Computational and Structural Biotechnology Journal 19 (2021) 5874–5887

COMPUTATIONAL ANDSTRUCTURAL BIOTECHNOLOGY JOURNAL

journal homepage: www.elsevier.com/locate/csbj

Automatic cell type identification methods for single-cell RNA sequencing

Bingbing Xie [a], Qin Jiang [b,*], Antonio Mora [c,*], Xuri Li [a,*]

[a] State Key Laboratory of Ophthalmology, Zhongshan Ophthalmic Center, Sun Yat-Sen University, Guangzhou 510060, Guangdong, China
[b] Affiliated Eye Hospital of Nanjing Medical University, Nanjing, China
[c] Joint School of Life Sciences, Guangzhou Medical University and Guangzhou Institutes of Biomedicine and Health (Chinese Academy of Sciences), Xinzao, Panyu District, Guangzhou 511436, Guangdong, China

ARTICLE INFO

Article history:
Received 27 April 2021
Received in revised form 23 September 2021
Accepted 18 October 2021
Available online 20 October 2021

Keywords:
Single-cell RNA sequencing (scRNA-seq)
Automatic identification
Cell type
Eager learning
Lazy learning
Marker learning

ABSTRACT

Single-cell RNA sequencing (scRNA-seq) has become a powerful tool for scientists of many research disciplines due to its ability to elucidate the heterogeneous and complex cell-type compositions of different tissues and cell populations. Traditional cell-type identification methods for scRNA-seq data analysis are time-consuming and knowledge-dependent for manual annotation. By contrast, automatic cell-type identification methods may have the advantages of being fast, accurate, and more user friendly. Here, we discuss and evaluate thirty-two published automatic methods for scRNA-seq data analysis in terms of their prediction accuracy, F1-score, unlabeling rate and running time. We highlight the advantages and disadvantages of these methods and provide recommendations of method choice depending on the available information. The challenges and future applications of these automatic methods are further discussed. In addition, we provide a free scRNA-seq data analysis package encompassing the discussed automatic methods to help the easy usage of them in real-world applications.

Contents

* Corresponding authors.
E-mail addresses: jqin710@vip.sina.com (Q. Jiang), antoniocmora@gzhmu.edu.cn (A. Mora), lixr6@mail.sysu.edu.cn (X. Li).

https://doi.org/10.1016/j.csbj.2021.10.027
2001-0370/

1. Introduction

Since the establishment of single-cell RNA sequencing (scRNA-seq) technology in 2009 [1], it has become a powerful tool for researchers in different fields of biological research. Compared with bulk RNA sequencing, which detects the average gene expression of samples, scRNA-seq can identify phenotypic heterogeneity of mixed cell populations in various biological samples [2]. With the continuous decrease of costs of sequencing and the advancements of sequencing technologies, scRNA-seq offers the opportunity to comprehensively sequence and annotate the cell types present in almost any tissue of a species [3–5], thus enabling the identification of the biological processes and molecular functions of known or new cell types. For example, two novel mouse retinal bipolar cell types, one of which has a non-canonical morphology, were identified through parallel scRNA-seq of approximately 25,000 bipolar cells [6]. In addition, Lavin *et al*. provided a detailed immune cell atlas of early-stage lung cancer [7] and observed significant decreases of CD8 effector T cells with an expansion of Tregs regulatory and exhausted T cells at the tumor site. Indeed, scRNA-seq has applications in numerous research fields, such as developmental biology, biomedical research, neuroscience, aging, etc. [8–10].

Cell type identification using scRNA-seq traditionally involves two steps. First, the cells are clustered using an unsupervised method, and then the clusters are annotated to different cell types based on canonical markers found in the differentially expressed genes of the cluster [11,12] (Fig. 1A). Many unsupervised scRNA-seq clustering methods have been proposed, including graph-based clustering [11,13,14], hierarchical clustering [15–17], and partition clustering [18–20]. However, even for the most commonly used graph-based clustering, many parameters, such as the nearest neighbor number in graph construction and resolution in community detection, must be manually defined by the user. This can significantly influence the outcomes [21]. Differences in clustering schemes also affect downstream interpretations [22]. Moreover, annotating each cluster can be a very time-consuming process, particularly, for users who do not have in-depth knowledge on the marker genes of different cell types, since this approach requires a manual search of literature and various databases. By contrast, the automatic cell type identification methods do not require manual annotation. Instead, they can be used to predict the cell types directly from the public resources of scRNA-seq data. As such, users without sufficient knowledge on cell markers could benefit greatly. Also, automatic methods are preferred when the datasets are large and when the re-analysis requires a large amount of resources [21].

Similar to the fields of trajectories [23] and ligand-receptor interactions [24] for single-cell sequencing data analysis, many cell-type identification methods have been established in recent years. In this review, we quantitatively discuss the performance of these cell-type identification methods, particularly, regarding prediction accuracy, F1-score, running time, and new cell-type identification. We also discuss some unresolved challenges of the automatic cell-type identification methods and highlight future research perspectives. Moreover, we integrate currently available automatic cell-type identification methods into an R package called AutomaticCellTypeIdentification, which may facilitate the use of these automatic methods in real-world applications.

2. Types of automatic cell-type identification methods

Thirty-two published automatic cell-type identification methods are systematically evaluated (Table 1). Based on the usage of training datasets, namely, gene-cell or cell-gene expression matrix or canonical cell markers, the automatic methods can be classified into three categories (Table 1): eager learning methods (ACTINN, CaSTLe, CHETAH, clustifyr, Garnett, MarkerCount, MARS, scCapsNet, scClassifR, SciBet, scID, scLearn, scmap-cluster, scMatch, scHPL, scPred, scPretrain, scVI, Seurat, SingleCellNet, SingleR and Superscan), lazy learning methods (CellAtlasSearch, CELLBLAST, CellFishing.jl and scmap-cell) and marker learning methods (CellAssign, DigitalCellSorter, MarkerCount, scCATCH, SCINA, SCSA and scTyper) (Fig. 1B). The lazy learning methods project cells based on the training datasets to identify the nearest neighbor cells, similar to the classical BLAST method [25], and the cell type is then determined according to the nearest neighbor cells. By contrast, eager learning methods gather cell-type information to group the training datasets first, and then map the testing cells to the closest pre-annotated group. Marker learning methods utilize canonical cell markers that are highly expressed in a given cell type to assign the testing datasets using a mathematic model.

Lazy learning methods include CELLBLAST [26], scmap-cell [27], CellFishing.jl [28], and CellAtlasSearch [29]. Eager learning methods account for the majority of the automatic methods, including scHPL [30], clustifyr [31], MARS [32], scPretrain [33], Superscan [34], Seurat [11,12], scLearn [35], scCapsNet [36], ACTINN [37], CaSTLe [38], CHETAH [39], SciBet [40], scID [41], scmap-cluster [27], scPred [42], SingleCellNet [43], SingleR [44], scVI [45], scMatch [46], scClassifR [47], and Garnett [48]. scClassifR and Garnett differ from the other eager learning methods in that they use both canonical markers and training datasets. scClassifR uses canonical markers to build a classifier for each cell type and Garnett uses canonical markers to identify representative cells to train a classifier. Since a pre-annotated training dataset is needed for scClassifR and Garnett, they are categorized as eager learning methods. Marker learning methods include scTyper [49], DigitalCellSorter [50], SCINA [51], SCSA [52], CellAssign [53], and scCATCH [54]. MarkerCount [55] contains eager learning and marker learning methods, so it is categorized in both.

Both eager learning and lazy learning methods require training datasets to train the classifier model. However, eager learning methods use training datasets that have been categorized into groups using cell types, which is not the case for lazy learning methods. Apart from the training datasets provided by the users themselves, expertly annotated datasets from public resources [56–58] are also available, thus allowing users to choose the most suitable training datasets. Canonical cell markers are required for marker learning methods, which can be found in public cell marker databases, such as PanglaoDB [59], CellMarker [60], and CancerSEA [61]. To facilitate automatic cell-type identification, scLearn, CELLBLAST, SciBet, SingleCellNet, scMatch, Superscan, and Garnett provide processed training datasets. Moreover, DigitalCellSorter, SCSA, scTyper, and scCATCH provide canonical cell markers for certain cell types. In addition, CELLBLAST (https://cblast.gao-lab.org) and SciBet (http://scibet.cancer-pku.cn) have built user-friendly web servers for querying cell types online, thus making it possible for researchers not familiar with programming to carry out related research.

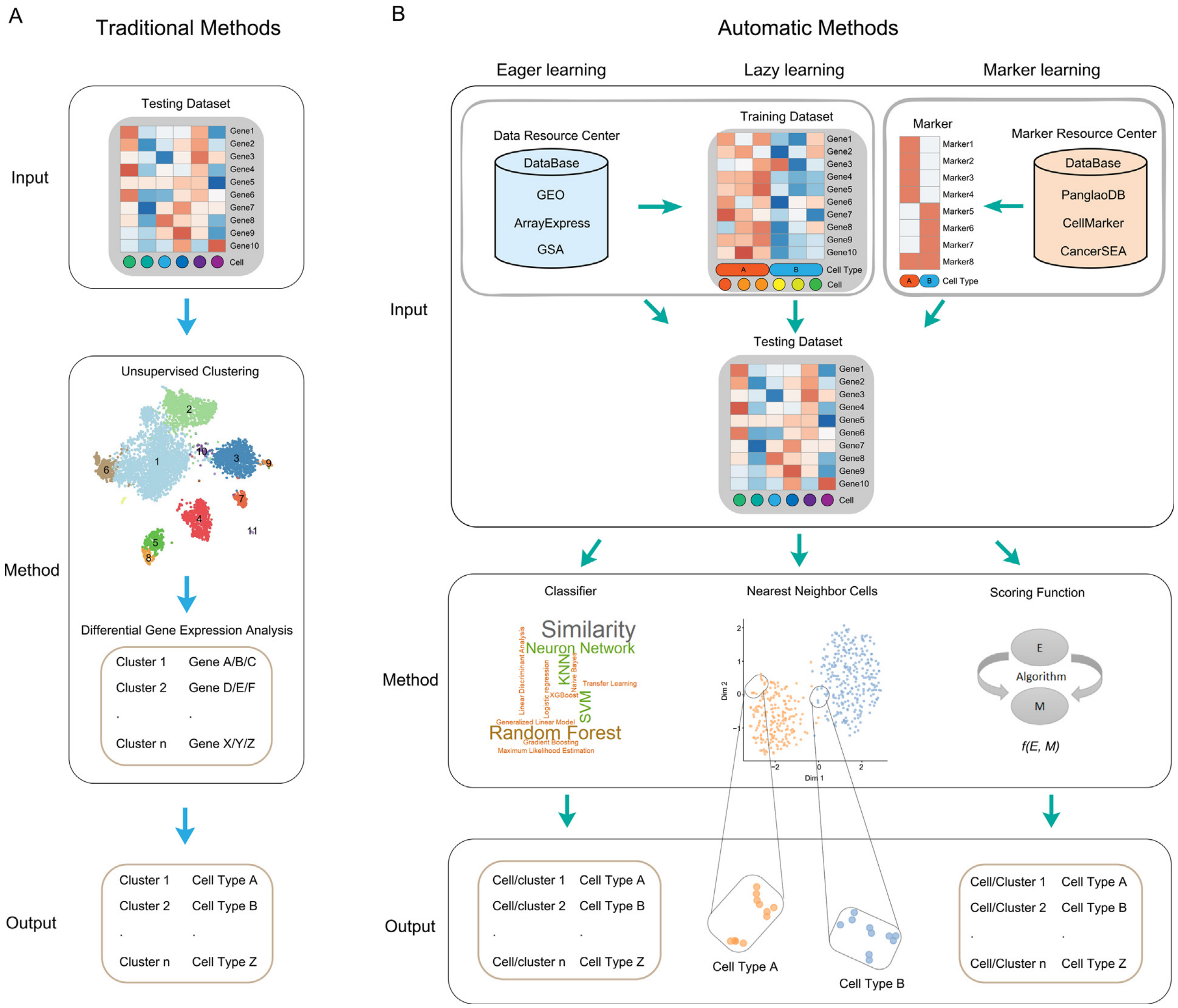

Fig. 1. Workflow of the traditional and automatic cell-type identification methods. A. The workflow of traditional cell-type identification methods showing that the input of traditional methods are the testing datasets. An unsupervised method is used to cluster the cells, and the differentially expressed genes of each cluster are detected. The cell types of each cluster are assigned by the canonical markers in the differentially expressed genes. B. The workflow of the automatic cell-type identification methods. The input of eager learning and lazy learning methods are the training datasets and testing datasets. The input of marker learning methods is the markers of each cell type and the testing datasets. The training datasets can be downloaded from the data resource centers (GEO, ArrayExpress and GSA). The markers of each cell type can be downloaded from the marker resource centers (PanglaoDB, CellMarker and CancerSEA). The methods used by eager learning, lazy learning and marker learning methods are classifiers, nearest neighbor cells, and the scoring functions, respectively. The cell types assigned by the automatic methods can be given to cells or clusters.

3. Features and models of automatic methods

Most automatic cell-type identification methods, except for scMatch and deep learning methods [32,33,36,37,45], have a feature selection procedure, which involves selection of the most beneficial or relevant features in model construction. This process makes the model more accurate, shortens the training time, avoids the curse of dimensionality, and enhances generalization. Features applied to automatic methods for cell-type assignment include “representative gene” and “transformed gene”. Many automatic methods utilize the “representative gene” feature directly through different gene selection strategies. For example, CellAtlasSearch, CHETAH, scID, and Garnett use highly expressed genes derived from bulk RNA-Seq or scRNA-seq data. High dropout genes [62], which are genes with a higher number of zero expression than expected, are used in scmap-cell, scmap-cluster, and scLearn. Genes with variance higher than the predefined threshold are regarded as features in Seurat and SingleR. Feature genes in CaSTLe are selected based on the highest gene expression and mutual information. Cell type-specific genes evaluating entropy differences between a specific cell type and other cell types are utilized in SciBet. Genes with a high expression rate in each cell type are selected in MarkerCount. Superscan uses the top 1000 genes calculated by the shap python package as features. Moreover, DigitalCellSorter, SCINA, SCSA, CellAssign, scClassifR, and scCATCH use classical genes obtained from literature as features. As an alternative to using genes as features directly, the “transformed gene” feature has also proven to be effective for cell-type identification. For example, principal components in scPred, embedding space calculated from the high-dimensional expression profile in CELL-BLAST, random singular values computed by using singular value decomposition in CellFishing.jl, and gene pairs obtained through expression relationships between genes in SingleCellNet can all yield a high prediction accuracy. clustifyr and scHPL don’t contain steps for feature selection, and high dropout genes and high variance genes are suggested as features.

Since there are many feature selection methods, it is important to know which features perform better than others. It has been

Table 1
Summary of automatic methods for cell-type identification.

Name of method	Type	Feature	Classifier	Prior knowledge provided	Ability to predict new cell type	Language	Input format
CellAtlasSearch	Lazy learning	Predefined genes with high fold change over respective median expression in at least one cell type	Nearest neighbor cell based on cosine similarity of Hamming locality-sensitive hashing	Information not available	Yes	Information not available	Count matrix
CELLBLAST	Lazy learning	Embedding space calculated from highly variable genes in Seurat	Nearest neighbor cell based on Euclidean distance and Wasserstein distance	Yes	Yes	Python	Count matrix
CellFishing.jl	Lazy learning	Random singular value calculated from filtered genes, whose maximum count across cells exceeds 10% quantile	Nearest neighbor cell based on cosine similarity of Hamming locality-sensitive hashing	No	Yes	Julia	Count matrix
scmap-cell	Lazy learning	High dropout genes (higher number of zero expression than expected) obtained from the M3Drop package	Nearest neighbor cell based on cosine similarity	No	Yes	R	Count matrix
ACTINN	Eager learning	All genes except outlier genes (the highest 1% and the lowest 1%)	Neural network with three hidden layers (100, 50, 25 nodes)	No	No	Python	Count matrix
CaSTLe	Eager learning	Genes with high expression and mutual information	XGBoost	No	Yes	R	Count matrix
CHETAH	Eager learning	200 selected genes that had the largest absolute fold change between the selected cell type and other cell types in a different hierarchical branch	Hierarchical determination based on feature gene expression profile	No	Yes	R	Count matrix
clustifyr	Eager learning	User defined features (features calculated by Seurat and M3Drop are recommended)	Cell type similarity based on Spearman correlation	No	Yes	R	Count or normalized matrix
Garnett	Eager learning	Selected genes whose expression is higher than the 90% quantile of each cell type	Grouped multinomial elastic-net regularized (α = 0.3) generalized linear model	Yes	Yes	R	Count matrix
MarkerCount	Eager learning/ Marker learning	Eager learning: Genes with a high expression rate in each cell type Marker learning: user defined markers	Self-defined score function	No	Yes	Python	Count matrix
MARS	Eager learning	All genes	Neural network	No	No	Python	Count matrix
scCapsNet	Eager learning	All genes (feature selection is embedded in the network)	Capsule network	No	No	Python	Normalized matrix
scClassifR	Eager learning	User defined markers of each cell type	SVM (RBF kernal)	No	Yes	R	Count matrix
SciBet	Eager learning	Cell type-specific genes evaluating entropy differences between a specific cell type and other cell types	Maximum likelihood estimation	Yes	Yes	R	Normalized matrix
scID	Eager learning	Genes specifically upregulated in the cluster of interest with estimated discriminative weights	Fisher's linear discriminant analysis	No	Yes	R	Count matrix
scLearn	Eager learning	High dropout genes (higher number of zero expressions than expected) obtained from the M3Drop package	Cell type similarity based on the transformation matrix from discriminative component analysis	Yes	Yes	R	Count matrix
scmap-cluster	Eager learning	High dropout genes (higher number of zero expression than expected) obtained from the M3Drop package	Cell type similarity based on cosine similarity, Pearson correlation and Spearman correlation	No	Yes	R	Count matrix
scMatch	Eager learning	All genes	Nearest cell type (the expression profiles of cell types from FANTOM5) by Spearman correlation	Yes	No	Python	Count matrix

(*continued on next page*)

Table 1 (*continued*)

Name of method	Type	Feature	Classifier	Prior knowledge provided	Ability to predict new cell type	Language	Input format
scHPL	Eager learning	User defined features (features calculated by Seurat is recommended)	Linear SVM	No	Yes	Python	Normalized matrix
scPred	Eager learning	Highly variable genes in Seurat	Support vector machine with a radial kernel or other models in the caret package (e.g., logistic regression, decision trees, bagging, neural networks)	No	Yes	R	Count matrix
scPretrain	Eager learning	All genes	Neural network	No	No	Python	Count matrix
scVI	Eager learning	All genes	Neural network	No	No	Python	Count matrix
Seurat	Eager learning	Highly variable genes, for which the variance of genes is higher than the threshold	Transfer learning	No	No	R	Count matrix
SingleCellNet	Eager learning	Gene pairs from genes preferentially expressed in each cell type	Random forest with 1000 trees	Yes	Yes	R	Count or normalized matrix
SingleR	Eager learning	Highly differentially expressed genes among each cell type	Nearest cell type (expression profile of the cell type could be from microarray, bulk RNA-seq or scRNA-seq data) by Spearman correlation	No	No	R	Count or normalized matrix
Superscan	Eager learning	1000 genes calculated by the “shap” python package	XGBoost	Yes	Yes	Python	Count matrix
CellAssign	Marker learning	User defined markers	Expectation-maximization inference	No	Yes	R	Count matrix
DigitalCellSorter	Marker learning	User defined markers	Voting algorithm	Yes	Yes	Python	Count matrix
scCATCH	Marker learning	Canonical markers of cell types from CELLMatch	Evidence-based score	Yes	Yes	R	Normalized matrix
SCINA	Marker learning	User defined markers	Expectation-maximization inference	No	Yes	R	Normalized matrix
SCSA	Marker learning	Canonical markers of cell types from CellMarker or CancerSEA	Self-defined score function	Yes	Yes	Python	Differentially expressed gene of clusters
scTyper	Marker learning	Canonical markers of cell types from CellMarker/ scTyper or user defined markers	Nearest cell type by cosine distance or Pearson correlation	No	Yes	R	Normalized matrix

shown that for CellFishing.jl, when the features selected by scmap-cell were used, the Cohen’s kappa scores increased [63]. On the other hand, for scmap-cell, when the features selected by CellFishing.jl were used, the Cohen’s kappa scores decreased [28]. These observations thus indicate that the scmap-cell features are better than the CellFishing.jl ones. scMatch recommends using all genes derived from scRNA-seq, rather than manually defined cell type-specific genes or highly expressed genes, as classifier features [46]. However, according to scmap-cell, high dropout genes perform better than highly variable genes or randomly selected genes in seventeen datasets across different platforms or species [27]. SciBet investigated feature performance of entropy differences, *F*-test, and scmap-cell using the same classifier among fourteen datasets, and the results showed that genes with high entropy differences yielded the highest classification accuracy [40]. In summary, the genes selected through entropy differences perform better than the other “representative gene” features.

Similar to the feature selection methods, there are many different predict models that can be used for automatic cell-type identification methods. The most commonly used model is the comparison of the similarity of the query cells with the cells or cell-type groups in the training datasets. For example, nearest neighbor cells are selected by cosine similarity in scmap-cell and CellFishing.jl, and the cell type is determined by these nearest neighbor cells. Cosine similarity is used in CellAtlasSearch and scTyper, and the cell type is determined using the similarity value. scMatch, SingleR, clustifyr, and CHETAH use Spearman correlation to determine cell types, whereas SingleR uses the 80th percentile of correlation values in each cell type to avoid heterogeneity [44]. CHETAH uses a self-defined “confidence score” calculated by the Spearman correlation to assign cell types [39]. Pearson correlation is used by default in scLearn, and Spearman correlation, cosine similarity, and Euclidean distance are supported in scLearn [35]. The scmap-cluster calculates cosine similarity, Spearman correlation, and Pearson correlation simultaneously to define cell types by method agreement, in which at least two of the similarities must be in agreement with high confidence [27]. In contrast to directly using gene expression to calculate similarity, CELLBLAST determines the similarity of cells based on Euclidean distance in low-dimensional embedding space and Wasserstein distance on

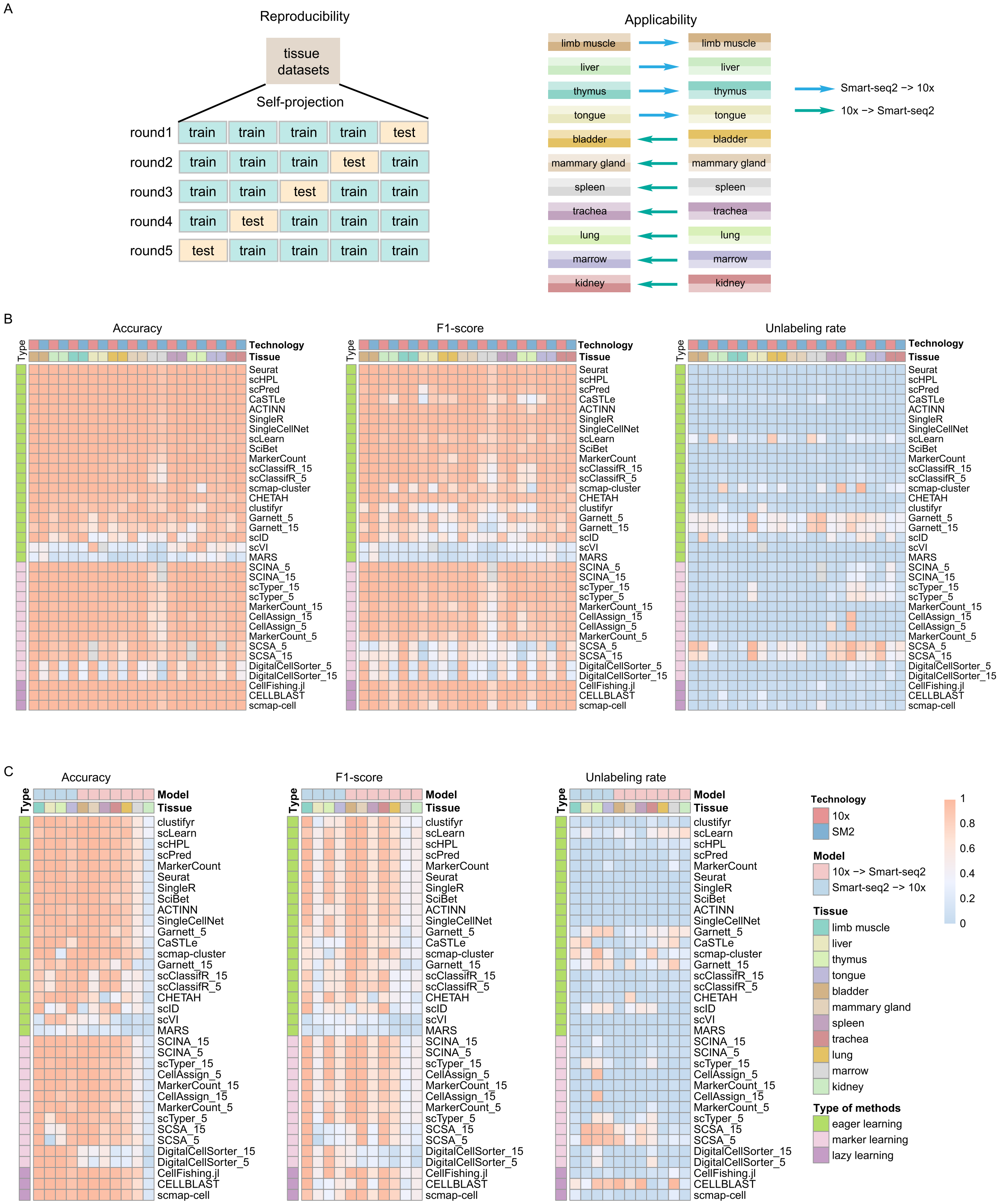

Fig. 2. Performance of the automatic cell-type identification methods using the Tabula Muris datasets. A. Schematic illustration of the automatic methods regarding reproducibility and applicability. Eleven mouse tissues (limb muscle, liver, thymus, tongue, bladder, mammary gland, spleen, trachea, lung, marrow and kidney) were used to test the self-projection. In applicability, the Smart-seq2 dataset of limb muscle, liver, thymus, and tongue is used as training datasets. The 10x datasets of bladder, mammary gland, spleen, trachea, lung, marrow and kidney are used as training datasets. B. The accuracy, F1-score and unlabeling rate in eleven mouse tissues. The heatmap is ordered by the accuracy in all three types of automatic methods. C. The accuracy, F1-score and unlabeling rate across different platforms. The heatmap is ordered by the accuracy in all the three types of automatic methods. The labels '5' and '15' in the marker learning methods and some of the eager learning methods indicate that they use the top 5 or top 15 differentially expressed markers.

posterior distributions [26]. General classifiers are also applied in the automatic methods, such as random forest in SingleCellNet, linear SVM in scHPL, SVM with a radial kernel in scPred and scClassifR, Fisher's linear discriminant analysis in scID, XGBoost in CaSTLe and Superscan, maximum likelihood estimation in SciBet, generalized linear model in Garnett, and transfer learning in Seurat. Popular deep learning methods, such as fully connected neural networks (ACTINN, MARS, scPretrain, and scVI) and capsule networks (scCapsNet), have also been used. For the marker learning methods, Expectation-Maximization algorithm in CellAssign and SCINA and Score function based on marker genes in DigitalCellSorter, SCSA, MarkerCount and scCATCH are used to predict cell types.

4. Cell type prediction performance of automatic methods

There are many evaluation criteria in cell type assignment, for example, accuracy [12,36,40], F1-score [35,42,53], Cohen's kappa [27,28,43], AUC [38,42], unlabeling rate [64], and mean-balanced [26]. CellAssign evaluates the performance of unsupervised methods (Seurat, SC3 [15], PhenoGraph [65], densityCut [66], dynamicTreeCut [67]), eager learning methods (scmap-cluster, correlation-based [68]) and marker learning methods (SCINA) on simulated data. In terms of accuracy and F1-score, CellAssign performs better than the above mentioned methods [53]. Compared with ACTINN, scmap, Seurat, SingleR and SVM, clustifyr achieves the highest accuracy using the Tabula Muris dataset [31]. The accuracy of SciBet outperforms Seurat and scmap in fourteen datasets [40]. Abdelaal et al found that linear SVM with rejection has the highest F1-score by comparing twenty-two automatic methods across tissues and platforms [64]. scLearn shows better accuracy than linear SVM with rejection in pancreas and PBMC datasets [35]. Here, we compare currently available automatic methods and discuss their performance. Reproducibility and applicability are used to evaluate the performance of automatic methods on tissue datasets. Reproducibility inspects whether the automatic methods have preference on specific tissues using self-projection (see Methods). Applicability verifies whether the automatic methods can predict the cell types of the testing dataset using training datasets (Fig. 2A). The self-projection of eleven tissues of the Tabula Muris dataset shows that automatic methods do not seem to have a bias on specific tissues (Fig. 2B). Most automatic methods achieved high accuracy and F1-score and low unlabeling rate across tissues except for marrow, since the marrow dataset consists of mostly immune cells with deep annotation, which is consistent in the PBMC data with deep annotation [64]. In eager learning methods, the mean accuracy of Seurat, scHPL, scPred, CaSTLe, ACTINN, SingleR, and SingleCellNet is greater than 0.98. The mean F1-score of Seurat, scHPL, SingleR, SingleCellNet, ACTINN is higher than 0.95. The unlabeling rates of these top automatic methods are low (<7%). In terms of accuracy, F1-score and unlabeling rate, the performance of the lazy learning methods CellFishing.jl, CELLBLAST and scmap-cell is similar to those of the top eager learning methods. Regarding accuracy and F1-score, the marker learning methods SCNIA, scTyper, MarkerCount and CellAssign have similar performance. Compared with the top eager learning methods, the mean accuracy and F1-score of the marker learning methods are slightly lower (∼4%). The unlabeling rate of the top marker learning methods ranges from 1% (MarkerCount) to 18% (scTyper).

Eleven tissues in the Tabula Muris dataset contain both 10x and Smart-seq2 platform datasets, which could be used to test the applicability of the automatic methods. In terms of self-projection, some tissues have relatively low accuracy and F1-score (Fig. 2C), mainly because the cell types in the testing datasets were not covered in the training datasets (Supplement Table 1). In kidney, marrow and lung, the numbers of cells in the missing cell types were high, resulting in low accuracy and F1-score, whereas in liver, tongue and spleen, the numbers of cells in the missing cell types were low, resulting in a low F1-score (Fig. 2C). Despite the effect of unequal cell types between training datasets and testing datasets, the cross-platform prediction achieves good performance like self-projection. These results imply that a comprehensive training dataset may lead to a better performance. In eager learning methods, clustifyr, scLearn and scHPL perform better than the other methods regarding accuracy, and scHPL, SciBet and SingleR have the highest F1-score. Among the top automatic methods, only scLearn has a high unlabeling rate. In lazy learning methods, CellFishing.jl performs better than CELLBLAST and scmap-cell regarding F1-score, and CELLBLAST has a higher unlabeling rate. In marker learning methods, SCNIA, scTyper, CellAssign and MarkerCount achieve high performance.

Since the Tabula Muris dataset used only two platforms, it may not reflect the properties of automatic methods regarding cross-platform prediction. The PBMC data, however, use training datasets from seven platforms and testing datasets from six platforms, and therefore may be more suitable to test the cross-platform prediction of automatic methods [69]. The three indicators (accuracy, F1-score and unlabeling rate) show a similar pattern across platforms, suggesting that automatic methods may not be significantly affected by different platforms (Fig. 3A). Regarding the three indicators, marker-based methods (all marker learning methods and Garnett/scClassify in eager learning methods) differ significantly from the other methods using 10x-V3 and CEL-Seq2 as training datasets (Fig. 3A). When using the canonical maker, however, their performance becomes better (Fig. 3A). The poor performance of differentially expressed markers was observed previously [64]. In eager learning methods, the top three methods in terms of mean accuracy are clustifyr, scLearn and scPred, and the top three methods in terms of mean F1-score are scPred, Seurat and SingleCellNet. In marker learning methods, CellAssign, SCSA and MarkerCount perform better than the other methods in terms of accuracy and F1-score. Compared with the top eager learning methods, the mean accuracy and F1-score of marker learning methods are slightly lower. In lazy learning methods, CellFishing.jl and scmap-cell perform better than CELLBLAST regarding the three indicators.

Merely using normal datasets to predict cell types might limit the application of automatic methods. Therefore, it needs to be evaluated whether automatic methods can detect malignant cells in tumorous tissues. In one study, the scRNA data of normal lung and late-stage lung cancer tissues were used as training and testing dataset respectively to test the sensitivity and specificity of automatic methods to predict malignant tumor cells [70]. At the same time, the ability to predict other types of cells was also compared, and the unlabeled cells predicted by the automatic methods were regarded as malignant tumor cells. The clustifyr method showed better performance than the other automatic methods in detecting malignant tumor cells with nearly 100% specificity and sensitivity (Fig. 3B). However, the F1-score of clustifyr for normal cells was not within the top methods, which are consistent with the Tabula Muris dataset and PBMC dataset (Fig. 2C, Fig. 3A). CellAssign, CellFishing.jl and scTyper have high sensitivity and low specificity, which means that the predicted malignant tumor cells have a high confidence (Fig. 3B). The marker learning methods and lazy learning methods achieved better performance in the tumor dataset (Fig. 3B), different from the result using normal tissue or PBMC datasets.

5. Comparison of the speed of the automatic methods

Massive amount of scRNA-seq data are being continuously generated. For one example, the Human Cell Atlas project aims to create reference maps of all human cells [71]. Thus, the speed of the automatic cell-type identification methods is a critical factor to be considered. Using a mouse brain dataset [72], we varied the training and testing datasets randomly to test the speed of the automatic methods (Fig. 4A). The fastest method appears to be scmap-cluster, and the other fast methods include SCINA, SingleCellNet, SciBet, SingleR and scHPL. The computation time of all automatic methods increased with larger training or testing

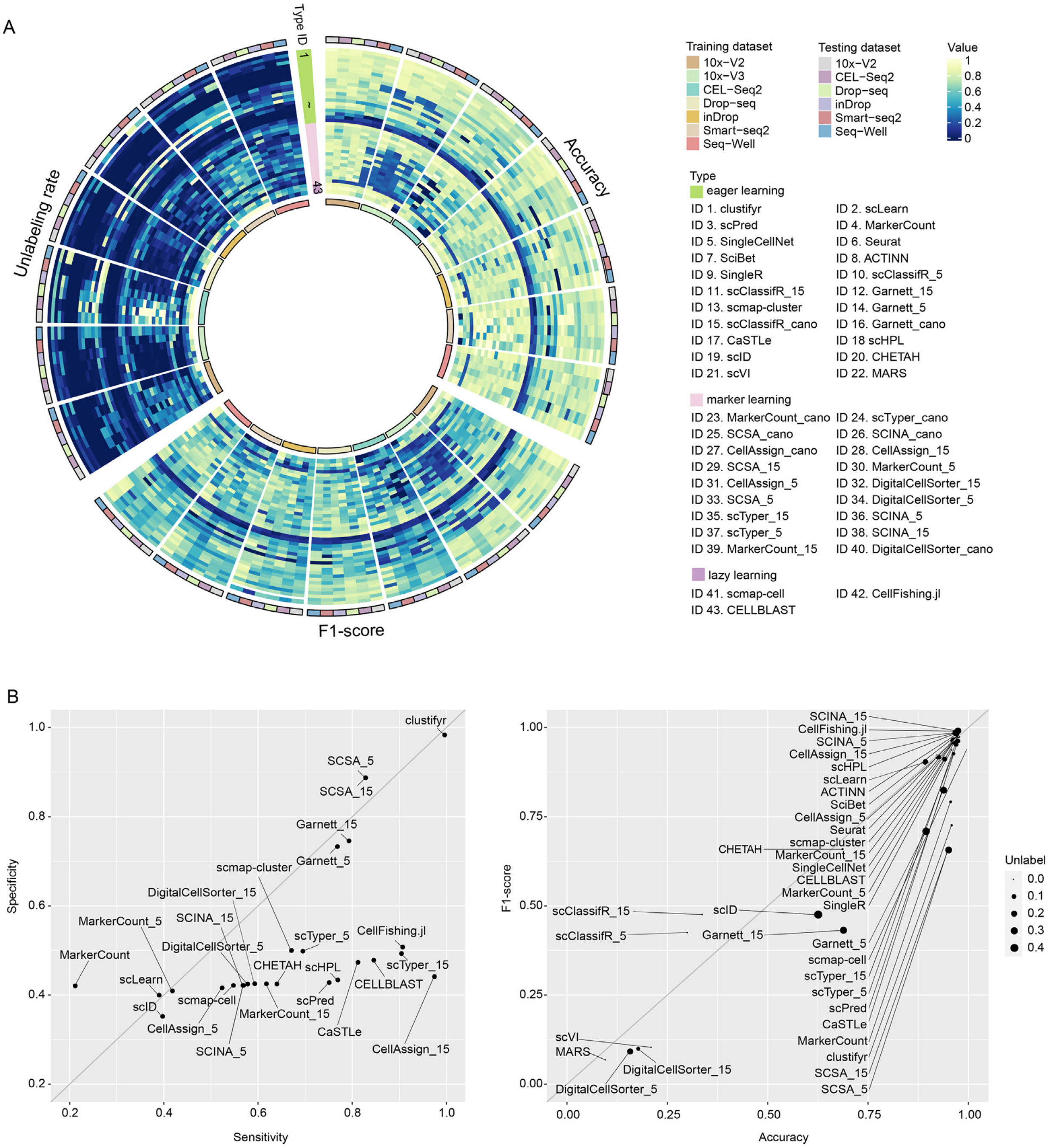

Fig. 3. Performance of the automatic cell-type identification methods using PBMC and tumor datasets. A. Circos plot shows the accuracy, F1-score and unlabeling rate of the PBMC datasets. The methods are ordered by the accuracy in all three types of automatic methods. "Cano" in marker learning methods or some of the eager learning methods: canonical markers. B. The performance of the automatic methods using human normal lung data to predict Tabula Muris lung data. As ACTINN, MARS, SciBet, scVI, Seurat and SingleR did not predict unlabeled cells, they are not included the calculation of sensitivity and specificity of tumor cells. scClassifR and SingleCellNet are not included since they did not predicate any unlabeled cells.

dataset (Fig. 4B, C). The increase of computing time of SingleCellNet and CELLBLAST is much smoother, suggesting that the computation time does not increase exponentially. As expected, the neural network-based methods (scVI, MARS, CellAssign and CELLBLAST) need more time than the other methods, since a huge number of parameters is needed to train the model. Overall, automatic methods show excellent performance in terms of speed with more than half of the methods needing no more than 100 s. The performance of eager learning methods is better than those of lazy learning and marker-based methods.

6. New cell type prediction by different automatic methods

The unlabeled cells are designed to catch the new cell types that do not exist in the training dataset. The goal of the automatic methods in predicting new cell types is to lower the false prediction rate and to increase the true prediction rate. For lazy learning methods, CellFishing.jl and scmap-cell use similar strategies to identify new cell types through nearest-neighbor cells [27,28]. When the cell type of the *K*-nearest neighbor (usually 10) cells is not exactly the same, the cells are assigned to the "unlabeled" cat-

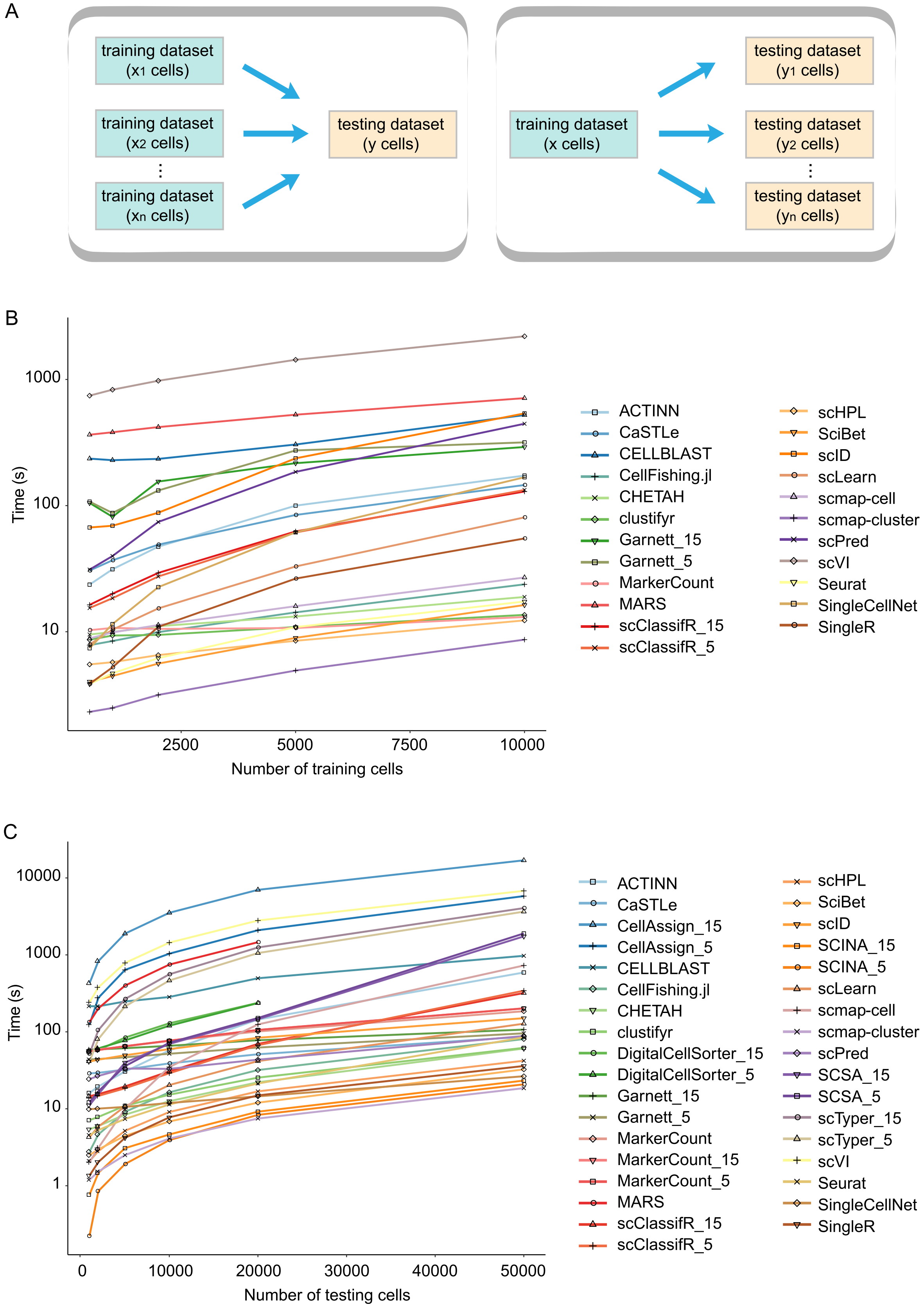

Fig. 4. Speed of automatic cell-type identification methods. A. Speed of the automatic methods. A fixed size of the testing dataset and varying sizes of the training datasets are used to test the computation time using different training datasets. Also, a fixed size of the training dataset and varying sizes of testing datasets are used to test the computation time using different testing datasets. B. The computation time of the automatic methods with the training dataset set at 500, 1000, 2500, 5000 and 10,000 cells, and the testing dataset set at 5000 cells. The marker learning methods are not included since they do not require training datasets. C. The computation time of the automatic methods with the training dataset set at 700 cells, and the testing dataset set at 1000, 2000, 5000, 10,000, 20,000 and 50,000 cells.

egory. scmap-cell has an additional condition when the maximum similarity of the value of the *K*-nearest neighbor cells is lower than the empirical threshold [27]. CELLBLAST first searches the nearest neighbor cells in a low-dimensional embedding space, then computes its significance on Wasserstein distance [26]. Because Wasserstein distance of nearest neighbor cells needs to be significant in multiple models, some predicted cells may not have nearest neighbor cells, thus leading to a relatively high unlabeling rate.

For the eager learning methods, scmap-cluster, CHETAH, clustifyr, and scLearn unlabel a cell when the correlation value is less than the empirical threshold [27,35,39]. Due to the inherent heterogeneity and complexity of scRNA-seq data and cell types [73], it is not suitable to apply one empirical threshold to identify novel cell types [35]. scLearn solves this issue by learning the thresholds of the last 1% of the similarity distribution for each cell type from the training dataset[35]. scPred, CaSTLe, scClassifR, and scID use posterior probability instead of correlation to assign unlabeled cells. However, posterior probability may misclassify cells into similar cell types. Therefore, the unlabeling rate could be underestimated [64]. SingleCellNet randomly selects a small part of the training dataset as a “random” cell type [43]. Since the “random” cell type usually does not have a unique gene expression profile, testing datasets thus may exhibit a low chance of showing high correlation with the “random” cell type, thus reflecting a considerably low unlabeling rate. SciBet uses a group of datasets as the background datasets, in which the training dataset is not included. The predicted cells are classified as unlabeled if they express more marker genes in the background datasets than in the training dataset [40]. scHPL computes the distance between original data and inverse transformed data from the training dataset’s PC space. If the distance is higher than the threshold determined on the training data, the cell is considered an unlabeled cell. Instead of predicting unlabeled cells, Superscan assign cell types with high, medium, and low confidence based on the entropy value of predicting the probability of each cell type.

For the marker learning methods, scTyper assumes that the expression of a cell type specific marker is 1, and the expression of the other markers 0. Then the value of cosine distance or correlation of each cell type is calculated. If the value is not significantly higher than that of the randomly generated sample for each cell type, it will be assigned as an unlabeled cell. CellAssign, MarkerCount, and DigitalCellSorter use posterior probabilities to assign unlabeled cells. The cell cluster in SCSA and scCATCH is categorized as “unlabeled” when the marker genes of the cluster do not match the canonical markers [52,54]. SCINA assumes that the unknown cell types do not express any canonical markers [51]. Overall, the unlabeling rates of automatic methods do not appear to be satisfying in terms of accuracy, which remains a challenge to be addressed in the future.

7. Summary and outlook

Automatic cell-type identification methods emerged only in recent years in 2018 [27]. However, the growing usage and rapid data production of scRNA-seq technology have made scRNA-seq data analysis a major challenge in the field. In this review, we systematically compared the features, classifiers, models, predictive performance, speed and the new cell-type prediction ability of currently available automatic cell-type identification methods. Regarding accuracy, F1-score, unlabeling rate, specificity and sensitivity of tumor cells and speed, the best performing methods among the three types of automatic methods (eager learning, lazy learning and marker learning) give similar outcomes (Fig. 5). Among the eager learning methods, clustifyr, scHPL and scPred show good performance across all indicators. SingleCellNet, SciBet and Seurat perform well on accuracy, F1-score and speed. Among the lazy learning methods, CellFishiing.jl appears to be the best method. For the marker learning methods, SCSA, SCINA, scTyper and CellAssign show good performance.

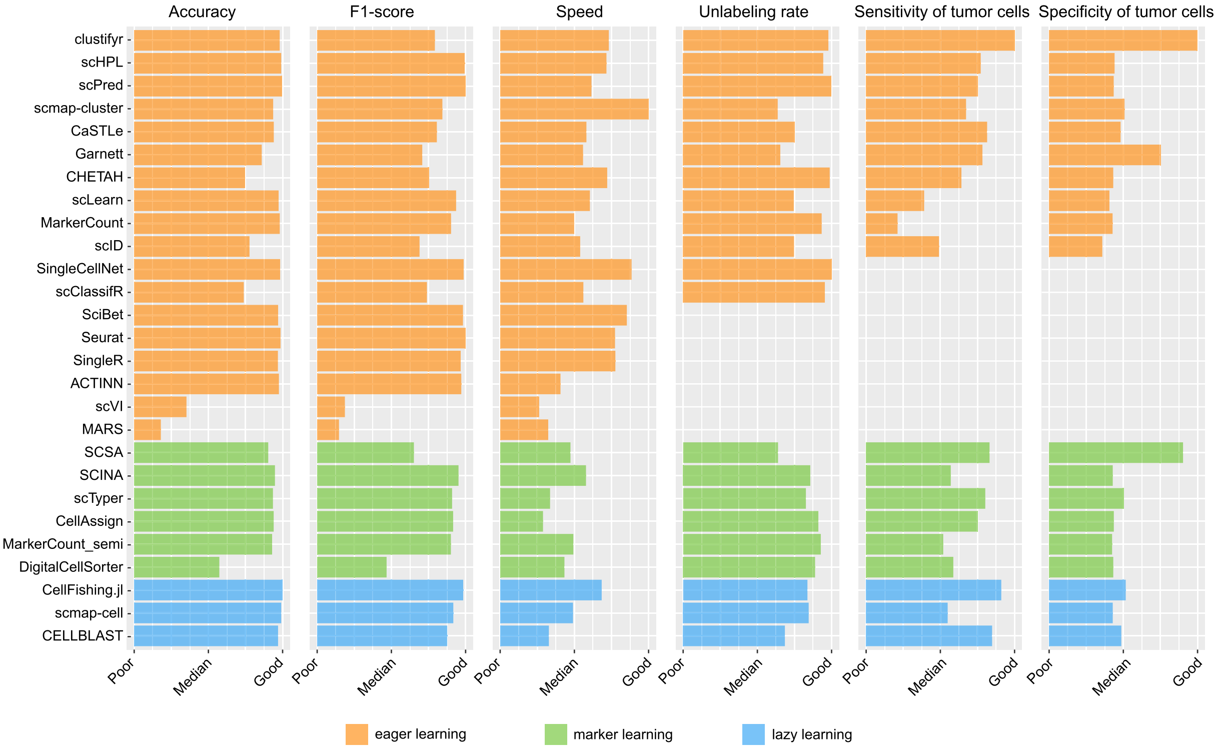

Fig. 5. Summary of performance of the automatic cell-type identification methods. Bar graphs of the automatic cell-type identification methods with six evaluation criteria indicated. For each evaluation criteria, the length of the bars shows the performance of the automatic method: poor, median or good. The automatic methods are sorted based on the mean performance of the evaluation criteria. No bar: not evaluated.

Although the automatic methods can predict cell types of scRNA-seq data automatically, they still require prior knowledge (training datasets or canonical cell markers) regarding cell types, like the traditional marker-based methods. For eager learning and lazy learning methods, gene expression data of each cell type are still needed. Marker learning methods require canonical markers for each cell type. When pre-annotated datasets of tissue or organs are available [3–5], eager learning and lazy learning methods are preferable in terms of accuracy and speed. When label information is not provided in the training datasets or when training datasets are not available, marker-based (marker learning) methods are recommended.

Moreover, prior knowledge, such as the number of cell types in the training dataset, significantly affects the performance of the automatic methods. To avoid such an issue, a more comprehensive training dataset is required. In scRNA-seq studies, researchers might focus on specific types of cells in a tissue, e.g., immune cells [74] or stromal cells [75]. This may lead to the loss of information on the other types of cells in the tissue. As such, these subset datasets should not be used to predict whole tissue datasets. Furthermore, the ability of the automatic methods to predict new cell types may not be sufficient at this stage. The unlabeled cell classification is designed to identify new cell types. However, this classification could be assigned to cells of similar types. Existing similarity evaluation, posterior probability evaluation, and design of pseudo cell types are insufficient for new cell type prediction, and better approaches are still needed.

In addition, another challenge is the application of automatic cell-type identification methods in processing datasets involving embryonic development [76] and tumor studies [77]. For embryo development datasets, the expression profiles of cell types may display a high degree of similarity, thus can lead to a high unlabeling rate. In addition, if the time points, in which the cell types fall into during embryonic development, are considered, processing such datasets becomes even more complicated. scLearn provides a solution that assigns cells with synthetic labels (time point and cell type) by combining two label types into one combined label [35]. The combination is not restricted to two labels and multiple labels can be combined, which may improve single-cell data analysis. In tumors, immune cells can undergo functional changes during the transition of normal tissue to malignancy in tumors [78]. It is therefore important to label heterogeneous immune cells (normal and abnormal). In CellAssign, malignant B cells were found to lose IGKC expression and upregulate IGLC compared with normal B cells. Therefore, CellAssign uses this differentially expressed marker gene to automatically identify malignant or normal B cells [53]. Detecting these heterogeneous immune cells through automatic cell-type identification can be highly useful for early-stage tumor detection. In recent years, other types of single-cell sequencing technology data, such as spatially resolved transcriptomic data [79] and scATAC-seq data [80], have been integrated with scRNA-seq data. Based on these data, substantial new computational developments are expected to further improve the automatic cell-type identification methods.

The automatic cell-type identification methods are being continuously developed. Meanwhile, the quality of prior knowledge has become a more and more important factor for automatic cell type prediction. scLearn, SciBet, SingleCellNet, scMatch, Garnett and DigitalCellSorter provide well-trained models or markers that can facilitate the use of these methods. However, the prior knowledge of these methods often comes from a single dataset, which may miss certain cell types due to the limitations of different sequencing platforms and experimental approaches [69,81]. CELLBLAST provides more comprehensive training datasets by integrating multiple datasets, which subsequently contains more cell types compared with a single training dataset [26]. Such integrated datasets therefore may be more commonly used as training datasets in the future. Yet, for integrated datasets, there are still two issues to be solved. The first is to try to avoid the influences of different sequencing technologies during the process of data integration, for example, by using MNN [82], CCA [12], LIGER [83], Scanorama [84], et al. The second is to try to unify the currently inconsistent annotation levels in the training datasets, for example, by the joint usage of multiple training datasets [85], or by manual curation of each training dataset. These processes may lead to a more comprehensive and integrated datasets that can be a valuable label resource for marker learning methods. In summary, due to the potential advantages of good accuracy, fast prediction, and effective usage of available datasets, automatic cell-type identification methods have the potential of replacing traditional cell-type identification methods for normal tissues, and may be more widely used by researchers in different research fields.

8. Methods

8.1. Data preprocessing

The count matrix and manually annotated labels are downloaded from public resources (see Data and code availability). For the Tabula Muris scRNA-seq dataset, genes not expressed in any cells were filtered. Cells with the below features were filtered: the number of expressed genes in the cells is less than 200, the number of expressed genes in the cells exceed twice the standard deviation of the mean, the expression of mitochondrial genes in the cells is more than 20%. Cells with no cell type information were deleted. Some tissues and organs do not have both 10x and Smart-seq2 data, such as aorta, brain myeloid, brain non-myeloid, diaphragm, fat, heart, large intestine and skin tissues. They therefore were not included in the analysis. Tissues containing more cell types were selected as training datasets for cross-platform prediction. The human normal lung and lung tumor datasets are downsampled (~10,000 cells) with an equal proportion of each cell type by the ‘createDataPartition’ function in the ‘caret’ R package.

8.2. Self-projection

The Tabula Muris scRNA-seq datasets were divided into five sections subsets according to the cell type labels using the ‘createFolds’ function in the ‘caret’ R package. Four subsets were used as training datasets, and the remaining subset was used as the testing dataset. After five rounds of cross-validation, the mean values of accuracy, F1-score and unlabeling rates were used to evaluate the reproducibility of the automatic cell-type identification methods.

8.3. Evaluation indicators

Accuracy is the ratio of correctly predicted cells divided by the total labeled cells (excluding the unlabeled cells). F1-score is the harmonic mean of the precision and recall. In multiple cell type assignment, F1-score is the mean value of each cell type as listed below:

$$\sum_{i}^{n} \frac{1}{n} \times \frac{2 \times precision_i \times recall_i}{precision_i + recall_i}$$

Unlabeling rate is the ratio of unlabeled cells (‘unknown’, ‘unassigned’, etc.) divided by the total cells. Specificity is the proportion of negatives in a binary classification that is correctly identified. Sensitivity is the proportion of positives in a binary classification that is correctly identified.

8.4. Markers

The 'Seurat' pipeline is used to find cell type specific markers. The count matrix is normalized using the NormalizeData function, the markers are selected using the FindAllMarkers function using the manually annotated labels as 'active.ident'. The 'only.pos' is set as true. The other parameters are set to default.

The top 5 and 15 genes are used in the marker learning methods and some of the eager learning methods (Garnett and scClassify). Canonical markers of PBMC datasets are downloaded from the published reports (https://bitbucket.org/jerry00/scumi-dev/src/master/R/marker_gene/human_pbmc_marker.rda). Highly expressed genes of each cell type are used as canonical markers.

8.5. Overall performance score

Overall performance of automatic methods consists of accuracy, F1-score, speed, unlabeling rate, sensitivity and specificity. The mean accuracy, F1-score and unlabeling rate are calculated using the mouse tissue datasets, PBMC datasets and human normal lung datasets. The performance score of mean accuracy and F1-score is calculated by dividing the maximum value into all methods. The '1 – unlabeling rate' is used as the unlabeling rate performance score. The speed of predicting 20,000 cells in the testing dataset is calculated from brain datasets. The reciprocal of the base 10 logarithms of computing time is used as the speed performance score. The sensitivity and specificity of tumor cells are calculated using the lung tumor dataset. The performance score of sensitivity and specificity are calculated by dividing the maximum value into all methods. The automatic methods based on 5 and 15 markers are merged together by computing their mean value. The prediction of the canonical markers is not included in the overall score.

8.6. Data and code availability

The scRNA-seq data used in this work are available from public resources: mouse tissues (https://tabula-muris.ds.czbiohub.org/), PBMC (SCP424 in Single Cell Portal), human normal lung and lung tumors (GSE131907 in GEO) and brain (GSE116470 in GEO). To ensure the reproducibility and extensibility of the automatic cell-type identification functions described in this work, we integrated these methods into the R package AutomaticCellTypeIdentification with the same usage formats. All codes are available at https://github.com/xiebb123456/AutomaticCellTypeIdentification.

Automatic cell-type identification methods evaluated

Name of method	Version	URL
CELLBLAST	v0.3.8	https://github.com/gao-lab/Cell_BLAST
CellFishing.jl	v0.3.2	https://github.com/bicycle1885/CellFishing.jl
scmap-cell	v1.6.0	https://github.com/hemberg-lab/scmap
ACTINN	master	https://github.com/mafeiyang/ACTINN
CaSTLe	v1.0.0.2	https://github.com/yuvallb/CaSTLe
CHETAH	v1.2.0	https://github.com/jdekanter/CHETAH
Garnett	v0.1.19	https://github.com/cole-trapnell-lab/garnett
SciBet	v0.1.0	https://github.com/zwj-tina/scibetR
scID	v2.1	https://github.com/BatadaLab/scID
scLearn	v1.0	https://github.com/bm2-lab/scLearn
scmap-cluster	v1.6.0	https://github.com/hemberg-lab/scmap
scPred	v1.9.0	https://github.com/powellgenomicslab/scPred
scVI	v0.4.1	https://github.com/YosefLab/scvi-tools
Seurat	v3.2.2	https://github.com/satijalab/seurat
SingleCellNet	v0.1.0	https://github.com/pcahan1/singleCellNet
SingleR	v1.1.1	https://github.com/dviraran/SingleR
CellAssign	v0.99.21	https://github.com/Irrationone/cellassign
DigitalCellSorter	v1.1	https://github.com/sdomanskyi/DigitalCellSorter
SCINA	v1.2.0	https://github.com/jcao89757/SCINA
SCSA	master	https://github.com/bioinfo-ibms-pumc/SCSA
scTyper	v0.1.0	https://github.com/omicsCore/scTyper
scHPL	V0.0.2	https://github.com/lcmmichielsen/scHPL
MARS	master	https://github.com/snap-stanford/mars
clustifyr	v1.5.0	https://github.com/rnabioco/clustifyr
scClassifR	v1.1.1	https://github.com/grisslab/scClassifR
MarkerCount	master	https://github.com/combio-dku/MarkerCount/tree/master

CRediT authorship contribution statement

Bingbing Xie: Conceptualization, Methodology, Data curation, Formal analysis, Writing-original draft, Writing-review & editing. **Qin Jiang:** Conceptualization, Methodology, Supervision, Funding acquisition, Writing-review & editing. **Antonio Mora:** Conceptualization, Methodology, Supervision, Writing-review & editing. **Xuri Li:** Conceptualization, Methodology, Supervision, Funding acquisition, Writing-review & editing.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

This study is supported by the State Key Laboratory of Ophthalmology, Zhongshan Ophthalmic Center at the Sun Yat-Sen University, the National Natural Science Foundation of China (82070983 and 81870679 to Q.J.), and a Key Program of Guangzhou Scientific Research Plan (201804020010).

Appendix A. Supplementary data

Supplementary data to this article can be found online at https://doi.org/10.1016/j.csbj.2021.10.027.

References

[1] Tang F, Barbacioru C, Wang Y, Nordman E, Lee C, Xu N, et al. mRNA-Seq whole-transcriptome analysis of a single cell. Nat Methods 2009;6:377–82.
[2] Islam S, Kjallquist U, Moliner A, Zajac P, Fan J-B, Lonnerberg P, et al. Characterization of the single-cell transcriptional landscape by highly multiplex RNA-seq. Genome Res 2011;21:1160–7.
[3] Schaum N, Karkanias J, Neff NF, May AP, Quake SR, et al. Single-cell transcriptomics of 20 mouse organs creates a Tabula Muris. Nature 2018;562:367.
[4] Han X, Wang R, Zhou Y, Fei L, Sun H, Lai S, et al. Mapping the mouse cell atlas by microwell-seq. Cell 2018;173:1307.
[5] Han X, Zhou Z, Fei L, Sun H, Wang R, Chen Y, et al. Construction of a human cell landscape at single-cell level. Nature 2020;581:303–9.
[6] Shekhar K, Lapan SW, Whitney IE, Tran NM, Macosko EZ, Kowalczyk M, et al. Comprehensive classification of retinal bipolar neurons by single-cell transcriptomics. Cell 2016;166:1308–1323.e30.
[7] Lavin Y, Kobayashi S, Leader A, Amir E-A, Elefant N, Bigenwald C, et al. Innate immune landscape in early lung adenocarcinoma by paired single-cell analyses. Cell 2017;169:750–765.e17.
[8] Paik DT, Cho S, Tian L, Chang HY, Wu JC. Single-cell RNA sequencing in cardiovascular development, disease and medicine. Nat Rev Cardiol 2020;17:457–73.
[9] Johnson MB, Walsh CA. Cerebral cortical neuron diversity and development at single-cell resolution. Curr Opin Neurobiol 2017;42:9–16.
[10] Ma S, Sun S, Geng L, Song M, Wang W, Ye Y, et al. Caloric restriction reprograms the single-cell transcriptional landscape of Rattus norvegicus aging. Cell 2020;180:984–1001.e22.
[11] Butler A, Hoffman P, Smibert P, Papalexi E, Satija R. Integrating single-cell transcriptomic data across different conditions, technologies, and species. Nat Biotechnol 2018;36:411–20.
[12] Stuart T, Butler A, Hoffman P, Hafemeister C, Papalexi E, Mauck WM, et al. Comprehensive integration of single-cell data. Cell 2019;177:1888–1902.e21.
[13] Xu C, Su Z. Identification of cell types from single-cell transcriptomes using a novel clustering method. Bioinformatics 2015;31:1974–80.
[14] Wang B, Zhu J, Pierson E, Ramazzotti D, Batzoglou S. Visualization and analysis of single-cell RNA-seq data by kernel-based similarity learning. Nat Methods 2017;14:414–6.
[15] Kiselev VY, Kirschner K, Schaub MT, Andrews T, Yiu A, Chandra T, et al. SC3: consensus clustering of single-cell RNA-seq data. Nat Methods 2017;14 (5):483–6.
[16] Zhang JM, Fan J, Fan HC, Rosenfeld D, Tse DN. An interpretable framework for clustering single-cell RNA-Seq datasets. BMC Bioinf 2018;19:93.
[17] Zeisel A, Muñoz-Manchado AB, Codeluppi S, Lönnerberg P, La Manno G, Juréus A, et al. Brain structure. Cell types in the mouse cortex and hippocampus revealed by single-cell RNA-seq. Science 2015;347:1138–42.
[18] Zurauskiene J, Yau C. pcaReduce: hierarchical clustering of single cell transcriptional profiles. BMC Bioinf 2016;17:140.
[19] Zhang H, Lee CAA, Li Z, Garbe JR, Eide CR, Petegrosso R, et al. A multitask clustering approach for single-cell RNA-seq analysis in Recessive Dystrophic Epidermolysis Bullosa. PLoS Comput Biol 2018;14:e1006053.
[20] Grün D, Muraro M, Boisset J-C, Wiebrands K, Lyubimova A, Dharmadhikari G, et al. De Novo prediction of stem cell identity using single-cell transcriptome data. Cell Stem Cell 2016;19:266–77.
[21] Kiselev VY, Andrews TS, Hemberg M. Challenges in unsupervised clustering of single-cell RNA-seq data. Nat Rev Genet 2019;20:273–82.
[22] Duò A, Robinson MD, Soneson C. A systematic performance evaluation of clustering methods for single-cell RNA-seq data. F1000Res 2018;7:1141.
[23] Saelens W, Cannoodt R, Todorov H, Saeys Y. A comparison of single-cell trajectory inference methods. Nat Biotechnol 2019;37:547–54.
[24] Armingol E, Officer A, Harismendy O, Lewis NE. Deciphering cell-cell interactions and communication from gene expression. Nat Rev Genet 2021;22:71–88.
[25] Altschul SF, Gish W, Miller W, Myers EW, Lipman DJ. Basic local alignment search tool. J Mol Biol 1990;215:403–10.
[26] Cao ZJ, Wei L, Lu S, Yang DC, Gao G. Searching large-scale scRNA-seq databases via unbiased cell embedding with Cell BLAST. Nat Commun 2020;11:1–13.
[27] Kiselev VY, Yiu A, Hemberg M. scmap: projection of single-cell RNA -seq data across data sets. Nat Methods 2018;15:359–62.
[28] Sato K, Tsuyuzaki K, Shimizu K, Nikaido I. Cell Fishing.jl: an ultrafast and scalable cell search method for single-cell RNA sequencing. Genome Biol 2019;20:1–23.
[29] Srivastava D, Iyer A, Kumar V, Sengupta D. Cell AtlasSearch: a scalable search engine for single cells. Nucleic Acids Res 2018;46:W141–7.
[30] Michielsen L, Reinders MJT, Mahfouz A. Hierarchical progressive learning of cell identities in single-cell data. Nat Commun 2021;12:2799.
[31] Fu R, Gillen AE, Sheridan RM, Tian C, Daya M, Hao Y, et al. clustifyr: an R package for automated single-cell RNA sequencing cluster classification. F1000Res 2020;9:223.
[32] Brbić M, Zitnik M, Wang S, Pisco AO, Altman RB, Darmanis S, et al. MARS: discovering novel cell types across heterogeneous single-cell experiments. Nat Methods 2020;17:1200–6.
[33] Zhang R, Luo Y, Ma J, Zhang M, Wang S (2020) scPretrain: Multi-task self-supervised learning for cell type classification. bioRxiv.
[34] Shasha C, Tian Y, Mair F, Miller HER Gottardo R (2021) Superscan: Supervised Single-Cell Annotation. bioRxiv.
[35] Duan B, Zhu C, Chuai G, Tang C, Chen X, Chen S, et al. Learning for single-cell assignment. Sci Adv 2020;6:eabd0855.
[36] Wang L, Nie R, Yu Z, Xin R, Zheng C, Zhang Z, et al. An interpretable deep-learning architecture of capsule networks for identifying cell-type gene expression programs from single-cell RNA-sequencing data. Nat Mach Intell 2020;2:693–703.
[37] Ma FY, Pellegrini M. ACTINN: automated identification of cell types in single cell RNA sequencing. Bioinformatics 2020;36:533–8.
[38] Lieberman Y, Rokach L, Shay T. CaSTLe - Classification of single cells by transfer learning: Harnessing the power of publicly available single cell RNA sequencing experiments to annotate new experiments. PLoS ONE 2018;13: e0208349.
[39] De Kanter JK, Lijnzaad P, Candelli T, Margaritis T, Holstege FCP. CHETAH: a selective, hierarchical cell type identification method for single-cell RNA sequencing. Nucleic Acids Res 2019;47. e95-e96.
[40] Li C, Liu B, Kang B, Liu Z, Liu Y, Chen C, et al. SciBet as a portable and fast single cell type identifier. Nat Commun 2020;11:1–8.
[41] Boufea K, Seth S, Batada NN. scID uses discriminant analysis to identify transcriptionally equivalent cell types across single-cell RNA-Seq data with batch effect. Iscience 2020;23:100914.
[42] Alquicira-Hernandez J, Sathe A, Ji HP, Nguyen Q, Powell JE. scPred: accurate supervised method for cell-type classification from single-cell RNA-seq data. Genome Biol 2019;20:737–46.
[43] Tan Y, Cahan P. SingleCellNet: A computational tool to classify single cell RNA-Seq data across platforms and across species. Cell Systems 2019;9:207.
[44] Aran D, Looney AP, Liu L, Wu E, Fong V, Hsu A, et al. Reference-based analysis of lung single-cell sequencing reveals a transitional profibrotic macrophage. Nat Immunol 2019;20:163–72.
[45] Lopez R, Regier J, Cole MB, Jordan MI, Yosef N. Deep generative modeling for single-cell transcriptomics. Nat Methods 2018;15:1053–8.
[46] Hou R, Denisenko E, Forrest ARR. scMatch: a single-cell gene expression profile annotation tool using reference datasets. Bioinformatics 2019;35:4688–95.
[47] Nguyen V & Griss J (2020) scClassifR: Framework to accurately classify cell types in single-cell RNA-sequencing data. bioRxiv.
[48] Pliner HA, Shendure J, Trapnell C. Supervised classification enables rapid annotation of cell atlases. Nat Methods 2019;16:983.
[49] Choi JH, In Kim H, Woo HG. scTyper: a comprehensive pipeline for the cell typing analysis of single-cell RNA-seq data. BMC Bioinf 2020;21:342.
[50] Domanskyi S, Szedlak A, Hawkins NT, Wang J, Paternostro G, Piermarocchi C. Polled Digital Cell Sorter (p-DCS): Automatic identification of hematological cell types from single cell RNA-sequencing clusters. BMC Bioinf 2019;20:1–6.
[51] Zhang Z, Luo DN, Zhong X, Choi JH, Ma YQ, et al. SCINA: A semi-supervised subtyping algorithm of single cells and bulk samples. Genes 2019;10:531.
[52] Cao Y, Wang X, Peng G. SCSA: A cell type annotation tool for single-cell RNA-seq data. Front Genet 2020;11:490.
[53] Zhang AW, O'Flanagan C, Chavez EA, Lim JLP, Ceglia N, McPherson A, et al. Probabilistic cell-type assignment of single-cell RNA-seq for tumor microenvironment profiling. Nat Methods 2019;16:1007–15.
[54] Shao X, Liao J, Lu X, Xue R, Ai N, Fan X. scCATCH: automatic annotation on cell types of clusters from single-cell RNA sequencing data. Iscience 2020;23:100882.
[55] Hanbyeol K, Joongho L, Keunsoo K, Seokhyun Y. MarkerCount: A stable, count-based cell type identifier for single cell RNA-Seq experiments. Res Square 2021.
[56] Edgar R, Domrachev M, Lash AE. Gene Expression Omnibus: NCBI gene expression and hybridization array data repository. Nucleic Acids Res 2002;30:207–10.
[57] Athar A, Fullgrabe A, George N, Iqbal H, Huerta L, et al. ArrayExpress update - from bulk to single-cell expression data. Nucleic Acids Res 2019;47:D711–5.
[58] Wang Y, Song F, Zhu J, Zhang S, Yang Y, Chen T, et al. GSA: Genome Sequence Archive. Genomics Proteomics Bioinformatics 2017;15:14–8.
[59] Franzen O, Gan LM, Bjorkegren JLM. PanglaoDB: a web server for exploration of mouse and human single-cell RNA sequencing data. Database - J Biol Databases Curation 2019.
[60] Zhang XX, Lan YJ, Xu JY, Quan F, Zhao EJ, et al. Cell Marker: a manually curated resource of cell markers in human and mouse. Nucleic Acids Res 2019;47: D721–8.
[61] Yuan H, Yan M, Zhang G, Liu W, Deng C, et al. CancerSEA: a cancer single-cell state atlas. Nucleic Acids Res 2019;47:D900–8.
[62] Andrews TS, Hemberg M. M3Drop: dropout-based feature selection for scRNASeq. Bioinformatics 2019;35:2865–7.
[63] Cohen J. Weighted kappa: nominal scale agreement with provision for scaled disagreement or partial credit. Psychol Bull 1968;70:213–20.

[64] Abdelaal T, Michielsen L, Cats D, Hoogduin D, Mei H, Reinders MJT, et al. A comparison of automatic cell identification methods for single-cell RNA sequencing data. Genome Biol 2019;20:194.

[65] Levine J, Simonds E, Bendall S, Davis K, Amir E-ad D, Tadmor M, et al. Data-driven phenotypic dissection of AML reveals progenitor-like cells that correlate with prognosis. Cell 2015;162:184–97.

[66] Ding J, Shah S, Condon A. densityCut: an efficient and versatile topological approach for automatic clustering of biological data. Bioinformatics 2016;32:2567–76.

[67] Langfelder P, Zhang B, Horvath S. Defining clusters from a hierarchical cluster tree: the Dynamic Tree Cut package for R. Bioinformatics 2008;24:719–20.

[68] Zheng GXY, Terry JM, Belgrader P, Ryvkin P, Bent ZW, Wilson R, et al. Massively parallel digital transcriptional profiling of single cells. Nat Commun 2017;8:14049.

[69] Ding JR, Adiconis X, Simmons SK, Kowalczyk MS, Hession CC, et al. Systematic comparison of single-cell and single-nucleus RNA-sequencing methods. Nat Biotechnol 2020;38. 756-756.

[70] Kim N, Kim HK, Lee K, Hong Y, Cho JH, Choi JW, et al. Single-cell RNA sequencing demonstrates the molecular and cellular reprogramming of metastatic lung adenocarcinoma. Nat Commun 2020;11:2285.

[71] Regev A, Teichmann SA, Lander ES, Amit I, Benoist C, et al. The human cell atlas. Elife 2017;6.

[72] Saunders A, Macosko EZ, Wysoker A, Goldman M, Krienen FM, de Rivera H, et al. Molecular diversity and specializations among the cells of the adult mouse brain. Cell 2018;174:1015–1030.e16.

[73] Skinnider MA, Squair JW, Foster LJ. Evaluating measures of association for single-cell transcriptomics. Nat Methods 2019;16:381–6.

[74] Zilionis R, Engblom C, Pfirschke C, Savova V, Zemmour D, Saatcioglu HD, et al. Single-cell transcriptomics of human and mouse lung cancers reveals conserved myeloid populations across individuals and species. Immunity 2019;50:1317–34.

[75] Xie T, Wang Y, Deng N, Huang G, Taghavifar F, Geng Y, et al. Single-cell deconvolution of fibroblast heterogeneity in mouse pulmonary fibrosis. Cell Rep 2018;22:3625–40.

[76] Zhong S, Zhang S, Fan X, Wu Q, Yan L, Dong Ji, et al. A single-cell RNA-seq survey of the developmental landscape of the human prefrontal cortex. Nature 2018;555:524–8.

[77] Zhang AW, McPherson A, Milne K, Kroeger DR, Hamilton PT, Miranda A, et al. Interfaces of malignant and immunologic clonal dynamics in ovarian cancer. Cell 2018;173:1755–1769.e22.

[78] Andor N, Simonds EF, Czerwinski DK, Chen J, Grimes SM, et al. Single-cell RNA-Seq of follicular lymphoma reveals malignant B-cell types and coexpression of T-cell immune checkpoints. Blood 2019;133:1119–29.

[79] Medaglia C, Giladi A, Stoler-Barak L, De Giovanni M, Salame TM, Biram A, et al. Spatial reconstruction of immune niches by combining photoactivatable reporters and scRNA-seq. Science 2017;358:1622–6.

[80] Cusanovich DA, Hill AJ, Aghamirzaie D, Daza RM, Pliner HA, et al. A single-cell atlas of in vivo mammalian chromatin accessibility. Cell 2018;174:1309–24.

[81] Mereu E, Lafzi A, Moutinho C, Ziegenhain C, McCarthy DJ, Álvarez-Varela A, et al. Benchmarking single-cell RNA-sequencing protocols for cell atlas projects. Nat Biotechnol 2020;38:747–55.

[82] Haghverdi L, Lun ATL, Morgan MD, Marioni JC. Batch effects in single-cell RNA-sequencing data are corrected by matching mutual nearest neighbors. Nat Biotechnol 2018;36:421–7.

[83] Welch JD, Kozareva V, Ferreira A, Vanderburg C, Martin C, Macosko EZ. Single-cell multi-omic integration compares and contrasts features of brain cell identity. Cell 2019;177:1873–1887.e17.

[84] Hie B, Bryson B, Berger B. Efficient integration of heterogeneous single-cell transcriptomes using Scanorama. Nat Biotechnol 2019;37:685–91.

[85] Lin Y, Cao Y, Kim HJ, Salim A, Speed TP, et al. scClassify: sample size estimation and multiscale classification of cells using single and multiple reference. Mol Syst Biol 2020;16:e9389.