scientific reports

Check for updates

OPEN

CASi: A framework for cross-timepoint analysis of single-cell RNA sequencing data

Yizhuo Wang[1], Christopher R. Flowers[2], Michael Wang[2], Xuelin Huang[1]✉ & Ziyi Li[1]✉

Single-cell RNA sequencing (scRNA-seq) technology has been widely used to study the differences in gene expression at the single cell level, providing insights into the research of cell development, differentiation, and functional heterogeneity. Various pipelines and workflows of scRNA-seq analysis have been developed but few considered multi-timepoint data specifically. In this study, we develop CASi, a comprehensive framework for analyzing multiple timepoints' scRNA-seq data, which provides users with: (1) cross-timepoint cell annotation, (2) detection of potentially novel cell types emerged over time, (3) visualization of cell population evolution, and (4) identification of temporal differentially expressed genes (tDEGs). Through comprehensive simulation studies and applications to a real multi-timepoint single cell dataset, we demonstrate the robust and favorable performance of the proposal versus existing methods serving similar purposes.

In recent years, the emergence of the single-cell RNA sequencing (scRNA-seq) technique has enabled researchers to study cellular compositions and transcriptomic profiles with unprecedented precision. After finishing the preprocessing and quality control steps, the downstream analysis of scRNA-seq data includes dimension reduction, unsupervised clustering, differential expression analysis and so on. The popular pipelines for these analyses include scran[1], Seurat[2], and SINCERA[3]. Recently, a number of supervised cell type annotation algorithms which can potentially identify novel cells have been developed, including scmap[4], CHETAH[5], and singleR and CAMLU[6]. In addition to recognizing the cell types, detecting differentially expressed genes (DEGs) is another common task in scRNA-seq analysis and lots of methods have been developed for this task. Simple statistical tests (i.e., the t-test and the Wilcoxon Rank Sum test) are used in pipelines such as Seurat. But more complex model-based frameworks have been shown to achieve better performance, e.g., DESeq2[7] and MAST[8].

Although many existing methods have been developed, almost all of them focus on analyzing data collected from cross-sectional studies, i.e., the scRNA-seq experiments are performed on samples collected at a single time point. In the setting of studying disease development and progression, following patients for a period of time and collecting data from continuing experiments is actually a natural choice. Recent years have witnessed a growing number of multi-timepoint studies. For example, Ravindra et al. studied the SARS-CoV-2 infection by performing scRNA sequencing experiments on infected human bronchial epithelial cells at four time points to track patients' immune responses to the virus[9]. In another study, Zhang et al. followed patients with ovarian cancer for five years and collected tissue samples before and after chemotherapy to study their stress-promoted chemoresistance[10]. Multi-timepoint scRNA-seq experiments provide a powerful tool for studying the dynamics of gene expression in individual cells and how it changes in response to various stimuli or disease conditions. At present, there are no available computational tools to comprehensively analyze multiple timepoints' scRNA-seq data. Most of the current transcriptomic single-cell studies with the design of different timepoints still use methods and workflows, such as Seurat[2], which do not specifically take the time changes into consideration. Ramazzotti et al. proposed a framework called LACE that processes Single-cell mutational profiles, which are generated by calling variants from scRNA-seq data collected at different time points[11]. However, LACE specifically focuses on building somatic mutational profiles and reconstructing longitudinal clonal tree in tumor cells, and is not a general framework designed for multi-timepoint scRNA-seq data.

There are several reasons why cross-timepoint data can be more challenging to analyze compared with cross-sectional data. First, samples from the same subject are likely to be correlated. The information from previous time points are generally helpful for analyzing the data from later time points, and this should be considered for information borrowing. Such correlation also complicates differential signal detection. Second, new cell types

[1]Department of Biostatistics, The University of Texas MD Anderson Cancer Center, Houston 77030, USA. [2]Department of Lymphoma/Myeloma, The University of Texas MD Anderson Cancer Center, Houston 77030, USA. ✉email: xlhuang@mdanderson.org; zli16@mdanderson.org

may emerge over the experimental course. Ignoring the potential for new cell types can lead to inaccurate cell annotation and misleading results. Such newly emerged cells may also be the key to disease progression and treatment outcome, and thus they should be highlighted in the analysis procedures. Third, the single cell data are usually collected gradually over a long period of time. It is preferable for the analysis pipeline to take such collection schedules into consideration, when able, to allow for stepwise analysis and the incorporation of future data without changing the current results dramatically.

To address these challenges, we develop a comprehensive framework specifically to analyze scRNA-seq data from multi-timepoint experiments. Similar to other pipelines, the first step of our method is to annotate cell labels. But our method allows for information borrowing from earlier time points, as well as step-wise analysis. Our framework implements a neural network classifier to assign cell labels in a supervised way, and it can achieve high accuracy in the cross-timepoint setting. Nevertheless, one challenge faced by all the supervised annotation methods is identification of the novel cell type, which is defined as a cell type that is not present in the initial or earlier time points and that only exists in the newer collections. Along with the supervised annotation, we propose and implement a novel methodology pipeline in the framework to identify new cell types that have emerged over time. Once the annotation is done, we provide visualizations to illustrate the cell population evolution. Additionally, we add one key downstream analysis, temporal differential expressed gene (tDEG) detection, to our framework. The aim is to identify genes with wildly increasing/decreasing behavior over time and with different changing patterns from group to group. For example, our framework is able to select the genes whose expression increases over time for responders but decreases over time for non-responders. The whole framework is named the Cross-timepoint Analysis of Single-cell RNA sequencing data (CASi) method.

Methods and materials

An overview of CASi

An overview of the whole CASi pipeline are presented in Fig. 1. CASi takes scRNA-seq data collected from different timepoints as the input. To simplify the discussion, we consider an scRNA-seq dataset collected at three time points: t_0, t_1, and t_2. Assuming t_0 data is pre-labeled, i.e., t_0 data is provided with cell types, the labels can be obtained using existing unsupervised or supervised approaches for cross-sectional data. The first step of CASi is to perform a supervised annotation for unlabeled data t_1 and t_2 using an artificial neural network classifier. However, if the labels for the t_0 data are not provided, CASi instead will perform an unsupervised clustering and use the clustering number as the cell type labels for t_0 data. Then the classifier can be trained with t_0 data and applied to annotate t_1 and t_2 data. Here we prefer an artificial neural network over other machine learning methods because a few recent works[12,13] showed its superior accuracy and favorable computational performance

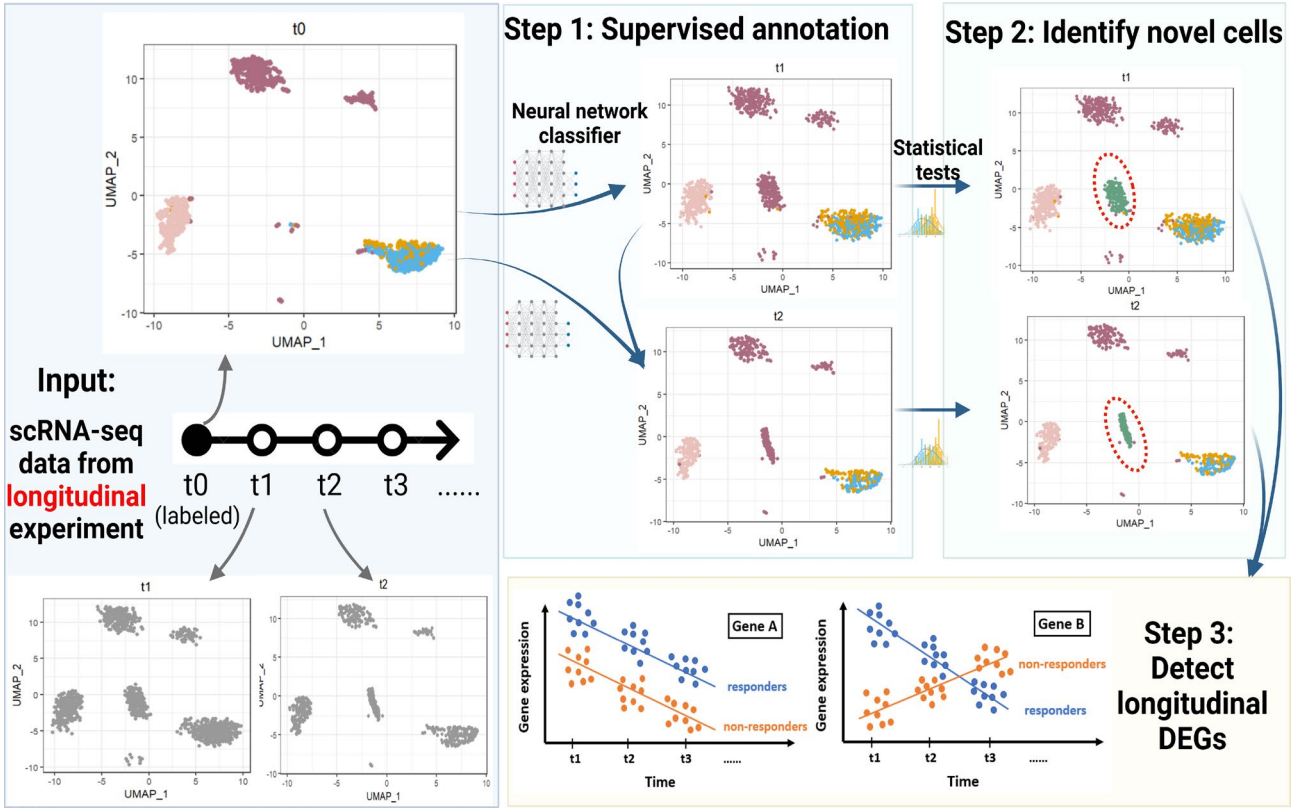

Figure 1. An overview of the CASi framework. The input is scRNA-seq data from different timepoints' experiments. CASi mainly consists of three steps: (**1**) cross-time points cell annotation, (**2**) detection of potential novel cell types, (**3**) identification of temporal differentially expressed genes.

in the task of cell type annotation. Additionally, besides analyzing the transition from t_0 to t_1 and t_0 to t_2, CASi allows to analyze the transition from t_1 to t_2 as well. Timepoint of interest is subjective to users' selection in the framework. If they are interested in the transition from t_1 to t_2, they could input the t_0 data and unlabeled t_1 data first, then combine the original t_0 data and the labeled t_1 data which will the output of CASi, and use this combined t_0/t_1 cells and unlabeled t_2 cells as the new inputs to re-do the analysis. In this way, users will be able to look into the transition from t_1 to t_2.

The second step of CASi is to identify any novel cell type that have emerged over time. When data of interest is collected from tumor samples, it is highly possible that the tumor cells will differentiate and new cell types (e.g., a distinct subclone of tumor cells) might appear. Assuming the cells of t_1 and t_2 data are a mixture of known and unknown cell types, we have designed a computational pipeline to distinguish these new cell types from existing cell types. The pipeline starts with a feature selection procedure to select a smaller set of informative features, followed with a dimension reduction procedure to better extract information from these features. Many previous works use large correlation as a criterion to identify the same cell type, such as scmap[4], CHETAH[5], NeuCA[14], SingleR[15], etc. Here we use the two-sample t-test to compare the correlations between features in known cell types and features in unknown cell types, and thus we are able to identify novel cells. Intuitively, after using the neural network to assign cell types for t_1 and t_2 data, if the cells actually belong to a new, unknown cell type, the levels of similarity with the known cell types will be lower than all of the other cells. Additionally, it should be noted that, CASi allows for the detection of multiple novel cell types. We achieve this by providing users with both the cluster-labeled UMAP (Uniform Manifold Approximation and Projection) plot and the correlation-labeled UMAP plot, which visually, directly reveals the possibility of multiple novel cell types when the UMAP plots indicate different levels of correlation.

The final step of CASi is to perform differential analysis tailored to multi-timepoint scRNA-seq data. We combine a generalized linear model with iterative feature selection to select genes that have apparent increasing/decreasing behavior over time and genes that behave differently along time in different groups.

Artificial neural networks (ANNs)

Denote the scRNA-seq expression matrix of the training data by $\boldsymbol{X}_0$ where $\boldsymbol{X}_0$ is a p by n_0 matrix with p being the total number of measured genes and n_0 is the number of cells, and the corresponding training cell label, $\boldsymbol{Y}_0$, is a n_0 by 1 vector. Similarly, denote the expression matrix of the testing data by $\boldsymbol{X}_1$, which has dimensions p by n_1 and the labels by $\boldsymbol{Y}_1$. After the standard min-max normalization, we select the top 2000 most variable genes from X_0 and the same set of features from $\boldsymbol{X}_1$. Next, using Keras[16], an open-source software library that provides a Python interface for artificial neural networks, we train a neural network model with one input layer, one output layer, and three hidden layers:

$$
\begin{aligned}
\Pr\left(\boldsymbol{y_0} \mid \boldsymbol{X}, \boldsymbol{\theta}\right) &= \eta\left(\boldsymbol{Z}_{\text{out}} \boldsymbol{W}_{out} + \boldsymbol{\beta}_{out}\right) \\
\boldsymbol{Z}_{\text{out}} &= \sigma\left(\boldsymbol{Z}_3 \boldsymbol{W}_3 + \boldsymbol{\beta}_3\right) \\
\boldsymbol{Z}_3 &= \sigma\left(\boldsymbol{Z}_2 \boldsymbol{W}_2 + \boldsymbol{\beta}_2\right) \\
\boldsymbol{Z}_2 &= \sigma\left(\boldsymbol{Z}_1 \boldsymbol{W}_1 + \boldsymbol{\beta}_1\right) \\
\boldsymbol{Z}_1 &= \sigma\left(\boldsymbol{X}_0 \boldsymbol{W}_0 + \boldsymbol{\beta}_0\right)
\end{aligned}
$$

The parameter set $\theta = \{\boldsymbol{W}_0, \boldsymbol{W}_1, \cdots, \boldsymbol{W}_3, \boldsymbol{\beta}_0, \boldsymbol{\beta}_1, \cdots, \boldsymbol{\beta}_3, \boldsymbol{\beta}_{\text{out}}, \boldsymbol{W}_{\text{out}}\}$ will be estimated during the training process. And z_l for $l = \{1, 2, 3\}$ are the hidden neurons with corresponding weight $\boldsymbol{W}_1$, and bias $\boldsymbol{\beta}_1$.$\sigma(\cdot)$ is the activation function, which can be a sigmoid, a rectified linear unit (ReLU), a hyperbolic tangent, etc. We choose to use the ReLU function in our hidden layers because the neural networks based on an ReLU function are generally easier to train and can avoid the vanishing gradient problem during optimization[17]; it is mathematically expressed as $\sigma_{\text{ReLU}}(x) = \max(x, 0)$. The SoftMax function will be used in the output layer. This is because the number of output categories is more than two, and it converts the values of the output layer into the predicted probabilities of each label. The number of neurons in the three hidden layers is selected as $\{256, 128, 64\}$. The model is trained using a stochastic gradient descent (SGD)-based algorithm with the mean squared error loss function $\mathscr{L}\left(\Pr(\boldsymbol{y_0}), Y_0\right) = \left\|\Pr(\boldsymbol{y_0}) - Y_0\right\|^2$. We use Adam as the optimization algorithm[18], and the mini-batch training strategy[19], which randomly trains a small proportion of samples and validates the rest of the samples in each iteration to improve training efficiency. By monitoring the loss, we implement the early stopping rule in Keras to avoid overfitting. Once the model performance stops improving for a couple epochs, the training process will stop. Additionally, to further prevent the overfitting issue, we add a dropout step with the dropping rate of 0.4 for each hidden layer to randomly drop units from the neural network during training.

Identify novel cell types

Let $Y^k, k = \{1, \ldots, K-1, K\}$ be the cell labels, and Y^K is the novel cell type that only appears in the testing data X_1. Then when annotating the cells in X_1, all Y^K cells will be wrongly labeled as $Y^k, k = \{1, \ldots, K-1\}$ by the neural network classifier. To address this issue, we design a pipeline. For each cell type, we implement an iterative clustering step using the Louvain algorithm[20]. We iterate over a series of resolution parameters and the algorithm stops only when the cell population is clustered into two clusters. Then we apply t-test on these two clusters using cell-type-specific correlations. If passing the t-test, the cluster with smaller correlation values will be considered 'suspicious' (novel) cells. The key idea of this pipeline is that the similarity (correlation) of the new/unknown cell type with the known cell types will be different (usually smaller), which can be captured by the two-sample t-test.

The final cell identities of the identified new cell types will be confirmed with external biological knowledge and expert opinions.} In detail, the pipeline consists of the following steps:

1. Reduce dimension:
   $X^{Y^k}_{0[2000\times n_0]} \xrightarrow{UMAP} U^{Y^k}_{[20\times n_0]},\ X^{Y^k}_{1[2000\times n_1]} \xrightarrow{UMAP} V^{Y^k}_{[20\times n_1]},\ k = \{1,\ldots,K-1\}$
2. Obtain the mean of UMAP vectors for each cell type:
   $m^{Y^k}_{[20\times 1]} = \frac{\sum U^{Y^k}_{[20\times n_0]}}{n_0},\ k = \{1,\ldots,K-1\}$
3. Obtain the Pearson correlation between cells in X_1 and the mean:
   $r^{Y^k}_{[1\times n_1]} = \mathrm{corr}\left(V^{Y^k}_{[20\times n_1]}, m^{Y^k}_{[20\times 1]}\right),\ k = \{1,\ldots,K-1\}$
4. Re-cluster each cell type in X_1 into two groups using the Louvain algorithm[20]:
   $X^{Y^k}_{1[2000\times n_1]} \xrightarrow{\text{Louvain}} X_1^{Y^k,\text{group 1}}, X_1^{Y^k,\text{group 2}},\ k = \{1,\ldots,K\}$
5. For each cell type Y^k apply two-sample t test to the two groups of cells using their cell type-specific correlation values. If significant, we will designate the group of cells that has a smaller mean correlation as the potential new cell type.
6. For each cell type Y^k, the t-test will assign a small group of cells as the new cell type. And in the end, we combine those significant groups of cells and annotate them as the new cell type Y_K.

Identify temporal differentially expressed genes (tDEGs)

For each gene in the dataset of interest, we build a negative binomial GLM model using a log link function. The negative binomial model is a generalization of the Poisson model such that the count Y_i still adopts a Poisson, but the expected count μ_i^* is a gamma-distributed random variable with mean μ_i and constant scale parameter ω, i.e., the mean and variance are not equal anymore. Mathematically, the count Y_i follows a negative-binomial distribution[21]:

$$p(y_i) = \frac{\Gamma(y_i+\omega)}{y!\Gamma(\omega)} \times \frac{\mu_i^{y_i}\omega^{\omega}}{(\mu_i+\omega)^{\mu_i+\omega}}$$

where the expected value is $E(Y_i) = \mu_i$ and the variance is $V(Y_i) = \mu_i + \mu_i^2/\omega$.

The covariates that we put into the GLM model are the time variable, the strata of interest (e.g., the response status), and the interaction term of time and strata as independent variables, and the gene expression count is Y_i. Note that for the existing methods, including DESeq2[7], MAST[8], and the Wilcoxon Rank Sum test offered by the FindMarker() function in Seurat, only one binary covariate (i.e., the strata of interest) can be used in the analysis. This means that multiple covariates and the interaction term are not under consideration when finding tDEGs. And when we fit the negative binomial GLM model, both the regression coefficients and ω will be estimated by the method of maximum likelihood. For each model, the p-value of each term is obtained from testing the null hypothesis of its coefficient being equal to zero. Then for all genes, we extract the interaction terms' p-values and use the Bonferroni correction to account for multiple-test effect.

Evaluation metrics

Assuming three time points: t_0, t_1, t_2, to evaluate the cell annotation step, we use two metrics, accuracy and adjusted rand index (ARI), when comparing with other methods for the cell annotation step. The accuracy is defined as (number of correctly labeled cells at time t_1 and t_2) / (total cell number of t_1 and t_2 data). And ARI is a widely used metric to evaluate clustering performance. Rand Index looks at similarities between any two clustering methods and ARI is the corrected-for-chance version of the Rand index[22]. In this study, the existing, supervised clustering methods to compare with include scmap[4], CHETAH[5], and scPred[23].

For the detection of tDEGs step, we use true discovery rate (TDR) to assess our method's performance. TDR is a measure of accuracy when multiple hypotheses are being tested at once and mathematically equals to (1-false discovery rate). The existing, differential expression analysis methods to compare with include DESeq2[7], MAST[8], and the Wilcoxon Rank Sum test offered by the FindMarker() function in Seurat[2].

Results

Simulation study

The simulation data are generated by assuming three sampling time points: t_0, t_1, t_2. To fully evaluate our method, we designed three scenarios: 1) t_0, t_1, and t_2 data contain the same cell types but with different cell type compositions; 2) a cell type in t_0 disappears in t_1 and t_2 data, i.e., t_0 data have one more cell type than t_1 and t_2 data; 3) a new cell type appears in t_1 and t_2 data, i.e., t_1 and t_2 data have one more cell type than t_0 data. We obtain a publicly available dataset of peripheral blood mononuclear cells (PBMC)[24], containing more than 60,000 sorted cells from eight immune cell types. We randomly extract cells from five cell types and use different cell type compositions for different scenarios. Detailed settings of these three scenarios can be found in the supplementary file.

CASi facilitates cross time points cell annotation with high accuracy

CASi is suitable when data of interest are collected from multiple timepoints as it borrows information across time points and allows for accurate cell annotation when data becomes gradually available. Using the neural network classifier, CASi achieves a high accuracy when mapping the cell labels of the initial time point onto later

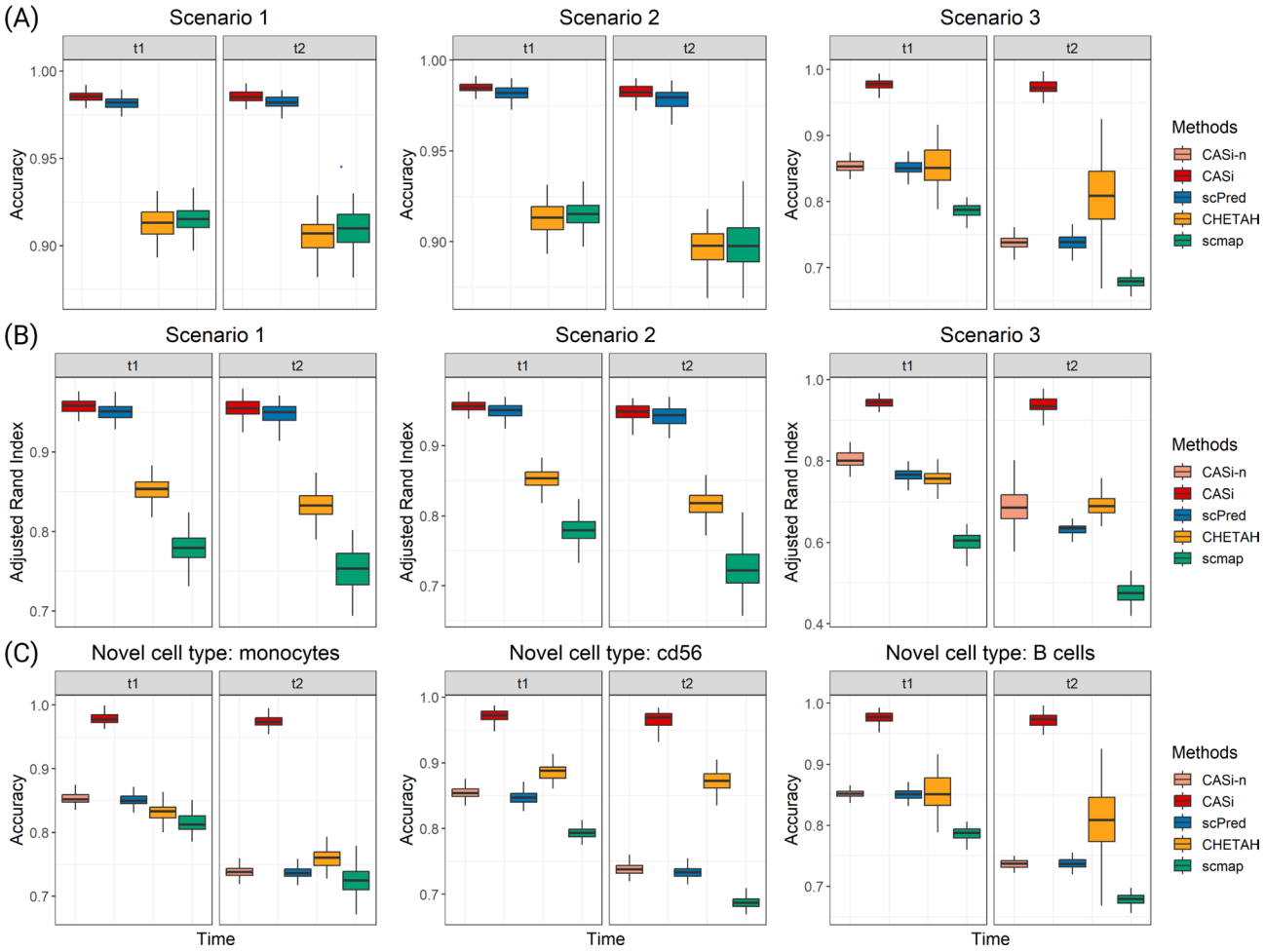

Figure 2. Results of PBMC simulation study based on 200 repetitions. CASi-naive (CASi-n) refers to only the neural network classifier without our follow-up pipeline. (**A**) accuracy of the cross-time points cell annotation. (**B**) ARI of the cross-time points cell annotation. (**C**) accuracy of scenario 3 using different cell types as the new cell types emerged in t_1 and t_2.

time points. The accuracy and ARI of each scenario based on 200 Monte Carlo experiments are shown in Fig. 2. It can be observed that, for scenario 1 and 2, our method has the highest accuracy and the highest ARI, but the advantage is not significant when compared to the scPred method. For scenario 3, we report the performance of CASi-n, which refers to only the neural network classifier, and the performance of CASi, which refers to the neural network classifier in alignment with the identify-novel-cell pipeline. As we expected, the disappearance of one cell type (scenario 2) does not affect the classifier's performance, while the appearance of one novel cell type (scenario 3) causes trouble to the classifier. After our pipeline identifies and labels the novel cells, both the accuracy and the ARI improve significantly, and CASi demonstrates state-of-art performance when novel cell types appear. For scenario 3, we vary the setting of scenario 3 and use different cell types as the novel cell type. In Panel C, we report the accuracy of these settings and observe a similar pattern that CASi outperforms other existing methods.

CASi addresses possibility of novel cell types appeared in later time points

CASi uses the levels of similarity between the known cell types and new cells to identify potential novel cell types that may have appeared at later time points. The full pipeline of identifying novel cells is described in the Method section. We provide users with the UMAP plots displaying cell type clusters and the UMAP plots displaying the correlation. Fig. 3 shows the UMAP plots of t_1 data. In the left figure, t, the purple group represents novel cells that appeared in t_1. And we can see that, in the right figure, the correlation between this new cell type and the existed cell type is very low. The UMAP clustering of t_1 data cell labels, the UMAP clustering of t_2 data cell labels, the UMAP clustering of t_2 data highlighting the novel cell type, and the UMAP clustering of t_2 data showing correlation levels can be found in the supplementary file (Figure 2~5). Here for scenario 3 in our simulation, we only assign one new cell type. When > *one* new cell type appears in later time points, users will be able to recognize them separately as new cells will be clustered into multiple groups on the UMAP plots.

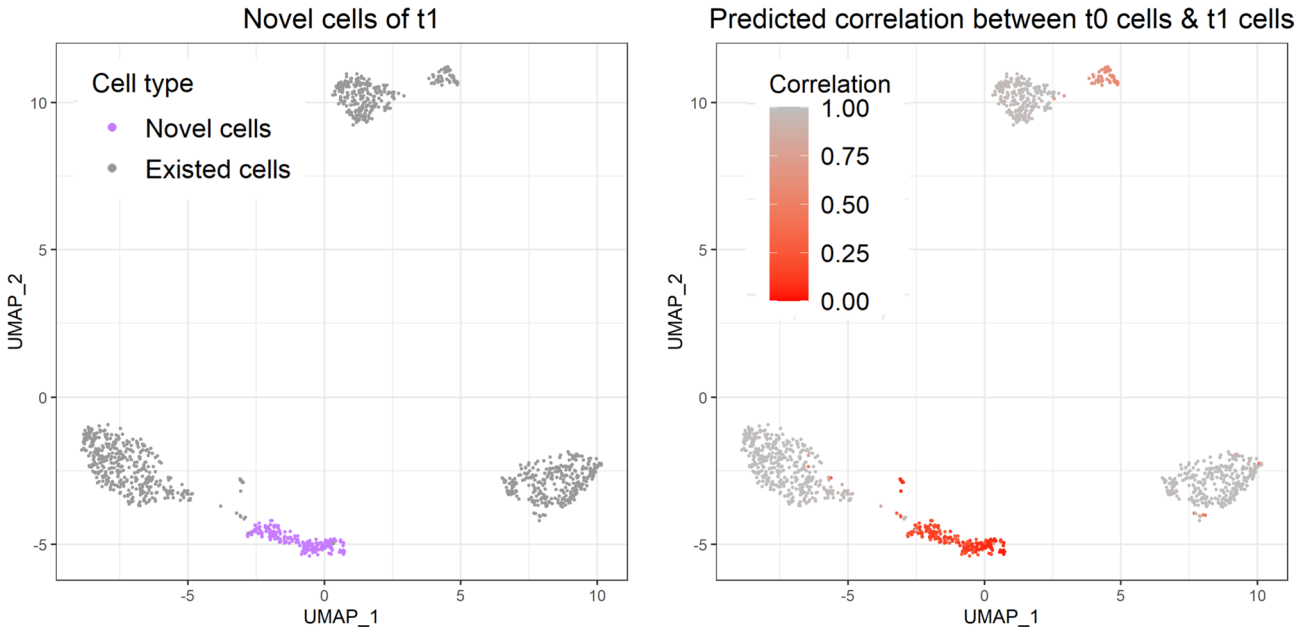

Figure 3. UMAP plots of the PBMC simulation study: intuition behind identifying novel cell types. The left figure is the clustering that highlights the novel cell type group, and the right figure is the clustering of correlation levels between existing cell types in t_0 data and new cell types in t_1 data.

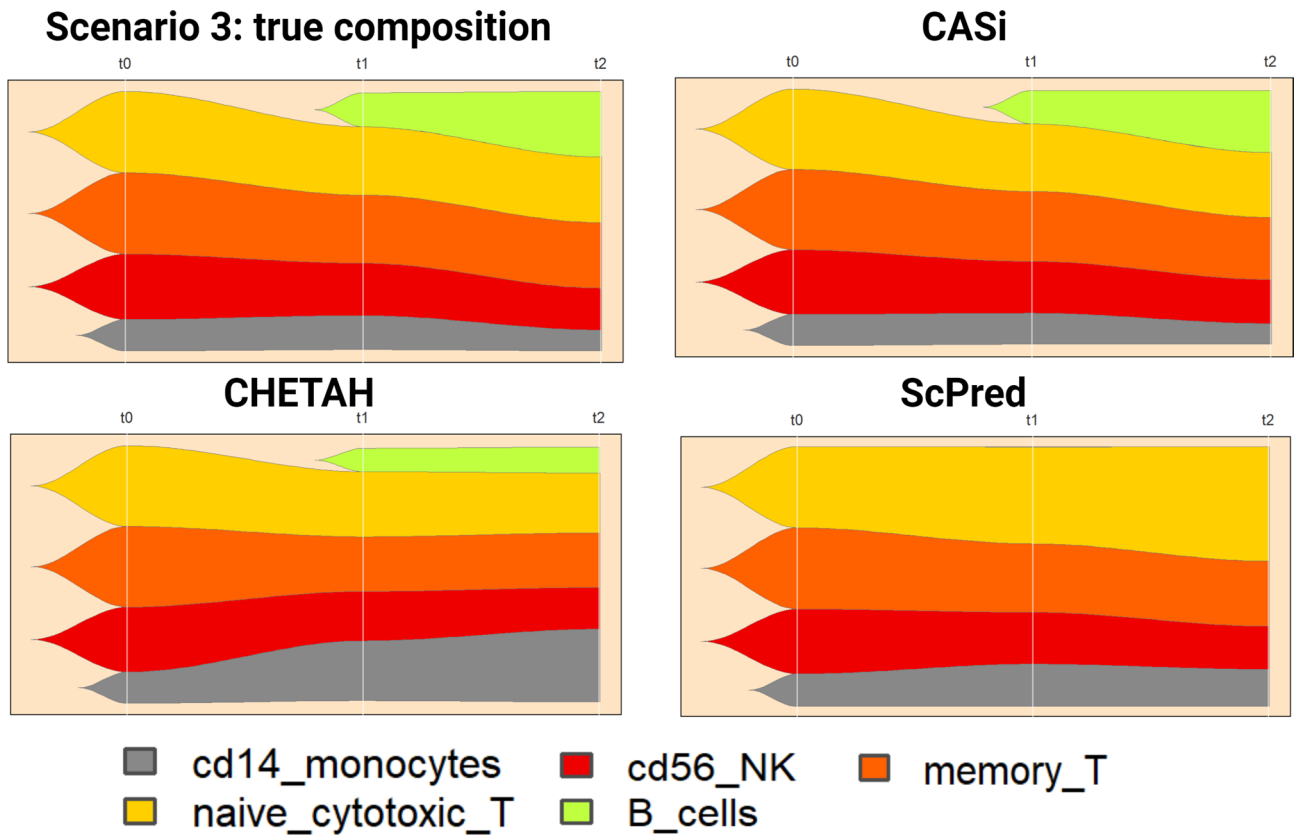

Figure 4. Fish plots of scenario 3 in which a novel cell type appears. We use B cells as the novel cell type which shows in green. The first figure is the ground truth of cell population and CASi is able to annotate the cells with high accuracy compared with other methods.

CASi provides visualization of cell type evolution over time

To track the changes of the complex cell population, an appropriate form of visualization is of importance. We use a combination of UMAP and fish plots to illustrate the dynamic changes in the cell type proportions over time. Traditional UMAP/scatter plots visually show how separable the clusters/cell types are using selected features. The fish plot developed by Miller et al. is a relatively new tool displaying changes in populations of cells over time[25], which is a great tool to visualize the temporal cellular proportion evolution. In fish plots, each color represents one cell type; the first plot in Fig. 4 shows the true cell type population of Scenario 3, where a novel cell type appears in t_1 and t_2. We can see that the CASi prediction is highly similar to the ground truth and able to identify most novel cells, while other methods either only identify a small portion of novel cells (CHETAH) or fail to capture any novel cells (scPred and scmap, the scmap result can be found in the supplementary file, Figure 6).

CASi identifies tDEGs with high TDR

A natural interest in multi-timepoint scRNA-seq data analysis is to detect genes that change vividly over time, and different groups might have different changing directions. We refer to this kind of gene as tDEGs. We randomly select 2000 genes and 900 monocyte cells from the PBMC dataset. Among 2000 genes, 300 genes are randomly selected to be tDEGs. We design three levels of time effect: weak, medium, and strong. Detailed procedures of how we assign tDEGs can be found in the supplementary file. Figure 5 shows that for all three time effect levels, CASi has the highest TDR at different thresholds, which means that CASi is able to capture tDEGs at the greatest extent. When the time effect increases, the TDR of CASi increases as well, but the TDR of other methods decreases.

A real-world multi-timepoint dataset

We apply CASi to a real-world mantle cell lymphoma (MCL) dataset[26] where all patients received ibrutinib, the current standard of care treatment for MCL, but responded differently to the treatment. Three patients are ibrutinib-responsive (patients V, C, and D) and two patients are non-responsive (patients B and E). The MCL dataset requires cross-timepoint analysis: it includes measurements of 21 specimens collected at baseline, during treatment, and/or at disease remission/progression. Since the number of measurements and the timing of measurements vary from patient to patient, we manually binarize the time variable into two groups: pre-treatment and post-treatment, which also aligns with the analysis in the original paper of the MCL data.

Cross-timepoint annotation

We again use accuracy and ARI to compare CASi with existing methods. From Fig. 6, in Panel A, we can see that when mapping the cell labels of pre-treatment data to the cell labels of post-treatment data, CASi achieves the highest accuracy and the highest ARI. The true cell composition of post-treatment and the predicted cell composition by CASi are displayed in the supplementary file, Figures 7 and 8, and we can see that the two of which are visually identical. In Fig. 6 Panel B, we show UMAP plots of cell types (top) and correlation (bottom). The tumor cells are separated into two clusters and the correlation between tumor cells in post-treatment data and tumor cells in pre-treatment data are not the same for these two clusters. The cluster with a correlation of 0.5–1 is more similar to the tumor cells in pre-treatment data, while the cluster with a correlation of 0–0.25 is very different from the tumor cells in pre-treatment data. This is a very interesting observation since tumor cells appear to be more genetically unstable than normal cells and may evolve over the treatment course. Over time, tumor cells divide more rapidly and become less dependent on signals from other cells. This is probably why we observe two clusters for post-treatment tumor cells with two levels of correlation.

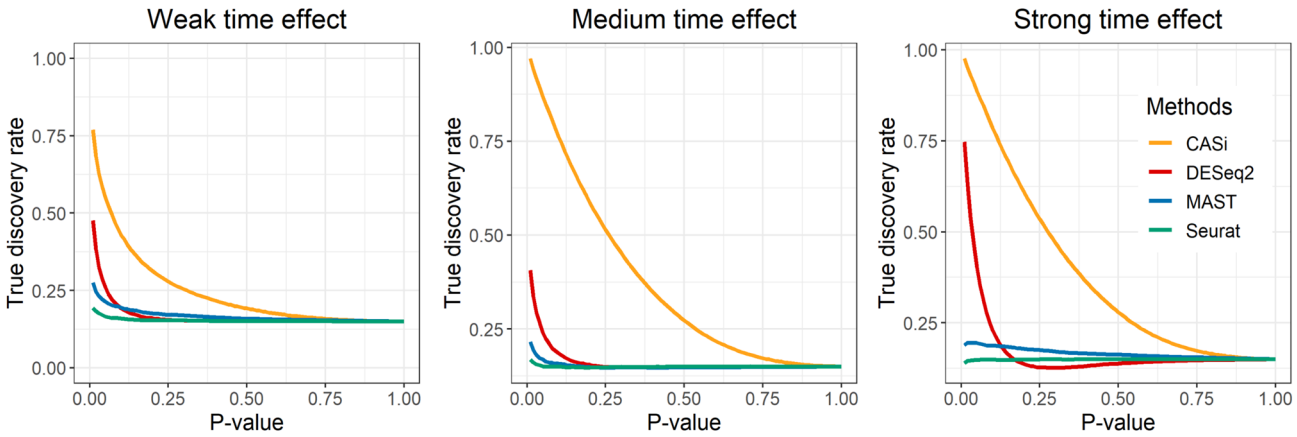

Figure 5. True discovery rate of identifying temporal differentially expressed genes. The results are averaged based on 200 repetitions. Three simulation settings are being considered here: weak time effect (left), medium time effect (middle), strong time effect (right).

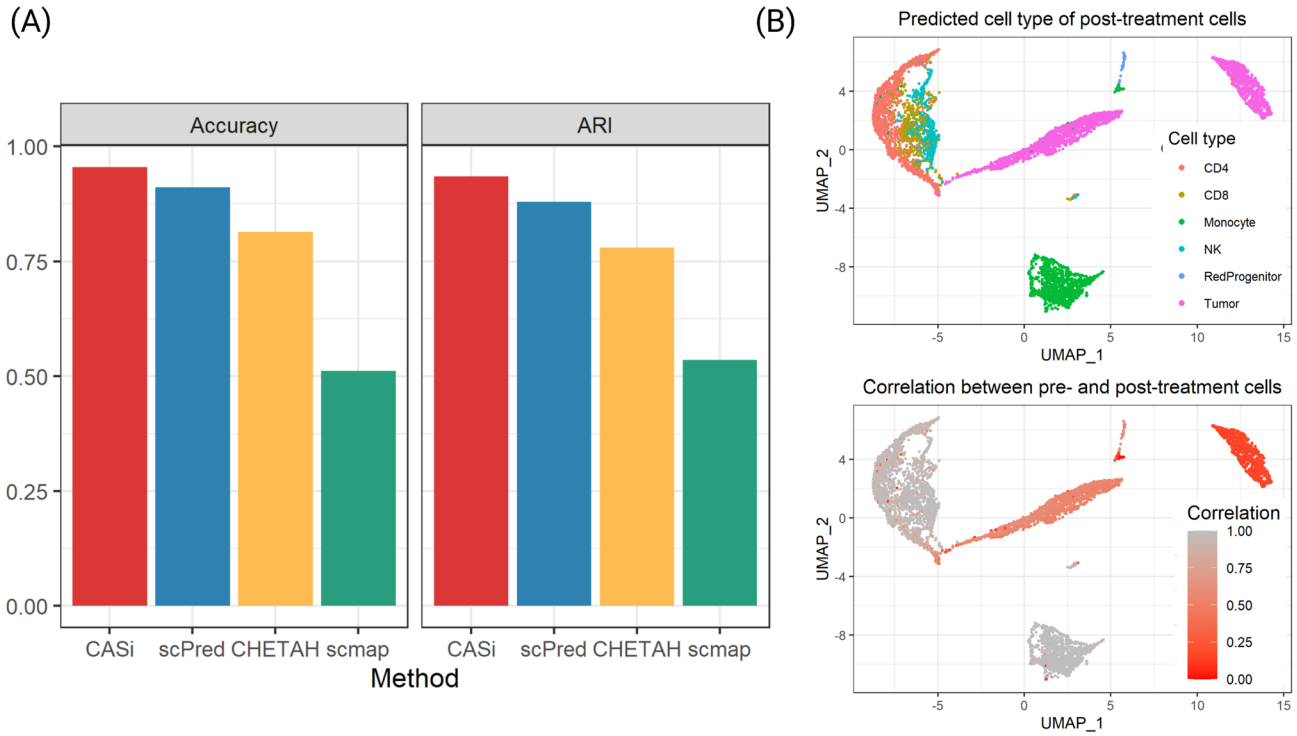

Figure 6. Annotation results of real-world data application. (**A**) shows the accuracy (left) and adjusted rand index (right) of the supervised clustering step. (**B**) shows the clustering of cell labels (left) and the clustering of correlations between pre-treatment cells and post-treatment cells (right) in which red represents a low correlation level.

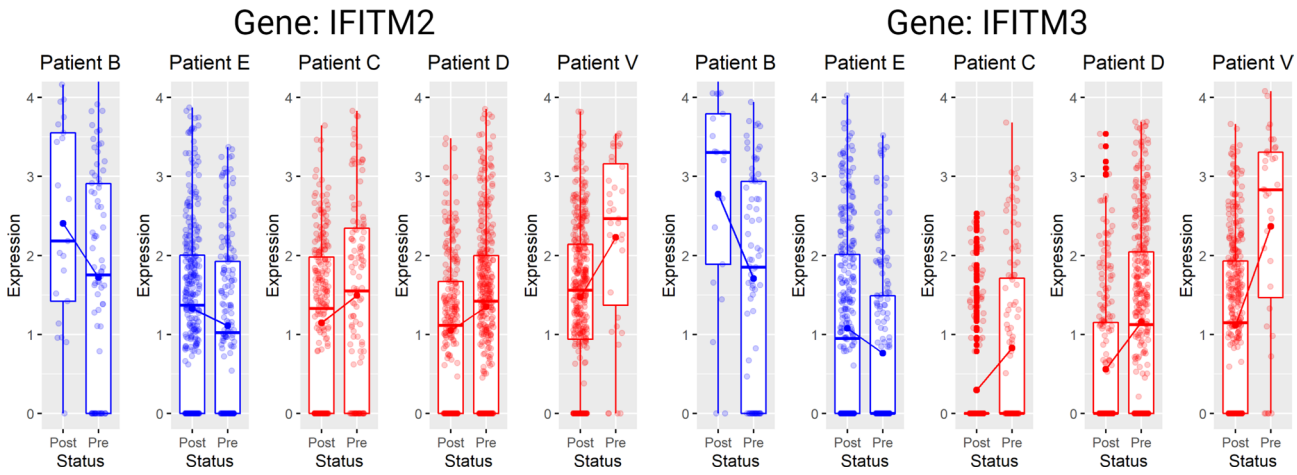

Figure 7. Differential expression analysis results of real-world data application. The changing gene expression profiles of IFITM2 (left) and IFITM3 (right), which are temporal differentially expressed genes identified by CASi but not Seurat. Non-responders are shown in blue; responders are shown in red. Totally opposite patterns can be seen for responders and non-responders.

Identification of tDEGs

tDEGs refer to the genes that change vividly over time, and the changing patterns are distinct for different groups. We apply CASi and Seurat at the same time to this dataset to find tDEGs. Using $p = 0.05$ as a cutoff, among a total of 1469 genes, CASi identifies 268 tDEGs and Seurat identifies 75 DEGs. There are 21 overlapped genes for tDEGs found by CASi and DEGs found by Seurat. CASi is able to identify a few tDEGs that Seurat failed to capture their significance. In Fig. 7 Panel A, we illustrate the two top tDEGS identified by CASi but not Seurat, IFITM2 (the left plot) and IFITM3 (the right plot), and we draw their gene expression patterns for all patients. Patients B and E are non-responders and are drawn in blue; patients C, D, and V are responders and are drawn in red. Clearly, for responders, the gene expression of the identified genes increases after the ibrutinib treatment, while for non-responders, the gene expression decreases after the ibrutinib treatment. This finding is supported by a

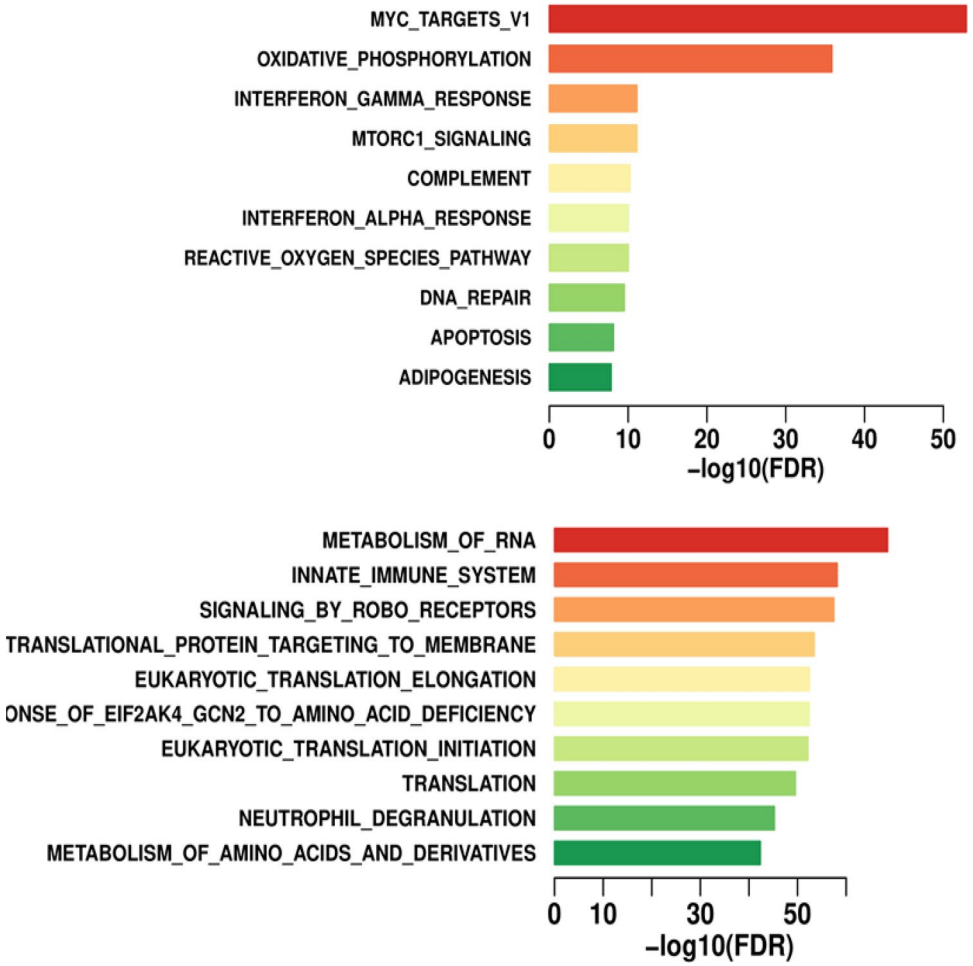

Figure 8. Enrichment analysis results of real-world data application using 500 temporal differentially expressed genes identified by CASi: the hallmark pathway results (top), the reactome pathway results (bottom).

recent study in which Lee et al. discovered that deletion of IFITM3 in MCL cells reduced competitive fitness and proliferation, and the upregulation of IFITM2 might suggest a compensation mechanism for the loss of IFITM3[27].

Additionally, using the top significant genes selected by CASi, we perform enrichment analysis, which identifies biological pathways that are enriched in this gene list more than just by chance. The results of the Hallmark pathway (top) and the Reactome pathway (bottom) are displayed in Fig. 8. For the top two hallmark pathways, MYC and oxidative phosphorylation, multiple studies have reported that MYC is frequently expressed in MCL, and targeting MYC provide a novel therapeutic strategy for MCL patients[28,29]; multiple studies have found that the MCL cancer cells can be effectively targeted with a small-molecule inhibitor of oxidative phosphorylation as a therapeutic strategy[30,31]. For reactome pathways, several researchers have findings that match our own; for example, RNA metabolism pathway is significantly enriched in MCL[32] and the ibrutinib resistance of MCL relates to the receptors initiating the innate immune system[33].

Discussion

Although a series of clustering and annotating methods have been developed for scRNA-seq data, these methods mostly focus on cross-sectional studies. A systematic analysis tool designed specifically for cross-timepoint scRNA-seq datasets is lacking. With a rapidly growing number of studies conducting experiments at different timepoints, there is a great need for methods to analyze multi-timepoint single-cell data. In this study, we present CASi, a comprehensive framework to provide a full analysis pipeline for analyzing scRNA-seq data from multi-timepoint designs, ultimately creating an informative profile of dynamic cellular changes.

The first step of CASi uses the neural network classifier to achieve cross-time points cell annotation with high accuracy. And as a supervised learning method, it efficiently avoids the overclustering issue. Overclustering often appears in unsupervised clustering. When the total number of cells increases, the cells of one type will be separated into two or even more clusters. Using the same scenario settings of the simulation, we compare ARI of unsupervised clustering implemented in the Seurat package with ARI of CASi using supervised clustering. The results of three scenarios are included in the supplementary file Figure 11. It can be observed that our method's

ARI increases with the cell number increasing, while the Seurat ARI decreases with the cell number increasing. This indicates that the supervised clustering method of CASi avoids the overclustering problem.

However, supervised clustering methods also have a universal disadvantage when dealing with novel cell types. When the new/unknown cell types appear only in the testing data, the classifier will not be able to distinguish them and will assign these cells to an existing but wrong cell type. In continuing experiments, with the progression of the tumor, some existing cell types might disappear and some new/unknown cell types that are not present at the beginning might appear later. Another innovation of our work is the detection of novel/unknown cell types that emerge over time. We designed a pipeline using correlation and t-test in a reliable way to distinguish novel cell types. We have demonstrated state-of-the-art power in both simulation studies and real-world data application compared with existing methods serving a similar purpose.

Differential expression analysis is one of the most common tasks in scRNA-seq studies, and it is also a critical step in CASi. Several methods have been proposed to detect DEGs, including BPSC[34], MAST[8], and Monocle[35]. These methods are designed for two group comparisons using scRNA-seq data, and they may have difficulty accounting for the time effect and interaction terms across different timepoints' scRNA-seq data. CASi allows users to detect tDEGs, which refers to genes that express wildly changing behavior over time. To find tDEGs, we combine the generalized linear model with iterative feature selection. For each gene, a p-value is obtained from the GLM model to indicate evidence of differential expression. We initially started the analysis with the generalized linear mixed model (GLMM) to account for subject's random effect. However, GLMM is computationally expensive, and looping through each gene is infeasible in most scenarios. We then turned to GLM and have found that the regression analysis results, i.e., the coefficients and p-value, are very similar between GLM and GLMM models in the explored settings. Thus, taking into consideration the model complexity and the computational cost, we eventually chose GLM to model the multi-timepoint gene expression count data.

There are some scenarios that CASi is not at advantage with existing baselines. First, CASi builds up on the artificial neural network. If the initial timepoint's data is dramatically different from later timepoint's data, the neural network would fail to borrow useful information and could not be very well trained. In this case, CASi, which features the supervised clustering, will not be at advantage with existing baselines. And it would better to use unsupervised clustering. Additionally, if the users imply that time effect is not strong, for example, even the cells are extracted from different time points, patients may be at the status, i.e., nothing really happens to the cells over time. In this scenario, it's not very meaningful to use CASi over other existing methods. Current methods can be extended in a few ways. First, the novel cell detection framework may not work well when the novel cell is transcriptomically similar to the known cell types. In this scenario, it may be beneficial to incorporate other information, such as copy number variation or mutation data, into consideration. Second, the current cell type annotation step pools the data from all the samples together for training and applying the neural network. It is possible that the cell populations are quite distinct for subjects from different conditions. As a result, it may be beneficial to perform the annotation in subjects within the same condition if sample size allows. Finally, the current GLM framework can only detect linear effect, i.e., the linear change of gene expression by time or the interaction of time and condition. It helps if the model can be extended to allow the detection of nonlinear changes.

Data availability

The PBMC Single-cell expression data used for simulation study was obtained from the 10X website (https://support.10xgenomics.com/)[24]. The MCL Single-cell expression data used for real-world data application was downloaded from the European Genome-Phenome Archive (EGA) database with the accession code EGAS00001005019[26]. Source codes for running CASi and installing the R package are available on GitHub (https://github.com/yizhuo-wang/CASi).

Received: 10 December 2023; Accepted: 1 April 2024
Published online: 09 May 2024

References

1. L Lun, A. T., Bach, K. & Marioni, J. C. Pooling across cells to normalize single-cell RNA sequencing data with many zero counts. *Genome Biol.* **17**, 1–14 (2016).
2. Satija, R., Farrell, J. A., Gennert, D., Schier, A. F. & Regev, A. Spatial reconstruction of single-cell gene expression data. *Nat. Biotechnol.* **33**, 495–502 (2015).
3. Guo, M., Wang, H., Potter, S. S., Whitsett, J. A. & Xu, Y. Sincera: A pipeline for single-cell RNA-seq profiling analysis. *PLoS Comput. Biol.* **11**, e1004575 (2015).
4. Kiselev, V. Y., Yiu, A. & Hemberg, M. scmap: Projection of single-cell RNA-seq data across data sets. *Nat. Meth.* **15**, 359–362 (2018).
5. De Kanter, J. K., Lijnzaad, P., Candelli, T., Margaritis, T. & Holstege, F. C. Chetah: A selective, hierarchical cell type identification method for single-cell RNA sequencing. *Nucleic Acids Res.* **47**, e95–e95 (2019).
6. Li, Z., Wang, Y., Ganan-Gomez, I., Colla, S. & Do, K.-A. A machine learning-based method for automatically identifying novel cells in annotating single-cell RNA-seq data. *Bioinformatics* **38**, 4885–4892 (2022).
7. Love, M. I., Huber, W. & Anders, S. Moderated estimation of fold change and dispersion for RNA-seq data with deseq2. *Genome Biol.* **15**, 1–21 (2014).
8. Finak, G. *et al.* Mast: A flexible statistical framework for assessing transcriptional changes and characterizing heterogeneity in single-cell rna sequencing data. *Genome Biol.* **16**, 1–13 (2015).
9. Ravindra, N. G. *et al.* Single-cell longitudinal analysis of sars-cov-2 infection in human airway epithelium. *BioRxiv* (2020).
10. Zhang, K. *et al.* Longitudinal single-cell RNA-seq analysis reveals stress-promoted chemoresistance in metastatic ovarian cancer. *Sci. Adv.* **8**, eabm1831 (2022).
11. Ramazzotti, D. *et al.* Lace: Inference of cancer evolution models from longitudinal single-cell sequencing data. *J. Comput. Sci.* **58**, 101523 (2022).

12. Ma, W., Su, K. & Wu, H. Evaluation of some aspects in supervised cell type identification for single-cell RNA-seq: classifier, feature selection, and reference construction. *Genome Biol.* **22**, 1–23 (2021).
13. Li, Z. & Feng, H. A neural network-based method for exhaustive cell label assignment using single cell RNA-seq data. *Sci. Rep.* **12**, 1–12 (2022).
14. Li, Z. & Feng, H. A neural network-based method for exhaustive cell label assignment using single cell rna-seq data. *Sci. Rep.* **12** (2021).
15. Aran, D. *et al.* Reference-based analysis of lung single-cell sequencing reveals a transitional profibrotic macrophage. *Nat. Immunol.* **20**, 163–172. https://doi.org/10.1038/s41590-018-0276-y (2019).
16. Chollet, F. Building autoencoders in keras. *The Keras Blog* **14** (2016).
17. Eckle, K. & Schmidt-Hieber, J. A comparison of deep networks with relu activation function and linear spline-type methods. *Neural Netw.* **110**, 232–242 (2019).
18. Kingma, D. P. & Ba, J. Adam: A method for stochastic optimization. *arXiv preprint*arXiv:1412.6980 *(2014).*
19. Li, M., Zhang, T., Chen, Y. & Smola, A. J. Efficient mini-batch training for stochastic optimization. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, 661–670 (2014).
20. Blondel, V. D., Guillaume, J.-L., Lambiotte, R. & Lefebvre, E. Fast unfolding of communities in large networks. *J. Stat. Mech. Theory Exp.* **2008**, P10008 (2008).
21. McCullagh, P. & Nelder, J. A. *Generalized linear models* (Routledge, 2019).
22. Santos, J. M. & Embrechts, M. On the use of the adjusted rand index as a metric for evaluating supervised classification. In *International conference on artificial neural networks*, 175–184 (Springer, 2009).
23. Alquicira-Hernandez, J., Sathe, A., Ji, H. P., Nguyen, Q. & Powell, J. E. scpred: Accurate supervised method for cell-type classification from single-cell RNA-seq data. *Genome Biol.* **20**, 1–17 (2019).
24. Zheng, G. X. *et al.* Massively parallel digital transcriptional profiling of single cells. *Nat. Commun.* **8**, 1–12 (2017).
25. Miller, C. A. *et al.* Visualizing tumor evolution with the fishplot package for R. *BMC Genomics* **17**, 1–3 (2016).
26. Zhang, S. *et al.* Longitudinal single-cell profiling reveals molecular heterogeneity and tumor-immune evolution in refractory mantle cell lymphoma. *Nat. Commun.* **12**, 1–17 (2021).
27. Lee, J. *et al.* Ifitm3 functions as a pip3 scaffold to amplify pi3k signalling in B cells. *Nature* **588**, 491–497 (2020).
28. Smith, S. M., Anastasi, J., Cohen, K. S. & Godley, L. A. The impact of myc expression in lymphoma biology: Beyond burkitt lymphoma. *Blood Cells Mol. Dis.* **45**, 317–323 (2010).
29. Dai, B. *et al.* B-cell receptor-driven malt1 activity regulates myc signaling in mantle cell lymphoma. *Blood J. Am. Soc. Hematol.* **129**, 333–346 (2017).
30. Zhang, L. *et al.* Metabolic reprogramming toward oxidative phosphorylation identifies a therapeutic target for mantle cell lymphoma. *Sci. Transl. Med.* **11**, eaau1167 (2019).
31. Noble, R. A. *et al.* Simultaneous targeting of glycolysis and oxidative phosphorylation as a therapeutic strategy to treat diffuse large b-cell lymphoma. *Br. J. Cancer* 1–11 (2022).
32. Zhang, W. *et al.* Dysregulation of n6-methyladenosine regulators predicts poor patient survival in mantle cell lymphoma. *Oncol. Lett.* **18**, 3682–3690 (2019).
33. Wang, L. *et al.* Toll-like receptor-4 signaling in mantle cell lymphoma: Effects on tumor growth and immune evasion. *Cancer* **119**, 782–791 (2013).
34. Vu, T. N. *et al.* Beta-poisson model for single-cell RNA-seq data analyses. *Bioinformatics* **32**, 2128–2135 (2016).
35. Trapnell, C., Cacchiarelli, D. & Qiu, X. Monocle: Cell counting, differential expression, and trajectory analysis for single-cell rna-seq experiments (2017).

Acknowledgements

Z.L. was partially supported by NIH grant R03CA270725. X.H. was partially supported by NIH grants R01CA272806, U54CA096300, and the Dr. Mien-Chie Hung and Mrs. Kinglan Hung Endowed Professorship. The authors thank the anonymous reviewers for their valuable suggestions.

Competing interests

The authors declare that they have no competing interests. The corresponding author is responsible for submitting a competing interests statement on behalf of all authors of the paper. This statement must be included in the submitted article file

Additional information

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1038/s41598-024-58566-x.

Correspondence and requests for materials should be addressed to X.H. or Z.L.