OXFORD

Briefings in Bioinformatics, 2023, **24(3)**, 1–13
https://doi.org/10.1093/bib/bbad104
Advance access publication date 20 March 2023
Problem Solving Protocol

SoCube: an innovative end-to-end doublet detection algorithm for analyzing scRNA-seq data

Hongning Zhang†, Mingkun Lu†, Gaole Lin, Lingyan Zheng, Wei Zhang, Zhijian Xu and Feng Zhu

Corresponding author. Feng Zhu, Polytechnic Institute, The Second Affiliated Hospital, College of Pharmaceutical Sciences, Zhejiang University School of Medicine, Zhejiang University, Hangzhou 310058, China. Tel.: +86-189-8946-6518; Fax: +86-571-8820-8444; E-mail: zhufeng@zju.edu.cn

†The authors wish it to be known that, in their opinion, should be regarded as Joint First Authors.

Abstract

Doublets formed during single-cell RNA sequencing (scRNA-seq) severely affect downstream studies, such as differentially expressed gene analysis and cell trajectory inference, and limit the cellular throughput of scRNA-seq. Several doublet detection algorithms are currently available, but their generalization performance could be further improved due to the lack of effective feature-embedding strategies with suitable model architectures. Therefore, SoCube, a novel deep learning algorithm, was developed to precisely detect doublets in various types of scRNA-seq data. SoCube (i) proposed a novel 3D composite feature-embedding strategy that embedded latent gene information and (ii) constructed a multikernel, multichannel CNN-ensembled architecture in conjunction with the feature-embedding strategy. With its excellent performance on benchmark evaluation and several downstream tasks, it is expected to be a powerful algorithm to detect and remove doublets in scRNA-seq data. SoCube is freely provided as an end-to-end tool on the Python official package site PyPi (https://pypi.org/project/socube/) and open-source on GitHub (https://github.com/idrblab/socube/).

Keywords: scRNA-seq, doublet detection, omics, feature embedding

INTRODUCTION

High-throughput single-cell RNA sequencing (scRNA-seq) has emerged as a widely applicable and powerful technology that has evolved into a vital tool enabling revolutionary discoveries in biomedical sciences [1–7]. However, the existence of doublets (or multiplets), which contain two or multiple cells in a droplet due to technical defects [8], seriously interfere with single-cell transcriptomics studies in several ways. First, doublets change the real distribution of cells and genes, which is vital for many downstream tasks, such as cell clustering; second, doublets limit the cell throughput of scRNA-seq because the proportion of doublets in droplets, which is positively correlated with cell concentration, follows a Poisson distribution [9]. There are some experimental techniques to detect doublets, such as cell hashing (doublets are droplets whose barcodes are associated with more than one oligo-tagged antibody) [10], Demuxlet (doublets are droplets whose barcodes are associated with mutually exclusive sets of SNPs) [11], Species mixture (doublets are droplets whose barcodes are associated with more than one species) [12] and MULTI-seq (doublets are droplets whose barcodes are associated with more than one lipid-tagged index) [13].

However, the experimental techniques have many limitations. First, the experimental techniques often require special experimental preparation, long experimental periods and extra cost [14]. Second, these techniques are not applicable in all scenarios. Identically, for cell hashing, if the cell of interest does not express the surface proteins we need, it will result in failure to assign each cell to its original sample [10]. Additionally, some techniques cannot detect doublets formed by cells from the same sample indices. More importantly, all these experimental techniques cannot be applied to existing scRNA-seq datasets from public platforms, such as GEO [15], DISCO [16], LINCS 1000 [17] and KPMP atlas [18], which are widely used by biological data scientists who often do not have enough experimental resources.

Hongning Zhang is working toward the M.S. degree in project "Artificial Intelligence in Medicine" at Polytechnic Institute, Zhejiang University. He received the B.S. degree in Pharmaceutical Science from Zhejiang University in 2020. His research interests include machine learning and deep learning with applications to single cell omics, protein and antibody design, and RNA interaction.

Mingkun Lu is working toward the M.S. degree in project "Artificial Intelligence in Medicine" at Polytechnic Institute, Zhejiang University. He is Research Assistant of Innovation Institute for Artificial Intelligence in Medicine, Zhejiang University. His research interests include protein function and structure prediction, multi-omics integration, and drug-target interaction prediction.Gaole Lin is working toward the B.S. degree at College of Pharmaceutical Sciences, Zhejiang University. He is Research Assistant of Innovative Drug Research and Bioinformatics Group. His research interests include scRNA-seq analysis and bulk RNA-seq analysis.

Lingyan Zheng is Ph.D. candidate in Medicine at University School of Medicine, Zhejiang University. She is Research Assistant of Alibaba-Zhejiang University Joint Research Center of Future Digital Healthcare. Her research interests include deep-learning-based protein function and structure prediction, protein drug design and optimization, and single cell proteomics.

Wei Zhang is working toward the M.S. degree at College of Pharmaceutical Sciences, Zhejiang University. His research interests include machine learning and deep learning with applications to multiomics.

Zhijian Xu is Researcher of Shanghai Institute of Materia Medica, Chinese Academy of Sciences. He is Co-mentor of "Artificial Intelligence in Medicine" at Polytechnic Institute, Zhejiang University. His research interests include drug design, Computational Chemistry and Computational Biology.

Feng Zhu is a tenured professor of College of Pharmaceutical Sciences in Zhejiang University, China. He is principal investigator of Innovative Drug Research and Bioinformatics Group (https://idrblab.org/) and this group has been working in the fields of AI-aided drug discovery and published many relative work.

Received: January 10, 2023. **Revised:** February 17, 2023. **Accepted:** February 28, 2023

Therefore, the development of automatic doublet detection computational algorithms has received increasing attention [7, 18–22].

Currently, some computational doublet detection algorithms have been developed based on distinct algorithms [12, 23–29]. According to their different feature-embedding strategies and prediction models, the above algorithms can be divided into two categories: traditional machine learning algorithms and deep learning algorithms. Traditional machine learning algorithms, with DoubletFinder (DF) [24] and scDblFinder [30] as representative algorithms, mainly use principal component analysis (PCA) for feature embedding and then use kNN [24], gradient boosting [25] or other algorithms to detect doublets. Deep learning algorithms represented by Solo [23] mainly use Variational Autoencoder (VAE), implemented in the Python package scVI [31], for feature vector embedding and then detect doublets by deep neural network. It is common sense that feature embedding plays a vitally important role in omics tasks [32–37]. However, the feature-embedding strategies of the above doublet detection algorithms have two limitations: (i) biological associations between different genes, such as gene regulation, are important potential features [38], but PCA and VAE lack the ability to characterize gene associations while also lacking biological interpretation; and (ii) the feature dimensionality of scRNA-seq is often very high [39], and reduction by PCA or VAE is a normal strategies to cope with the 'curse of dimensionality' [32] but results in the loss of much useful information. Studies in bulk RNA-seq have proven that 2D matrix representations can better preserve information while alleviating the 'curse of dimensionality' compared to the 1D vector embedded by PCA [39].

In our research, a novel deep learning algorithm, SoCube, was proposed to detect doublets from a user-given scRNA-seq UMI matrix, and previous limitations were addressed by innovation in feature-embedding strategy and model architecture (workflow is shown in Figure 1). For feature-embedding innovation, SoCube is the first to apply a 3D feature-embedding strategy to scRNA-seq, which works by embedding gene biological associations into the first two dimensions and embedding gene-specific features into the third dimension. This strategy alleviates the 'curse of dimensionality' without losing the original information but also mines new information with better biological interpretation, such as biological associations of genes. For model architecture innovation, SoCube innovates with an ensemble model architecture based on a multikernel, multichannel convolutional neural network (CNN) in conjunction with the above novel feature-embedding strategy. With state-of-the-art (SOTA) generalization performance on benchmark evaluation and several downstream tasks, it is expected to be a powerful tool to detect doublets in scRNA-seq data.

MATERIALS AND METHODS

Workflows of SoCube

Feature embedding

Before embedding, genes with a mean expression less than 0.05 and droplets (cells) with library sizes less than 1000 were filtered to reduce the sparsity and dimensions of the raw scRNA matrix (droplet removal is only for feature embedding, these droplets still remain unless they are detected as doublets) [40]. Feature-embedding strategy is illustrated in Figure 2. The pairwise distances of genes were calculated based on the differential intercellular expression of genes. The distance metric is a hyperparameter, and the default setting is the Pearson correlation distance. Based on these pairwise distances, the genes were projected onto a 2D feature space as feature points by using UMAP [41].

These feature points embed the broadly learned correlation relationships of genes. The points were further assigned to a 2D grid gene map (*Gmap*) by using the Jonker-Volgenant (J-V) algorithm, which was implemented in the Python package lapjv v1.3.1 to solve the linear assignment problem [42]. In the process of building *Gmap*, g genes corresponding to the required grid with width $m = \lceil \sqrt{g} \rceil$ and height $n = \lceil \frac{g}{m} \rceil + 1$. The purpose of using J-V algorithm is to establish the optimal mapping between the UMAP feature space U of genes and the grid space G, and a grid in the *Gmap* represents the position of its corresponding gene in 2D space. Thus, 2D *Gmap* maintains the broadly learned correlation relationships of genes. As the number of grids will be slightly larger than the number of genes in most of the cases, the unmatched grids will be filled with zero.

Meanwhile, the genes' low-dimension embedding vectors (*Gvectors*) were calculated by using the PCA algorithm (default target dimension setting is 10). *Gmap* represents gene biological associations, while *Gvectors* represents gene specificity-based multichannel information. Finally, *Gmap* ($m \times n$) and *Gvectors* (d) were projected onto a 3D complex embedding (*G3D*, $m \times n \times d$) feature space, that is, *Gvectors* were assigned to the position of their corresponding genes according to the gene arrangement in *Gmap*. *G3D* is droplet-independent universal feature embedding, and each droplet's 3D feature embedding was calculated by multiplying its gene expression value with *G3D*.

Doublet simulation

Doublet simulation is required to perform doublet detection task because this task is not suitable for directly using models trained on other datasets, and thus SoCube provides three different doublet simulation strategies: balance simulation strategy, heterotypic-doublet-first simulation (HEFS) strategy and homotypic-doublet-first simulation (HOFS) strategy. Balance simulation strategy generates *in silico* doublets by taking the sum of two randomly chosen observed droplets, which is typical strategy widely used [12, 23, 24]. Suppose the number of *in silico* doublet is N, HEFS strategy randomly chooses $5N$ pairs of droplets, sorts them in descending order based on Pearson correlation distance, and selects the top N pairs of droplets to generated doublets. HOFS strategy works similarly with HEFS but sorts droplet pairs in ascending order (or selects the last N pairs). Doublets generated by HEFS contain more heterotypic doublets, which formed by cells from different cell types, and doublets generated by HOFS contain more homotypic doublets, which contain only one cell type. Balance simulation chooses droplets without any bias and generated doublets' number will be more balanced between the types. Users could select different simulation strategy according to their dataset features. Besides, SoCube provides an option to set the scaling factor (default is 1.0) for the ratio of the doublet unique molecular identifiers (UMI) level to the singlet UMI level if users have reason to believe that the doublet's UMI is less than twice the singlet's UMI on average. N *in silico* doublets were generated and mixed with observed droplets as the training set. N was calculated based on the proportion setting (default is 1.0) of positive and negative samples in the training set.

Model fitting and doublet removal

A multikernel, multichannel CNN-ensembled architecture, SoCubeNet, was constructed for the doublet detection task

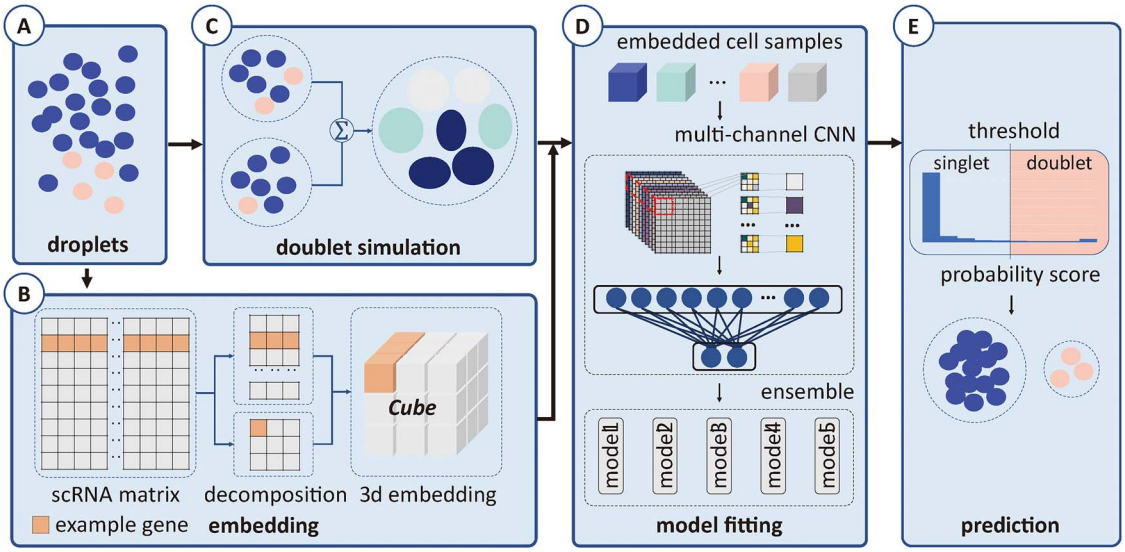

Figure 1. Overview of the SoCube workflow. (**A**) Input: scRNA-seq data mixed with singlets (marked as blue) and doublets (marked as orange). (**B**) Droplet embedding: each droplet's 1D gene expression vector is embedded into a 3D feature (SoCube) with gene similarity and uniqueness representation. (**C**) Doublet simulation: doublets used for model fitting are generated by taking the sum of randomly chosen observed droplets (regarded as putative singlets). The mixture of *in silico* doublets and putative singlets is used for model fitting. (**D**) Model fitting: a multikernel, multichannel CNN-ensembled architecture was designed for fitting embedded droplet data. An ensemble learning strategy was used to correct the bias result from dataset division. (**E**) Doublet prediction and removal: droplets whose probability scores are greater than the threshold will be regarded as doublets and removed.

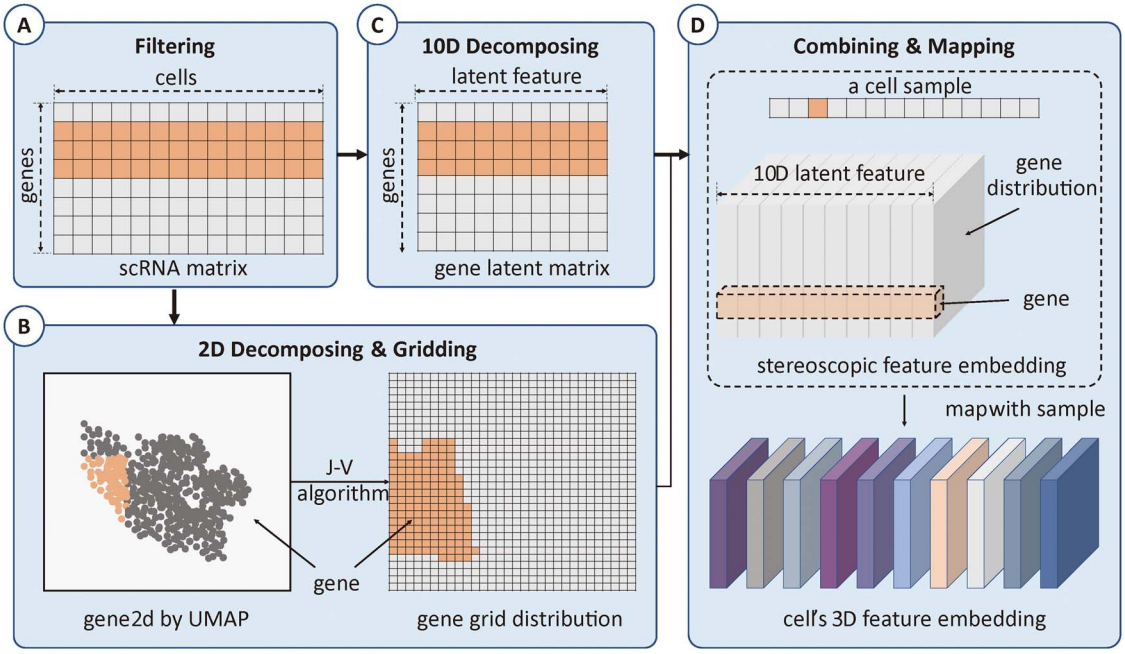

Figure 2. Procedure of droplet embedding. (**A**) Data filtering: genes with low mean expression (≤0.05) and droplets with low library size (≤1000) will be filtered. (**B**) 2D decomposing and gridding: filtered scRNA-seq will be projected onto a 2D feature space by UMAP and then assigned to a 2D grid map by the J-V algorithm and the unmatched grids will be filled with zero. (**C**) 10D decomposition: gene 10D latent features will be calculated by PCA. (**D**) Combining and droplet mapping: gene 10D latent features and a gene 2D grid map will be combined into a 3D embedding. Each droplet will be mapped into 3D embedding according to the gene expression value.

(illustrated in Figure 3). The base learner (also named as basic model) is formed with a feature extraction layer and classification layer. The feature extraction layer mainly contains two InceptionV1 blocks. Each block has three multichannel convolution kernels with different kernel sizes. Such a block can integrate multichannel input from droplets' 3D embedding features under different receptive fields. The classification layer is formed with three fully connected layers. Latent features extracted automatically by the feature extraction layer are passed to the classification layer, and the probability score of doublets is output. The binary cross entropy (BCE) defined in equation (1) was used as the loss function. The Adam optimizer was used to minimize the loss value, the initial learning rate was 0.001, and an exponential learning rate decay strategy was used with

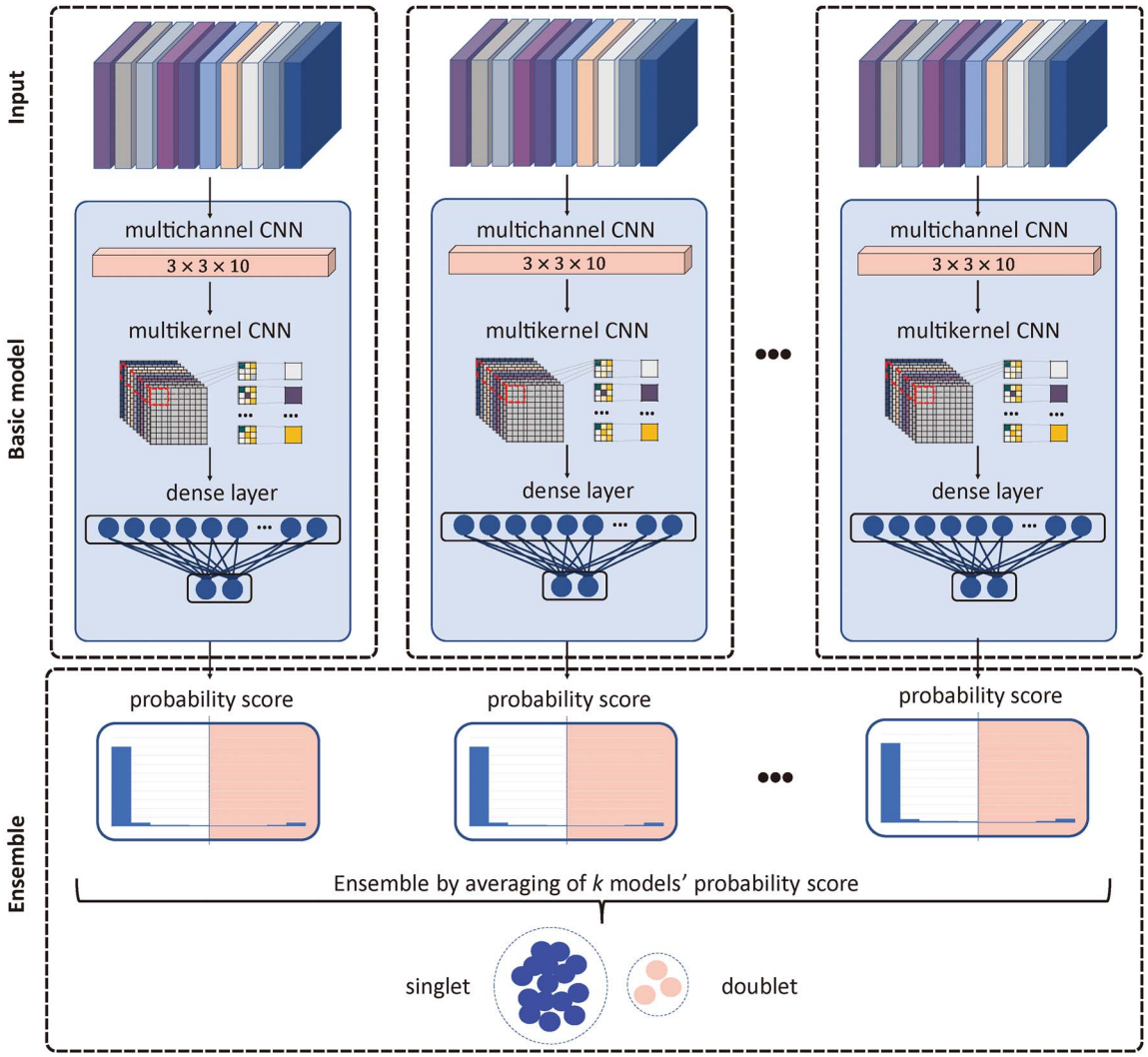

Figure 3. Architecture of SoCubeNet. It is an ensemble model architecture, which contains k basic models, each basic model contains two feature extraction blocks (multichannel CNN and multikernel CNN) and a classification block (dense layer) and will be trained independently by k-fold cross-validation sampling.

hyperparameter $\gamma = 0.99$.

$$\mathrm{BCE} = -\left(y \log \tilde{y} + (1-y) \log\left(1-\tilde{y}\right)\right) \quad (1)$$

where y is the numerical label and $\tilde{y}$ is the predicted doublet probability score.

Self-fitting by randomly dividing the dataset into training and validation sets results in bias. Thus, Bagging, a parallel ensemble learning strategy, was applied in SoCube to minimize this bias [43, 44]. Datasets generated from the doublet simulation step will be used to fit k (default setting is 5) models independently by k-fold cross-validation sampling. Unlike the traditional Bagging strategy, which uses bootstrap sampling, k-fold cross-validation sampling can ensure that all droplets will be sampled. The final doublet probability score is the average of k models' probability scores. Based on the predicted doublet probability score, SoCube removes the droplets whose scores are greater than the threshold. As with most models, the default probability threshold is 0.5. Users can remove a specific proportion of droplets according to the probability score or just apply the default setting.

Quantification analysis

The existence of doublets in scRNA-seq influences many scRNA-seq downstream analyses. Therefore, SoCube performance was evaluated from the following perspectives. Previous representative algorithms Solo, DoubletFinder and scDblFinder were selected as comparison algorithms. The official recommended parameter settings of the other three algorithms were used when performing the following evaluations.

Benchmark evaluation

Sixteen benchmark datasets were collected from previous publication studies to evaluate the performance of SoCube for detecting doublets [10, 11, 13, 14, 23]. The various formats of the original data were uniformly converted into 'h5ad' format, a hierarchical scRNA-seq data format implemented in the Python package anndata, to be readable by SoCube. AUPRC and AUROC, two metrics of the overall accuracy of a binary classification algorithm implemented in the Python package scikit-learn, were used to evaluate the overall doublet detection accuracy.

Each algorithm was tested five times in parallel on 16 datasets, respectively, and the average value was taken as the final result. And then the top-performing dataset number of each algorithm

was counted based on test result. Besides, the prediction performance of SoCube for droplets of different library sizes in a dataset was evaluated on these datasets and used the average value as final result again. For details, droplets of each dataset were sorted in ascending order according to cell library size and were divided into ten equal-sized bins after predicted by SoCube. The first bin contained droplets in the top 10% of the library size, the second bin contained droplets with library sizes between 10 and 20%, and so on. The assessment of model generalization is relative, but the absolute metrics such as AUPRC are different for different datasets. In order to treat each dataset equally in terms of its contribution to generalization assessment, the result of each bin was scaled by maximum-minimum normalization.

Cell clustering analysis

To evaluate the performance of SoCube for removing spurious cell clusters, a real scRNA-seq dataset 'scPred_pbmc_1', which contains eight different cell types, was fetched from previous research [45]. Based on the balance doublet simulation strategy mentioned in the SoCube workflow, which also widely adopted by previous algorithm, doublets were introduced with a 20% doublet rate. Louvain clustering [46] implemented in the R package Seurat (v4.1.0) was used to identify cell clusters on its clean version without doublets ('clean dataset') and its post-doublet-detection version after each doublet detection algorithm was applied (droplets whose doublet probability scores were greater than the default threshold were removed). The cell cluster result was marked on the same graph as the clean dataset, and the clean dataset result was used as a positive control for benchmarking spurious cell cluster removal on the post-doublet-detection datasets.

Differentially expressed gene analysis

To evaluate four doublet detection algorithms from this perspective, a synthetic scRNA-seq dataset with two cell types and 1126 between-cell-type DE genes (6% of a total of 18 760 genes) was fetched from previously published research [14]. MAST and the Wilcoxon rank-sum test implemented in the R package Seurat (v 4.1.0) were applied to this dataset ('contaminated dataset') and its post-doublet-detection version after each doublet detection algorithm was applied [47–49]. After each DE method was applied to every dataset, genes whose Bonferroni-corrected *P* values were under 0.05 were identified as DE genes. Three accuracy measures—precision, recall, and TNR—were calculated for every set of identified DE genes. For each DE method, its accuracy on the 'contaminated dataset' was used as the negative control for benchmarking its accuracy on the post-doublet-detection datasets.

Cell trajectory inference

Two synthetic scRNA-seq datasets were fetched from previous research [14]. Both datasets contained 1000 genes. The first dataset consisted of 100 doublets and 500 singlets following a bifurcating trajectory, whose two branches had 250 singlets each, and the second dataset consisted of 250 doublets and 1000 singlets from a conjunction of three sequential trajectories. Each dataset was expanded into a suite, including its original version ('contaminated dataset'), its clean version without doublets ('clean dataset'), and its post-doublet-detection version after each doublet detection algorithm was applied (droplets whose doublet probability scores were greater than the default threshold of 0.5 were removed). The cell trajectories of the two datasets were constructed by the minimum spanning tree (MST) algorithm implemented in the R package slingshot (v2.2.1) [50]. Trajectories constructed from the contaminated datasets and the clean datasets were used as the negative and positive controls for benchmarking the trajectories inferred from the post-doublet-detection datasets.

For temporally DE gene analysis, a synthetic dataset was fetched from previous research [14]. This dataset had 250 temporally DE genes (tDEG) out of a total of 750 genes. Slingshot and TSCAN were used to infer the pseudotime of each droplet on this dataset ('contaminated data') and its post-doublet-detection version after each doublet detection algorithm was applied (droplets whose doublet probability scores were greater than the default threshold were removed). Then, for each dataset, each gene's levels were regressed in all droplets based on inferred pseudotime by the general additive model (GAM), which was implemented in the R function gam, and a *P* value was obtained [51]. Similar to previous DE gene analysis, genes whose Bonferroni-corrected *P* values were less than 0.05 were identified as tDEGs. Three accuracy measures—precision, recall, and TNR—were calculated for every set of identified tDEGs. For each temporally DE method, its accuracy on the contaminated dataset was used as the negative control for benchmarking its accuracy on the post-doublet-detection datasets.

RESULTS AND DISCUSSION

To illustrate the difference of three doublet simulation strategies, dataset hm-6k [5] was used and visualized real singlets, real doublets annotated by cell hashing and *in silico* doublets generated by three strategies on UMAP 2d feature space (shown in Figure 4A). Dataset hm-6k contains two species (human and mouse) and heterotypic doublets are in majority. It can be observed that data distribution of *in silico* doublets generated by HEFS is similar to the distribution of real doublets, and by contrast, doublets generated by HOFS is absolutely separated with real doublets. As the purpose of generating *in silico* doublets is to simulate real doublet distribution, the simulation strategies will significantly influence detection result. Besides balance random simulation strategy, which widely adopted by previous algorithms, SoCube innovatively proposed heterotypic-doublet-first simulation and homotypic-doublet-first simulation to fit more datasets. For many datasets, a balanced simulation strategy is sufficient by default, but for severely type-imbalanced datasets like hm-6k, adopting a post-typically-biased strategy will significantly improve performance. For example, performance on hm-6k has ~10% improvement when using heterotypic simulation strategy. Users are free to choose the SoCube's simulation strategy according to their data situation.

From the benchmarking of real datasets, SoCube's innovative 3D feature-embedding strategy and ensembled multichannel CNN architecture were able to significantly improve the doublet detection result's accuracy with good robustness and generalization performance.

Optimum generalization performance on real benchmark datasets

The accuracy of doublet detection is the most direct and important evaluation criterion. Sixteen benchmark datasets were collected from previous publication studies to evaluate the performance of SoCube for detecting doublets (shown in Supplementary Table S1 online at http://bib.oxfordjournals.org/) [10, 11, 13, 14, 23]. These datasets have different cell types, dataset sizes (ranging from 500 to 26 426), and doublet ratios (ranging from 3.43 to 37.31%) and use four different doublet detection

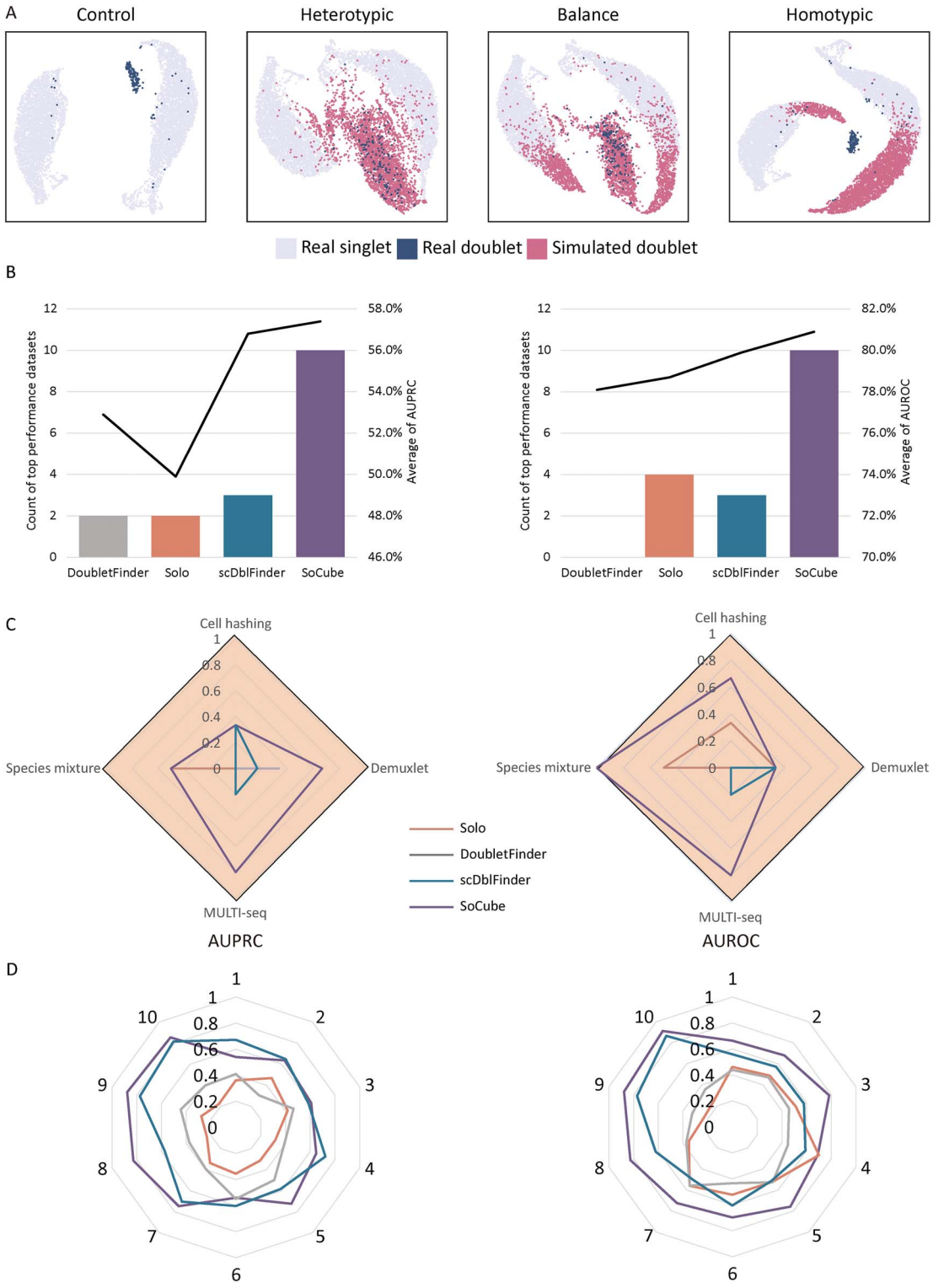

Figure 4. The performance of the four computational doublet detection algorithms on benchmark datasets. (**A**) Three doublet simulation strategies of SoCube and visualized on hm-6k dataset. (**B**) Generalization performance on 16 benchmark datasets. The bar chart indicates the count of datasets for which each algorithm achieved top performance, and the line chart indicates the average prediction result of 16 datasets for each algorithm. (**C**) Performance of the four algorithms on datasets annotated by Cell hashing, Demuxlet, MULTI-seq or Species mixture under AUPRC and AUROC. DoubletFinder is missing in AUROC because it has no top-performance dataset in any of the four types of annotation methods. (**D**) scaled average performance of the four algorithms stratified by cell library size on 16 benchmark datasets. The left radar map shows the performance in terms of AUPRC, and the right shows the AUROC. Each corner represents a bin, and the smaller the bin number, the smaller the cell library size of the sample it contains.

experimental methods (cell hashing, Demuxlet, MULTI-seq and Species mixture) to ensure the generalizability of evaluation. Doublets typically make up 5–20% of droplets according to previous studies [23, 52] and these collected datasets. Thus, the area under the precision-recall curve (AUPRC) and the area under the receiver operator curve (AUROC) were used to evaluate the

accuracy of the four doublet detection algorithms because those two metrics are suitable for imbalanced datasets. The association of AUROC and AUPRC has been mathematically proven [53], but AUPRC is sensitive to class distribution and is more concerned with doublet prediction precision, while AUROC is insensitive to class distribution and is more concerned with predictors' overall performance. Hence, both of them were used for the comprehensive evaluation of SoCube.

The accuracies of doublet detection of SoCube, DF, Solo and DoubletFinder on 16 benchmark datasets are shown in Supplementary Table S1 online at http://bib.oxfordjournals.org/ and Figure 4 (see method details in the 'Benchmark Evaluation' section). As shown in Figure 4B, SoCube performed best on most datasets, specifically, the best performance was achieved with the benchmark datasets of 10/16 under both AUPRC and AUROC. The performance of SoCube is also competitive on the remaining six datasets. Overall, on the average performance of 16 datasets, SoCube slightly outperforms the other three algorithms with 57.4% AUPRC and AUROC at 80.9%. Although there is no denying that scDblFinder also performs well on some datasets, which reinforces the famous 'No Free Lunch' theorem in machine learning [54]. The overall performance proved that SoCube had better generalizability and could be applied to many scRNA-seq data.

Different experimental doublet detection methods tend to have different preferences for doublets, which are determined by their mechanisms [14]. The doublets annotated by different experimental methods are actually different subsets of all doublets existing in the data. To evaluate the generalization performance of SoCube, the winning rates (i.e. the number percentage of top-performing datasets) of SoCube, DF, Solo and scDblFinder on datasets annotated by the four different experimental doublet detection methods were evaluated, as shown in Figure 4C. SoCube performed best on datasets annotated by all four experimental methods. The performance of the four algorithms was found to be comparable on the dataset annotated by Demuxlet under AUROC and on the dataset annotated by Cell hashing under AUPRC. The results illustrate SoCube's powerful generalization performance on a variety of different experimental annotation datasets to capture the real difference between singlets and doublets.

Cell library size, defined as the total sum of counts across all genes for each cell, has a great effect on the accuracy of doublet detection [25]. Hence, the prediction performance of SoCube for cells with different library sizes in each benchmark dataset was evaluated for comprehensive comparison and scaled result is shown in Figure 4D. It is seen that scDblFinder, as an improved version of DoubletFinder, has a very good performance on small library size samples (bin 1, 2, 3, 4) under AUPRC. But SoCube has a much better performance in samples with large library size. It may be that the information abundance of samples with small library size is lower and deep learning algorithms are more prone to overfitting. However, samples with small library size are often filtered in downstream analyses due to poor data quality, so the performance in samples with large library size is more critical. The AUROC of SoCube is almost the best on cells of different library sizes, which indicates that SoCube can well reduce the false positive rate, thus reducing the loss of information caused by normal cells being removed by mistaken identification as doublets. Therefore, it can be concluded that SoCube has better generalization performance for samples of different cell library sizes.

Effect on downstream analysis

More realistic cell clustering

The existence of heterotypic doublets will result in the misinterpretation of spurious cell clusters formed by annotating heterotypic doublets as novel cell types [19]. To evaluate the capacity of SoCube for removing spurious cell clusters, the test dataset 'scPred_pbmc_1' was collected from previous research [45] (see method details in the 'Cell Clustering Analysis' section). The visualization clustering result is illustrated in Figure 5A. To compared with the ground truth cell types distribution, the clustering results of the datasets processed by different doublet detection algorithms used the same clustering visualization points as the positive control group (clean data), and the cell clusters were distinguished by color annotation. Visually, the SoCube-processed group showed the most similar results to the positive control group. It can also be seen that the change in the number of clusters corroborates the previous assertion about the negative effects of doublets. There are 14 cell clusters in negative control group, significantly more than the positive control group with 8 cell clusters. The four computational algorithms effectively reduced the number of wrong clusters by removing the doublets, with SoCube and Solo obtaining the same number of clusters as the positive control group, scDblFinder obtaining 9 clusters, and DoubletFinder obtaining 10 clusters.

In addition, to quantitatively evaluate the reliability of cell clustering after using different detection algorithms, the Jaccard similarity coefficient (also named the Jaccard index) was calculated compared to ground truth cell types. Here, a higher Jaccard correlation coefficient indicates that the clustering-based cell pair set B is more consistent with the true type-based cell pair set A, which in turn indicates that the prediction results of the clustering algorithm are more realistic and reliable. The test shows that the Jaccard similarity coefficient of the positive control group without any doublet is the highest at 26.35%, which is normal and acceptable given that clustering algorithm is unsupervised learning algorithm. However, it can still be seen that the Jaccard similarity coefficient of SoCube group is the closest to the positive control group at 25.75%. In contrast, Solo and DoubletFinder were 22.47% and 20.28%, respectively, both at least 3% points lower than SoCube. scDblFinder has performed well in the previous benchmark tests, but the Jaccard similarity coefficient for SoCube is at least 3% points lower than SoCube. Although scDblFinder performed well in the previous benchmark test, its Jaccard similarity coefficient of 19.86% was seen in the cluster analysis, slightly inferior to DoubletFinder. It is obvious that SoCube was more consistent with the control group in terms of both the number of clusters and the Jaccard similarity coefficient. In summary, from the perspective of cell clustering analysis, SoCube is able to obtain more realistic cell clustering results from its processed data due to its excellent feature-embedding strategy, unique neural network model design and various doublet simulation strategies.

More differentially expressed genes reserved

The correct identification of differentially expressed genes (DEGs) between specific cell types is key to understanding phenotypic variation [55]. DEG analysis is based on the 'identical distribution' assumption that cells of the same type follow the same distribution of gene levels [19, 56]. For differential analysis of read count data, the Poisson distribution and the negative binomial distribution are the most commonly used models [56]. However, the existence of doublets in scRNA-seq will interfere with

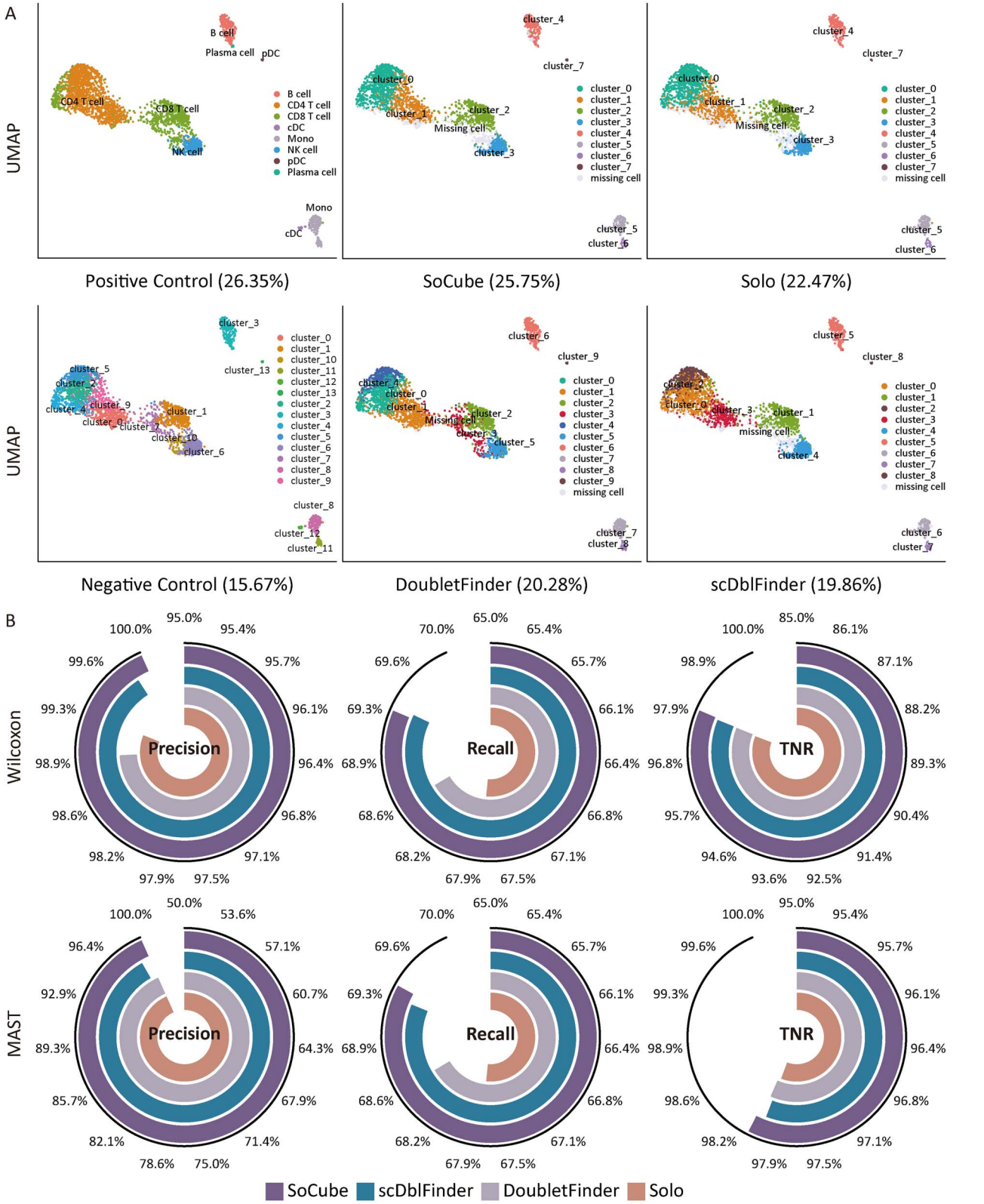

Figure 5. Cell clustering analysis and differential expression analysis cross-cell type. (**A**) Clustering results of datasets processed by the four computational doublet detection algorithms. The positive control group was clustered using scRNA-seq without any doublets and negative control group was clustered by original datasets. The category 'missing cell' means that these cells were misinterpreted as doublets and removed by the detection algorithms. The numerically labeled categories were obtained by the Louvain cluster algorithm. Each group's Jaccard index was calculated compared with the true cell types. (**B**) The effects of four computational doublet detection algorithms on DEG analysis. The Wilcoxon test and MAST were separately used as the DEG analysis method. The three polar plots show the precision, recall and TNR of four computational doublet detection algorithms.

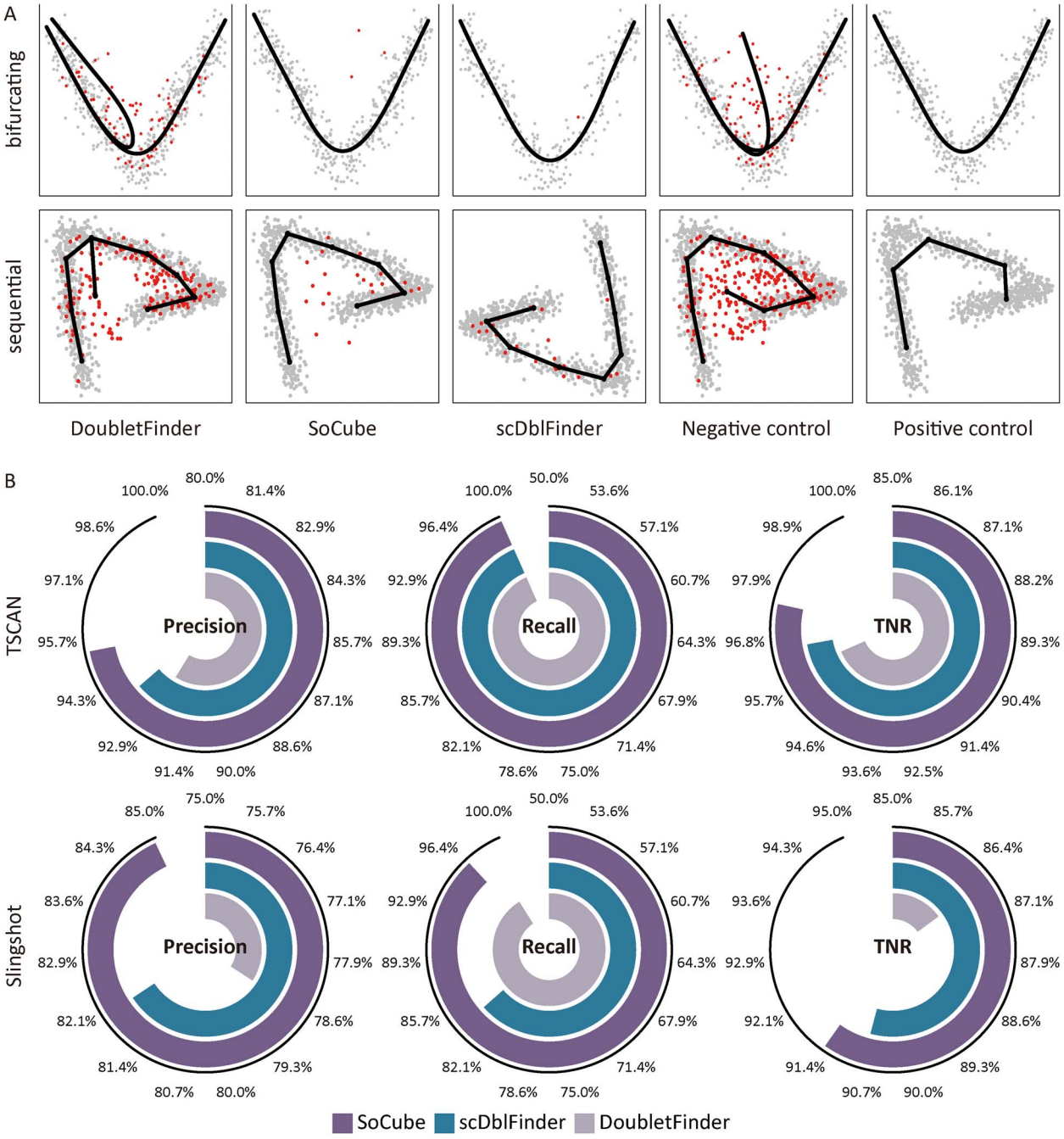

Figure 6. Cell trajectory inference and temporally DEG analysis of scRNA-seq datasets processed by SoCube, DoubletFinder and scDblFinder. (**A**) Bifurcating or sequential trajectory inference, 'negative control' results from raw scRNA-seq, 'positive control' results from clean scRNA-seq, and red points represent remaining doublets. (**B**) Temporally DEG analysis results of SoCube, DoubletFinder and scDblFinder using TSCAN or Slingshot as the cell trajectory inference method.

downstream DEG analysis by violating this assumption [19]. Therefore, if a doublet detection algorithm is effective in removing doublets, it will improve the accuracy of DEG analysis.

SoCube was benchmarked on a scRNA-seq dataset fetched from previous research [14]. The effect gains of DEG analysis on the contaminated dataset and its post-doublet detection version are shown in Figure 5B (see method details in the 'Differentially Expressed Gene Analysis' section). In general, SoCube achieves SOTA performance that is comparable to scDblFinder on both Wilcoxon and MAST differential test DEG analysis. In detail, the effect gains of different doublet detection algorithms on DEG analysis differed less in precision and TNR but differed significantly in recall. Higher precision and TNR indicate more accurate DEGs, while higher recall indicates fewer missed real DEGs, and the two are often incompatible. Researchers often prioritize higher precision and TNR by adjusting DEG analysis method parameters to obtain accurate differential genes, but this results in the loss of biological information contained in other unidentified DEGs [55]. Therefore, it makes sense to significantly improve recall without reducing high precision and TNR. SoCube and scDblFinder achieved the comparable gains of ~6% for metric recall compared with the negative control group, while DoubletFinder

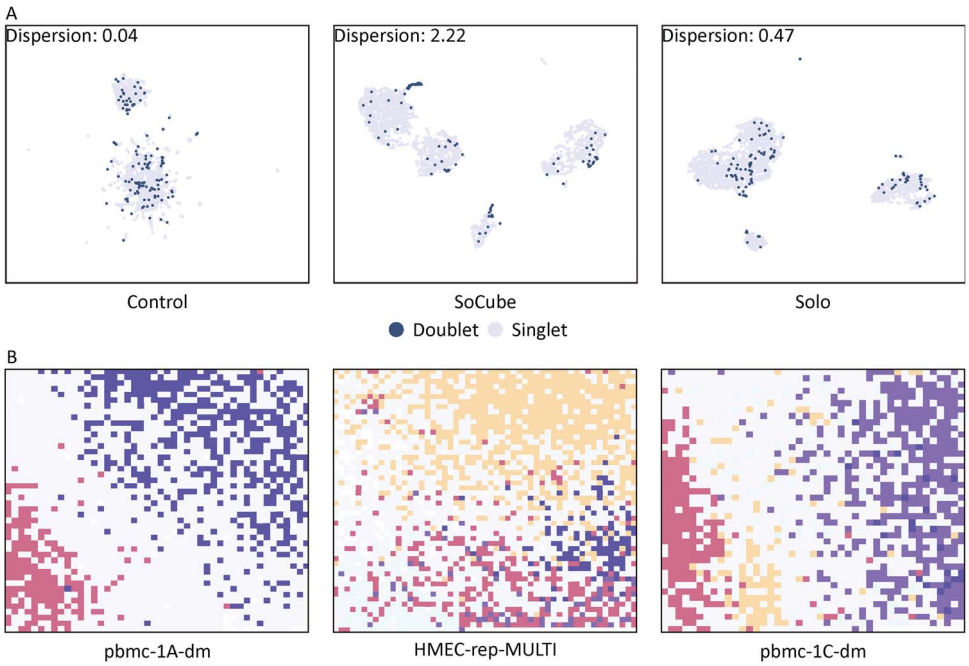

Figure 7. Visualization of interpretability analysis of SoCube. (**A**) Reduction and visualization of pbmc-1A-ch latent features extracted by SoCube and Solo. The 'Dispersion' value is the dispersion of doublets and singlets. The control group was visualized from the original features. (**B**) WGCNA was performed on three benchmark datasets. Heatmaps were plotted based on SoCube's 2d Gmap embedding feature. Each grid represents a gene. Genes marked with the same color are coexpressed genes. Light gray grids indicate that these genes do not belong to any coexpressed gene cluster.

and Solo achieved gains of only ~5% and ~4%, respectively. Given that there are over 1000 DEGs in this dataset, the impact of such a recall gain difference would be significant. Although SoCube did not significantly outperform scDblFinder in terms of differential expression analysis, the performance of SoCube was also at the level of SOTA, and SoCube's feature embedding strategy is highly biologically interpretable and transferable, so this does not negate the excellence of SoCube.

More reliable cell trajectory

Cell trajectory inference or pseudotemporal ordering is a computational technique used in single-cell transcriptomics to determine the pattern of a dynamic process experienced by cells and then arrange cells based on their progression through the process [57]. The cell trajectory corresponds to a cellular process, such as cell differentiation, and is based on the similarity of cells in terms of gene expression profiles [58]. Therefore, the existence of doublets in scRNA-seq also confounds cell trajectory inference [59]. Cell trajectory inference is biased by the existence of doublets because doublets may result in spurious branches in an inferred trajectory.

To evaluate the direct gains of cell trajectory inference after removing doublets by SoCube, it was tested on two datasets [14] containing different trajectories, and cell trajectories were inferred from original datasets and datasets processed by four doublet detection algorithms, as shown in Figure 6A (see method details in the 'Cell Trajectory Inference' section). The result from Solo was not obtained because Solo did not work properly on these two datasets and returned an 'NaN error' (a common error when intermediate values are close to infinity or zero), which showed Solo's limited robustness.

As shown in the first row of Figure 6A, one dataset contains a bifurcating trajectory and another dataset contains three sequential trajectories. Due to the interference of doublet on the statistical distribution of cells, a spurious branch deviating from normal cells was inferred in the negative control group (not perform doublet removal) of the dataset containing the bifurcating trajectory, which directly confirmed the aforementioned negative effect of doublet on cell trajectory inference. However, we need to note that doublet does not always lead to spurious branches, depending on the distribution of doublet relative to normal cells. As shown in the second row of Figure 6A, in the negative control group of another dataset containing three sequential trajectories, no spurious branch found because doublets were balanced distributed on both sides of the normal cells. It can also be seen that the results of both SoCube and scDblFinder groups are very similar to the cell trajectories of the positive control group (without any doublet), with few residual doublets (points marked in red) in both groups. This reveals the excellent performance of both detection algorithms. In contrast, DoubletFinder not only remove a limited number of doublets, but also led to the formation of spurious branches not found in the negative control group due to the unbalanced number of removed doublets on either side of the trajectory of the second dataset. An interesting phenomenon was observed, where the fit of the cell trajectory after application of SoCube was even better than that of the positive control. The few doublets that were not removed by SoCube caused an effect on the cell trajectory inference within the margin of error [14]. Therefore, both scDblFinder and SoCube are effective doublet detection algorithms in terms of cell trajectory inference.

Temporally differentially expressed gene (tDEG) analysis, a typical downstream task following cell trajectory inference, explores differential gene expression along the inferred cell trajectory and identify tDEGs [58]. Therefore, the existence of doublets also decreases the accuracy of tDEG identification. The evaluation results of the temporally DEG analysis shown in Figure 6B were similar to the cell trajectory inference (see method details in the 'Cell Trajectory Inference' section). Whether using Slingshot or TSCAN as the method for cell trajectory inference, the datasets with doublets removed by SoCube had the highest precision and

TNR value for temporally DEG analysis. DoubletFinder had a slight advantage in recall when using Slingshot as cell trajectory inference, but this did not make up for the significant gap with SoCube in other metrics. Solo did not work on this test dataset again due to the 'NaN error' returned. While scDblFinder is comparable in cell trajectory inference, SoCube showed better performance in temporally DEG analysis.

In summary, SoCube's novel feature-embedding strategy, doublet simulation strategies and model architecture are effective and make inferred cell trajectories more reliable with fewer spurious branches, which in turn improves the results of downstream task temporally DEG analysis.

Interpretability analysis

To explore the reason why SoCube is effective for doublet detection, the real dataset pbmc-1A-ch was used as test data, and its internal latent features produced by SoCube and Solo were downscaled and visualized (see Figure 7A). DoubletFinder and scDblFinder were not included because internal latent features could not be obtained from them. In addition, the dispersion of the result (defined as the relative Euclidean distance between the doublet average and the singlet average) was quantitatively evaluated. Doublets were mixed with singlets in a control group with very low dispersion, and doublets were gathered into the edge area of singlets after processing by SoCube and Solo. Quantitatively, the dispersion value of SoCube was greater than that of Solo. One of the most important advantages of SoCube is the representation ability of the gene–gene inner relationship, which was explained by a weighted gene coexpression network analysis (WGCNA) implemented in the R package WGCNA [60], a widely used data mining method for studying gene–gene correlations. Three benchmark datasets selected as case studies were preprocessed by Seurat, and the WGCNA pipeline [61] was applied to determine the modules of coexpressed genes. The results are visualized in Figure 7B. It is obvious that SoCube was able to gather coexpressed genes while embedding the original droplet feature into 2D Gmap. The distance of two genes in SoCube's Gmap could represent their biological relationship.

The most innovative change in SoCube compared with previous doublet detection algorithms was proposing a brand-new cell feature-embedding strategy. This embedding feature captures gene–gene correlations and gene uniqueness information into local areas. The CNN architecture model effectively uses this information to find a low nonlinear dimension space that can distinguish between singlets and doublets. Solo uses VAE to extract features directly from original data [23]. However, original scRNA-seq is very sparse, and the existence of many zeros hinders weight optimization of VAE [40]. In addition, original scRNA-seq data features are unordered [39], and VAE lacks the ability to discover internal feature relationships, such as coexpressed genes, which are important biological feature patterns related to biological process regulation [60, 62]. Thus, the dispersion between singlets and doublets in the low-dimensional space generated by SoCube was greater than that generated by Solo, which resulted in the success of SoCube. The key algorithm of scDblFinder and DoubletFinder use kNN to cluster droplets and then predict based on similarity to the *in silico* doublets. On the one hand, kNN as a linear algorithm does not extract complex biological information well [63], and on the other hand, the features used by these algorithms do not take well into account the uniqueness of each gene in a complex biological network, and this uniqueness is also a potential representation of the cell [64, 65].

CONCLUSIONS

In summary, the ability of SoCube to detect doublets facilitates the task of downstream analysis of single-cell transcriptomics. SoCube effectively improves the generalization performance of doublet detection for scRNA-seq data by adopting a new feature-embedding strategy, using a deep learning approach with CNN architecture, and supplemented with an ensemble learning strategy. Therefore, SoCube promises to be a powerful algorithm that can help researchers more effectively remove doublets from their data to improve the reliability of subsequent analysis results and reduce experimental cost. At the same time, the novel feature-embedding strategy proposed by SoCube can be transferred to other single-cell omics tasks.

Key Points

- This study proposed a novel AI-based doublet detection algorithm (named SoCube), which significantly improve generalization performance compared with previous SOTA algorithms.
- This study proposed a novel 3D composite feature-embedding strategy, which mined intrinsic associations between genes from high-dimensional, sparse and disordered raw features.
- This study proposed three doublet simulation strategies with different propensities (balance, heterotypic-doublet-first and homotypic-doublet-first) to accommodate different data.

SUPPLEMENTARY DATA

Supplementary data are available online at https://academic.oup.com/bib.

FUNDING

National Natural Science Foundation of China (81872798 and U1909208); Natural Science Foundation of Zhejiang Province (LR21H300001); Leading Talent of the 'Ten Thousand Plan'—National High-Level Talents Special Support Plan of China; Fundamental Research Fund for Central Universities (2018QNA7023); 'Double Top-Class' University Project (181201*194232101); Key R&D Program of Zhejiang Province (2020C03010). This work was supported by Westlake Laboratory (Westlake Laboratory of Life Sciences and Biomedicine); Alibaba-Zhejiang University Joint Research Center of Future Digital Healthcare; Alibaba Cloud; Information Technology Center of Zhejiang University.

AUTHOR CONTRIBUTIONS

F.Z. and H.Z. conceived the idea and designed the study; H.Z. and M.L. proposed SoCube's algorithm; H.Z., M.L. and G.L. performed the evaluation of SoCube; H.Z., M.L., L.Z., W.Z. and Z.X. collected related datasets and provided related biological support; H.Z. and M.L. wrote the manuscript; All authors reviewed and approved the manuscript.

DATA AVAILABILITY

SoCube was built as an end-to-end command line software, published on PyPi (https://pypi.org/project/socube/), the

official python package site. and was open-source on GitHub (https://github.com/idrblab/socube/). A docker image was provided on Docker Hub (https://hub.docker.com/r/gcszhn/socube/) for fast deployment. All data used in this study were publicly available as previously described. Details were listed in Supplementary Tables S2 and S3 online at http://bib.oxfordjournals.org/.

REFERENCES

1. Klein AM, Mazutis L, Akartuna I, *et al.* Droplet barcoding for single-cell transcriptomics applied to embryonic stem cells. *Cell* 2015;**161**:1187–201.
2. Cao J, Packer JS, Ramani V, *et al.* Comprehensive single-cell transcriptional profiling of a multicellular organism. *Science* 2017;**357**:661–7.
3. Gierahn TM, Wadsworth MH, 2nd, Hughes TK, *et al.* Seq-well: portable, low-cost RNA sequencing of single cells at high throughput. *Nat Methods* 2017;**14**:395–8.
4. Macosko EZ, Basu A, Satija R, *et al.* Highly parallel genome-wide expression profiling of individual cells using nanoliter droplets. *Cell* 2015;**161**:1202–14.
5. Zheng GX, Terry JM, Belgrader P, *et al.* Massively parallel digital transcriptional profiling of single cells. *Nat Commun* 2017;**8**:14049.
6. Rosenberg AB, Roco CM, Muscat RA, *et al.* Single-cell profiling of the developing mouse brain and spinal cord with split-pool barcoding. *Science* 2018;**360**:176–82.
7. Fava VM, Bourgey M, Nawarathna PM, *et al.* A systems biology approach identifies candidate drugs to reduce mortality in severely ill patients with COVID-19. *Sci Adv* 2022;**8**: eabm2510.
8. Sathyamurthy A, Johnson KR, Matson KJE, *et al.* Massively parallel single nucleus transcriptional profiling defines spinal cord neurons and their activity during behavior. *Cell Rep* 2018;**22**: 2216–25.
9. Andrews TS, Kiselev VY, McCarthy D, *et al.* Tutorial: guidelines for the computational analysis of single-cell RNA sequencing data. *Nat Protoc* 2021;**16**:1–9.
10. Stoeckius M, Zheng S, Houck-Loomis B, *et al.* Cell hashing with barcoded antibodies enables multiplexing and doublet detection for single cell genomics. *Genome Biol* 2018;**19**:224.
11. Kang HM, Subramaniam M, Targ S, *et al.* Multiplexed droplet single-cell RNA-sequencing using natural genetic variation. *Nat Biotechnol* 2018;**36**:89–94.
12. Wolock SL, Lopez R, Klein AM. Scrublet: computational identification of cell doublets in single-cell transcriptomic data. *Cell Syst* 2019;**8**:281–91 e289.
13. McGinnis CS, Patterson DM, Winkler J, *et al.* MULTI-seq: sample multiplexing for single-cell RNA sequencing using lipid-tagged indices. *Nat Methods* 2019;**16**:619–26.
14. Xi NM, Li JJ. Benchmarking computational doublet-detection methods for single-cell RNA sequencing data. *Cell Syst* 2021;**12**: 176–94 e176.
15. Barrett T, Wilhite SE, Ledoux P, *et al.* NCBI GEO: archive for functional genomics data sets—update. *Nucleic Acids Res* 2013;**41**:D991–5.
16. Li M, Zhang X, Ang KS, *et al.* DISCO: a database of deeply integrated human single-cell omics data. *Nucleic Acids Res* 2022;**50**:D596–602.
17. Subramanian A, Narayan R, Corsello SM, *et al.* A next generation connectivity map: L1000 platform and the first 1,000,000 profiles. *Cell* 2017;**171**:1437–52 e1417.
18. Hansen J, Sealfon R, Menon R, *et al.* A reference tissue atlas for the human kidney. *Sci Adv* 2022;**8**:eabn4965.
19. Luecken MD, Theis FJ. Current best practices in single-cell RNA-seq analysis: a tutorial. *Mol Syst Biol* 2019;**15**:e8746.
20. Russ DE, Cross RBP, Li L, *et al.* A harmonized atlas of mouse spinal cord cell types and their spatial organization. *Nat Commun* 2021;**12**:5722.
21. Argyriou A, Wadsworth MH, 2nd, Lendvai A, *et al.* Single cell sequencing identifies clonally expanded synovial CD4(+) T(PH) cells expressing GPR56 in rheumatoid arthritis. *Nat Commun* 2022;**13**:4046.
22. Sepulveda-Falla D, Sanchez JS, Almeida MC, *et al.* Distinct tau neuropathology and cellular profiles of an APOE3 Christchurch homozygote protected against autosomal dominant Alzheimer's dementia. *Acta Neuropathol* 2022;**144**:589–601.
23. Bernstein NJ, Fong NL, Lam I, *et al.* Solo: doublet identification in single-cell RNA-seq via semi-supervised deep learning. *Cell Syst* 2020;**11**:95–101 e105.
24. McGinnis CS, Murrow LM, Gartner ZJ. DoubletFinder: doublet detection in single-cell RNA sequencing data using artificial nearest neighbors. *Cell Syst* 2019;**8**:329–37 e324.
25. Bais AS, Kostka D. Scds: computational annotation of doublets in single-cell RNA sequencing data. *Bioinformatics* 2020;**36**:1150–8.
26. DePasquale EAK, Schnell DJ, Van Camp PJ, *et al.* DoubletDecon: deconvoluting doublets from single-cell RNA-sequencing data. *Cell Rep* 2019;**29**:1718–27 e1718.
27. Lun AT, McCarthy DJ, Marioni JC. A step-by-step workflow for low-level analysis of single-cell RNA-seq data with bioconductor. *F1000Res* 2016;**5**:2122.
28. Sun B, Bugarin-Estrada E, Overend LE, *et al.* Double-jeopardy: scRNA-seq doublet/multiplet detection using multi-omic profiling. *Cell Rep Methods* 2021;**1**:None.
29. Weber LL, Sashittal P, El-Kebir M. doubletD: detecting doublets in single-cell DNA sequencing data. *Bioinformatics* 2021;**37**:i214–21.
30. Germain PL, Lun A, Garcia Meixide C, *et al.* Doublet identification in single-cell sequencing data using scDblFinder. *F1000Res* 2021;**10**:979.
31. Lopez R, Regier J, Cole MB, *et al.* Deep generative modeling for single-cell transcriptomics. *Nat Methods* 2018;**15**:1053–8.
32. Pedersen HK, Forslund SK, Gudmundsdottir V, *et al.* A computational framework to integrate high-throughput '-omics' datasets for the identification of potential mechanistic links. *Nat Protoc* 2018;**13**:2781–800.
33. Fu J, Zhang Y, Wang Y, *et al.* Optimization of metabolomic data processing using NOREVA. *Nat Protoc* 2022;**17**:129–51.
34. Schiffman C, Petrick L, Perttula K, *et al.* Filtering procedures for untargeted LC-MS metabolomics data. *BMC Bioinform* 2019;**20**:334.
35. Taylor SL, Ruhaak LR, Kelly K, *et al.* Effects of imputation on correlation: implications for analysis of mass spectrometry data from multiple biological matrices. *Brief Bioinform* 2016;**18**: 312–20.
36. Zhang Z, Zhao Y, Liao X, *et al.* Deep learning in omics: a survey and guideline. *Brief Funct Genomics* 2019;**18**:41–57.
37. Zou Q, Xing P, Wei L, *et al.* Gene2vec: gene subsequence embedding for prediction of mammalian N(6)-methyladenosine sites from mRNA. *RNA* 2019;**25**:205–18.
38. Seninge L, Anastopoulos I, Ding H, *et al.* VEGA is an interpretable generative model for inferring biological network activity in single-cell transcriptomics. *Nat Commun* 2021;**12**:5684.
39. Shen WX, Liu Y, Chen Y, *et al.* AggMapNet: enhanced and explainable low-sample omics deep learning with

feature-aggregated multi-channel networks. *Nucleic Acids Res* 2022;**50**:e45.
40. Huang M, Wang J, Torre E, *et al.* SAVER: gene expression recovery for single-cell RNA sequencing. *Nat Methods* 2018;**15**:539–42.
41. McInnes L, Healy J, Melville J. UMAP: uniform manifold approximation and projection for dimension reduction. *arXiv* 2018;**2018**:arXiv.1802.03426.
42. Jonker R, Volgenant A. A shortest augmenting path algorithm for dense and sparse linear assignment problems. *Comput Secur* 1987;**38**:325–40.
43. Breiman L. Bagging predictors. *Mach Learn* 1996;**24**:123–40.
44. Liu XB, Liu ZT, Wang GJ, *et al.* Ensemble transfer learning algorithm. *IEEE Access* 2018;**6**:2389–96.
45. Alquicira-Hernandez J, Sathe A, Ji HP, *et al.* scPred: accurate supervised method for cell-type classification from single-cell RNA-seq data. *Genome Biol* 2019;**20**:264.
46. Blondel VD, Guillaume J-L, Lambiotte R, *et al.* Fast unfolding of communities in large networks. *J Stat Mech Theory Exp* 2008;**2008**:P10008.
47. Finak G, McDavid A, Yajima M, *et al.* MAST: a flexible statistical framework for assessing transcriptional changes and characterizing heterogeneity in single-cell RNA sequencing data. *Genome Biol* 2015;**16**:278.
48. Fay MP, Proschan MA. Wilcoxon-Mann-Whitney or t-test? On assumptions for hypothesis tests and multiple interpretations of decision rules. *Stat Surv* 2010;**4**:1–39.
49. Hao Y, Hao S, Andersen-Nissen E, *et al.* Integrated analysis of multimodal single-cell data. *Cell* 2021;**184**:3573–87 e3529.
50. Street K, Risso D, Fletcher RB, *et al.* Slingshot: cell lineage and pseudotime inference for single-cell transcriptomics. *BMC Genomics* 2018;**19**:477.
51. Wood SN. *Generalized Additive Models: An Introduction with R*. New York: Chapman and Hall/CRC, 2017.
52. Germain P-L, Sonrel A, Robinson MD. pipeComp, a general framework for the evaluation of computational pipelines, reveals performant single cell RNA-seq preprocessing tools. *Genome Biol* 2020;**21**:227.
53. Davis J, Goadrich M. The relationship between precision-recall and ROC curves. *ACM* 2006;**2006**:233–40.
54. Adam SP, Alexandropoulos S-AN, Pardalos PM, *et al.* No free lunch theorem: a review. In: Demetriou IC, Pardalos PM (eds). *Approximation and Optimization: Algorithms, Complexity and Applications*. Cham: Springer International Publishing, 2019, 57–82.
55. Costa-Silva J, Domingues D, Lopes FM. RNA-Seq differential expression analysis: an extended review and a software tool. *PloS One* 2017;**12**:e0190152.
56. Anjum A, Jaggi S, Varghese E, *et al.* Identification of differentially expressed genes in RNA-seq data of *Arabidopsis thaliana*: a compound distribution approach. *J Comput Biol* 2016;**23**:239–47.
57. Bacher R, Kendziorski C. Design and computational analysis of single-cell RNA-sequencing experiments. *Genome Biol* 2016;**17**:63.
58. Saelens W, Cannoodt R, Todorov H, *et al.* A comparison of single-cell trajectory inference methods. *Nat Biotechnol* 2019;**37**:547–54.
59. Tian L, Dong X, Freytag S, *et al.* Benchmarking single cell RNA-sequencing analysis pipelines using mixture control experiments. *Nat Methods* 2019;**16**:479–87.
60. Langfelder P, Horvath S. WGCNA: an R package for weighted correlation network analysis. *BMC Bioinform* 2008;**9**:559.
61. Feregrino C, Tschopp P. Assessing evolutionary and developmental transcriptome dynamics in homologous cell types. *Dev Dyn* 2022;**251**:1472–89.
62. Stuart JM, Segal E, Koller D, *et al.* A gene-coexpression network for global discovery of conserved genetic modules. *Science* 2003;**302**:249–55.
63. Alon U. *An Introduction to Systems Biology: Design Principles of Biological Circuits*. New York: Chapman and Hall/CRC, 2006.
64. Shao X, Yang H, Zhuang X, *et al.* scDeepSort: a pre-trained cell-type annotation method for single-cell transcriptomics using deep learning with a weighted graph neural network. *Nucleic Acids Res* 2021;**49**:e122.
65. Li X, Chen W, Chen Y, *et al.* Network embedding-based representation learning for single cell RNA-seq data. *Nucleic Acids Res* 2017;**45**:e166.