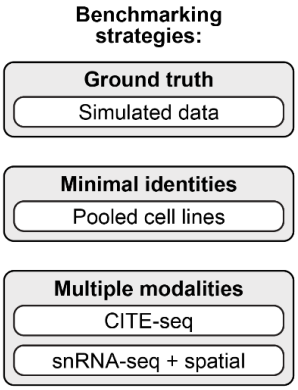
a. Schematic showing the types of input data accepted by CHOIR.

b. Schematic showing the CHOIR output locations by object type.

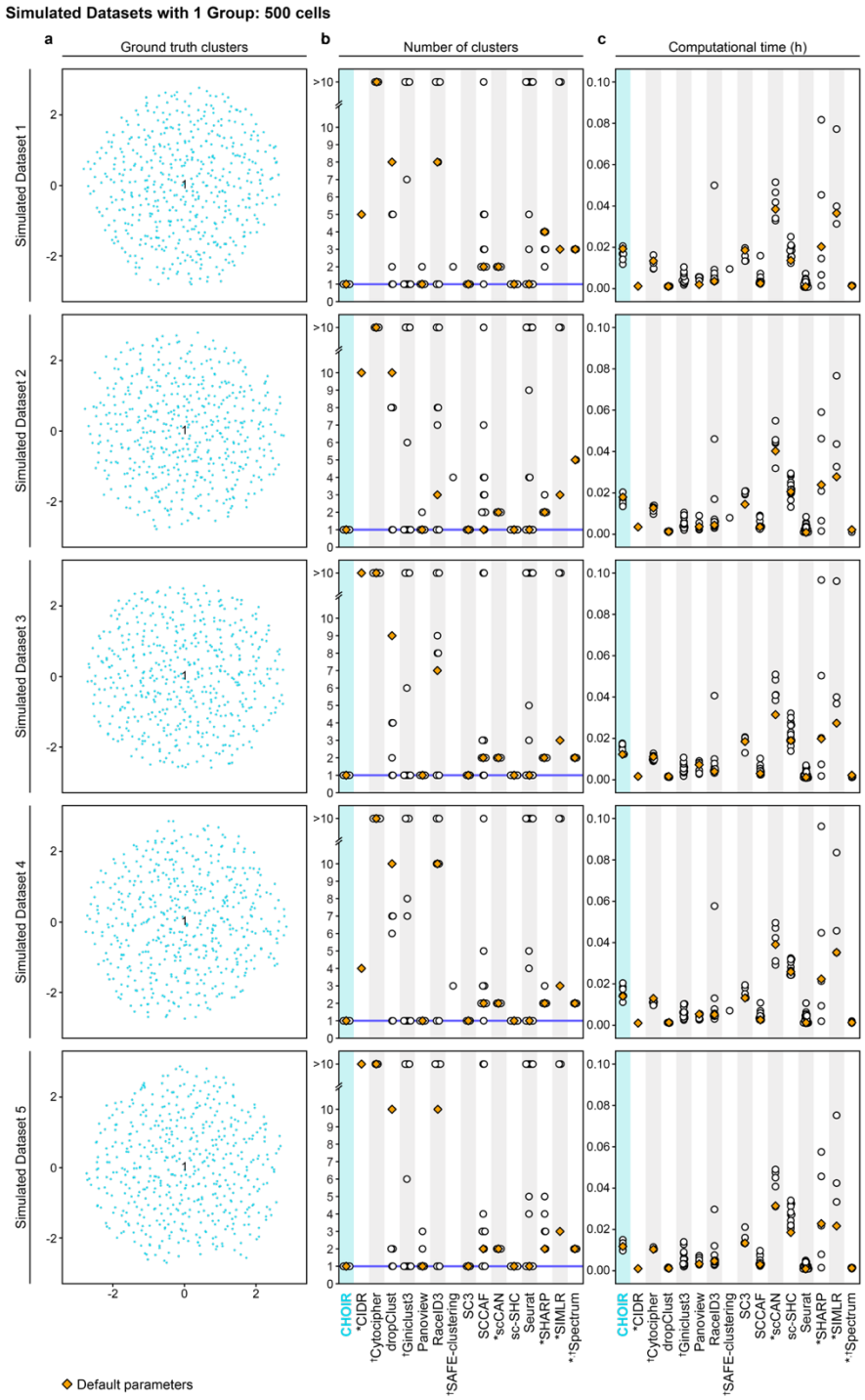


Supplementary Fig. 2.2: CHOIR is compatible with single- and multi-omic modalities
a. A schematic demonstrating how CHOIR constructs and iteratively prunes a hierarchical clustering tree using random forest classifiers (RFs) and statistical inference to prevent underclustering and overclustering, colored by which steps use separate count split matrices when count splitting is enabled. (Figure caption continued on the next page.)

(Figure caption continued from the previous page.) **b–i.** UMAP embedding colored according to the clusters identified by the default parameters of CHOIR (**b,f**), the default parameters of ArchR (**c,g**), the default parameters of ArchR without the maximum cluster number limitation (**d,h**), and the default parameters of Signac (**e,i**), with each method applied to the RNA and ATAC modalities together (**b–e**) or the ATAC modality only (**f–i**).
j–l. UMAP embedding colored according to the clusters identified by the default parameters of CHOIR (**j**), the default parameters of CHOIR with count splitting enabled (**k**), and the default parameters of Seurat (**l**), with each method applied to the RNA modality only.
m. Number of clusters identified using each parameter setting tested for each method on each selection of data modalities. ArchR, Signac, and Seurat all use Louvain or Leiden clustering as the core clustering method. ArchR and Signac were built for compatibility with scATAC-seq and multi-omic data, whereas Seurat was designed primarily for scRNA-seq data. Therefore, these methods were applied to the modalities for which they were designed. Count splitting was designed specifically for scRNA-seq data; therefore, it is shown for CHOIR applied to the RNA modality alone.



Supplementary Fig. 2.3: A three-tiered benchmarking approach was used to assess the performance of CHOIR relative to existing clustering methods

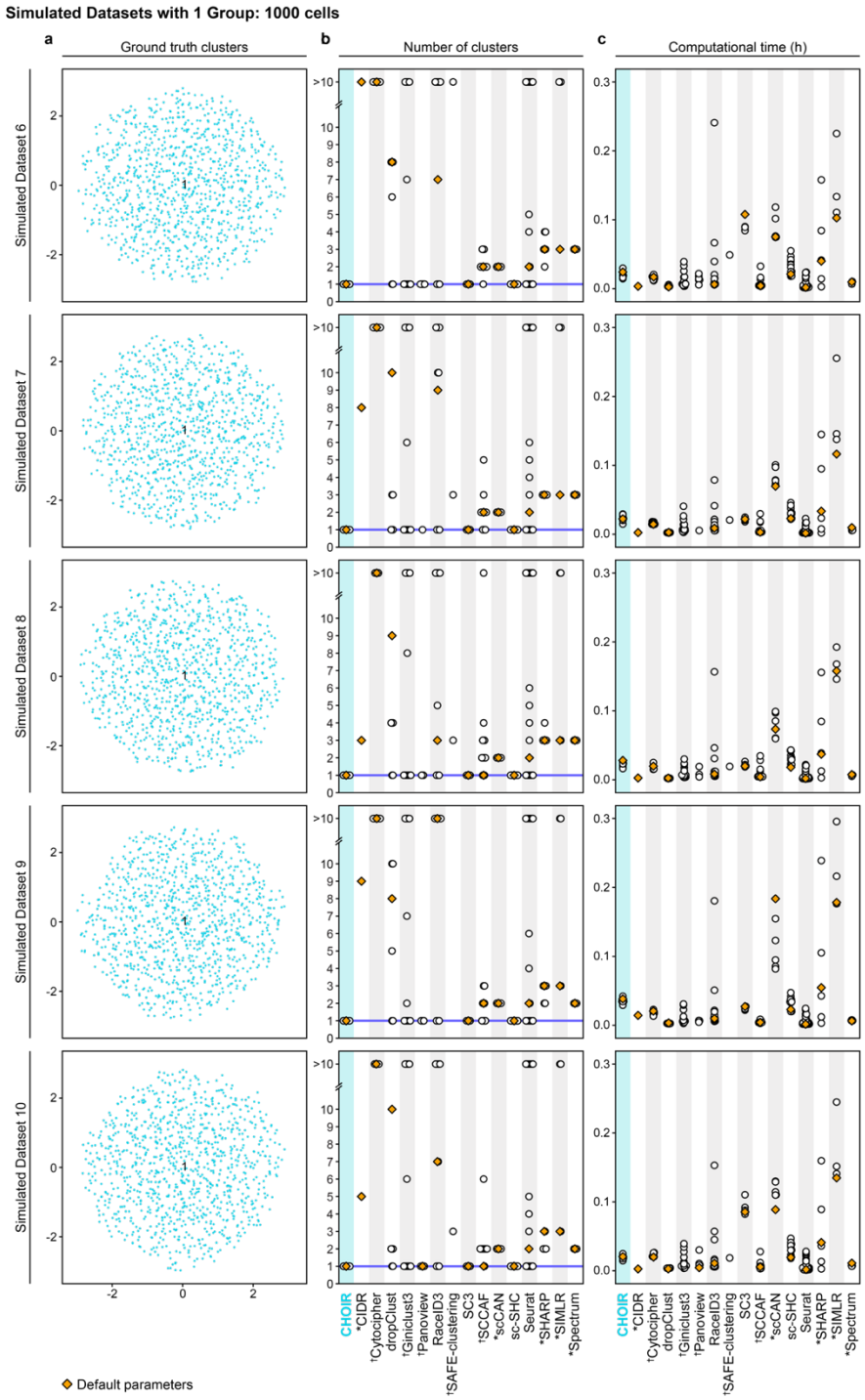


Supplementary Fig. 2.4: Detailed benchmarking results from simulated datasets with one ground truth group and 500 cells

a. UMAP embeddings of each dataset colored according to the ground truth groups.

b. Number of clusters identified using each parameter setting tested for each method. Symbols indicate clustering methods that *assumed ≥2 clusters, †had one or more parameter settings that failed to run, or ‡had one or more parameter settings that did not complete within the maximum allotted runtime of 96 hours for at least one dataset.

c. Computational time required by each parameter setting tested for each method.

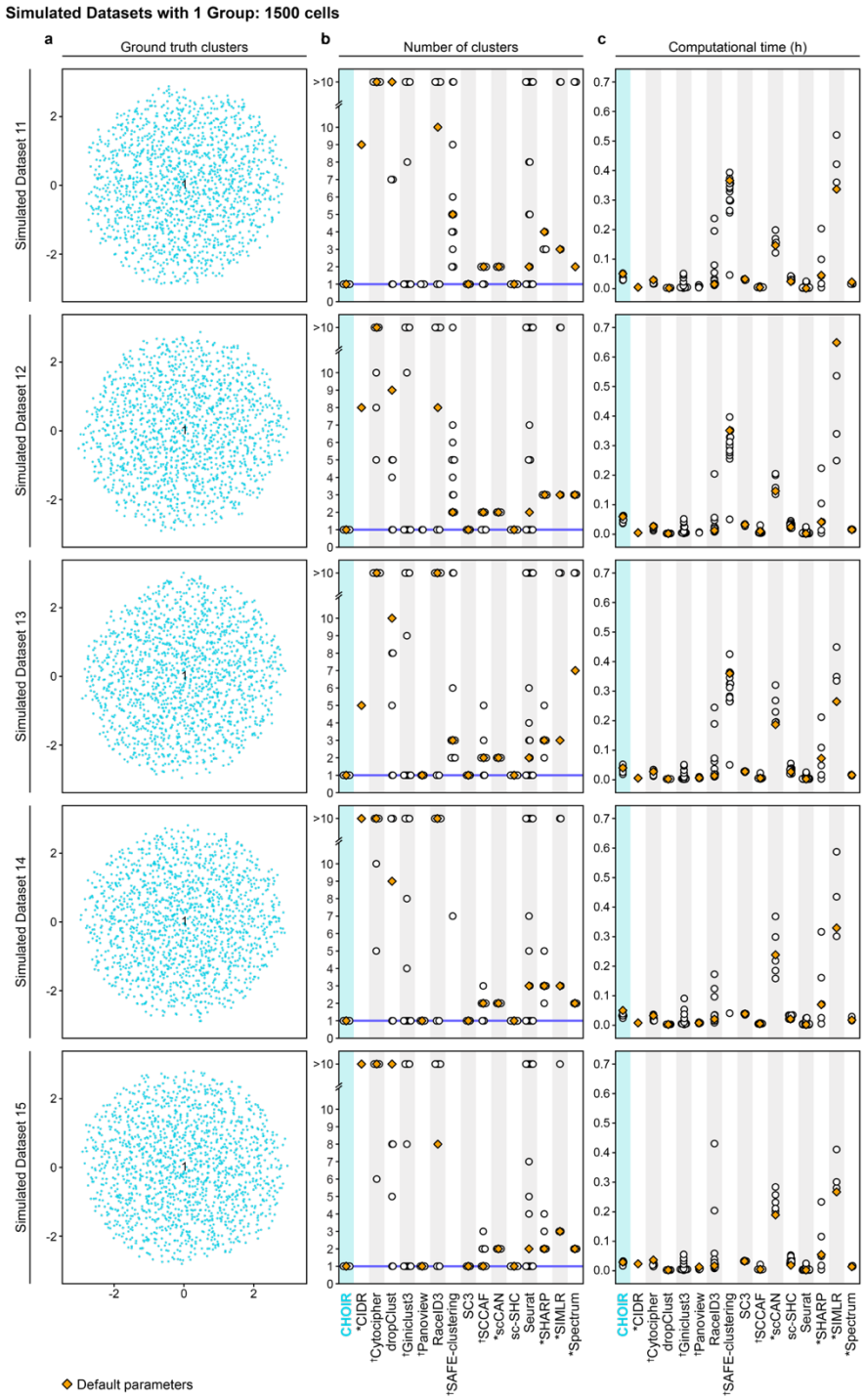


Supplementary Fig. 2.4: Detailed benchmarking results from simulated datasets with one ground truth group and 1000 cells

a. UMAP embeddings of each dataset colored according to the ground truth groups.

b. Number of clusters identified using each parameter setting tested for each method. Symbols indicate clustering methods that *assumed ≥2 clusters, †had one or more parameter settings that failed to run, or ‡had one or more parameter settings that did not complete within the maximum allotted runtime of 96 hours for at least one dataset.

c. Computational time required by each parameter setting tested for each method.

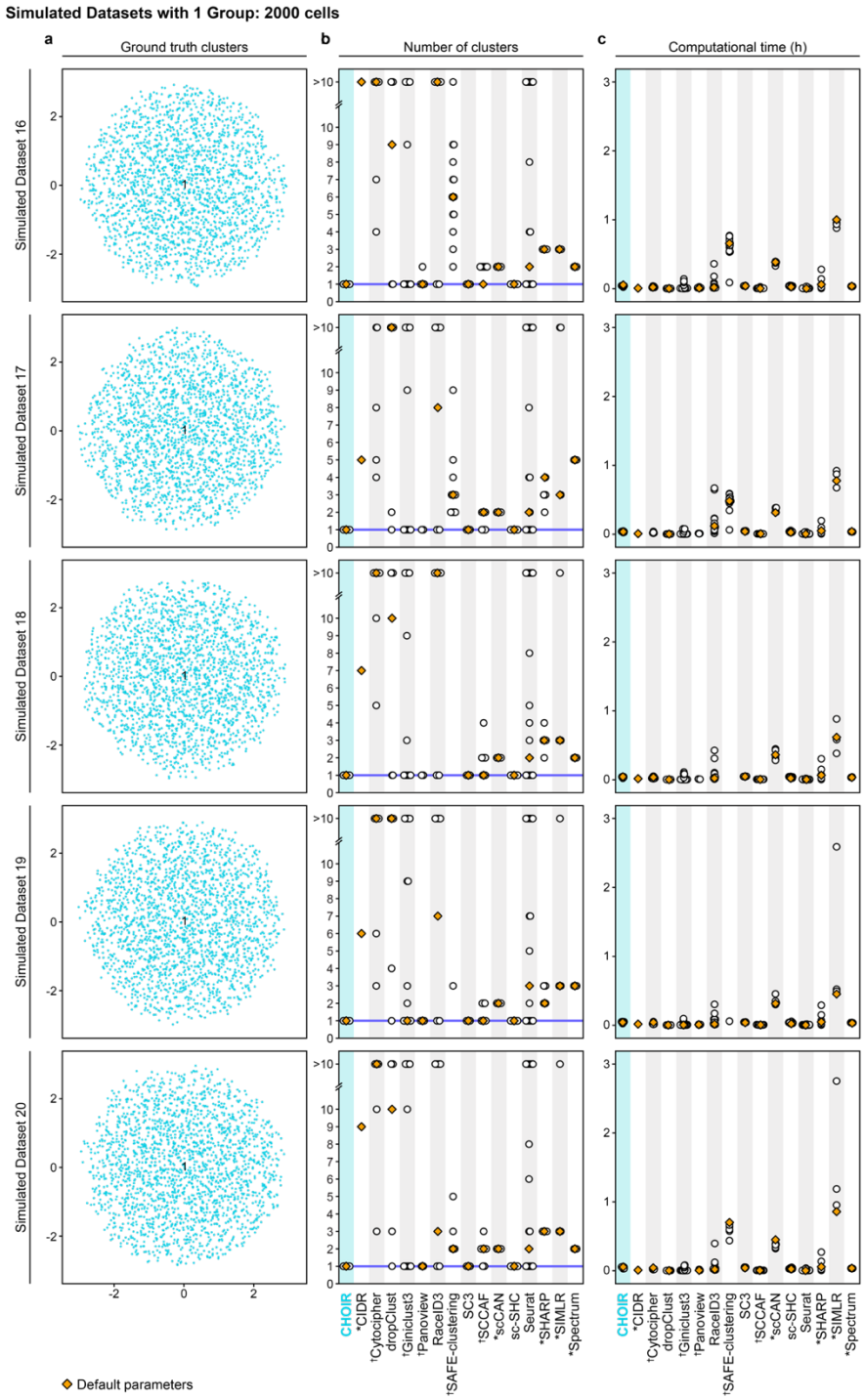


Supplementary Fig. 2.5: Detailed benchmarking results from simulated datasets with one ground truth group and 1500 cells

a. UMAP embeddings of each dataset colored according to the ground truth groups.

b. Number of clusters identified using each parameter setting tested for each method. Symbols indicate clustering methods that *assumed ≥2 clusters, †had one or more parameter settings that failed to run, or ‡had one or more parameter settings that did not complete within the maximum allotted runtime of 96 hours for at least one dataset.

c. Computational time required by each parameter setting tested for each method.

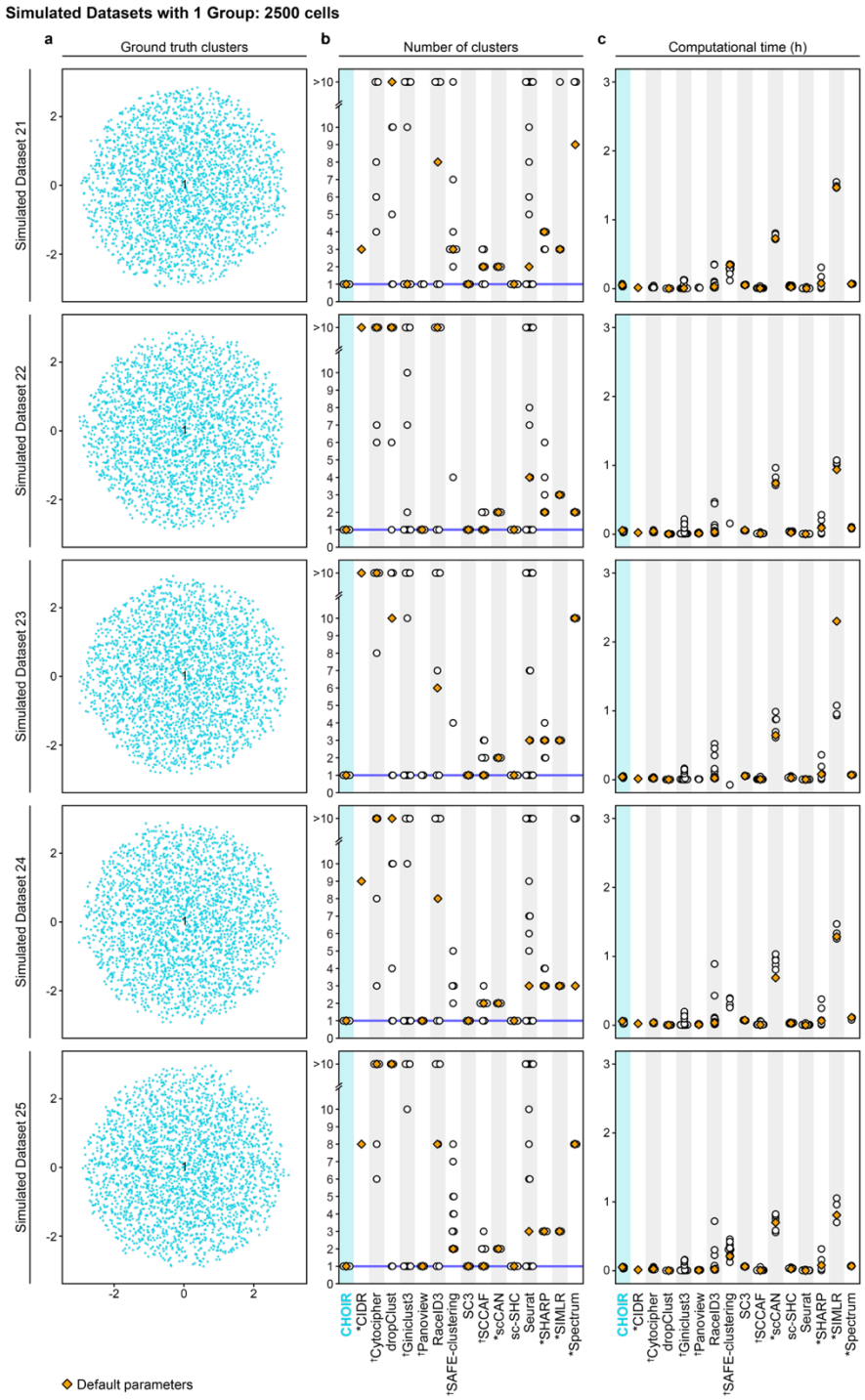


Supplementary Fig. 2.6: Detailed benchmarking results from simulated datasets with one ground truth group and 2000 cells

a. UMAP embeddings of each dataset colored according to the ground truth groups.

b. Number of clusters identified using each parameter setting tested for each method. Symbols indicate clustering methods that *assumed ≥2 clusters, †had one or more parameter settings that failed to run, or ‡had one or more parameter settings that did not complete within the maximum allotted runtime of 96 hours for at least one dataset.

c. Computational time required by each parameter setting tested for each method.



Supplementary Fig. 2.7: Detailed benchmarking results from simulated datasets with one ground truth group and 2500 cells

a. UMAP embeddings of each dataset colored according to the ground truth groups.

b. Number of clusters identified using each parameter setting tested for each method. Symbols indicate clustering methods that *assumed ≥2 clusters, †had one or more parameter settings that failed to run, or ‡had one or more parameter settings that did not complete within the maximum allotted runtime of 96 hours for at least one dataset.

c. Computational time required by each parameter setting tested for each method.



Supplementary Fig. 2.9: Detailed benchmarking results from simulated datasets with 5 ground truth groups and 5000 cells

a. UMAP embeddings of each dataset colored according to the ground truth groups.

b. Adjusted Rand Index resulting from each parameter setting tested for each method. Symbols indicate clustering methods for which one or more parameter settings †failed to run or ‡did not complete within the maximum allotted runtime of 96 hours for at least one dataset.

c. Number of clusters identified by each parameter setting tested for each method.

d. Computational time required by each parameter setting tested for each method.



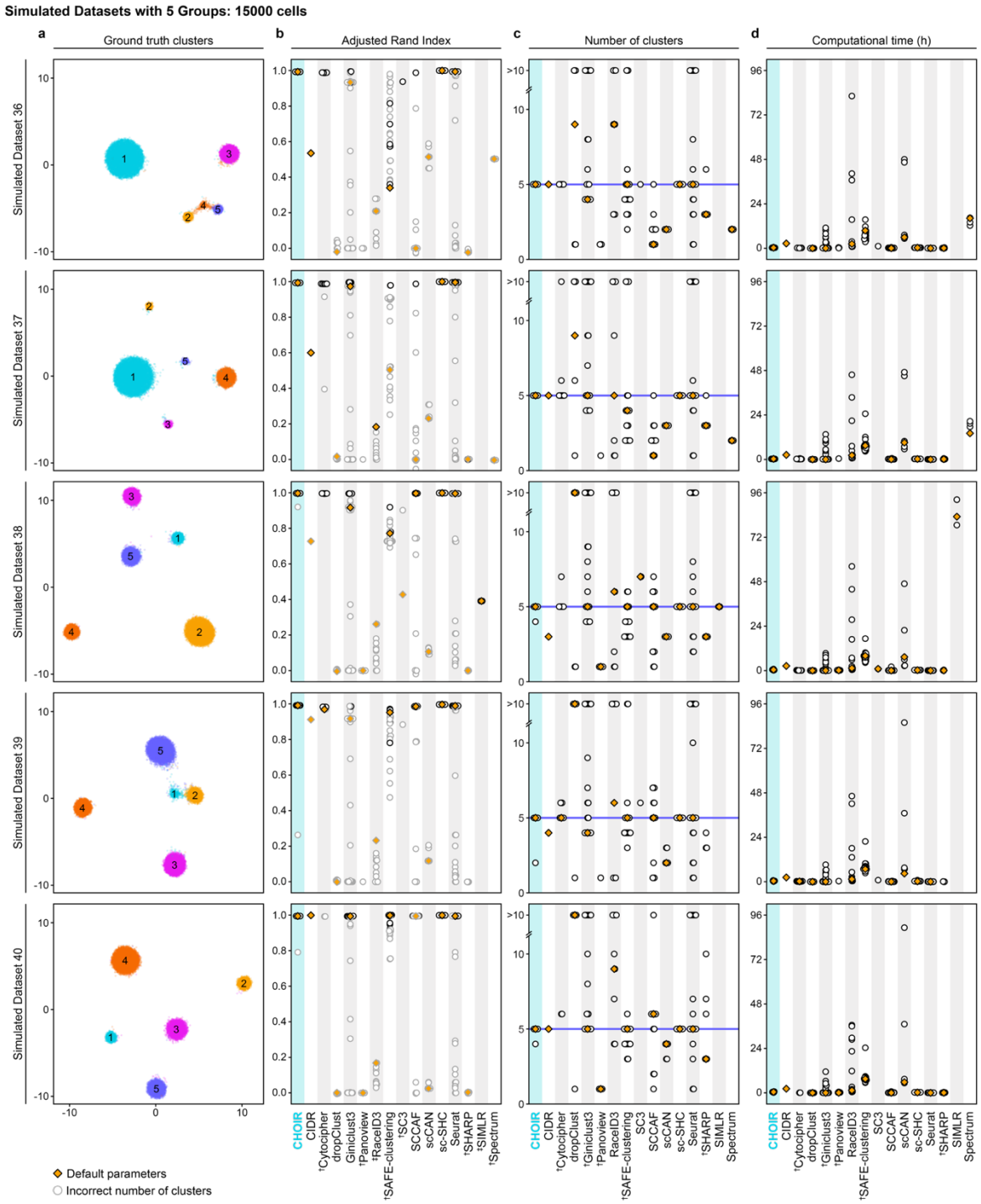
Supplementary Fig. 2.8: Detailed benchmarking results from simulated datasets with 5 ground truth groups and 10000 cells

a. UMAP embeddings of each dataset colored according to the ground truth groups.

b. Adjusted Rand Index resulting from each parameter setting tested for each method. Symbols indicate clustering methods for which one or more parameter settings †failed to run or ‡did not complete within the maximum allotted runtime of 96 hours for at least one dataset.

c. Number of clusters identified by each parameter setting tested for each method.

d. Computational time required by each parameter setting tested for each method.



Supplementary Fig. 2.9: Detailed benchmarking results from simulated datasets with 5 ground truth groups and 15000 cells

a. UMAP embeddings of each dataset colored according to the ground truth groups.

b. Adjusted Rand Index resulting from each parameter setting tested for each method. Symbols indicate clustering methods for which one or more parameter settings †failed to run or ‡did not complete within the maximum allotted runtime of 96 hours for at least one dataset.

c. Number of clusters identified by each parameter setting tested for each method.

d. Computational time required by each parameter setting tested for each method.



Supplementary Fig. 2.10: Detailed benchmarking results from simulated datasets with 5 ground truth groups and 20000 cells

a. UMAP embeddings of each dataset colored according to the ground truth groups.

b. Adjusted Rand Index resulting from each parameter setting tested for each method. Symbols indicate clustering methods for which one or more parameter settings †failed to run or ‡did not complete within the maximum allotted runtime of 96 hours for at least one dataset.

c. Number of clusters identified by each parameter setting tested for each method.

d. Computational time required by each parameter setting tested for each method.



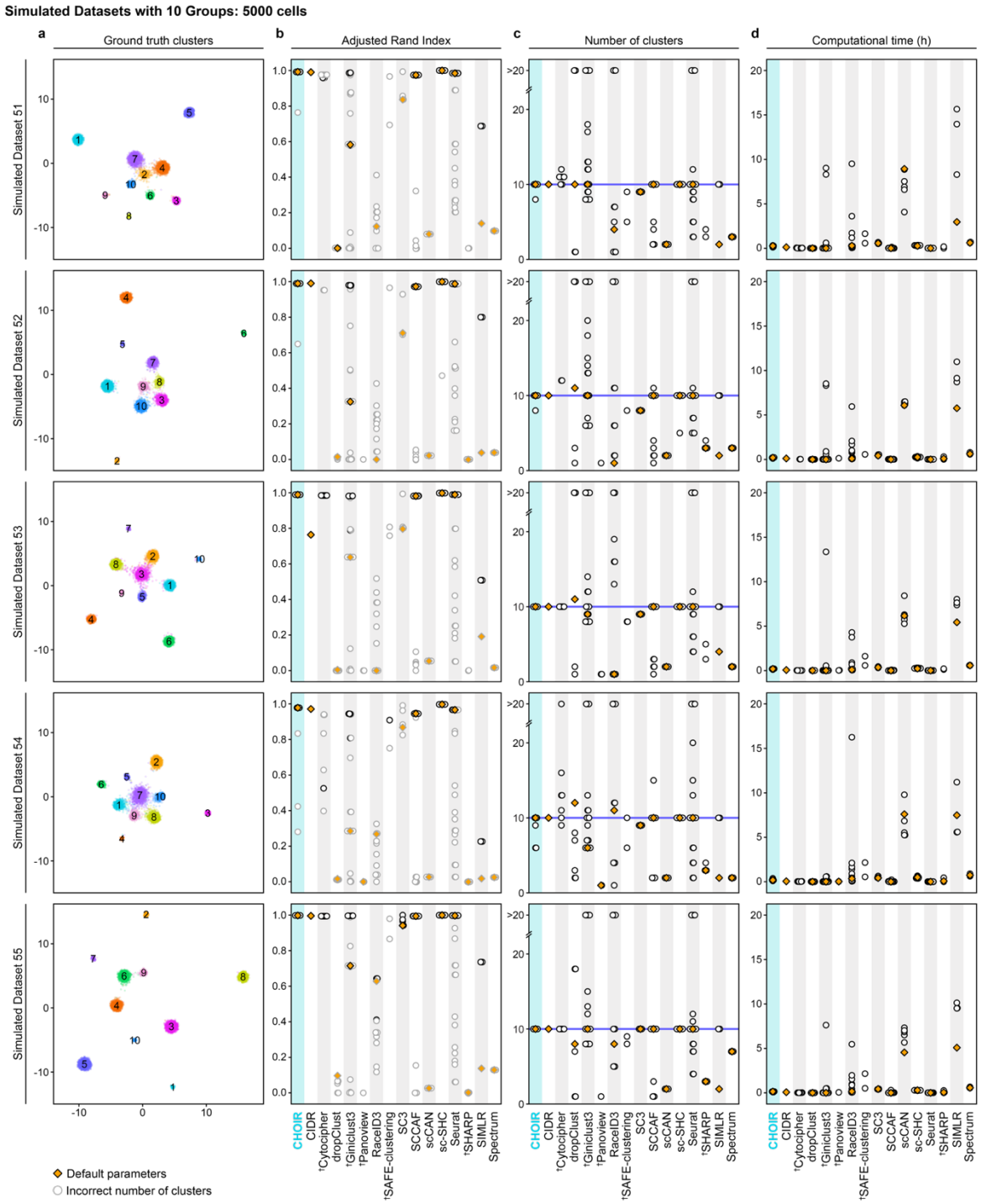
Supplementary Fig. 2.11: Detailed benchmarking results from simulated datasets with 5 ground truth groups and 25000 cells

a. UMAP embeddings of each dataset colored according to the ground truth groups.

b. Adjusted Rand Index resulting from each parameter setting tested for each method. Symbols indicate clustering methods for which one or more parameter settings †failed to run or ‡did not complete within the maximum allotted runtime of 96 hours for at least one dataset.

c. Number of clusters identified by each parameter setting tested for each method.

d. Computational time required by each parameter setting tested for each method.



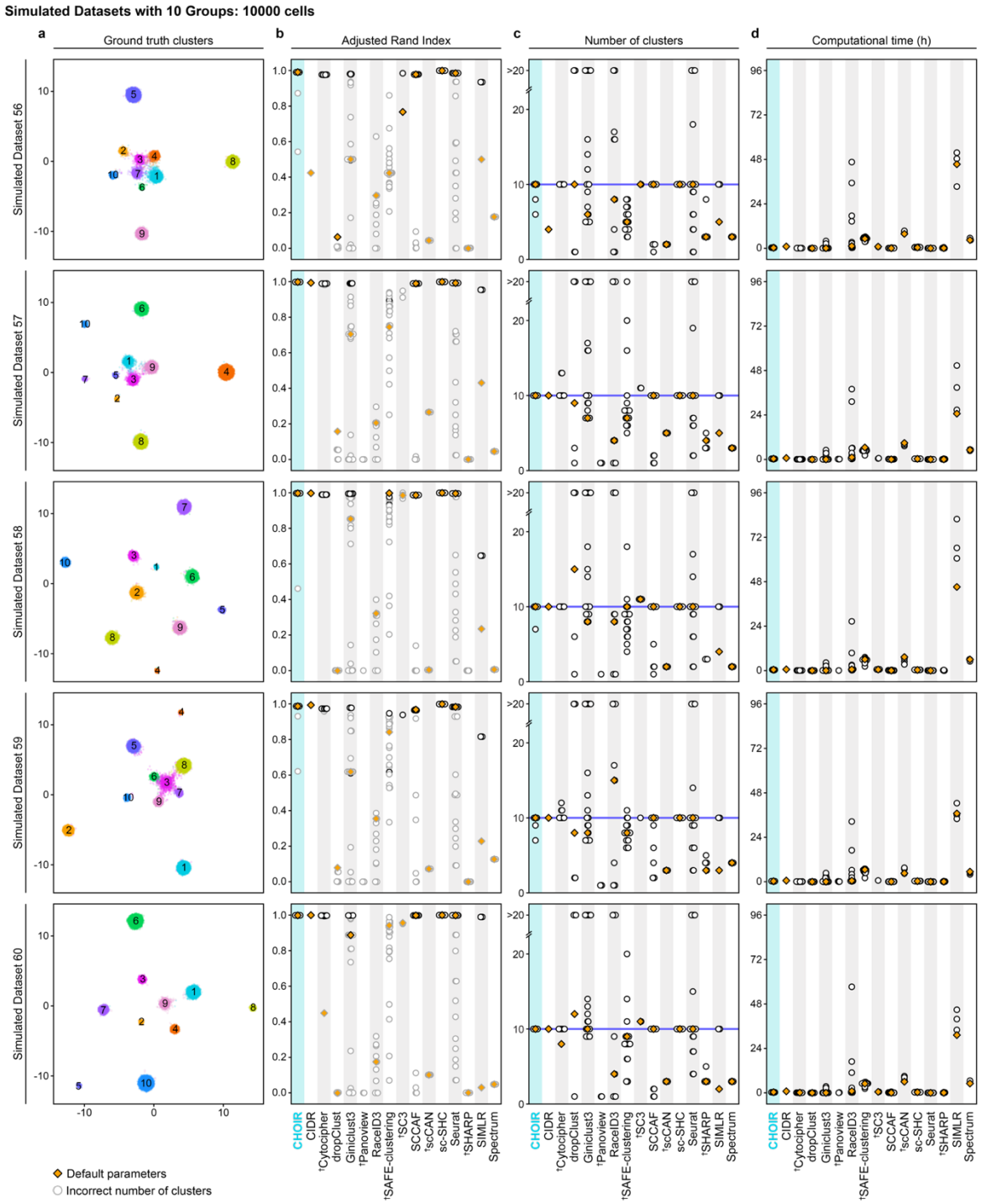
Supplementary Fig. 2.12: Detailed benchmarking results from simulated datasets with 10 ground truth groups and 5000 cells

a. UMAP embeddings of each dataset colored according to the ground truth groups.

b. Adjusted Rand Index resulting from each parameter setting tested for each method. Symbols indicate clustering methods for which one or more parameter settings †failed to run or ‡did not complete within the maximum allotted runtime of 96 hours for at least one dataset.

c. Number of clusters identified by each parameter setting tested for each method.

d. Computational time required by each parameter setting tested for each method.



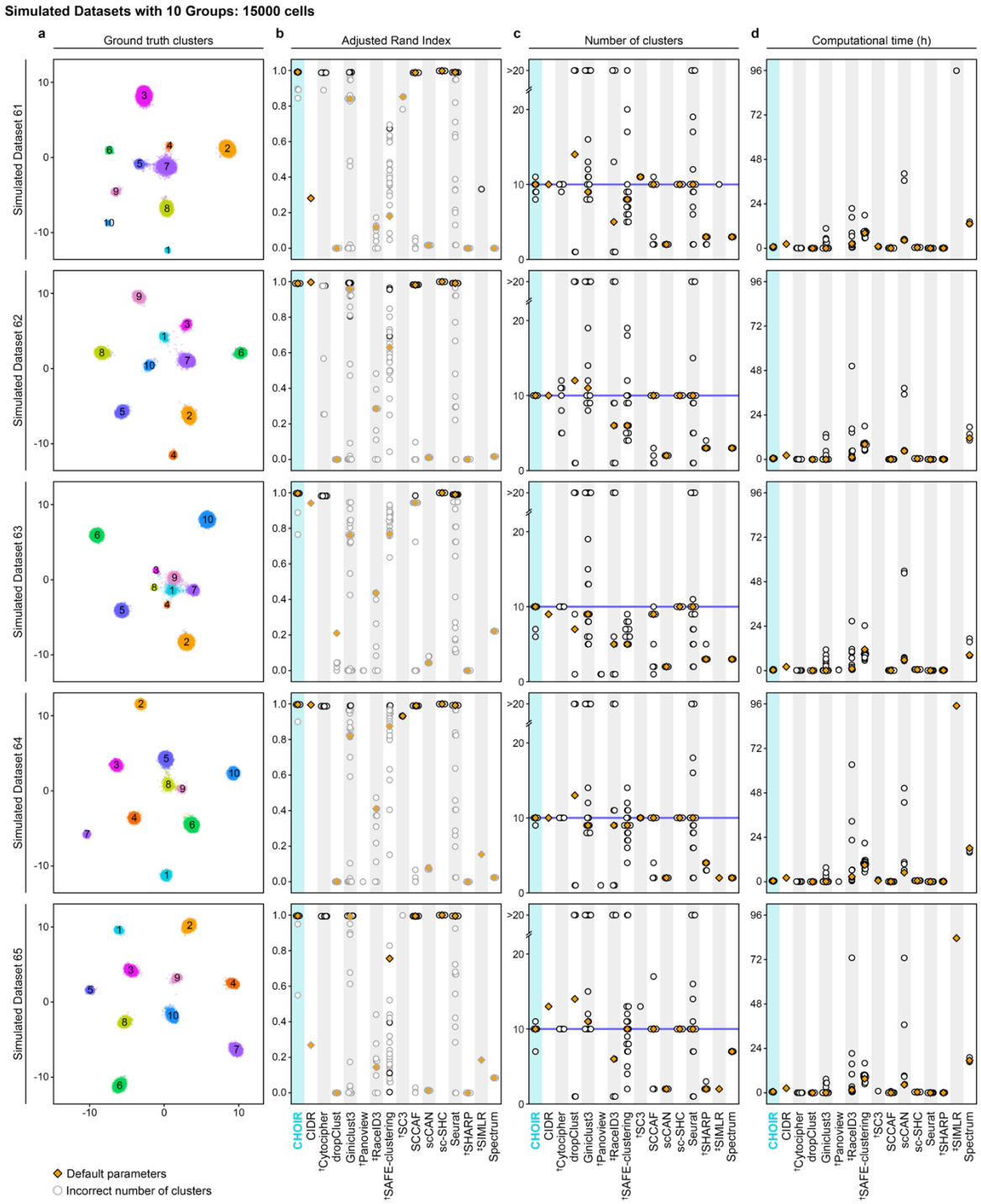
Supplementary Fig. 2.13: Detailed benchmarking results from simulated datasets with 10 ground truth groups and 10000 cells

a. UMAP embeddings of each dataset colored according to the ground truth groups.

b. Adjusted Rand Index resulting from each parameter setting tested for each method. Symbols indicate clustering methods for which one or more parameter settings †failed to run or ‡did not complete within the maximum allotted runtime of 96 hours for at least one dataset.

c. Number of clusters identified by each parameter setting tested for each method.

d. Computational time required by each parameter setting tested for each method.



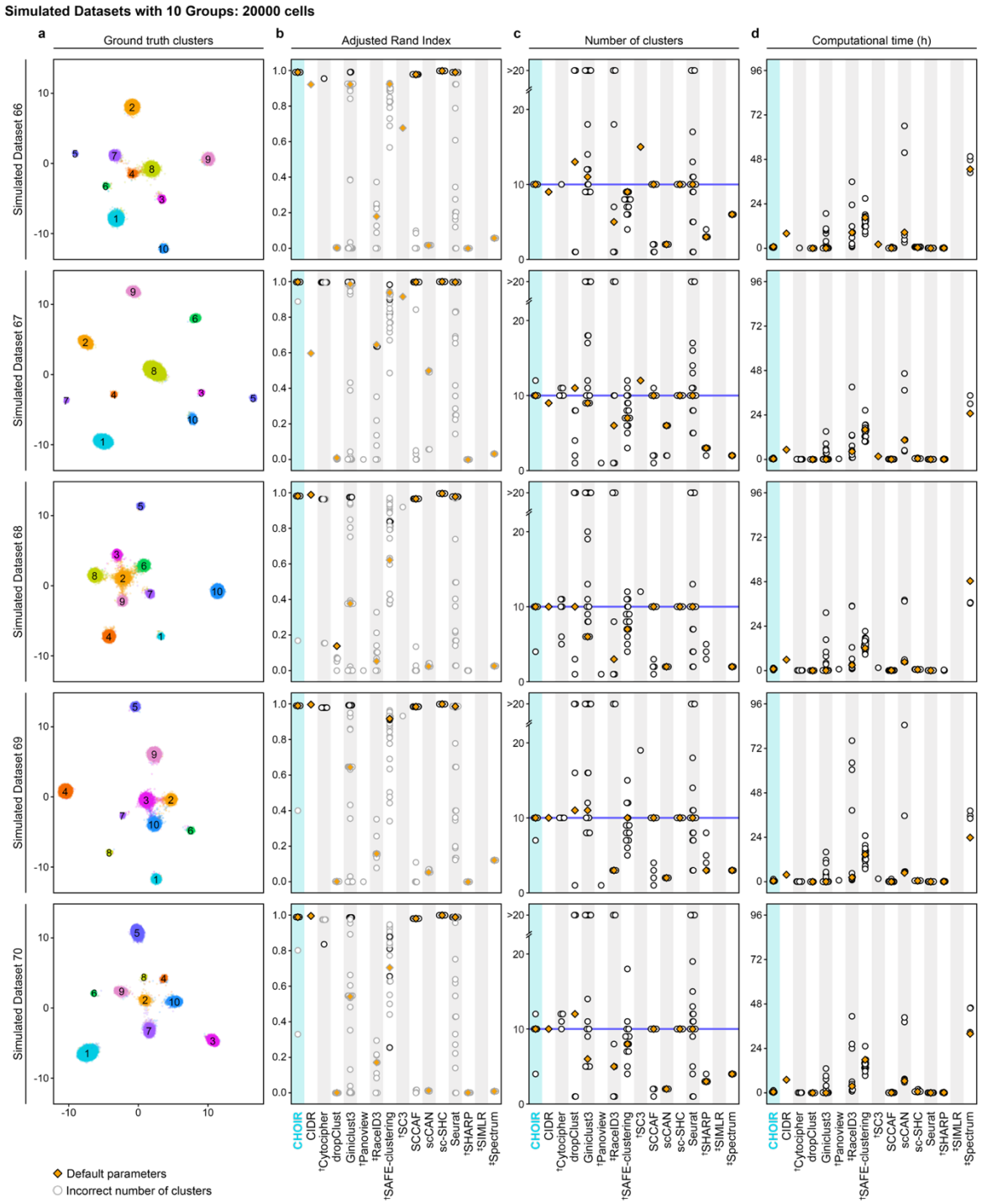
Supplementary Fig. 2.14: Detailed benchmarking results from simulated datasets with 10 ground truth groups and 15000 cells

a. UMAP embeddings of each dataset colored according to the ground truth groups.

b. Adjusted Rand Index resulting from each parameter setting tested for each method. Symbols indicate clustering methods for which one or more parameter settings †failed to run or ‡did not complete within the maximum allotted runtime of 96 hours for at least one dataset.

c. Number of clusters identified by each parameter setting tested for each method.

d. Computational time required by each parameter setting tested for each method.



Supplementary Fig. 2.15: Detailed benchmarking results from simulated datasets with 10 ground truth groups and 20000 cells

a. UMAP embeddings of each dataset colored according to the ground truth groups.

b. Adjusted Rand Index resulting from each parameter setting tested for each method. Symbols indicate clustering methods for which one or more parameter settings †failed to run or ‡did not complete within the maximum allotted runtime of 96 hours for at least one dataset.

c. Number of clusters identified by each parameter setting tested for each method.

d. Computational time required by each parameter setting tested for each method.



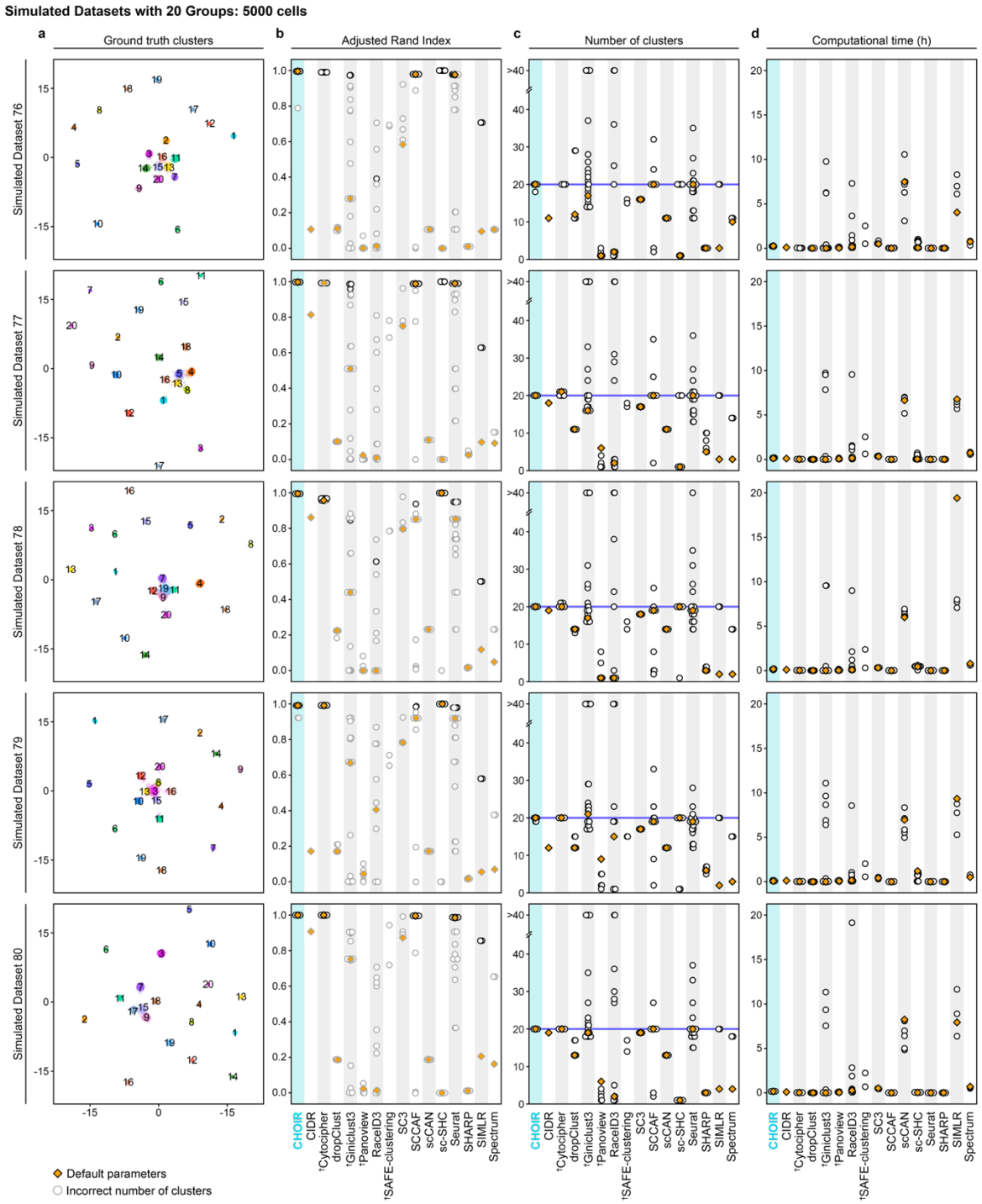
Supplementary Fig. 2.16: Detailed benchmarking results from simulated datasets with 10 ground truth groups and 25000 cells

a. UMAP embeddings of each dataset colored according to the ground truth groups.

b. Adjusted Rand Index resulting from each parameter setting tested for each method. Symbols indicate clustering methods for which one or more parameter settings †failed to run or ‡did not complete within the maximum allotted runtime of 96 hours for at least one dataset.

c. Number of clusters identified by each parameter setting tested for each method.

d. Computational time required by each parameter setting tested for each method.



Supplementary Fig. 2.17: Detailed benchmarking results from simulated datasets with 20 ground truth groups and 5000 cells

a. UMAP embeddings of each dataset colored according to the ground truth groups.

b. Adjusted Rand Index resulting from each parameter setting tested for each method. Symbols indicate clustering methods for which one or more parameter settings †failed to run or ‡did not complete within the maximum allotted runtime of 96 hours for at least one dataset.

c. Number of clusters identified by each parameter setting tested for each method.

d. Computational time required by each parameter setting tested for each method.



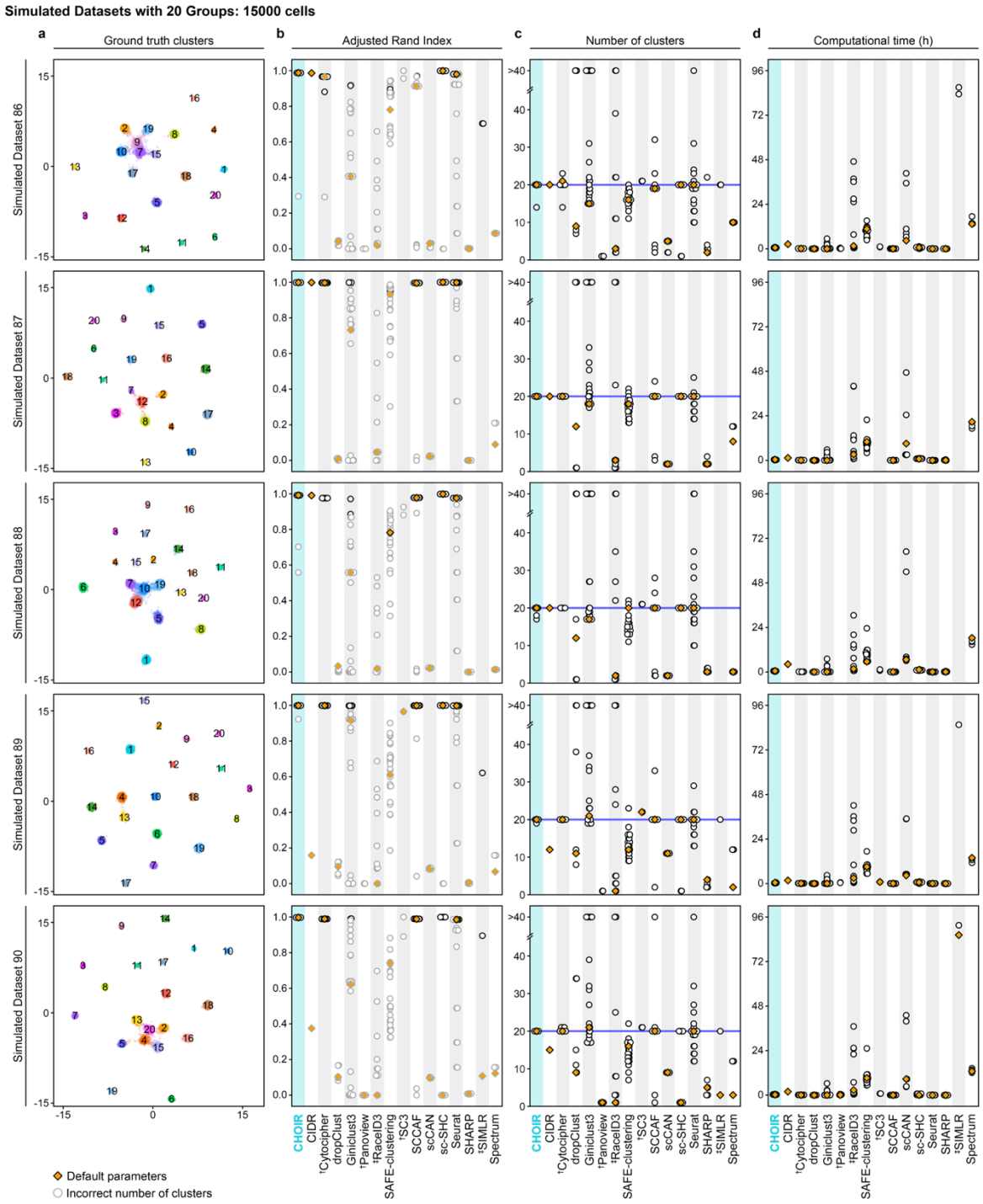
Supplementary Fig. 2.18: Detailed benchmarking results from simulated datasets with 20 ground truth groups and 10000 cells

a. UMAP embeddings of each dataset colored according to the ground truth groups.

b. Adjusted Rand Index resulting from each parameter setting tested for each method. Symbols indicate clustering methods for which one or more parameter settings †failed to run or ‡did not complete within the maximum allotted runtime of 96 hours for at least one dataset.

c. Number of clusters identified by each parameter setting tested for each method.

d. Computational time required by each parameter setting tested for each method.



Supplementary Fig. 2.19: Detailed benchmarking results from simulated datasets with 20 ground truth groups and 15000 cells

a. UMAP embeddings of each dataset colored according to the ground truth groups.

b. Adjusted Rand Index resulting from each parameter setting tested for each method. Symbols indicate clustering methods for which one or more parameter settings †failed to run or ‡did not complete within the maximum allotted runtime of 96 hours for at least one dataset.

c. Number of clusters identified by each parameter setting tested for each method.

d. Computational time required by each parameter setting tested for each method.



Supplementary Fig. 2.20: Detailed benchmarking results from simulated datasets with 20 ground truth groups and 20000 cells

a. UMAP embeddings of each dataset colored according to the ground truth groups.

b. Adjusted Rand Index resulting from each parameter setting tested for each method. Symbols indicate clustering methods for which one or more parameter settings †failed to run or ‡did not complete within the maximum allotted runtime of 96 hours for at least one dataset.

c. Number of clusters identified by each parameter setting tested for each method.

d. Computational time required by each parameter setting tested for each method.



Supplementary Fig. 2.21: Detailed benchmarking results from simulated datasets with 20 ground truth groups and 25000 cells

a. UMAP embeddings of each dataset colored according to the ground truth groups.

b. Adjusted Rand Index resulting from each parameter setting tested for each method. Symbols indicate clustering methods for which one or more parameter settings †failed to run or ‡did not complete within the maximum allotted runtime of 96 hours for at least one dataset.

c. Number of clusters identified by each parameter setting tested for each method.

d. Computational time required by each parameter setting tested for each method.



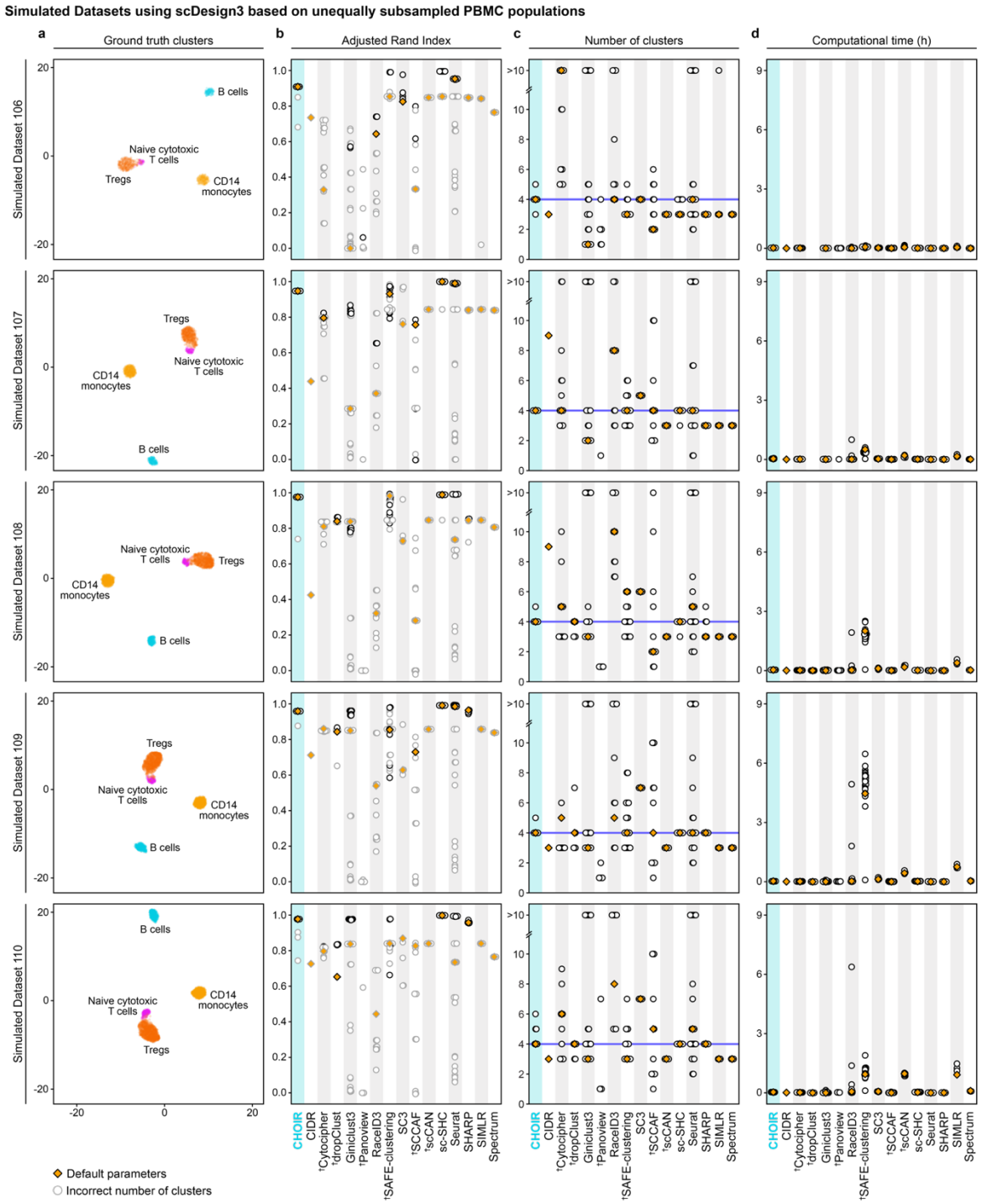
Supplementary Fig. 2.22: Benchmarking results from simulated datasets generated using scDesign3 based on equally subsampled PBMC populations

a. UMAP embeddings of each dataset colored according to the ground truth groups.

b. Adjusted Rand Index obtained using each parameter setting tested for each method. †Clustering methods for which one or more parameter settings failed to run for at least one dataset.

c. Number of clusters identified using each parameter setting tested for each method.

d. Computational time required by each parameter setting tested for each method.



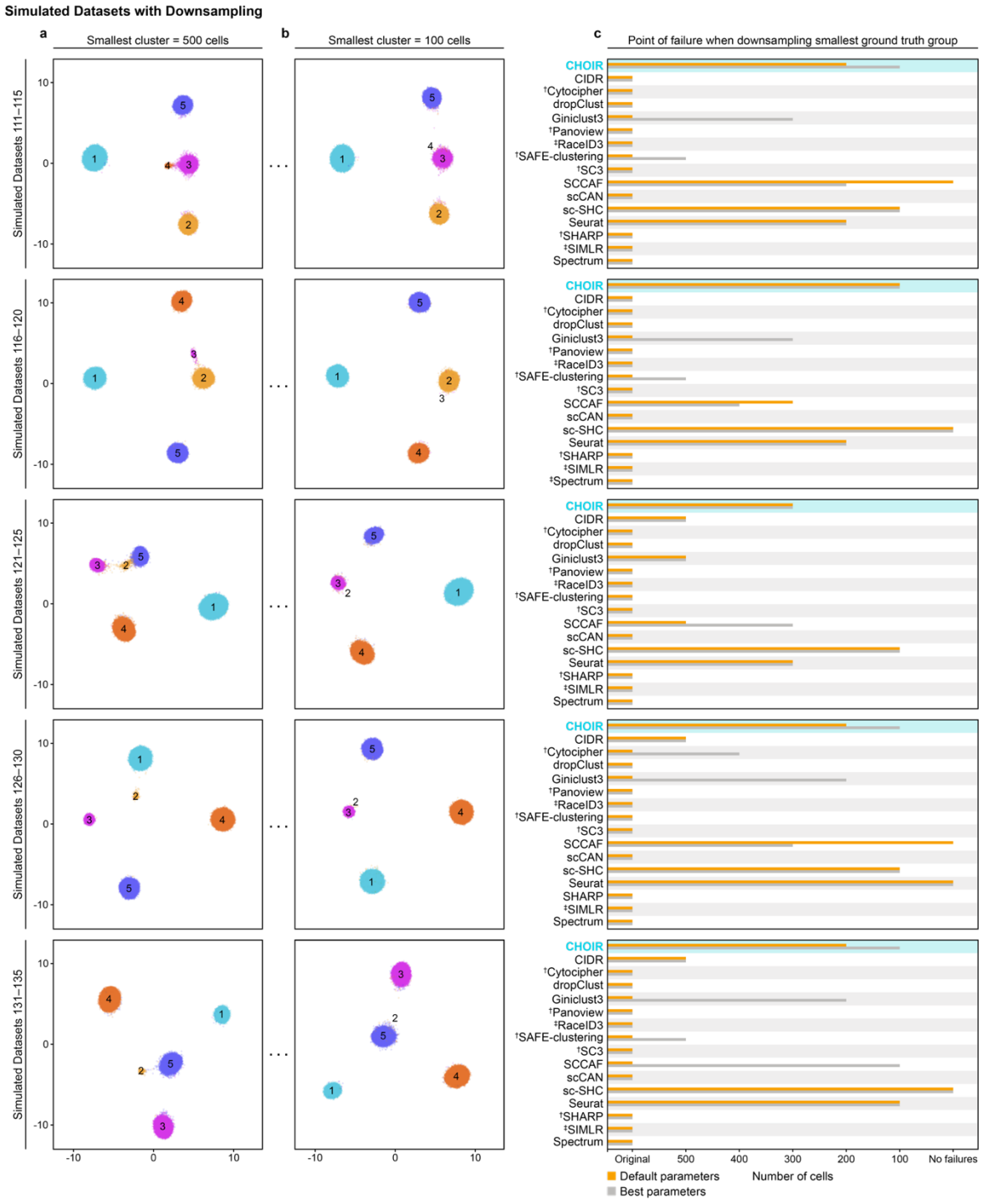
Supplementary Fig. 2.23: Benchmarking results from simulated datasets generated using scDesign3 based on unequally subsampled PBMC populations

a. UMAP embeddings of each dataset colored according to the ground truth groups.

b. Adjusted Rand Index obtained using each parameter setting tested for each method. †Clustering methods for which one or more parameter settings failed to run for at least one dataset.

c. Number of clusters identified using each parameter setting tested for each method.

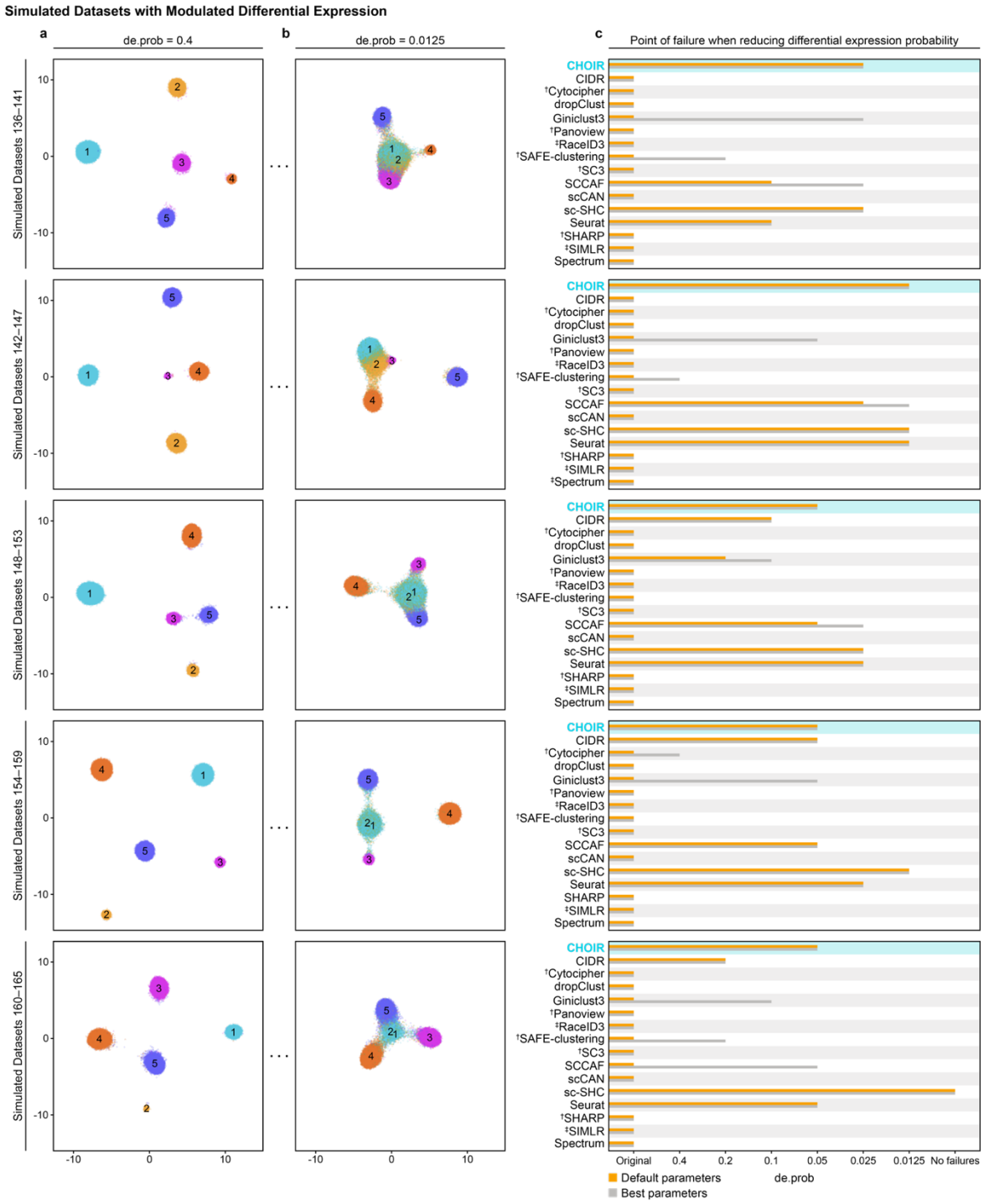
d. Computational time required by each parameter setting tested for each method.



Supplementary Fig. 2.24: Benchmarking results from simulated datasets with incremental downsampling of smallest ground truth group

a–b. UMAP embeddings of each dataset colored according to the ground truth groups, when the smallest ground truth group was downsampled to the first value of 500 cells (**a**) and the final value of 100 cells (**b**). These datasets are modifications of Simulated Datasets 46–50. (Figure caption continued on the next page.)

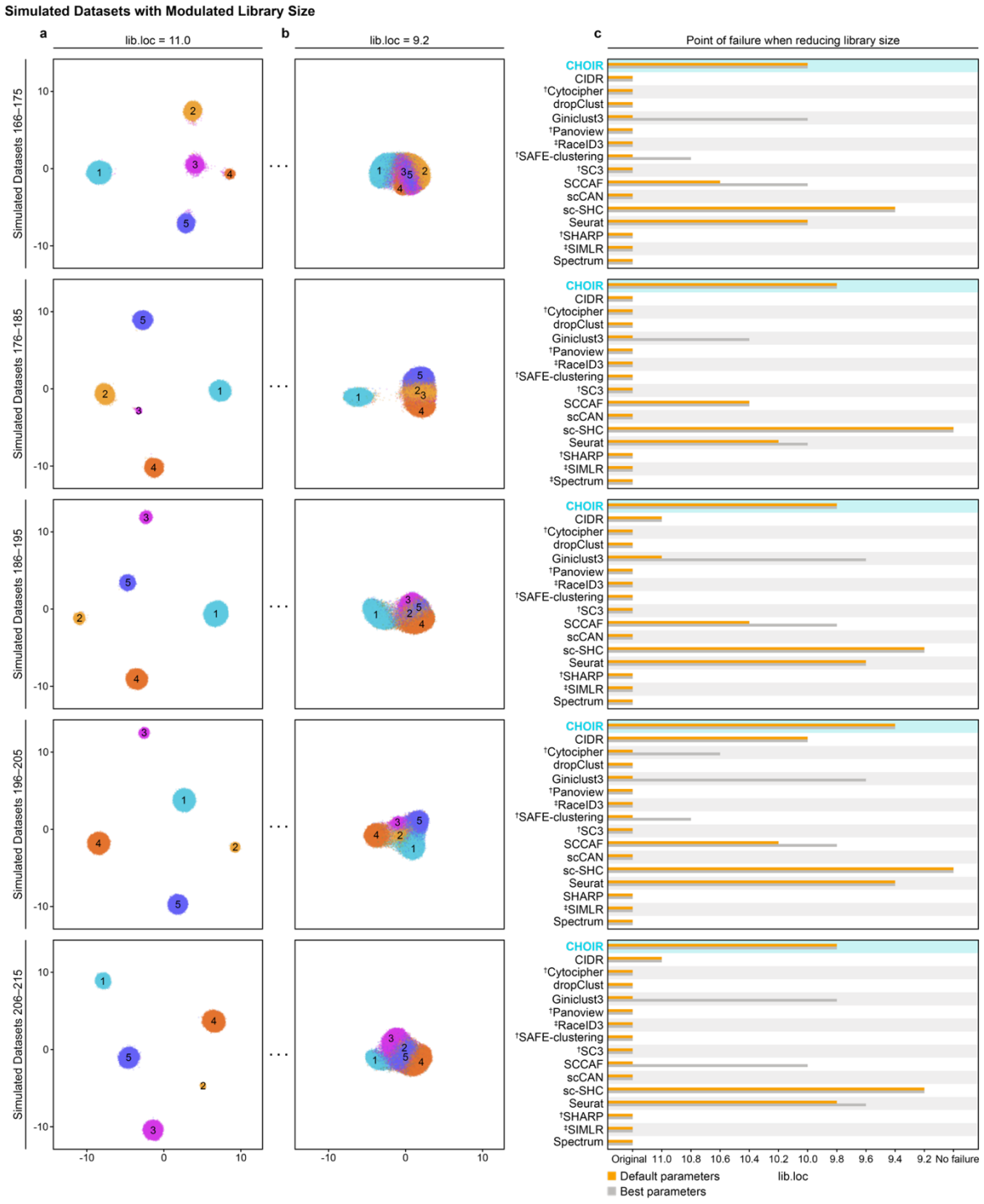
(Figure caption continued from the previous page.) **c.** The point of failure at which each method failed to achieve the correct number of clusters with an Adjusted Rand Index ≥ 0.9. The default parameter settings are shown, as well as the best-performing parameter settings, as evaluated across all 25 datasets shown. Symbols indicate clustering methods for which one or more parameter settings [†]failed to run or [‡]did not complete within the maximum allotted runtime of 96 hours for at least one dataset.



Supplementary Fig. 2.25: Benchmarking results from simulated datasets with incremental reductions of the level of differential expression

a–b. UMAP embeddings of each dataset colored according to the ground truth groups, when the simulation parameter "de.prob" (representing the level of simulated differential expression) for ground truth groups 1 and 2 was set to the first value of 0.4 (**a**) and the final value of 0.0125 (**b**). These datasets are modifications of Simulated Datasets 46–50. (Figure caption continued on the next page.)

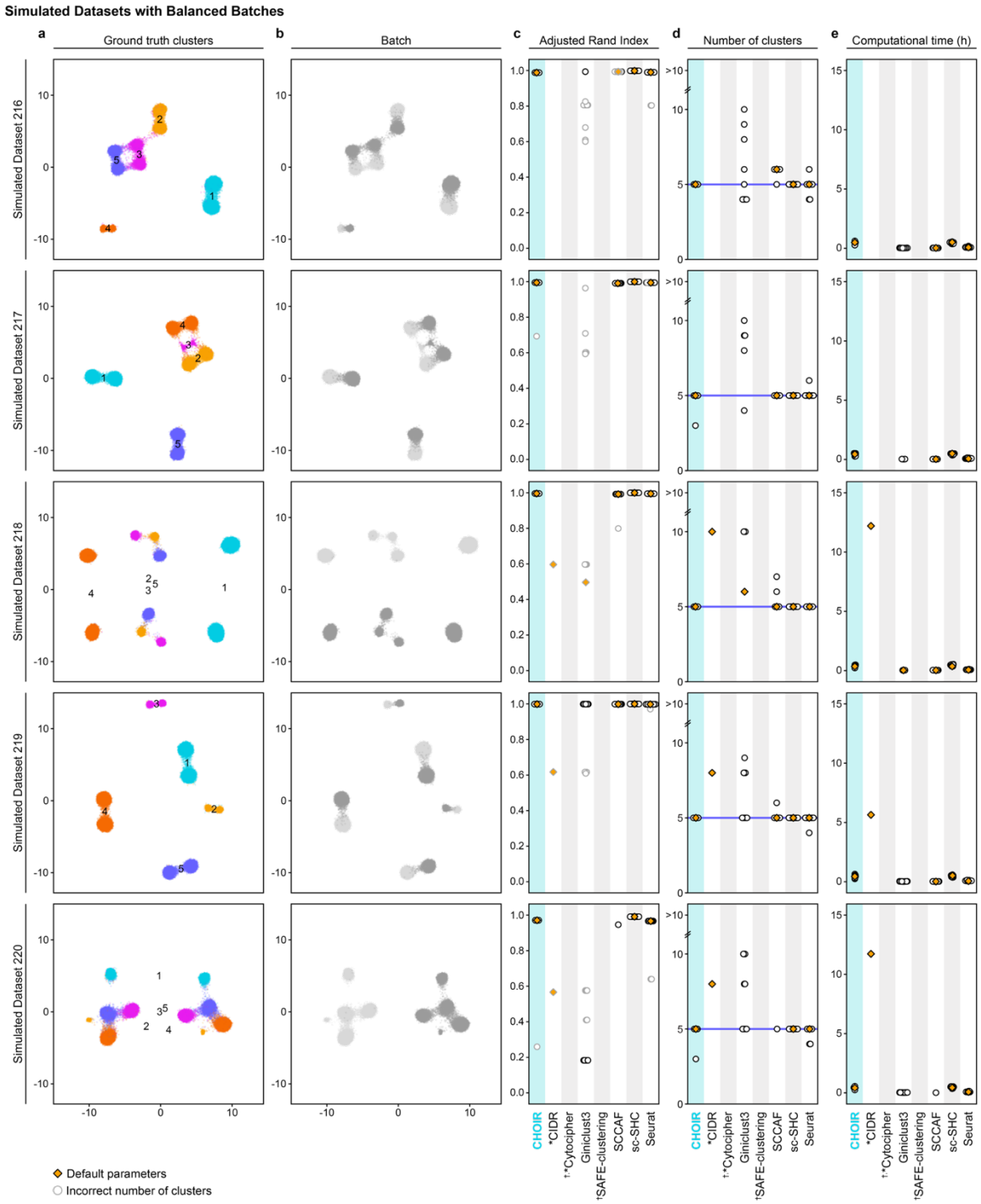
(Figure caption continued from the previous page.) **c.** The point of failure at which each method failed to achieve the correct number of clusters with an Adjusted Rand Index ≥ 0.9. The default parameter settings are shown, as well as the best-performing parameter settings, as evaluated across all 25 datasets shown. Symbols indicate clustering methods for which one or more parameter settings [†]failed to run or [‡]did not complete within the maximum allotted runtime of 96 hours for at least one dataset.



Supplementary Fig. 2.26: Benchmarking results from simulated datasets with incremental reductions of the library size

a–b. UMAP embeddings of each dataset colored according to the ground truth groups, when the simulation parameter "lib.loc" (representing the library size) was set to the first value of 11 (**a**) and the final value of 9.2 (**b**). These datasets are modifications of Simulated Datasets 46–50. (Figure caption continued on the next page.)

(Figure caption continued from the previous page.) **c.** The point of failure at which each method failed to achieve the correct number of clusters with an Adjusted Rand Index ≥ 0.9. The default parameter settings are shown, as well as the best parameter settings, as evaluated across all 25 datasets shown. Symbols indicate clustering methods for which one or more parameter settings [†]failed to run or [‡]did not complete within the maximum allotted runtime of 96 hours for at least one dataset.

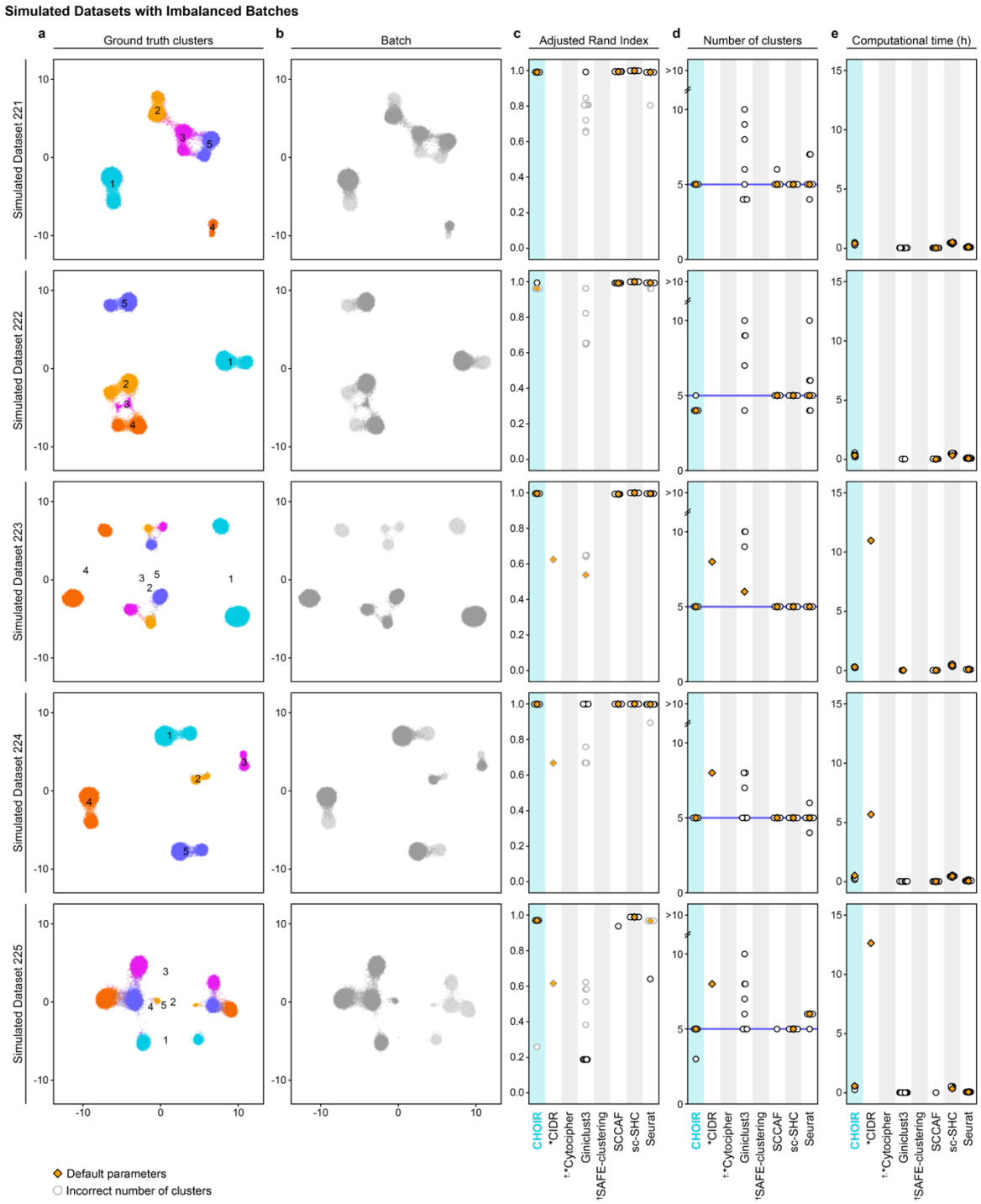


Supplementary Fig. 2.27: Benchmarking results from simulated datasets with two batches of equal size

a–b. UMAP embeddings of each dataset colored according to the ground truth groups (**a**) and batch labels (**b**). These datasets are modifications of Simulated Datasets 46–50.

c. Adjusted Rand Index obtained using each parameter setting tested for each method. Symbols indicate clustering methods for which *no parameter settings identified the correct number of clusters and reached an Adjusted Rand Index of >0.9 for the original clustering of Simulated Datasets 46–50, or †one or more parameter settings failed to run for at least one dataset. (Figure caption continued on the next page.)

(Figure caption continued from the previous page.) **d.** Number of clusters identified using each parameter setting tested for each method.
e. Computational time required by each parameter setting tested for each method.

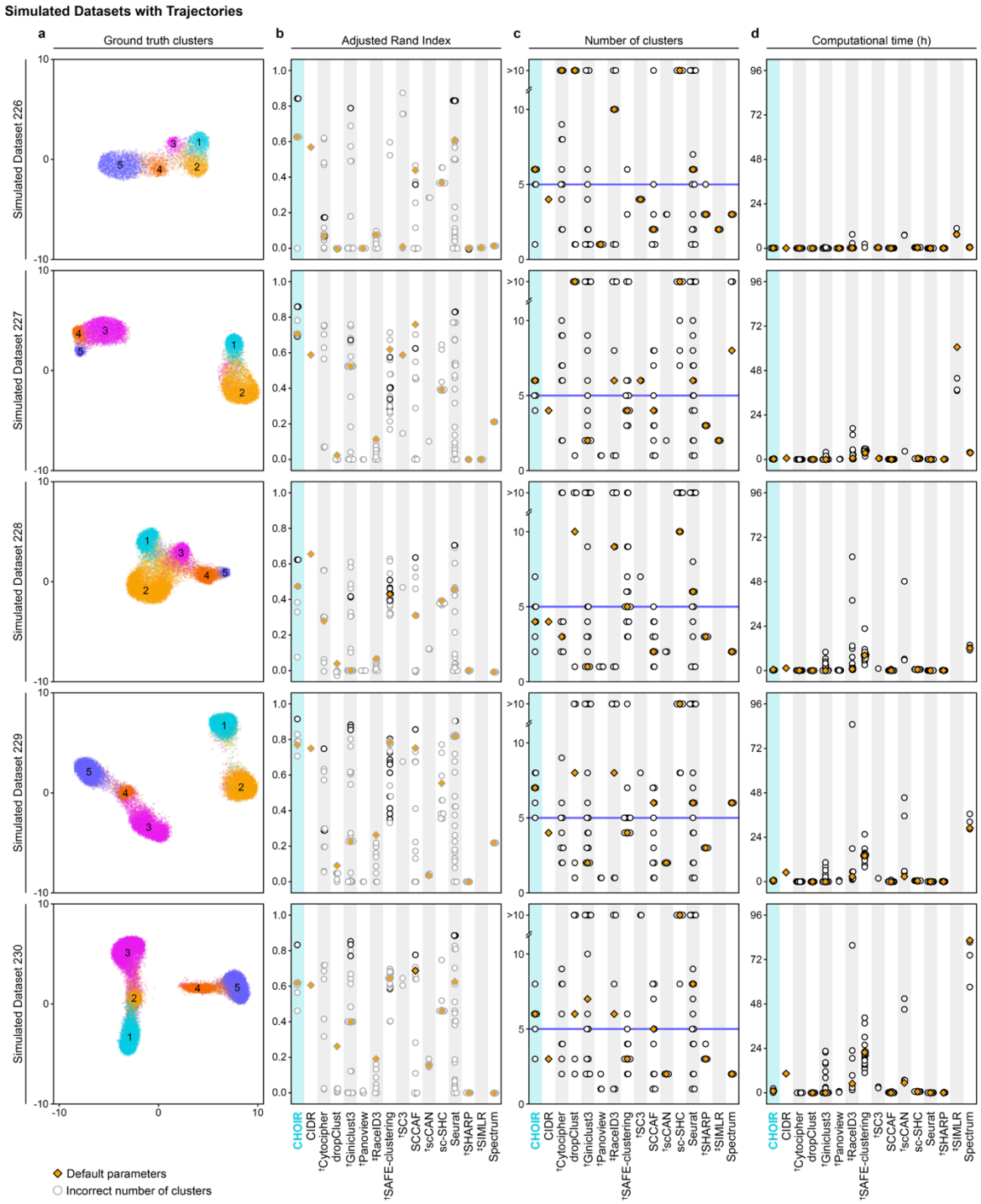


Supplementary Fig. 2.28: Benchmarking results from simulated datasets with two batches of unequal sizes

a–b. UMAP embeddings of each dataset colored according to the ground truth groups (**a**) and batch labels (**b**). These datasets are modifications of Simulated Datasets 46–50.

c. Adjusted Rand Index obtained using each parameter setting tested for each method. Symbols indicate clustering methods for which *no parameter settings identified the correct number of clusters and reached an Adjusted Rand Index of >0.9 for the original clustering of Simulated Datasets 46–50, or †one or more parameter settings failed to run for at least one dataset. (Figure caption continued on the next page.)

(Figure caption continued from the previous page.) **d.** Number of clusters identified using each parameter setting tested for each method.
e. Computational time required by each parameter setting tested for each method.



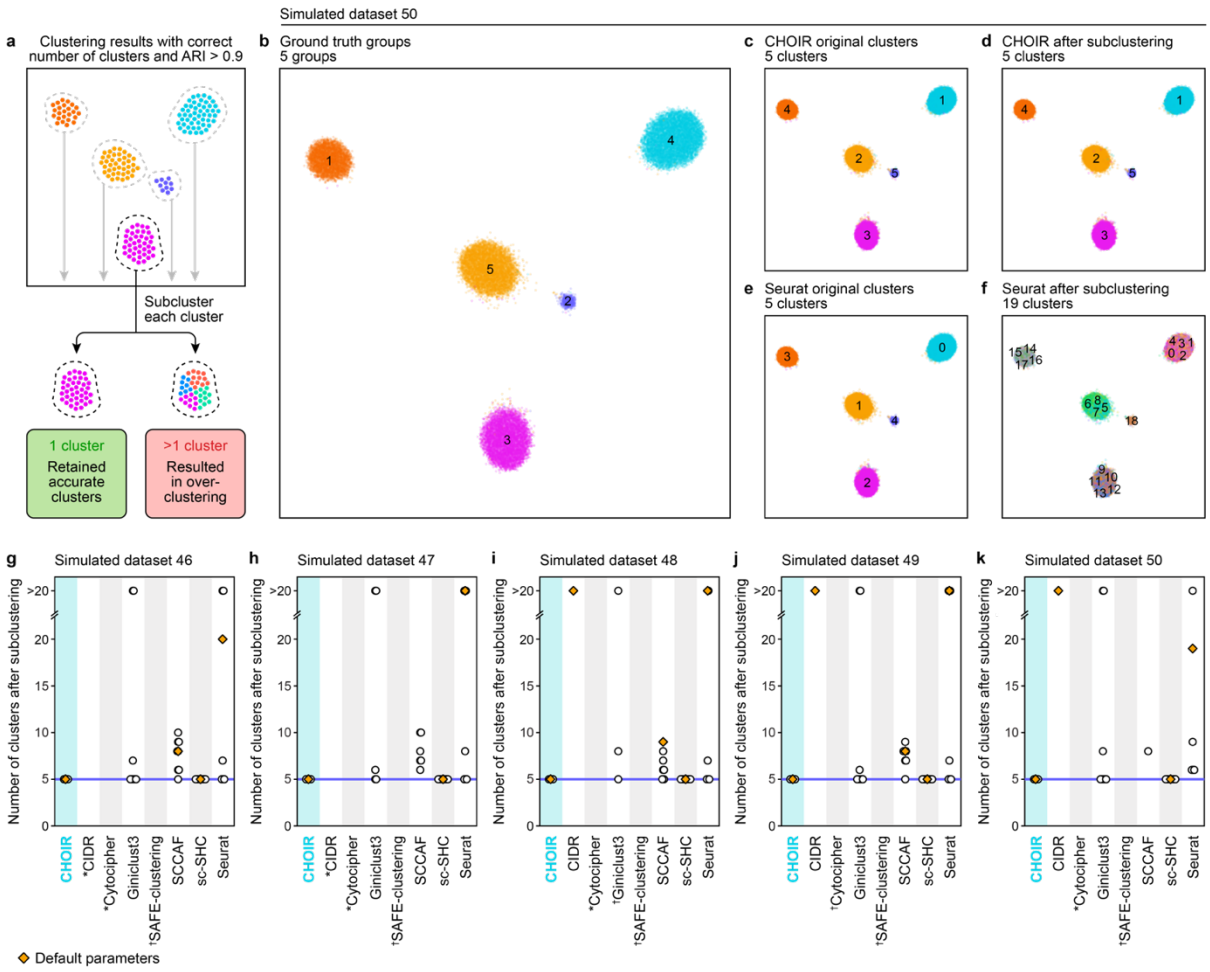
Supplementary Fig. 2.29: Benchmarking results from simulated cell differentiation trajectory datasets

a. UMAP embeddings of each dataset colored according to the ground truth groups.

b. Adjusted Rand Index obtained using each parameter setting tested for each method. Symbols indicate clustering methods for which one or more parameter settings †failed to run or ‡did not complete within the maximum allotted runtime of 96 hours for at least one dataset.

c. Number of clusters identified using each parameter setting tested for each method.

d. Computational time required by each parameter setting tested for each method.

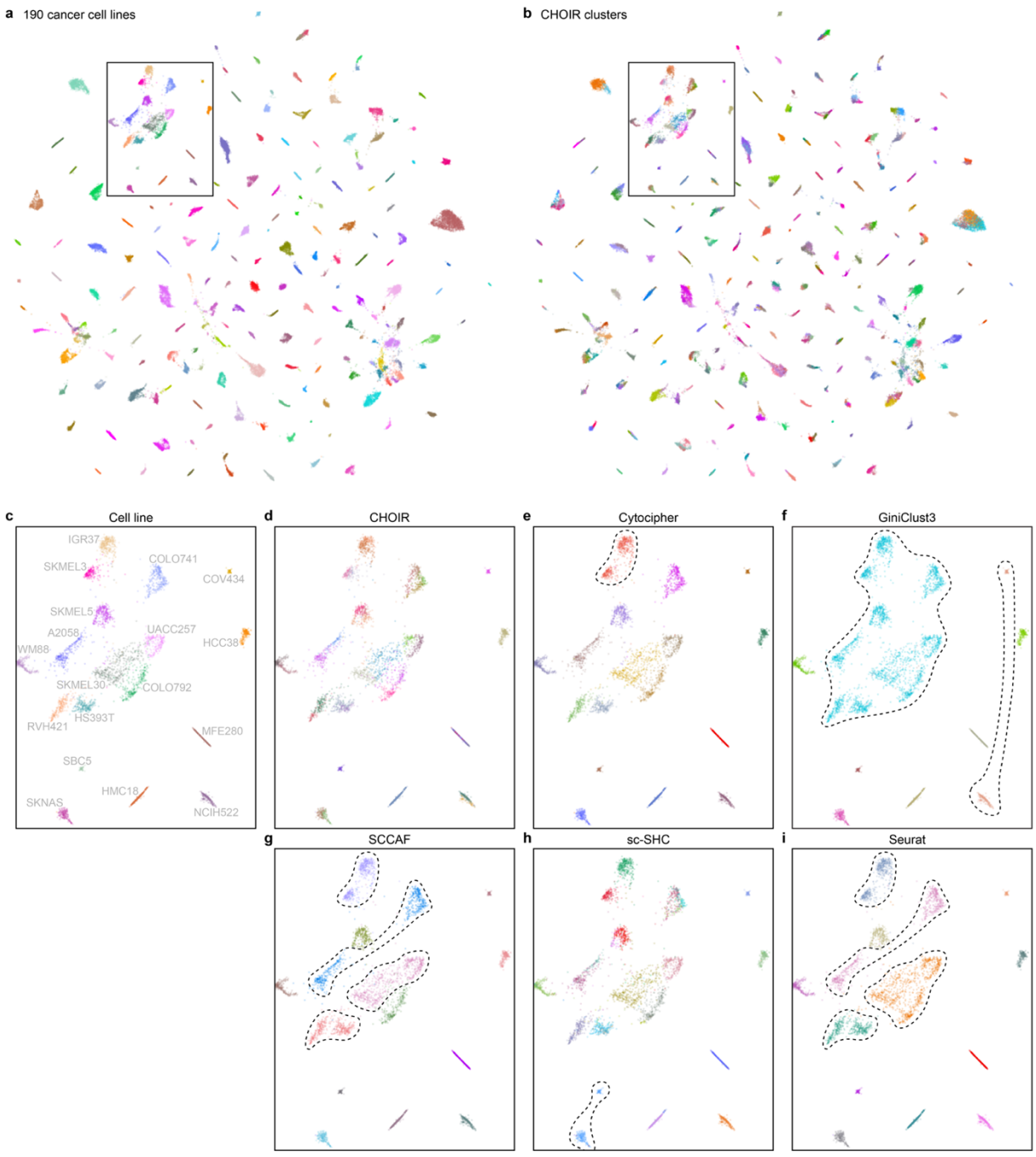


Supplementary Fig. 2.30: Clusters identified by CHOIR do not require further rounds of subclustering

a. Schematic showing the steps of the subclustering analysis.

b–f. UMAP embedding of Simulated Dataset 50 colored according to the ground truth groups (**b**), the clusters identified using the default parameters of CHOIR (**c**), the clusters identified after subclustering each of the clusters in (**b**) using the default parameters of CHOIR (**d**), the clusters identified using the default parameters of Seurat (**e**), and the clusters identified after subclustering each of the clusters in (**e**) using the default parameters of Seurat (**f**).

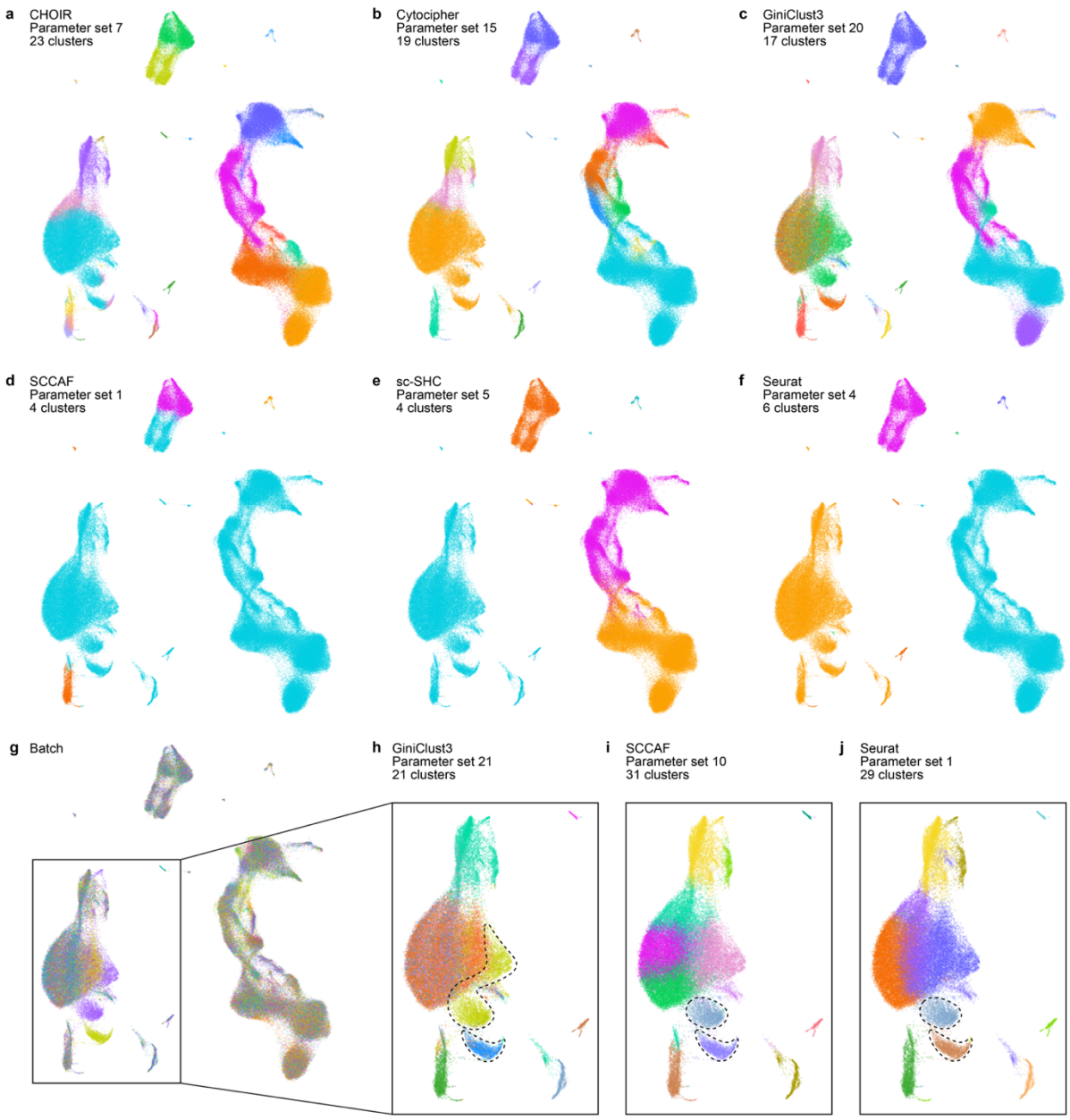
g–k. Dot plots showing the number of clusters obtained after subclustering for each parameter and method combination tested for simulated datasets 46–50. Parameter settings for each cluster method that resulted in the correct number of clusters and achieved an Adjusted Rand Index of >0.9 relative to the ground truth groups were selected <or: identified> here. Only eight of the clustering methods tested (CHOIR, CIDR, Cytocipher, GiniClust3, SAFE-clustering, SCCAF, sc-SHC, and Seurat) had any parameter settings meet this condition for any of datasets 46–50. For each such parameter setting, each cluster was subset and subclustered to identify the number of subclusters obtained. Symbols indicate clustering methods for which *no parameter settings identified the correct number of clusters and reached an Adjusted Rand Index of >0.9 for the original clustering of the dataset or †one or more parameter settings failed to run. For SAFE-clustering, no complete set of subclustered results ran without encountering at least one failure.



Supplementary Fig. 2.31: Additional evidence that CHOIR prevents underclustering in an scRNA-seq dataset consisting of pooled cell lines

a–b. UMAP embedding of a pooled cancer cell line scRNA-seq dataset from Kinker et al. 2020 consisting of 48,879 cells, colored according to the 190 cancer cell lines (**a**) and the clusters identified using the default parameters of CHOIR (**b**).

c–i. UMAP embeddings showing the cells within the highlighted rectangle in (**a–b**) colored by cell line (**c**) and the clusters identified by CHOIR (**d**), Cytocipher (**e**), GiniClust3 (**f**), SCCAF (**g**), sc-SHC (**h**), or Seurat (**i**), using the default parameters for each method. Dashed lines indicate instances where the clustering method failed to distinguish individual cell lines, resulting in multiple cell lines grouped within a single cluster.

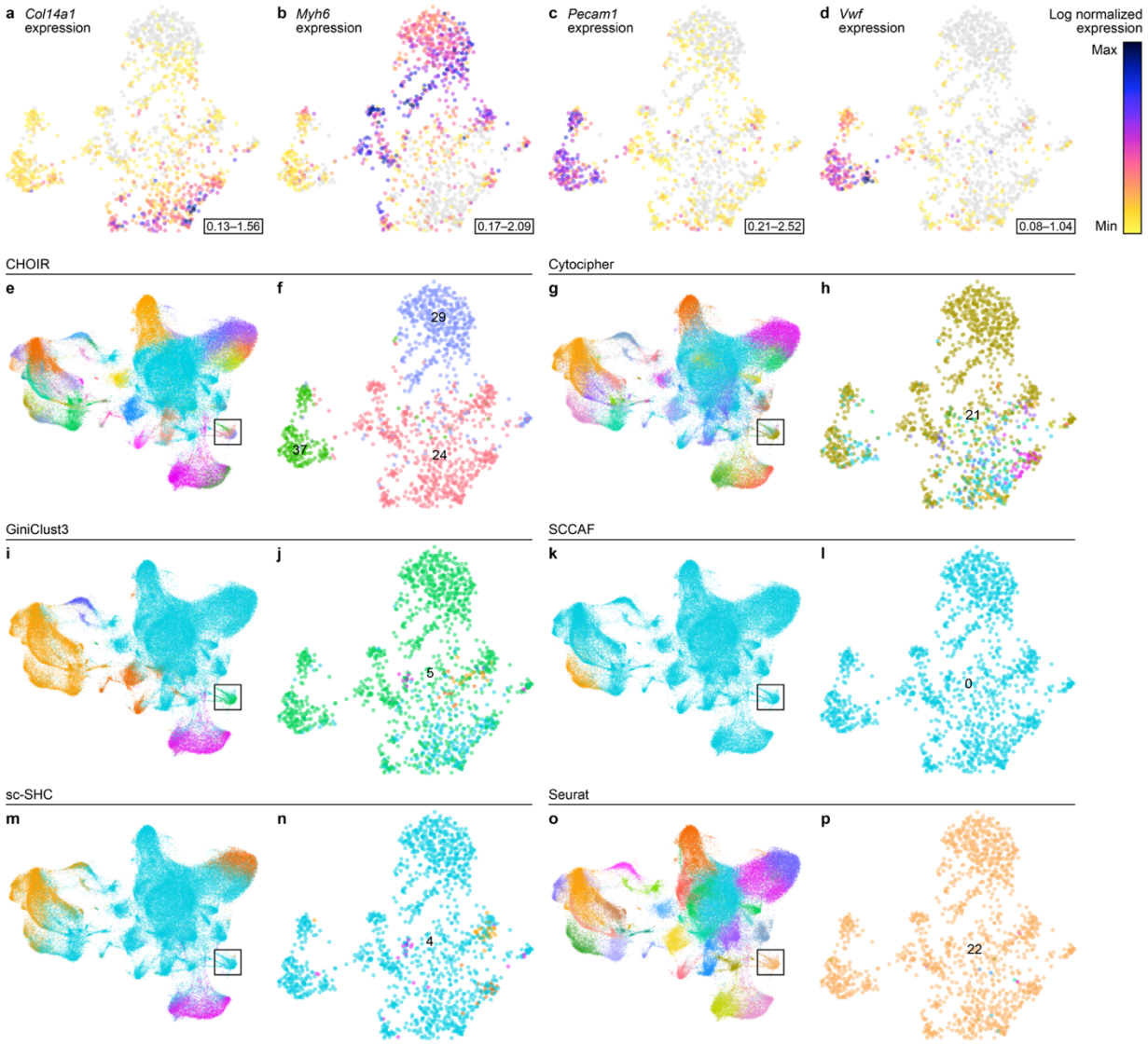


Supplementary Fig. 2.32: Comparison of results obtained with different clustering methods on the Hao et al. 2021 CITE-seq dataset

a–f. UMAP embeddings for the Hao et al. 2021 dataset colored according to the clusters identified by the parameter settings for CHOIR (**a**), Cytocipher (**b**), GiniClust3 (**c**), SCCAF (**d**), sc-SHC (**e**), or Seurat (**f**) that maximized the number of clusters while preventing overclustering as shown in **Fig. 2.4**.

g. UMAP embedding for the Hao et al. 2021 dataset colored according to batch. The highlighted box identifies a subset of cells where batch effects are observed.

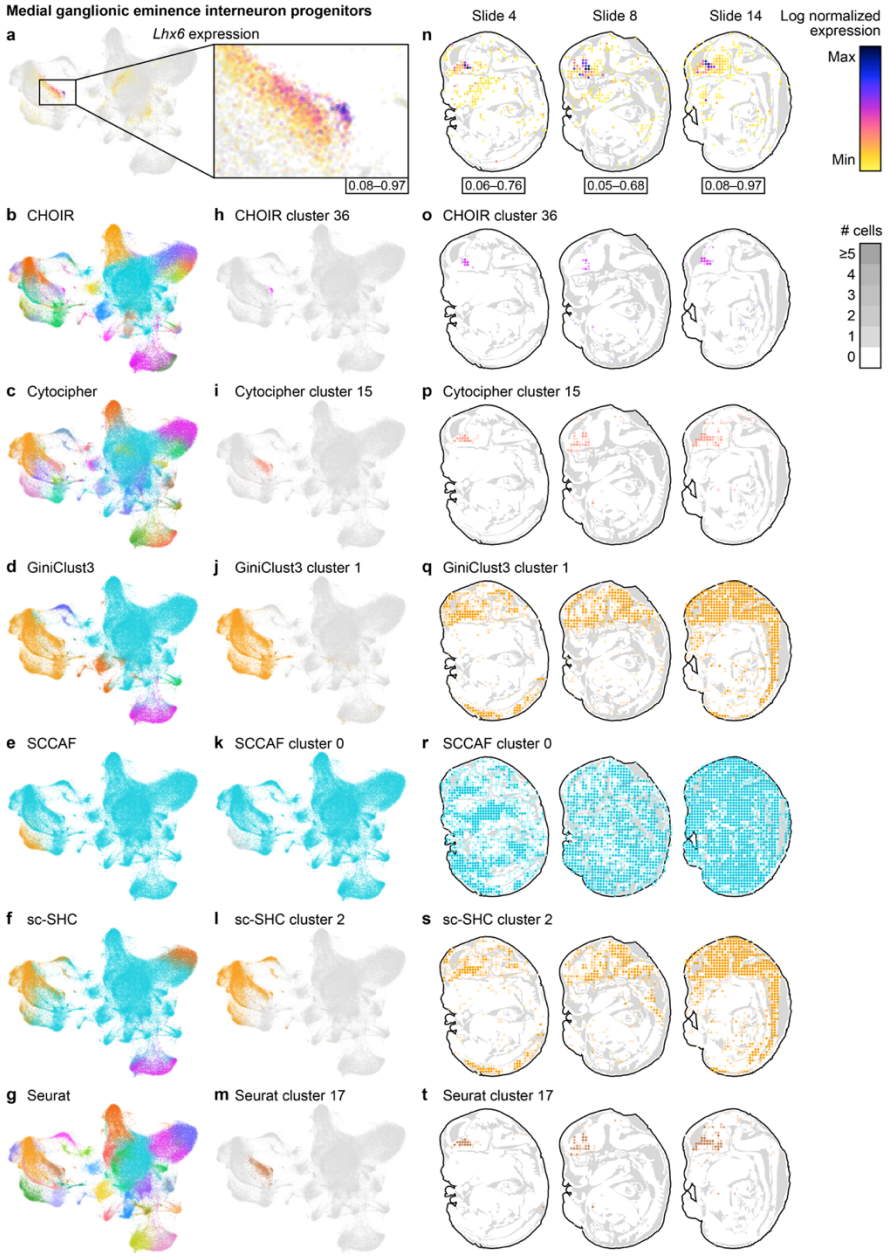
h–j. UMAP embeddings showing instances in which overclustering was due to batch effects for GiniClust3 (**h**), SCCAF (**i**), and Seurat (**j**). Dashed lines indicate clusters confounded by batch.



Supplementary Fig. 2.33: CHOIR identifies three functionally distinct cell types of the embryonic heart

a–d. UMAP embedding of the Srivatsan et al. 2021 whole mouse embryo sci-Space dataset computed on the subsetted dimensionality reduction of parent cluster P4, identified using the default parameters in CHOIR, and colored according to the expression levels of connective tissue marker gene *Col14a1* (**a**), cardiomyocyte marker gene *Myh6* (**b**), or endothelial cell marker genes *Pecam1* (**c**) and *Vwf* (**d**). For each plot, the minimum and maximum normalized expression values are shown at the bottom. Grey color indicates no detected expression.

e–p. UMAP embeddings of all cells (**e, g, i, k, m, o**) or heart cells (**f, h, j, l, n, p**) colored according to the clusters identified by the default parameters of CHOIR (**e–f**), Cytocipher (**g–h**), GiniClust3 (**i–j**), SCCAF (**k–l**), sc-SHC (**m–n**), or Seurat (**o–p**).

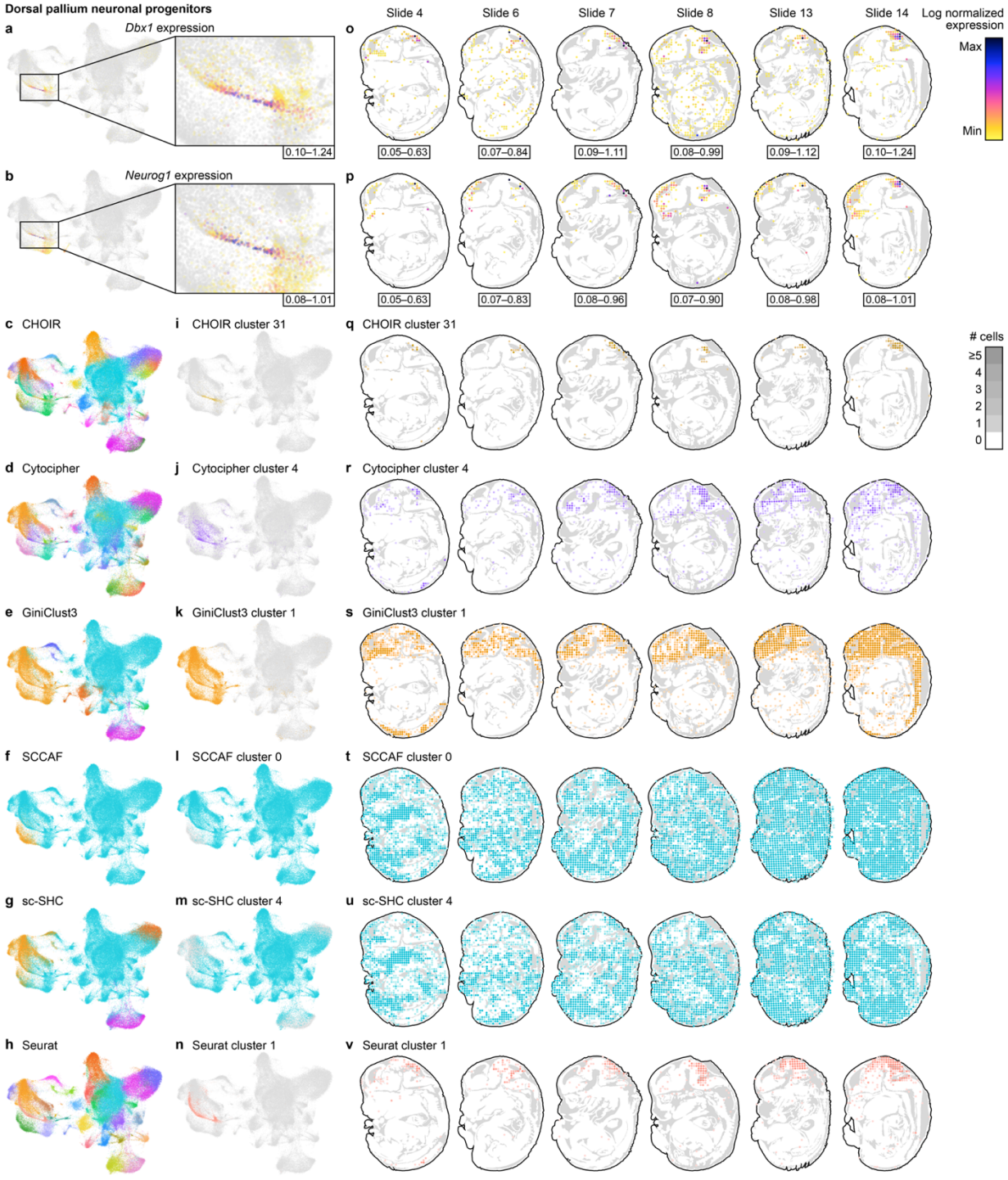


Supplementary Fig. 2.34: CHOIR cluster 36 represents the pool of medial ganglionic eminence interneuron progenitors characterized by high expression of Lhx6

a–g. UMAP embedding of the Srivatsan et al. 2021 whole mouse embryo sci-Space dataset, colored according to the expression level of the medial ganglionic eminence interneuron progenitor marker gene *Lhx6* (**a**), or the clusters identified by the default parameters of CHOIR (**b**), Cytocipher (**c**), GiniClust3 (**d**), SCCAF (e), sc-SHC (**f**), or Seurat (**g**). In panel (**a**), a zoom-in of *Lhx6* expression is shown to the right and the minimum and maximum normalized expression values are shown at the bottom of the zoom-in. Grey color indicates no expression detected.

h–m. UMAP embeddings of the Srivatsan et al. 2021 whole mouse embryo sci-Space dataset, colored by the cluster shown in panels (**a–g**) that harbors the *Lhx6*-expressing medial ganglionic eminence interneuron progenitors for each of the methods indicated. (Figure caption continued on the next page.)

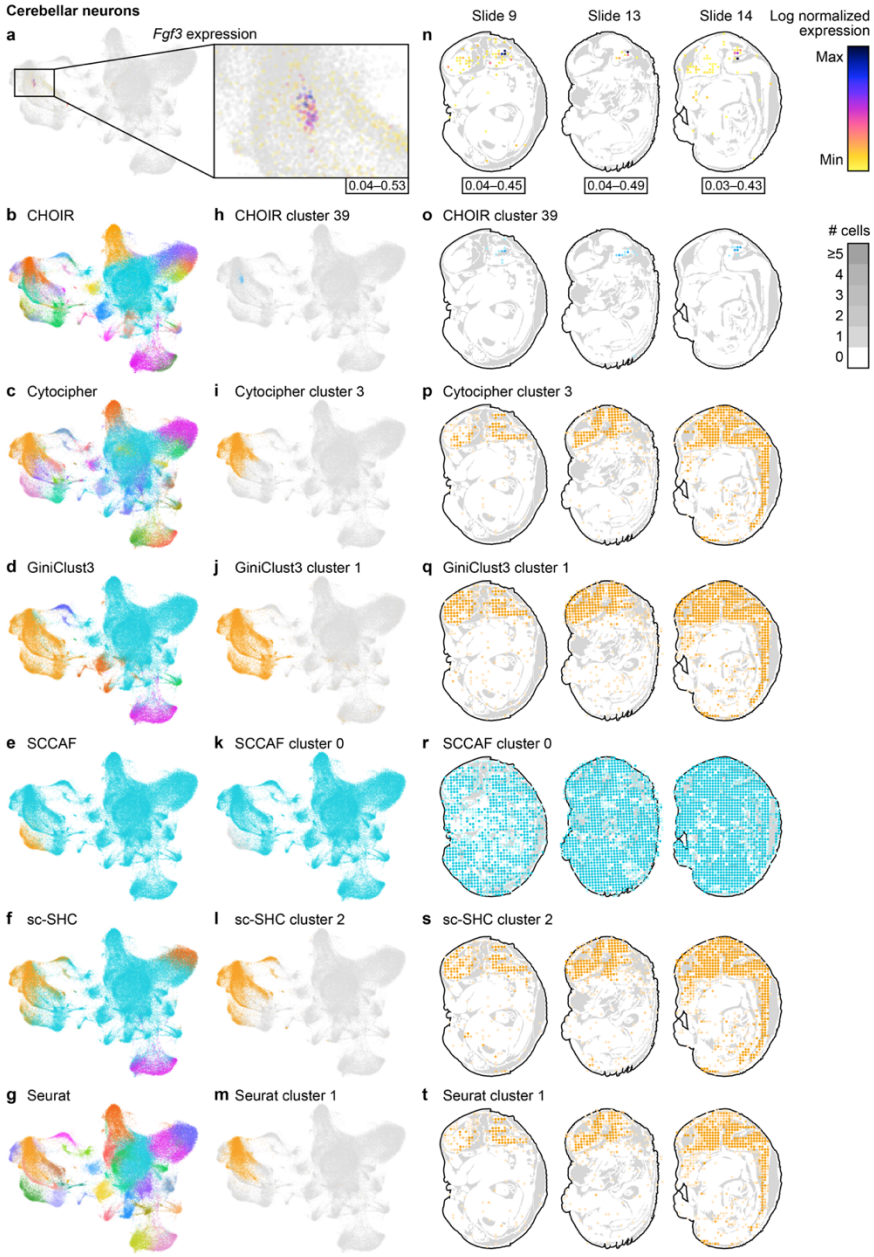
(Figure caption continued from the previous page.) **n.** Distribution of medial ganglionic eminence interneuron progenitor marker gene *Lhx6* in all sections with >25 cells belonging to CHOIR cluster 36 (medial ganglionic eminence interneuron progenitors).
o–t. Distribution of the method-specific cluster shown to the left in panels (**h–m**) across all sections shown in panel (**n**). The darker the shade of the cluster color, the more cells of the respective cluster were detected at the indicated location.



Supplementary Fig. 2.35: CHOIR cluster 31 represents a neuronal progenitor pool of the spatially restricted E14 dorsal pallium characterized by high expression of Dbx1 and Neurog1

a–h. UMAP embeddings of the Srivatsan et al. 2021 whole mouse embryo sci-Space dataset, colored according to the expression levels of dorsal pallium neuronal progenitor marker gene *Dbx1* (**a**) and *Neurog1* (**b**), or the clusters identified by the default parameters of CHOIR (**c**), Cytocipher (**d**), GiniClust3 (**e**), SCCAF (**f**), sc-SHC (**g**), or Seurat (**h**). In panels (**a–b**), a zoom-in of *Dbx1* and *Neurog1* expression is shown to the right and the minimum and maximum normalized expression values are shown at the bottom of the zoom-in. Grey color indicates no detected expression. (Figure caption continued on the next page.)

(Figure caption continued from the previous page.) **i–n.** UMAP embeddings of the Srivatsan et al. 2021 whole mouse embryo sci-Space dataset, colored by the cluster shown in panels (**a**–**h**) that harbors the *Dbx1- and Neurog1*-expressing dorsal pallium neuronal progenitors for each of the methods indicated.
o–p. Distribution of dorsal pallium neuronal progenitor marker genes *Dbx1* (**o**) and *Neurog1* (**p**) in all sections with >25 cells belonging to CHOIR cluster 31 (dorsal pallium neuronal progenitors). For each plot, the minimum and maximum normalized expression values are shown at the bottom. Grey color indicates no expression detected.
q–v. Distribution of the method-specific cluster shown to the left in panels (**i–n**) across all sections shown in panels (**o–p**). The darker the shade of the cluster color, the more cells of the respective cluster were detected at the indicated location.

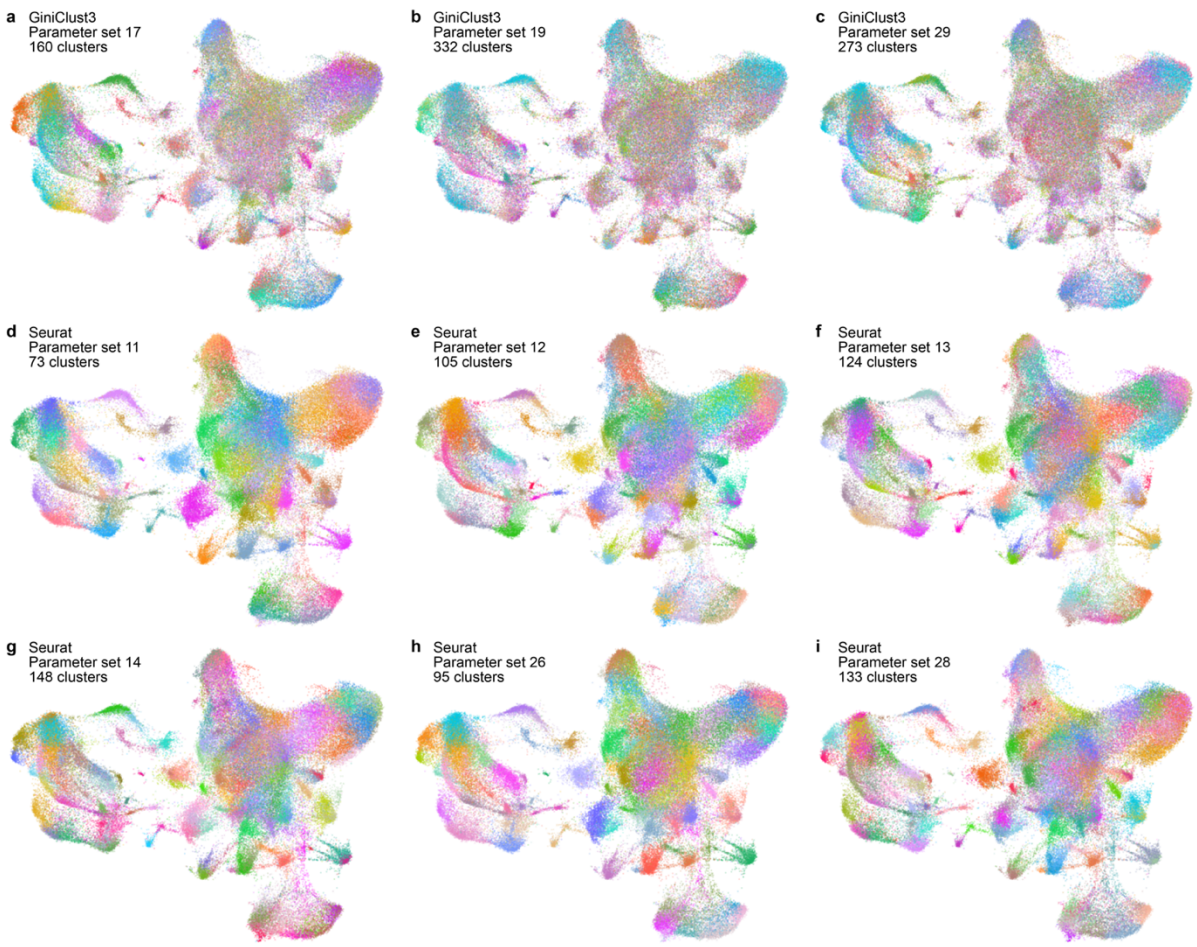


Supplementary Fig. 2.36: CHOIR cluster 39 represents a population of neurons localized within the developing cerebellum characterized by high expression of Fgf3

a–g. UMAP embedding of the Srivatsan et al. 2021 whole mouse embryo sci-Space dataset, colored according to the expression level of developing cerebellum marker gene *Fgf3* (**a**), or the clusters identified by the default parameters of CHOIR (**b**), Cytocipher (**c**), GiniClust3 (**d**), SCCAF (**e**), sc-SHC (**f**), or Seurat (**g**). In panel (**a**), a zoom-in of *Fgf3* expression is shown to the right and the minimum and maximum normalized expression values are shown at the bottom of the zoom-in. Grey color indicates no detected expression.

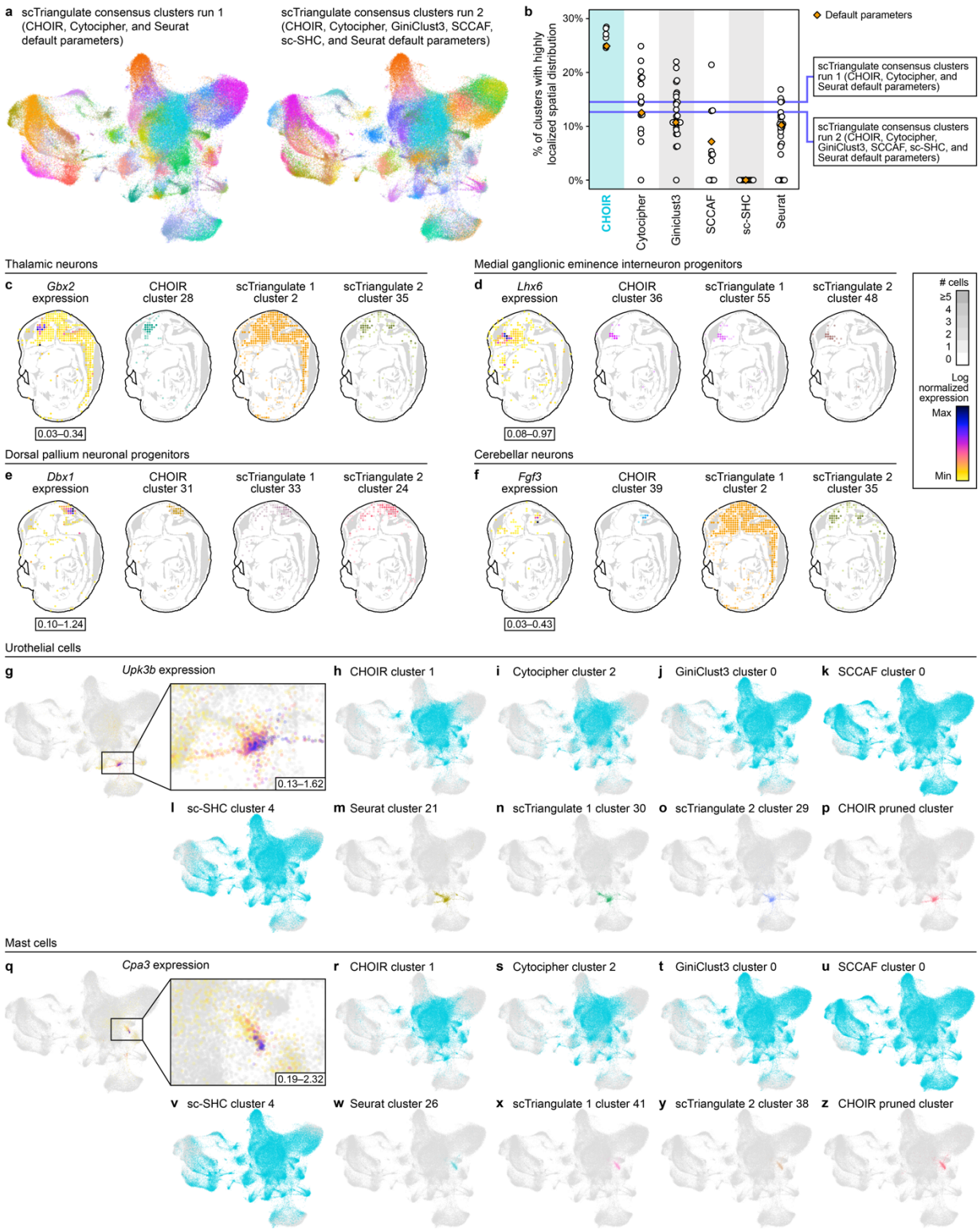
h–m. UMAP embeddings of the Srivatsan et al. 2021 whole mouse embryo sci-Space dataset, colored by the cluster shown in panels (**a–g**) that harbors the *Fgf3*-expressing developing cerebellum neurons for each of the methods shown to the left. (Figure caption continued on the next page.)

(Figure caption continued from the previous page.) **n.** Distribution of developing cerebellum marker gene *Fgf3* in all sections with >25 cells belonging to CHOIR cluster 39 (developing cerebellum neurons). For each plot, the minimum and maximum normalized expression values are shown at the bottom. Grey color indicates no expression detected.
o–t. Distribution of the method-specific cluster shown to the left in panels (**h–m**) across all sections shown in panel (**n**). The darker the shade of the cluster color, the more cells of the respective cluster were detected at the indicated location.



Supplementary Fig. 2.37: Non-default parameters for GiniClust3 and Seurat that identified certain anatomically restricted cell types resulted in severe overclustering of other cell types

a–i. UMAP embeddings of the Srivatsan et al. 2021 whole mouse embryo sci-Space dataset colored according to the clusters identified by GiniClust3 (**a–c**) or Seurat (**d–i**) using non-default parameters. Although these non-default parameters identified at least 3 of the 7 functionally distinct cell types in the heart or specific brain regions that were identified by CHOIR's default parameters, they resulted in severe overclustering of other cell types.



Supplementary Fig. 2.38: Consensus clustering using scTriangulate does not improve overall clustering results (Figure caption continued on the next page.)

(Figure caption continued from the previous page.) **a.** UMAP embeddings of the Srivatsan et al. 2021 whole mouse embryo sci-Space dataset colored according to the clusters identified when scTriangulate was applied to the default parameter settings of CHOIR, Cytocipher, and Seurat (left), or of CHOIR, Cytocipher, GiniClust3, SCCAF, sc-SHC, and Seurat (right).
b. Dot plot showing the mean percentage of all clusters per section that had a highly localized spatial distribution for all method and parameter combinations tested, annotated with blue lines corresponding to the performance of scTriangulate as discussed in (**a**). Same dot plot as depicted in **Fig. 2.5** but with additional annotations showing the performance of scTriangulate.
c–f. Spatial coordinate plots colored by expression levels of key cell type marker genes and the distributions of the corresponding CHOIR and scTriangulate clusters that harbor the cells expressing those marker genes for thalamic neurons (**c**), medial ganglionic eminence interneuron progenitors (**d**), dorsal pallium neuronal progenitors (**e**), and cerebellar neurons (**f**). For plots showing clusters, the darker the shade of any given color, the more cells of the respective cluster were detected at the indicated location, as shown in the inset legend next to panel (**d**). Slide 14 is shown throughout this figure because it had the highest number of recovered nuclei and positions.
g. UMAP embedding colored according to the expression level of the urothelial cell marker gene *Upk3b*. A zoom-in of *Upk3b* expression is shown to the right and the minimum and maximum normalized expression values are shown at the bottom of the zoom-in. Grey color indicates no expression detected.
h–p. UMAP embeddings colored by the cluster that harbors the urothelial cells expressing *Upk3b* in the final clusters identified by the default parameter settings of CHOIR (**h**), Cytocipher (**i**), GiniClust3 (**j**), SCCAF (**k**), sc-SHC (**l**), or Seurat (**m**), and clusters identified when scTriangulate was applied to the default parameter settings of CHOIR, Cytocipher, and Seurat (**n**) or of CHOIR, Cytocipher, GiniClust3, SCCAF, sc-SHC, and Seurat (**o**). A cluster corresponding to the urothelial cells expressing *Upk3b* was identified by CHOIR but pruned after correction for multiple comparisons (**p**).
q. UMAP embedding colored according to the expression level of the mast cell marker gene *Cpa3*. A zoom-in of *Cpa3* expression is shown to the right and the minimum and maximum normalized expression values are shown at the bottom of the zoom-in. Grey color indicates no expression detected.
r–z. UMAP embeddings colored by the cluster that harbors the mast cells expressing *Cpa3* in the final clusters identified by the default parameter settings of CHOIR (**r**), Cytocipher (**s**), GiniClust3 (**t**), SCCAF (**u**), sc-SHC (**v**), or Seurat (**w**), and clusters identified when scTriangulate was applied to the default parameter settings of CHOIR, Cytocipher, and Seurat (**x**) or of CHOIR, Cytocipher, GiniClust3, SCCAF, sc-SHC, and Seurat (**y**). A cluster corresponding to the mast cells expressing *Cpa3* was identified by CHOIR but pruned due to the variance significance threshold (**z**).

Publishing Agreement

It is the policy of the University to encourage open access and broad distribution of all theses, dissertations, and manuscripts. The Graduate Division will facilitate the distribution of UCSF theses, dissertations, and manuscripts to the UCSF Library for open access and distribution. UCSF will make such theses, dissertations, and manuscripts accessible to the public and will take reasonable steps to preserve these works in perpetuity.

I hereby grant the non-exclusive, perpetual right to The Regents of the University of California to reproduce, publicly display, distribute, preserve, and publish copies of my thesis, dissertation, or manuscript in any form or media, now existing or later derived, including access online for teaching, research, and public service purposes.

Signed by:
Catherine Saint
5B639968FF9F44A...
Author Signature

12/6/2024
Date