NAR Genomics and Bioinformatics, 2024, **6**, lqae174
https://doi.org/10.1093/nargab/lqae174
Advance access publication date: 18 December 2024
Pipelines and Workflows for Biological Data Analysis

IDclust: Iterative clustering for unsupervised identification of cell types with single cell transcriptomics and epigenomics

Pacôme Prompsy[1,2,3,4,*], Mélissa Saichi[1,2], Félix Raimundo[1,2,5] and Céline Vallot[1,2,*]

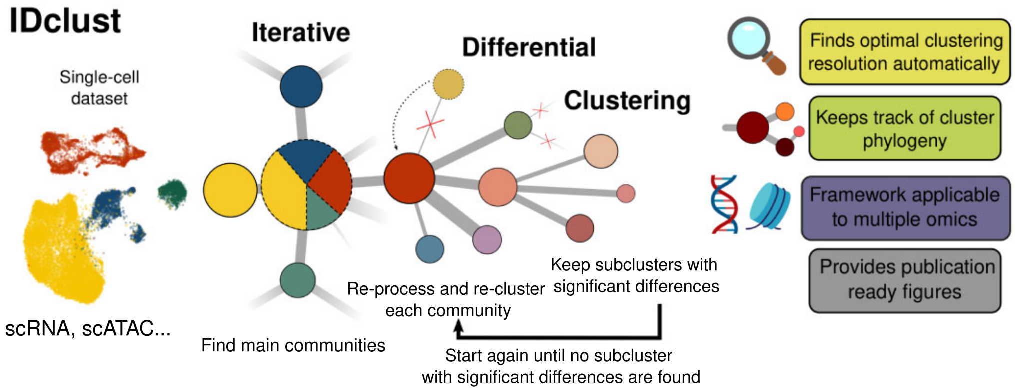
[1]CNRS UMR3244, Institut Curie, PSL Research University, 26 rue d'Ulm, 75005 Paris, France
[2]Department of Translational Research, Institut Curie, PSL Research University, 26 rue d'Ulm, 75005 Paris, France
[3]Department of Dermatology, Lausanne University Hospital (CHUV), Avenue de Beaumont 29, 1011Lausanne, Switzerland
[4]Faculty of Biology and Medicine, University of Lausanne, Rue du Bugnon 46, 1005 Lausanne, Switzerland
[5]Department of Genomics and Computational Biology, University of Massachusetts Chan Medical School, 55 N Lake Ave, Worcester, MA 01605, USA

[*]To whom correspondence should be addressed. Tel: +41 213140353; Email: pacome.promsy@chuv.ch
Correspondence may also be addressed to Céline Vallot. Email: celine.vallot@curie.fr

Abstract

The increasing diversity of single-cell datasets require systematic cell type characterization. Clustering is a critical step in single-cell analysis, heavily influencing downstream analyses. However, current unsupervised clustering algorithms rely on biologically irrelevant parameters that require manual optimization and fail to capture hierarchical relationships between clusters. We developed IDclust, a framework that identifies clusters with significant biological features at multiple resolutions using biologically meaningful thresholds like fold change, adjusted *P*-value and fraction of expressing cells. By iteratively processing and clustering subsets of the dataset, IDclust guarantees that all clusters found have significantly different features and stops only when no more interpretable cluster is found. It also creates a hierarchy of clusters, enabling visualization of the hierarchical relationships between different clusters. Analyzing multiple single-cell transcriptomic reference datasets, IDclust achieves superior clustering accuracy compared to state of the art algorithms. We showcase its utility by identifying previously unannotated clusters and identifying branching patterns in scATAC datasets. Using it's unsupervised nature and ability to analyze different -omics, we compare the resolution of different histone marks in multi-omic paired-tag dataset. Overall, IDclust automates single-cell exploration, facilitates cell type annotation and provides a biologically interpretable basis for clustering.

Graphical abstract


IDclust
Single-cell dataset
scRNA, scATAC...
Iterative
Differential
Clustering
Find main communities
Re-process and re-cluster each community
Keep subclusters with significant differences
Start again until no subcluster with significant differences are found
Finds optimal clustering resolution automatically
Keeps track of cluster phylogeny
Framework applicable to multiple omics
Provides publication ready figures

Introduction

With the advent of single-cell genomics, researchers now have access to measurements of messenger RNA (mRNA) expression, chromatin accessibility or chromatin modifications in thousands to millions of cells from tissue and even whole organisms. One key advantage of single-cell technologies is the ability to distinguish and discover 'cell types.' A 'cell type' can be defined as a group of cells that share morphological, phenological and functional features, yet the definition of 'cell type' is not fully resolved and remains open to debate (1,2). In practice, researchers exploring single-cell datasets use state-of-the-art processing softwares, such as the Monocle (3), Seurat (4)

Received: August 8, 2024. **Revised:** November 8, 2024. **Editorial Decision:** November 18, 2024. **Accepted:** November 19, 2024

or Scanpy (5) packages. Common practices consist in performing unsupervised clustering with the provided functions implemented in the packages using either the default parameters (desired number of clusters, clustering resolution, k neighbors, etc.) or manually adapting them until subjective satisfaction is achieved. However, this manual adjustment is done on an adhoc basis, lacks quantitative metrics for support and relies on abstract parameters that are not biologically interpretable.

Multiple unsupervised algorithms have been proposed to find clusters within single-cell datasets. The Louvain (6) algorithm, used under the hood in many of the packages mentioned above, has two main parameters, namely the resolution which controls the number of final clusters—the greater the resolution, the higher the number of final clusters—and the number of neighbors K used to create the shared nearest neighbor graph which monitors the number of cell communities from the graph. The manual adjustment of these parameters alters the clustering reproducibility and robustness. Other algorithms have been proposed to find the optimal number of clusters which represents the consensus number of clusters obtained from multiple subsampling steps of the dataset as MultiK (7), SC3 (8) and ConsensusClusterPlus (9). These approaches relying on consensus clustering are computationally heavy and hardly scalable on large datasets. Other approaches, such as Clustree (10) vary clustering resolution to help choose a more robust number of clusters but do not provide the optimal number of clusters. Finally, some algorithms rely on finding hierarchies of clusters, such as hierarchical clustering or TooManyCells (11). However, many of these approaches still depend on parameters that are not biologically relevant (e.g., resolution) or are only dedicated to scRNA-seq. We argue that no single set of clustering parameters can be universally applied to all single-cell datasets that vary in number of cells (from hundreds to millions), in nature (cell line, tissue, disease) and obtained using different technologies (10X, Smart-seq3, etc.).

Here, we present IDclust, iterative differential clustering, a clustering framework based on the definition of ‘cell type’ as ‘a homogeneous group of cells that share a common set of molecular differences versus other cells.’ IDclust is an unsupervised clustering algorithm that recursively finds all clusters with significantly different biological features compared to their neighbors. With IDclust, the parameters discriminating between meaningful clusters are classic differential analysis-based parameters: log2-fold change (FC), adjusted *P*-value and fraction of expressing cells, with an additional parameter, the number of differential features. Thus, the parameter optimization inherent to clustering is shifted to biologically relevant parameters.

This framework can be applied to single-cell RNA (scRNA) or single-cell epigenomic datasets, such as scATAC-seq or scCUT&Tag, but can easily be extended to other molecular assays. The recursive exploration allows us to preserve relationships between clusters and thus create a hierarchy of clusters. The framework guarantees that each cluster has a certain number of significantly different features and outputs the marker genes of each cluster, facilitating the annotation of the clusters. Finally, subparts of the dataset are reprocessed at each step, allowing to effectively ‘zoom in’ and discover clusters that might be missed when studying the entire dataset. IDClust is available as an R package at https://github.com/vallotlab/IDclust.

Materials and methods

Description of IDclust

IDclust (iterative differential clustering) is an algorithm that recursively processes a single-cell dataset and finds clusters with significantly different features. It automatically and repeatedly analyzes cells within a chosen cluster at each step, reclusters them if needed and avoids clusters that contain no genuine biological variations. It stops when all the clusters with enough significantly different features from the neighboring clusters have been found.

The algorithm returns hierarchical clustering of the cells, as well as the significantly differential features found at each step. The output of IDclust can be visualized as a hierarchical cluster tree or overlaid on 2D representations such as uniform manifold approximation and projection (UMAP) or principal component analysis (PCA).

The algorithm is conceived as a modulable framework in which the user can input its own processing and differential function that best fits its technology. In this way, we believe that IDclust could be used for other single-cell technologies (single-cell DNA methylation, single-cell proteomics, etc.) and adapted for future computational developments involving data processing and differential analysis.

IDclust algorithm

The algorithm takes as input either a SingleCellExperiment or a Seurat object, containing an unnormalized count matrix. It outputs:

1. The set of clusters with preserved relationship.
2. The set of marker features for each cluster.

```
Pseudo-code:
X = the raw matrix (features x cells)
  process(X) ← function(X) {
  normalize X
  dimensionality_reduction X
  returns embedding
}
find_clusters(X) ← function(X, force = FALSE) {
  returns a set of clusters using Louvain, adapting k to the number of cells in X
  if (force == TRUE):
  returns a set of clusters using Louvain between 2 and 6
}
is_valid_cluster(X, cluster, limit) ← function(X, cluster) {
  X_cluster ← X[cells in cluster]
  X_rest ← X[cells not in cluster]
    differential_features ← differential_analysis(X_cluster, X_rest)
  if length(differential_features) > minimum_number_of_genes:
    return(TRUE)
  else
  return(FALSE)
}
# Start of the algorithm
X ← process(X)
initial_clusters ← find_clusters(X, force = TRUE)
n ← 0
clusterisable_clusters = initial_clusters
while length(clusterisable_clusters) > 0 :
    n ← n + 1
    cluster ← clusterisable_clusters[n]
    X' ← X[cells in cluster]
    X' ← preprocess(X')
    subclusters ← find_clusters(X')
    for subcluster in subclusters:
      if is_valid_cluster(subcluster):
          clusterisable_clusters = clusterisable_clusters + subcluster
          assigned_cells += cells[subcluster]
      else
        subcluster is not a cluster, assign cells of subcluster to cluster
  done
  if assigned_cells / cells < 0.2 :
    clusterisable_clusters = clusterisable_clusters - subcluster in subclusters
  done
```

With ‘*minimum_number_of_genes*’ being the minimal number of differentially expressed genes to be considered different (5 by default), ‘*preprocess*’ the normalization and dimensionality reduction provided by the user, ‘*find_clusters*’ the clustering algorithm provided by the user.

IDclust processing and differential analysis functions

The framework underlying functions was implemented for either scRNA (cells x genes) or scEpigenomics data (cells x regions), but these functions can be supplied by the user if working with other categories of ‘single-cell omics.’ For scRNA, processing and clustering are performed by the ‘Seurat’ package, and for scEpigenomics, processing and clustering are performed by the ‘ChromSCape’ package. For both omics, the differential analysis was set as pseudobulk edgeR LRT, one of the top performing differential analysis methods in a recent differential analysis benchmark (12,13). If replicates are known, they can be specified as parameters to create pseudobulks; otherwise, pseudobulk replicates are created by grouping cells at random.

IDclust graphical functions

IDclust plots the iterative clustering through a network of clusters (nodes) separated by edges, where node size correlates with the number of cells in each cluster, and edge width represents the number of differentially expressed genes (DEGs) in the cluster. The central node contains all the cells, while the leaf nodes contain the final clusters. The node boundary can be either solid or dotted, indicating whether the cluster is present in the final set (solid line) or is a latent cluster that was fully resubclustered (dotted line).

Gene set enrichment analysis

Leveraging databases of gene sets, IDclust can automatically determine which cell type or gene set is specifically enriched in a given cluster using the DEGs found during clustering. This allows users to quickly obtain information about what cell type or in what cell stage might each cluster be. In this study, we used a combination of the PangLao, CellMarker and scTyper manually curated databases (14) and filtered them for the corresponding tissue type/organs.

Datasets for clustering algorithms benchmarking

To benchmark the various parameters of IDclust, we gathered datasets that were used in two recent clustering benchmarks to compare the clustering outputs of various algorithms with the ‘ground truths’ of known cell types, Duò *et al.* (15) and Yu *et al.* (16):

- The Koh/KohTCC (17), Kumar/KumarTCC/Kumar Simulation Easy/Kumar Simulation Easy (18), Trapnell/TrapnellTCC (19), Zhengmix/Zhengmix4eq/Zhengmix4uneq/Zhengmix48eq (20) dataset were the same as in Duò *et al.* (15) and retrieved using DuoClustering2018 Bioconductor package. ‘True cell types’ were taken in the ‘phenoid’ column.
- As in Yu *et al.* (16), the Tabula Muris (21) atlas from Droplet based technology or FACS-sorted technology was used to generate multiple types of datasets. The data were first downloaded from (Tabula Muris: Transcriptomic characterization of 20 organs and tissues from Mus musculus at single cell resolution) as Robjects. The cell type was composed of the concatenation of the organ and the ‘cell_ontology_class.’ The cell type with <200 cells were removed, and then each cell type was subsample to 200 cells. The following type of datasets were generated as in Yu *et al.*:
- TabulaMuris Droplet—‘balanced cell type’: A balanced mix of cell types from various organs (2–50 cell types).
- TabulaMuris FACS sorted—‘balanced cell type’: Same as the above for FACS-sorted datasets.
- TabulaMuris Droplet—‘unbalanced cell type’: An unbalanced mix of 10 cell types from various organs (10 cell types). In each batch, five cell types were subsampled progressively from 50 to 200 cells, 10 by 10 while the rest of the cell types were fixed to 200.
- TabulaMuris FACS sorted—“unbalanced cell type’: same as the above but with FACS-sorted data.

In addition to this, we generated a mix of organs:

- TabulaMuris Droplet—‘organs’: All the cell types of organs were mixed from 2 to 11 organs, in three replicates.
- TabulaMuris FACS sorted—‘organs’: Same as the above for FACS-sorted datasets.
- Each organ of the Tabula Sapiens dataset (22) was downloaded from cellxgene website as Seurat object (https://cellxgene.cziscience.com/collections/e5f58829-1a66-40b5-a624-9046778e74f5).

Benchmarking for clustering algorithms

To compare IDclust with state-of-the-art single-cell clustering methods, we ran Seurat and Monocle, among the best performing methods, on a recent benchmark 16 with default parameters. We also compared the results of TooManyCell, one of the most closely related clustering algorithms, which produces a hierarchical partition of cells. The ‘raw’ method is TooManyCells using the default parameters, and the ‘pruned’ method is described in the TooManyCells tutorial (https://gregoryschwartz.github.io/too-many-cells/) aimed at removing unwanted leaves using the parameters ‘–smart-cutoff 4 –min-size 1′. For Metacell2 and SEACells, we ran the default pipelines using recommended parameters respectively in https://github.com/dpeerlab/SEACells/blob/main/notebooks/SEACell_computation.ipynb and https://github.com/tanaylab/metacells-vignettes/blob/main/notebooks/one-pass.ipynb. Then, as these algorithms produce metacells, we thus further processed and clustered the metacells using Seurat default pipeline changing the number of neighbors during k-nearest neighbors (KNN) graph construction to $k = 2$ as the number of metacells is significantly lower than the number of cells.

Metrics to evaluate clustering results

To evaluate clustering solutions for datasets with known ground truth labels (cell types defined by the authors), we used two metrics: adjusted mutual information (AMI) and the adjusted Rand index (ARI). For real-world datasets, this approach is limited since authors usually rely on Louvain clustering to find clusters. However, we lack gold standards for large single-cell datasets and rely on in-depth exploration by the original authors.

The ARI is a corrected version of the Rand index (RI), which measures the similarity between two clusters by considering all pairs of samples and determining whether they are consistently assigned to the same or different clusters in both the predicted and true clusters. The raw RI score is then adjusted for chance to produce the ARI score using the following formula:

$$ARI = \frac{RI - E\,(RI)}{\max\,(RI) - E\,(RI)}$$

where RI is the pre-computed RI, and E (RI) E (RI) E (RI) is the expected RI.

MI is a measure of the mutual dependence between two variables. It quantifies the amount of information obtained about one variable through the other variable. For clustering, MI measures the agreement between the predicted clustering and the true clustering. The MI value is computed according to the following formula:

$$MI\,(U, V) = \sum_{i=1}^{|U|} \sum_{j=1}^{|V|} \frac{|U_I \cap V\,j|}{N} \log \frac{N|U_i \cap V\,j|}{|Ui||V\,j|}$$

where $Ui|Ui|Ui$ is the number of samples in cluster. $Ui|Ui|Ui$, $Vj|Vj|Vj$ is the number of samples in cluster $Vj|Vj|Vj$, and NNN is the total number of samples.

Results

Overview of IDclust

The workflow of IDclust is illustrated in Figure 1A, and the algorithm is described in detail in the 'Materials and methods' section. IDclust takes as input a single-cell matrix (cells x genes) in the form of a Seurat (4) or SingleCellExperiment (23) object. The framework first processes the matrix to find a coarse clustering, running the Louvain algorithm with a high number of neighbors (starting K number of neighbors = 100) (Figure 1A, panel 0). The algorithm recursively explores each cluster by reprocessing the dataset and finding subclusters within each original cluster (Figure 1A, panels 1–2). At each resolution, a 'one versus all' differential analysis is performed between all subclusters. This enables the distinction of relevant subclusters, i.e., those with meaningful differences compared to the rest of the cells, from clusters with no or too few differences. The irrelevant clusters, with insufficient differences, are then assigned to the parent cluster and are considered 'already explored' (Figure 1A, panels 3–4). The algorithm stops when no additional subclusters with significant markers are found within all unexplored clusters. IDclust discovers clusters with significant variations that are not identified by analyzing the entire dataset. Moreover, the marker features of clusters and subclusters are remembered and marked as 'clade-specific' or 'cluster-specific.' For a given parent cluster, the 'clade-specific' markers that were then found to be 'cluster-specific' in one of its child clusters were removed from the 'common parent markers' to keep only marker genes common to all child clusters.

The clustering is done by default with the Louvain algorithm. At the first iteration, we constrained the algorithm to divide the dataset into 2 to 6 clusters to begin with, in order to start with rather large entities with a reasonable number of cells in each cluster. For the next iterations, the K parameter used to generate the shared nearest neighbor (SNN) graph—a higher K will yield less but larger clusters—is calculated as a percentage of the number of cells but was constrained between 10 and 50 for computational reasons. The processing and differential analysis steps are entirely customizable for any omics with cell by features matrices but default functions are provided for scRNA and scEpigenomics (scATAC-seq, scCUT&Tag, scChIP-seq, scDNA methylation, …). For the processing of scRNA datasets, the Seurat package (4) with default parameters was used and ChromSCape package (24) was used for scEpigenomics. The default differential analysis for both omics methods is performed by the pseudobulk edgeR *glmLRT* (likelihood ratio test), one of the top performing differential analyses in two recent benchmarks (12,13). The phylogenies of the clusters can be reconstructed and visualized as illustrated in Figure 1B.

The steps of the algorithms described above are recapitulated by taking the example of 'red' cluster processing. Within this cluster, four subclusters were found, but the 'yellow' cluster did not harbor any significantly overexpressed markers compared to the other subclusters and was thus assigned to the 'red' parent cluster. The 'red' cluster is therefore not entirely reclustered and will be displayed in the graph with solid lines as opposed to dotted lines for transient clusters. In the example, two new clusters are found within the 'green' subcluster, but both do not have enough markers, and the algorithm assigns them to their 'green' parent cluster. In contrast, three new clusters with significant differences were found in the 'salmon' subcluster. Finally, the edge width reflects the number of significantly overexpressed markers in each subcluster, while the node area is proportional to the number of cells.

Overall, IDclust combines dimensionality reduction, clustering and identification of marker features in a single function. This approach guarantees that every cluster possesses marker genes that are significantly overexpressed. It proposes a different visualization than commonly used 2D representations, such as UMAP or t-SNE, in which cluster closeness can be better represented.

IDclust parameter sweep

Relying on two recent single-cell clustering benchmarks (Duò *et al.* (15) and Yu *et al.* (16)), we collected scRNA datasets of various nature: generated *in silico*, mixed *in silico*, fluorescence-activated cell sorting (FACS) sorted, droplet-based, human or mouse (17–21) (see methods for further details). The datasets comprised known ground truth (cell type)

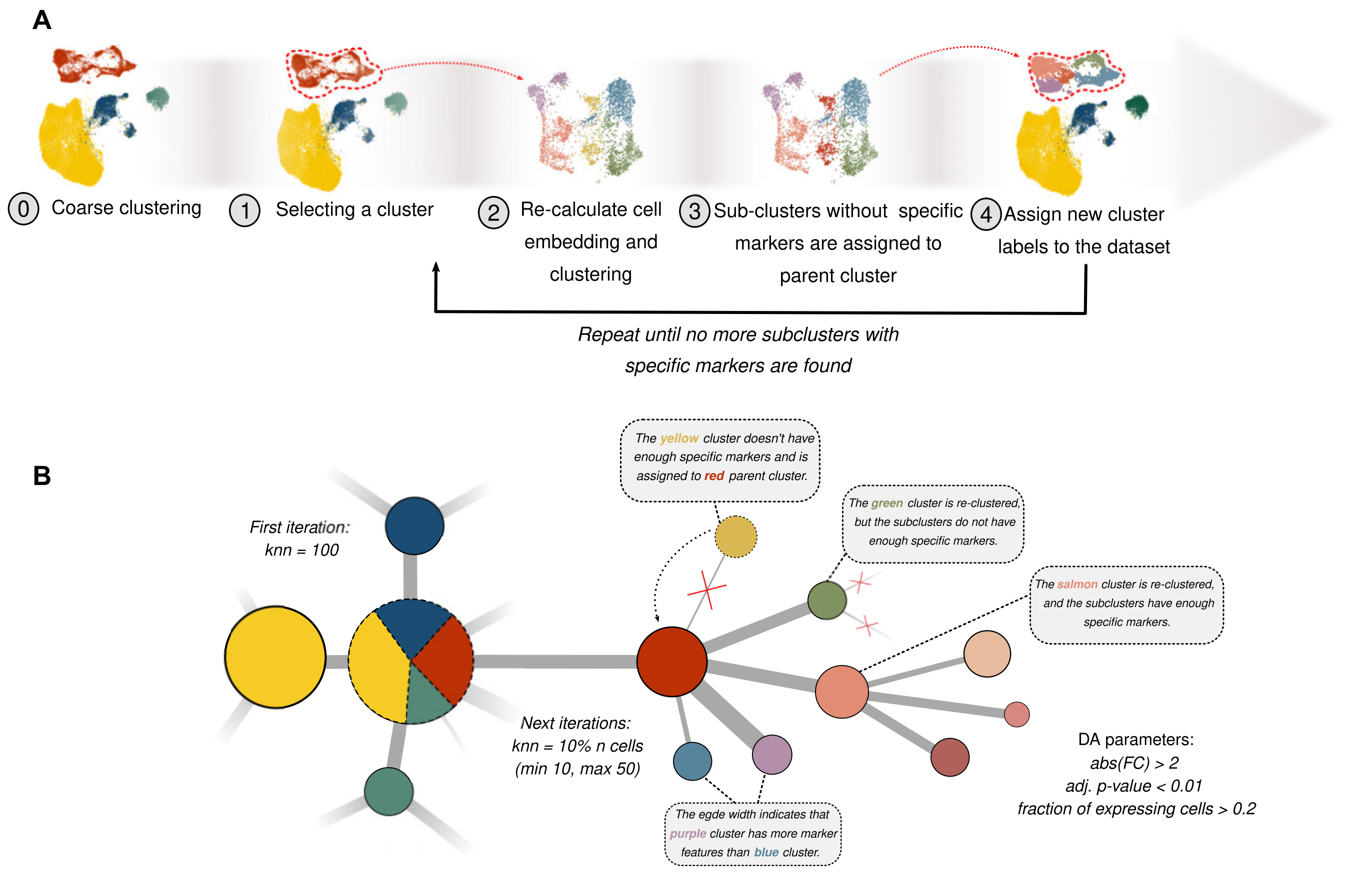

A
0 Coarse clustering
1 Selecting a cluster
2 Re-calculate cell embedding and clustering
3 Sub-clusters without specific markers are assigned to parent cluster
4 Assign new cluster labels to the dataset
Repeat until no more subclusters with specific markers are found
B
First iteration: knn = 100
Next iterations: knn = 10% n cells (min 10, max 50)
The yellow cluster doesn't have enough specific markers and is assigned to red parent cluster.
The green cluster is re-clustered, but the subclusters do not have enough specific markers.
The salmon cluster is re-clustered, and the subclusters have enough specific markers.
The egde width indicates that purple cluster has more marker features than blue cluster.
DA parameters:
abs(FC) > 2
adj. p-value < 0.01
fraction of expressing cells > 0.2

Figure 1. Schematic representation of the IDclust algorithm and tree visualization. (**A**) Schematic representation of the iterative differential clustering (IDclust) method used to identify clusters of cells in an unsupervised way. Cells are first processed and clustered at low resolution. Then, each cluster is recursively reprocessed and subclustered. Only biologically relevant clusters bearing a defined amount of significantly different features from the other subclusters are kept as 'true clusters.' The subclusters bearing not enough overexpressed markers are assigned to the parent cluster that is not reclustered. The iterative path of clustering and the significantly different genes define a hierarchy that is represented in the upper part. (**B**) Schematic representation of the hierarchical cluster tree visualization. Each node is a cluster, either transient (dotted line) or final (solid line). The hierarchical relationships describe the iterative clustering at each iteration of the algorithm. The pie chart represents the proportion of each subcluster in a given transient parent cluster but could also be colored according to known cell type or percentage of cells activating a feature of interest. The width of the edges is proportional to the number of marker features of the child cluster, and the size of the nodes is proportional to the number of cells in each cluster.

data. We first benchmarked the differential analysis parameters of IDclust in a non combinatorial way. The parameters tested were the FC, adjusted *P*-value, number of differential features required and fraction of expressing cells (Figure 2A; Supplementary Figure S1a and b). We used three metrics to compare performances across the parameter sweep: AMI, ARI and number of predicted versus ground-truth cell types. The AMI and ARI are standard evaluation metrics that compare two sets of annotation, the ground truth and predicted ones. We observed that the clustering was quite stable to varying differential analysis threshold changes. Therefore, we set the default parameters to the commonly used biological thresholds: FC $\geq$ 2, adjusted *P*-value $<$ 0.01 and an FC of 2, an adjusted *P*-value of 0.01 and a fraction of expressing cells of 20%.

Next, we benchmarked the clustering parameters that were also susceptible to influence on the final clustering: linear coefficient to calculate K neighbors for Louvain clustering, maximum K neighbors, K neighbors in the first round, resolution in the first round and resolution in the next rounds (Supplementary Figure S2a and b). Varying these parameters had a weak impact on the AMI and ratio of predicted clusters to true cell types, except for the resolution at the next rounds. Pushing the resolution at very low values, for example, 1e–05, decreased the AMI (Supplementary Figure S2a), and increasing the resolution at high values (>0.2) predicted too many clusters compared to the ground truth (Supplementary Figure S2b).

Overall, we observed that the algorithm performed better in terms of the AMI and ratio of predicted clusters versus ground truth cell types for all the parameters when the number of cell types was >10.

Benchmarking IDClust performances

We next compared the performance of IDclust with that of other state-of-the-art clustering tools, namely, Seurat, Monocle and TooManyCells. Seurat and Monocle provide clustering at one single resolution, while TooManyCells iteratively identifies clusters at various resolutions, such as IDclust. For TooManyCells, two default sets of parameters are proposed either by taking the graph as is ('TooManyCells_raw') or by pruning the graphs ('TooManyCells_pruned'). Additionally, we compared our method with Metacell2 (25) and SEACells

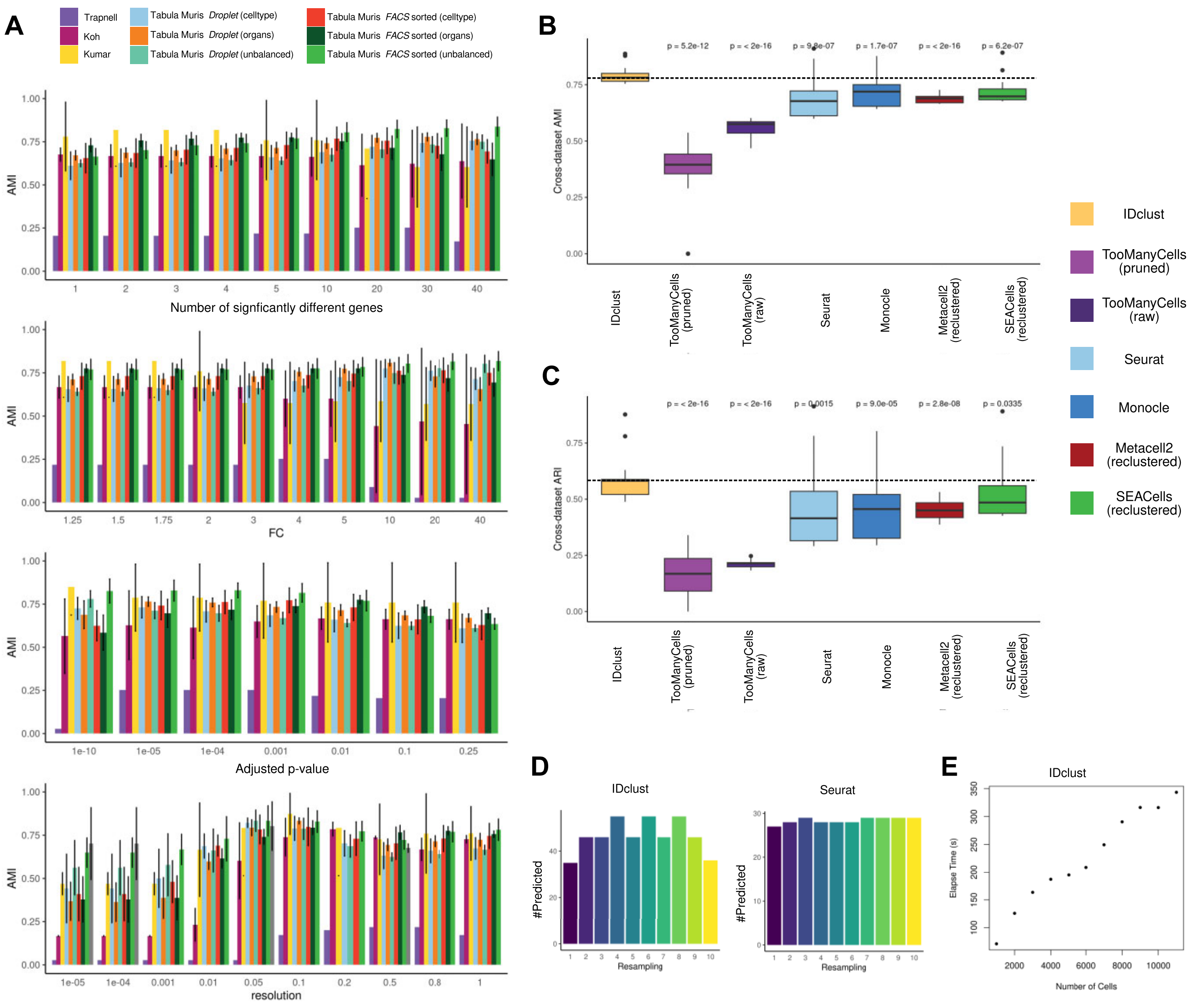


Figure 2. Comparing IDclust performances on Tabula Muris 'organs' datasets with those of other clustering tools. (**A**) AMI calculated between known cell types and clusters predicted by IDclust on datasets of variable numbers of cell types across different thresholds of adjusted *P*-value, FC, number of differential genes required and resolution of the iterative louvain clustering colored according to the different datasets. The Trapnell, Koh and Kumar datasets were downloaded from the DuoClustering2018 package, and the Tabula Muris dataset (droplet of FACS sorted) was divided into various cell types at random ('cell type'), unbalanced cluster composition ('unbalanced') or by mixing different organs together at random ('organs'). (**B**) Distribution boxplots of AMI calculated between known cell types and clusters on the 'TabulaMuris-organs' dataset of variable numbers of organs for IDclust, TooManyCells either by pruning or not pruning the leaves, Seurat, Monocle, Metacell2 and SEACells with default parameters, as Metacell2 and SEACells produce. Two-sided *T*-tests were used to compare all tools with IDclust. (**C**) Same as (B) but for ARI. (**D**) Stability of the clustering, depicted by the number of clusters predicted by randomly subsampling 80% of the cells in the Tabula Muris-Organs (10 Organs) dataset; this process was repeated 10 times. (**E**) Elapsed time of IDclust running on an increasing number of cells.

(26) which implement a high-granularity bottom-up approach aggregating single-cell into 'metacells,' for example, groups of cells with uniform profiles. These two algorithms are not clustering algorithms per se but rather aim at improving signal by aggregating cells together before downstream tasks such as clustering or trajectory analysis. This is why we re-clustered metacells obtained with both tools using Seurat with a lower number of neighbors during KNN-graph construction ($k = 2$) in order to compare clustering results.

Altogether, we showed that IDclust achieved overall comparable or better AMI (Figure 2B and Supplementary Figure S3a) and ARI (Figure 2C and Supplementary Figure S3b) than other algorithms and estimated the number of 'true' cell types with greater accuracy (Supplementary Figure S3c). Interestingly, the difference in performance increases when the complexity of the datasets increases. Too many cells is an iterative clustering algorithm similar to IDclust, but as shown in Supplementary Figure S3c, the raw version produces a very large number of clusters, which is not the case for IDclust. Pruning with default parameters produces fewer clusters than expected, and thus, optimization is needed.

We compared IDclust and Seurat with default parameters on a more recent dataset from which we did not estimate parameters from IDclust, the TabulaSapiens22 dataset containing 483 152 cells over 24 organs (Supplementary Figure S4a). We found that, compared with those of Seurat, the ARI and

AMI of 22/24 organs significantly and substantially increased (paired *T*-test, *P*-value = 2.3e–05 and 1e–05 for AMI and ARI, respectively). Although IDclust tends to predict slightly fewer clusters than defined by the authors, contrary to Seurat, it manages to not separate large homogeneous clusters (Supplementary Figure S4b and c).

Finally, we compare the stability of IDclust and Seurat by randomly subsampling 80% of the dataset 10 times. Overall, IDclust was slightly less stable than Seurat across subsamples (Sd = 7.15 and 0.70, respectively, for IDclust and Seurat) (Figure 2D). Finally, we benchmark the running time for an increasing number of cells (Figure 2E) and show that the running time scales linearly with the number of cells and that for a 10 000 cell dataset, IDclust runs in ∼5 min.

IDclust automatically reconstructs the hierarchy of the mouse skin and identified additional cell populations

To showcase the use of IDclust on a complex dataset, we ran IDclust with default parameters on the single-cell transcriptomics datasets from a SHARE-seq experiment on mouse skin (27). By comparing the clusters obtained ($n = 22$) with author-defined cell types ($n = 22$), we observed that the majority of cell types were automatically found by IDclust, with an AMI of 0.698 (Figure 3A). In addition, the IDclust reconstructs the hierarchy within the mouse skin and separates cell entities (Figure 3B and C), such as the epidermis (A1), the regenerative part (A2) and the permanent part (A3) of the hair follicle, and within the dermis, such as the hair papilla (A4) and the endothelial compartment (A5). IDclust was able to identify both rare cell types—dermal papilla, sebaceous gland, melanocytes and macrophage DCs (2.15%, 0.56%, 0.55% and 0.8%, respectively—and more common cell types—TAC-1 and granular epithelium (10.1% of all cells)). Additional pertinent information given by the network is the order in which the cells were clustered. For instance, cells from K6 + Bulge C. Layers emerge from the epithelium rather than the Bulge/Isthmus entity, indicating that its transcriptional profile might be closer to that of epithelial cells than cells from the Bulge/Isthmus. IDclust enables rapid visualization of the marker features supporting the separation of cell types, as exemplified in Figure 3D, with dopachrome tautomerase (Dct), an enzyme involved in melanin production and marking melanocytes, or collagen I (Col1a2), which marks dermal fibroblasts.

Owing to the hierarchical information between clusters, IDclust can retrieve marker genes with different expression patterns among clusters (Figure 3E): (i) genes that are expressed in multiple downstream clusters (labeled ‘A2 versus all’) and (ii) genes that are specifically expressed in individual clusters A2_B1 to A2_B5.

IDclust performs iterative clustering until no significant marker genes can be identified, independent of any human intervention. This can lead to the identification of potential new groups of cells within an annotated dataset. For example, we found two subpopulations within the originally annotated ‘Macrophage DC’ subtype (Figure 3F). Subcluster A5_B3_C1 was enriched for hallmark macrophage markers such as antigen F4/80 (Adgre1), Cd163, maltose receptor 1 (Mrc1) and colony stimulating factor 1 receptor (*Csf1r*) (28–30) while *A5_B3_C2* was enriched for T-cell-specific markers such as T-cell receptor zeta (Cd247) and thymocyte selection associated high mobility group box (Tox). To validate whether these marker lists were random or were synonymous with potential biological differences between cell populations, we predicted which transcription factor could drive the expression of these different genes according to the ChEA3 database (31). We showed that *Mafb*, a factor involved in macrophage lineage commitment (32) and *Tbx21* and *Tcf7* (33), two factors involved in activation and survival of lymphocyte T cells, are the most highly enriched TFs and could drive the expression programs of cells from A5_B3_C1 and A5_B3_C2, respectively (Figure 3G).

IDclust reveals progressive acquisition of open chromatin regions during differentiation of the mouse skin

To showcase the use of IDclust for other data types, we ran it on the corresponding single-cell ATAC-seq dataset of the same mouse skin dataset (27). IDclust allows the user to input the processing and differential functions to work with regions instead of genes. IDclust recapitulates most of the author’s annotations, which were achieved using both the RNA and ATAC signals, with an AMI of 0.543 (Figure 4A, $n = 31$ clusters). Similar to the scRNA dataset, the main cell entities were the epidermis (A2), the regenerative part (A4) and the permanent part (A3) of the hair follicle and the dermis layer (A1) (Figure 4B and C). This shows that the major cell entities are identical in terms of both the epigenome and the transcriptome.

IDclust identifies cases of gradual differentiation, where an entire cell population acquires new features without differentiating into different cell subpopulations. For example, during bulge/isthmus differentiation (A3), cells gradually acquire open chromatin features in the regenerated part of the hair follicle (Figure 4D). Successive generations retain features from their parents but also gain new open chromatin features. In addition, leveraging information from multiple cell marker databases, the cell clusters can be automatically annotated based on the number of markers found in the databases, as shown for the dermal portion of the dataset in Figure 4F. In this case, the ‘Dermal Sheath’ and ‘Dermal Papilla’ subtypes were not present in the database, which explains the incorrect annotations.

We then compared the IDclust results between RNA and ATAC. More clusters were found from an open chromatin ($n = 31$) than from a gene expression perspective ($n = 22$), showing that cells with similar expression profiles sometimes had distinct chromatin accessibility, for example, RNA cluster A1 (Figure 4F). Examining the ‘ORS’ cell type, IDclust revealed three clusters related to chromatin accessibility (one parent cluster, A4_B4 and two child clusters: A4_B4_C2 and A4_B4_C3) but only one in RNA (A2_B4) (Figure 4D). In ATAC, compared to the remaining A4 clusters, the parent cluster A4_B2 had 79 significantly more accessible regions, which were also accessible in the two child clusters (Figure 4D). Among these regions, the TSSs of 19 overlapping genes, 8 of which were found to be overexpressed in the corresponding RNA cluster A2_B4, showed agreement between chromatin openness and gene expression. However, from a chromatin accessibility perspective, the two child populations bear common and distinct open regions, while from a gene expression perspective, this looks like a unique homogenous population. These two child clusters with open chromatin might be popu-

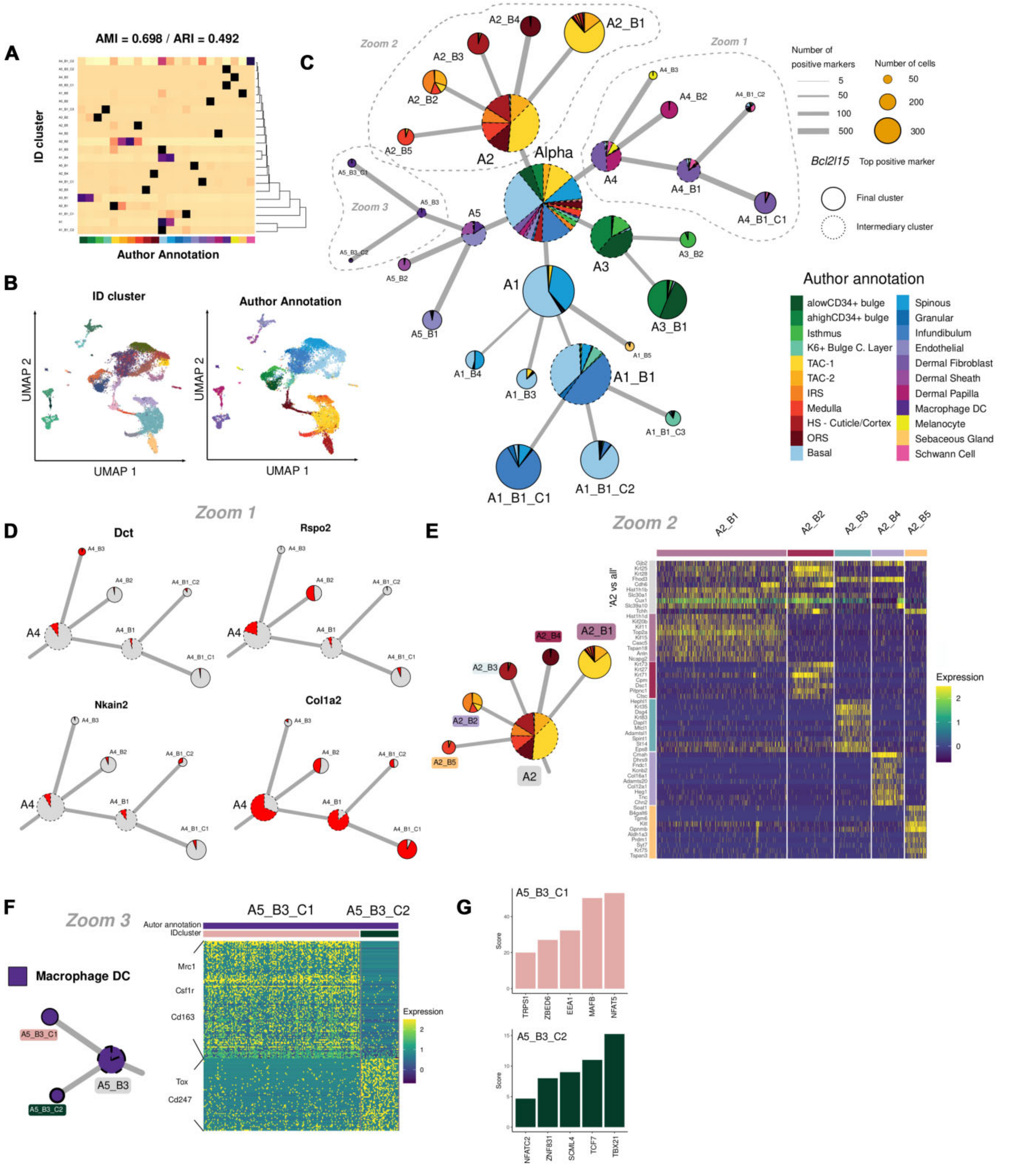


Figure 3. IDclust identifies meaningful clusters in an unsupervised way in mouse skin epigenomes (SHARE-seq-RNA). (**A**) Association heatmap of IDclust 22 clusters (rows) compared to author-defined 22 clusters (columns). Darker cells represent strong intersections, while lighter cells represent no intersections. (**B**) UMAP of the expression modality of SHARE-seq of mouse skin colored by IDclust clusters (upper panel) and author manual annotation (lower panel). (**C**) IDclust tree colored according to the author's original manual annotation for comparison. Each node is a cluster, with a solid line representing final clusters and a dashed line representing intermediary clusters that were fully subclusters. The size of the circles is proportional to the number of cells, and the width of the edges is proportional to the number of marker genes for the downstream node. Zooms are indicated for reference to (**D**–**F**). (D) Subset of the IDclust tree (Zoom1) colored according to one of the top marker genes identified during clustering. The red color in the pie charts represents the proportion of cells in a cluster active for the given feature. (E) (left) Subset of the IDclust tree (Zoom 1) colored according to the author annotation. (Right) Associated heatmap of the top 10 marker genes for each IDclust cluster. The first 10 markers (gray bar) are the top markers of transient cluster 'A2' compared to all other cells. (F) Same as (E) but for the author cell type 'Macrophage DC,' which was subclustered into 2 distinct clusters by IDclust (Zoom 1). (**G**) ChEA3 inverted mean rank score for transcription factor enrichment of marker genes in A5_B3_C1 (top) and A5_B3_C2 (bottom).

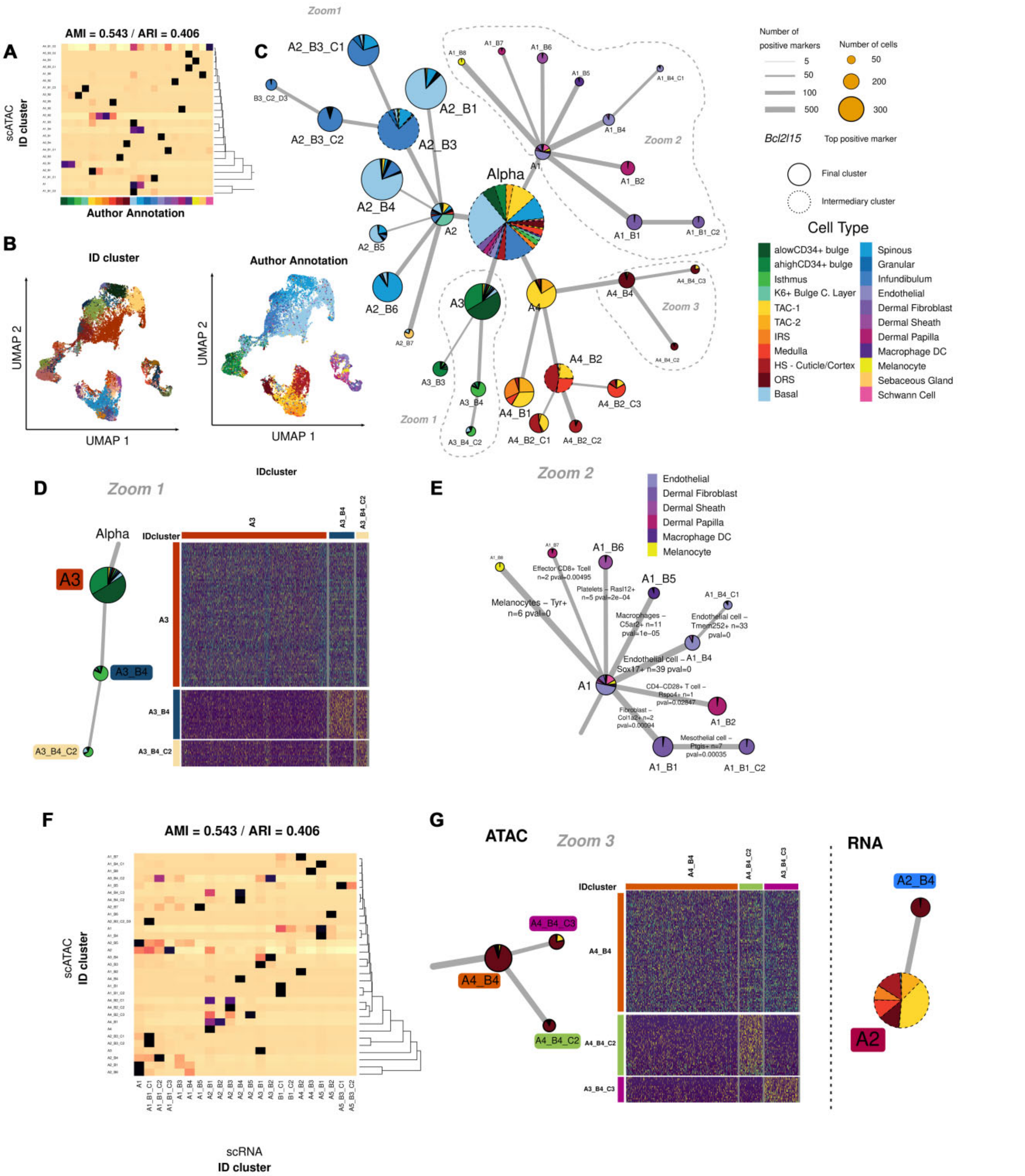


Figure 4. IDclust identifies meaningful clusters in an unsupervised way in mouse skin epigenomes (SHARE-seq-ATAC). (**A**) Association heatmap of IDclust 31 clusters (rows) compared to author-defined 22 clusters (columns). Darker cells represent strong intersections, while lighter cells represent no intersections. (**B**) UMAP of the ATAC modality of SHARE-seq of mouse skin colored by IDclust clusters (upper panel) and author annotation (lower panel). (**C**) IDclust tree colored by the proportion of cells in each cell type of the author annotation. Each node is a cluster, with a solid line representing final clusters and a dashed line representing intermediary clusters that were fully clustered. The size of the circles is proportional to the number of cells, and the width of the edges is proportional to the number of marker features for the downstream node. Zooms are indicated for reference to (**D–F**). (D) (left) Subset of the IDclust tree (Zoom 1) colored by author annotation. (Right) Associated heatmap of the top marker features for each IDclust cluster. (E) Subset of the IDclust tree (Zoom 2) colored by author annotation and with the top cell marker for each edge. The edges are annotated automatically with cell type using the enrichment of genes associated with marker features in combination with cell marker databases. The top most differentially expressed genes, the *P*-value associated with the hypergeometric enrichment test and the number of genes intersecting the list are also indicated (see 'Materials and methods' section). The number on the right indicates the number of differential markers supporting the match. (F) Association heatmap of IDclust ATAC-based clusters (rows) compared to IDclust RNA-based clusters (columns). Darker cells represent strong intersections, while lighter cells represent no intersections**.** (**G**) (left) Same as (D) but for Zoom 2, which contains a hierarchical acquisition of features. (Right) Corresponding subset of the RNA-based IDclust tree containing the outer root sheath (ORS) cell type.

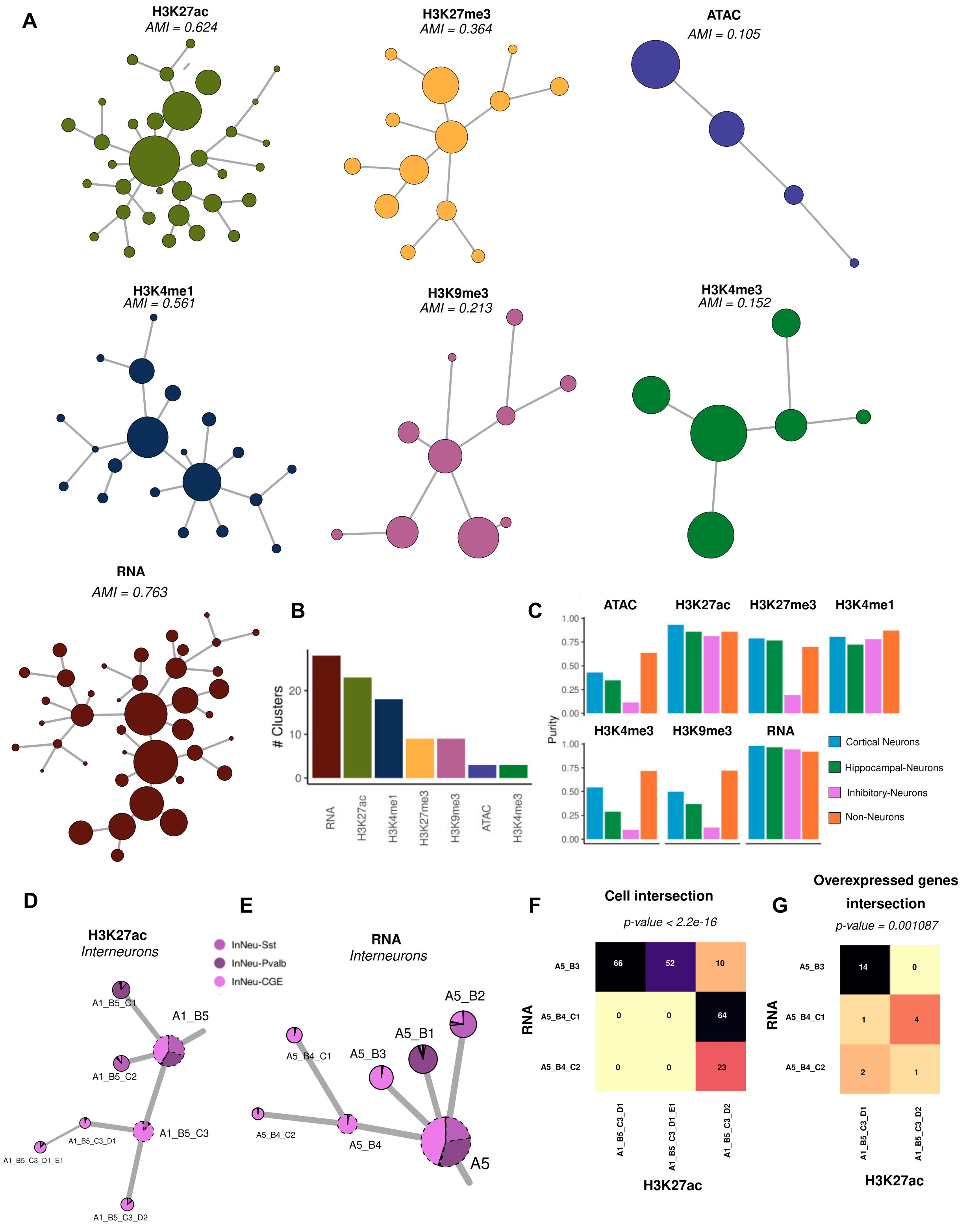


Figure 5. Granularity analysis of different histone marks in the mouse brain (paired-tag). (**A**) IDclust network of the paired-tag mouse brain dataset for various single-cell sequencing methods: RNA, open chromatin and histone marks, including H3K4me3, H3K27me3, H3K27ac, H3K4me1 and H3K9me3. The AMI was calculated between the IDclust clusters and author cell type annotations. (**B**) Total number of clusters found by IDclust in the various molecular profiles. (**C**) Distribution of the number of differential features per cluster in the various molecular profiles. (C) 'Within-cluster weighted purity' of the main brain cell clades, as defined by the authors, for each molecular profile. (**D**) IDclust subnetwork of the H3K27ac paired-tag mouse brain dataset focusing on the 'Interneuron' cell types contained within clade A1_B5. (**E**) Same as (D) but for the RNA part, focusing on the 'Interneuron' cell types contained within clade A5. In Neu-Sst, InNeu-Pvalb and InNeu-CGE (caudal ganglionic eminence) are author-defined clusters. (**F**) Association heatmap of H3K27ac- and RNA-defined clusters for cells specific to the 'CGE' author-defined cell type. The *P*-value was calculated with the chi-squared test, with the null hypothesis indicating the independence of variables. (**G**) Association heatmap of H3K27ac- and RNA-defined 'cluster-specific' marker genes common to both modalities. The *P*-value was calculated with the chi-squared test, with the null hypothesis indicating the independence of variables.

lations primed with different expression programs, which are not yet activated by transcription factors.

Cell type annotation and new cluster discovery using IDclust in the mouse brain (paired-tag)

IDclust provides an unbiased clustering approach to reveal all clusters with private molecular features within a dataset from a single-cell dataset. It provides a framework to compare the granularity of multiple omic datasets of one cell population. To illustrate this, we applied IDclust to the paired-tag mouse brain dataset (34) containing single-cell sequencing of RNA, open chromatin and histone marks, including H3K4me3, H3K27me3, H3K27ac, H3K4me1 and H3K9me3, on the same sample (Figure 5A). We found that in the mouse brain, the most granular molecular profiles were obtained via RNA and enhancer profiling with H3K27ac and H3K4me1 mapping consistent with what we previously found (35). In RNA, IDclust combined with gene set enrichment (see 'Materials and methods' section) was able to precisely retrieve the cell types defined by the authors (Supplementary Figure S5a). Interestingly, chromatin accessibility and the active promoter marker H3K4me3 presented the lowest overall granularity, which was lower than that of the repressive marks H3K27me3 and H3K9me3 (Figure 5B). By overlaying the four major 'cell clades' present in the mouse brain, as originally defined by the authors, for example, cortical, hippocampal, inhibitory neurons and non-neurons, we were able to distinguish general patterns (Figure 5C). For instance, overall, non-neurons were the easiest cells to differentiate from the rest, while inhibitory neurons were more often mixed with other cell types than were cortical or hippocampal neurons.

We observed that in both the H3K27ac and RNA profiles, the CGE interneuron cell type was divided into two distinct clusters (Figure 5D and E). To determine whether these clusters were consistent across the modalities, we selected the clusters of interest for each modality on the same cells. We found that the two main subclusters in RNA indeed significantly matched (chi-squared test, P-value < 2.2e-16) the two H3K27ac clusters with high purity (92.1% and 89.6% for A5_B3 and A5_B4, respectively) (Figure 5F). By comparing the marker genes associated with each cluster (Figure 5G), we found a significant overlap for the same pair of clusters (chi-squared test P-value = 0.001). Among the 14 genes present in the A5_B3-A1_B5_C3_D1 pair, multiple neuron-related genes, such as the Sidekick genes Sdk1 and Sdk2, which are involved in the formation and maintenance of neural circuits in the retina (36) and the Cannabinoid receptor Cnr1 were identified. This shows that this division of cells performed by IDclust *de novo* clusters is reliable, as it is found in a parallel manner in the two molecular profiles.

Discussion

We developed IDclust, a framework for unsupervised recursive differential clustering that relies on biologically relevant thresholds. Additionally, IDclust provides visualization tools, top marker genes, cell type and gene set annotation functionalities, allowing the user to make sense of the clusters. We have shown that the framework is applicable to multiple kinds of -omics, as the user can set its own clustering and differential analysis functions to be run by IDclust, but it has ready to use functions for single-cell transcriptomics and epigenomics. We provide the algorithm as an R package with simple instructions for users at https://github.com/vallotlab/IDclust.

Using IDclust, we were able to identify the main cell types in the scRNA, scATAC and scHistone datasets while also discovering new clusters with meaningful heterogeneity. In the scRNA dataset, we found a subcluster previously not found by the authors that expressed T-cell markers and was enriched for T-cell transcription factor cells. Additionally, IDclust hierarchically arranges the clusters and finds genes common to each branch as well as genes specific to each subcluster. This allows the biologist to obtain a two-step view of the marker genes to answer the following questions: what is specific to this branch, and what is specific to this subbranch? In the scATAC dataset, we found that IDclust reveals two different kinds of hierarchical structures: branching or nested clusters. In the branching hierarchy, the parent cluster has marker regions of both infants, each of which has specific markers. In the nested hierarchy, the infants progressively gained specific markers compared to their parents. Finally, in a challenging multiomics dataset such as Paired-tag, IDclust allows us to find and compare the granularity provided by different histone marks, showing that in the mouse brain, H3K27ac and H3K4me1 provide more granular views than ATAC or H3K4me3.

The limitations of this algorithm are the inner clustering parameters, which impact the accuracy and precision. This is because when 'zooming' on a cluster occurs, if the resolution of the Louvain algorithm is too high, many small clusters will be found, some of which differ from each other. If the resolution is too low, only 1 subcluster will be found, and the cluster will not be further divided. To palliate this, we force the first round of clustering to be between 2 and 6 so that the starting resolution is not too impactful. Additionally, the number of neighbors K needed to create the SNN graph for the Louvain algorithm decreases with the number of cells and is fixed between a minimum of $K = 10$ and $K = 50$. This allows a reasonable number of clusters depending on the number of cells. The biological limitation of this approach is that within each cell cluster, the cell types may ultimately be separated by cell phase (G1, S and G2/M).

Overall, we believe that IDclust is a useful tool for deciphering complex datasets that will enable scientists to find clusters based on biologically meaningful parameters and enable easier interpretation of the clustering results. Moreover, since IDclust reprocesses embedding for each iteration, it finds sources of variation that would be too subtle when looking at the whole dataset. These subtle but real variations may be overshadowed by the largest differences, but are important for fine clustering. This tool will help researchers automatically find all the populations that are significantly different from each other and then refine the clustering by interpreting the gene sets enriched in each cluster. Finally, IDclust's hierarchy of clusters represents a useful representation of the true link between cell populations.

Data availability

The R package IDclust, tutorials and scripts used for this manuscript are publicly available on GitHub (https://github.com/vallotlab/IDclust) under the GPL-3 license. The code for the analysis of the data presented in this paper can be found on Zenodo (https://doi.org/10.5281/zenodo.12772866). The mouse TabulaMuris (21) scRNA-seq dataset is downloaded from Tabula Muris: Transcriptomic characterization of 20 organs and tissues from Mus musculus at single cell resolution. The human organs dataset from the Tabula Sapiens dataset (22) was downloaded from cellxgene website as Seurat object (https://cellxgene.cziscience.com/collections/e5f58829-1a66-40b5-a624-9046778e74f5). The Koh/KohTCC (17), Kumar/KumarTCC/Kumar Simulation Easy/Kumar Simulation Easy (18), Trapnell/TrapnellTCC (19) and Zhengmix/Zhengmix4eq/Zhengmix4uneq/Zhengmix48eq (20) dataset were retrieved using DuoClustering2018 Bioconductor package. The paired-tag (34) dataset for all six modalities was downloaded from GSE152020.

Supplementary data

Supplementary Data are available at NARGAB Online.

Acknowledgements

We thank Hélène Salmon and Abdenour Abbas for discussion about cell type identification.

Author Contributions: P.P. and C.V. conceived the study. P.P. performed the method development and computational analysis. M.S. and F.R. tested the tool. P.P. and C.V. wrote the manuscript together with input from all authors.

Funding

SiRIC-Curie program [INCa-DGOS-4654, INCa-DGOS-Inserm-12554 to C.V.]; H2020 program [948528-ChromTrace to C.V.].

Conflict of interest statement

None declared.

References

1. Trapnell,C. (2015) Defining cell types and states with single-cell genomics. *Genome Res.*, **25**, 1491–1498.
2. (2017) What is your conceptual definition of ‘cell type’ in the context of a mature organism? *Cell Syst*, **4**, 255–259.
3. Cao,J., Spielmann,M., Qiu,X., Huang,X., Ibrahim,D.M., Hill,A.J., Zhang,F., Mundlos,S., Christiansen,L., Steemers,F.J., *et al.* (2019) The single-cell transcriptional landscape of mammalian organogenesis. *Nature*, **566**, 496–502.
4. Hao,Y., Hao,S., Andersen-Nissen,E., Mauck,W.M., Zheng,S., Butler,A., Lee,M.J., Wilk,A.J., Darby,C., Zager,M., *et al.* (2021) Integrated analysis of multimodal single-cell data. *Cell*, **184**, 3573–3587.
5. Wolf,F.A., Angerer,P. and Theis,F.J. (2018) SCANPY: large-scale single-cell gene expression data analysis. *Genome Biol.*, **19**, 15.
6. Blondel,V.D., Guillaume,J.-L., Lambiotte,R. and Lefebvre,E. (2008) Fast unfolding of communities in large networks. *J. Stat. Mech. Theory Exp.*, **2008**, P10008.
7. Liu,S., Thennavan,A., Garay,J.P., Marron,J.S. and Perou,C.M. (2021) MultiK: an automated tool to determine optimal cluster numbers in single-cell RNA sequencing data. *Genome Biol.*, **22**, 232.
8. Kiselev,V.Y., Andrews,T.S. and Hemberg,M. (2019) Challenges in unsupervised clustering of single-cell RNA-seq data. *Nat. Rev. Genet.*, **20**, 273–282.
9. Wilkerson,M.D. and Hayes,D.N. (2010) ConsensusClusterPlus: a class discovery tool with confidence assessments and item tracking. *Bioinformatics*, **26**, 1572–1573.
10. Zappia,L. and Oshlack,A. (2018) Clustering trees: a visualization for evaluating clusterings at multiple resolutions. *GigaScience*, **7**, giy083.
11. Schwartz,G.W., Zhou,Y., Petrovic,J., Fasolino,M., Xu,L., Shaffer,S.M., Pear,W.S., Vahedi,G. and Faryabi,R.B. (2020) TooManyCells identifies and visualizes relationships of single-cell clades. *Nat. Methods*, **17**, 405–413.
12. Soneson,C. and Robinson,M.D. (2018) Bias, robustness and scalability in single-cell differential expression analysis. *Nat. Methods*, **15**, 255–261.
13. Squair,J.W., Gautier,M., Kathe,C., Anderson,M.A., James,N.D., Hutson,T.H., Hudelle,R., Qaiser,T., Matson,K.J.E., Barraud,Q., *et al.* (2021) Confronting false discoveries in single-cell differential expression. *Nat. Commun.*, **12**, 5692.
14. Franzén,O., Gan,L.-M. and Björkegren,J.L.M. (2019) PanglaoDB: a web server for exploration of mouse and human single-cell RNA sequencing data. *Database*, **19**, baz046.
15. Duò,A., Robinson,M.D. and Soneson,C. (2018) A systematic performance evaluation of clustering methods for single-cell RNA-seq data. *F1000Res.*, **7**, 1141.
16. Yu,L., Cao,Y., Yang,J.Y.H. and Yang,P. (2022) Benchmarking clustering algorithms on estimating the number of cell types from single-cell RNA-sequencing data. *Genome Biol.*, **23**, 49.
17. Koh,P.W., Sinha,R., Barkal,A.A., Morganti,R.M., Chen,A., Weissman,I.L., Ang,L.T., Kundaje,A. and Loh,K.M. (2016) An atlas of transcriptional, chromatin accessibility, and surface marker changes in human mesoderm development. *Sci. Data*, **3**, 160109.
18. Kumar,R.M., Cahan,P., Shalek,A.K., Satija,R., Jay DaleyKeyser,A., Li,H., Zhang,J., Pardee,K., Gennert,D., Trombetta,J.J., *et al.* (2014) Deconstructing transcriptional heterogeneity in pluripotent stem cells. *Nature*, **516**, 56–61.
19. Trapnell,C., Cacchiarelli,D., Grimsby,J., Pokharel,P., Li,S., Morse,M., Lennon,N.J., Livak,K.J., Mikkelsen,T.S. and Rinn,J.L. (2014) The dynamics and regulators of cell fate decisions are revealed by pseudotemporal ordering of single cells. *Nat. Biotechnol.*, **32**, 381–386.
20. Zheng,G.X.Y., Terry,J.M., Belgrader,P., Ryvkin,P., Bent,Z.W., Wilson,R., Ziraldo,S.B., Wheeler,T.D., McDermott,G.P., Zhu,J., *et al.* (2017) Massively parallel digital transcriptional profiling of single cells. *Nat. Commun.*, **8**, 14049.
21. The Tabula Muris Consortium, Overall coordination, Logistical coordination, Organ collection and processing, Library preparation and sequencing, Computational data analysis, Cell type annotation, Writing group, Supplemental text writing group, and Principal investigators (2018) Single-cell transcriptomics of 20 mouse organs creates a Tabula Muris. *Nature*, **562**, 367–372.
22. The Tabula Sapiens Consortium, Jones,R.C., Karkanias,J., Krasnow,M.A., Pisco,A.O., Quake,S.R., Salzman,J., Yosef,N., Bulthaup,B., Brown,P., *et al.* (2022) The Tabula Sapiens: a multiple-organ, single-cell transcriptomic atlas of humans. *Science*, **376**, eabl4896.
23. Amezquita,R.A., Lun,A.T.L., Becht,E., Carey,V.J., Carpp,L.N., Geistlinger,L., Marini,F., Rue-Albrecht,K., Risso,D., Soneson,C., *et al.* (2020) Orchestrating single-cell analysis with Bioconductor. *Nat. Methods*, **17**, 137–145.
24. Prompsy,P., Kirchmeier,P., Marsolier,J., Deloger,M., Servant,N. and Vallot,C. (2020) Interactive analysis of single-cell epigenomic landscapes with ChromSCape. *Nat. Commun.*, **11**, 5702.
25. Ben-Kiki,O., Bercovich,A., Lifshitz,A. and Tanay,A. (2022) Metacell-2: a divide-and-conquer metacell algorithm for scalable scRNA-seq analysis. *Genome Biol.*, **23**, 100.
26. Persad,S., Choo,Z.-N., Dien,C., Sohail,N., Masilionis,I., Chaligné,R., Nawy,T., Brown,C.C., Sharma,R., Pe’er,I., *et al.*

(2023) SEACells infers transcriptional and epigenomic cellular states from single-cell genomics data. *Nat. Biotechnol.*, **41**, 1746–1757.

27. Ma,S., Zhang,B., LaFave,L.M., Earl,A.S., Chiang,Z., Hu,Y., Ding,J., Brack,A., Kartha,V.K., Tay,T., *et al.* (2020) Chromatin potential identified by shared single-cell profiling of RNA and Chromatin. *Cell*, **183**, 1103–1116.
28. Hume,D.A. (2015) The many alternative faces of macrophage activation. *Front. Immunol.*, **6**, 370.
29. Mosser,D.M. and Edwards,J.P. (2008) Exploring the full spectrum of macrophage activation. *Nat. Rev. Immunol.*, **8**, 958–969.
30. Fabriek,B.O., Dijkstra,C.D. and van den Berg,T.K. (2005) The macrophage scavenger receptor CD163. *Immunobiology*, **210**, 153–160.
31. Keenan,A.B., Torre,D., Lachmann,A., Leong,A.K., Wojciechowicz,M.L., Utti,V., Jagodnik,K.M., Kropiwnicki,E., Wang,Z. and Ma'ayan,A. (2019) ChEA3: transcription factor enrichment analysis by orthogonal omics integration. *Nucleic Acids Res.*, **47**, W212–W224.
32. Sarrazin,S., Mossadegh-Keller,N., Fukao,T., Aziz,A., Mourcin,F., Vanhille,L., Kelly Modis,L., Kastner,P., Chan,S., Duprez,E., *et al.* (2009) MafB restricts M-CSF-dependent myeloid commitment divisions of hematopoietic stem cells. *Cell*, **138**, 300–313.
33. Szabo,S.J., Kim,S.T., Costa,G.L., Zhang,X., Fathman,C.G. and Glimcher,L.H. (2000) A novel transcription factor, T-bet, directs Th1 lineage commitment. *Cell*, **100**, 655–669.
34. Zhu,C., Zhang,Y., Li,Y.E., Lucero,J., Behrens,M.M. and Ren,B. (2021) Joint profiling of histone modifications and transcriptome in single cells from mouse brain. *Nat. Methods*, **18**, 283–292.
35. Raimundo,F., Prompsy,P., Vert,J.-P. and Vallot,C. (2023) A benchmark of computational pipelines for single-cell histone modification data. *Genome Biol.*, **24**, 143.
36. Yamagata,M. (2020) Structure and functions of sidekicks. *Front. Mol. Neurosci.*, **13**, 139.

Received: August 8, 2024. **Revised:** November 8, 2024. **Editorial Decision:** November 18, 2024. **Accepted:** November 19, 2024