Genome Biology

RESEARCH

Open Access

Benchmarking single-cell cross-omics imputation methods for surface protein expression

Chen-Yang Li[1†], Yong-Jia Hong[1†], Bo Li[1,2] and Xiao-Fei Zhang[1,2*]

[†]Chen-Yang Li and Yong-Jia Hong contributed equally to this work.

*Correspondence: zhangxf@ccnu.edu.cn

[1] School of Mathematics and Statistics, and Hubei Key Lab–Math. Sci., Central China Normal University, Wuhan 430079, China
[2] Key Laboratory of Nonlinear Analysis & Applications (Ministry of Education), Central China Normal University, Wuhan 430079, China

Abstract

Background: Recent advances in single-cell multimodal omics sequencing have facilitated the simultaneous profiling of transcriptomes and surface proteomes within individual cells, offering insights into cellular functions and heterogeneity. However, the high costs and technical complexity of protocols like CITE-seq and REAP-seq constrain large-scale dataset generation. To overcome this limitation, surface protein data imputation methods have emerged to predict protein abundances from scRNA-seq data.

Results: We present a comprehensive benchmark of twelve state-of-the-art imputation methods across eleven datasets and six scenarios. Our analysis evaluates the methods' accuracy, sensitivity to training data size, robustness across experiments, and usability in terms of running time, memory usage, popularity, and user-friendliness. With benchmark experiments in diverse scenarios and a comprehensive evaluation framework of the results, our study offers valuable insights into the performance and applicability of surface protein data imputation methods in single-cell omics research.

Conclusions: Based on our results, Seurat v4 (PCA) and Seurat v3 (PCA) demonstrate exceptional performance, offering promising avenues for further research in single-cell omics.

Keywords: Single-cell multimodal omics, Single-cell RNA-seq, Surface protein expression, Cross-omics imputation, Benchmark

Background

Recent advances in single-cell multimodal omics (scMulti-omics) sequencing have revolutionized our ability to simultaneously profile multiple molecular layers within individual cells, offering comprehensive insights into cellular functions and heterogeneity [1–4]. Protocols such as cellular indexing of transcriptomes and epitopes (CITE-seq) and RNA expression and protein sequencing assay (REAP-seq) enable the concurrent quantification of transcriptomes and surface proteomes within the same cell, effectively

bridging the gap between gene expression and protein functionality [5, 6]. These integrated approaches have the potential to reveal cellular diversity that single-cell RNA sequencing (scRNA-seq) alone might overlook [7, 8].

While CITE-seq and REAP-seq represent groundbreaking technologies with immense potential, their prohibitive costs and intricate technical requirements, compared to scRNA-seq, present obstacles to the widespread generation of large-scale public datasets essential for unraveling the complexities of diverse tissues [9]. Given that genes are the blueprints for protein synthesis and that a correlation exists between transcriptomes and proteomes [10, 11], a promising solution is to leverage large reference datasets to learn the relationship between RNA and proteins. This relationship can then be used to predict protein abundances in cells measured only by scRNA-seq. Several recent studies have explored this possibility, leading to the development of various surface protein data imputation methods.

These imputation methods utilize datasets generated by CITE-seq or REAP-seq, which include both surface protein and gene expression data, as training data to develop machine learning models. These models are then used to predict surface protein expression in cells measured by scRNA-seq alone (test data). The imputation methods can be broadly categorized into three types: traditional machine learning-based methods and two types of deep learning-based methods. The first type of methods, including Seurat v3 (CCA) [12], Seurat v3 (PCA) [12], Seurat v4 (CCA) [13], and Seurat v4 (PCA) [13], first identify mutual nearest neighbors between training and test datasets in a shared low-dimensional space and then transfer surface protein data from the training dataset to the test based on the identified mutual nearest neighbors. The other two types are both based on deep learning, differing in their network structures. The first type, including cTP-net [14], sciPENN [15], scMOG [16], and scMoGNN [17], employs deep neural networks to directly learn a mapping between transcriptomic and proteomic data from the training dataset, which is then used to make imputations for the test dataset. The second type, including TotalVI [18], Babel [19], moETM [20], and scMM [21], is based on an encoder-decoder framework. These methods first use an encoder to embed both transcriptomic and proteomic data into a joint latent representation, and then use a decoder to make predictions for the proteomic data.

Although these methods have demonstrated good performance in various scenarios, predicting protein expression from gene expression data remains challenging due to post-transcriptional and post-translational modifications, as well as differences in protein stability and localization [22–25]. Therefore, a comprehensive evaluation of these methods in practical applications is essential. In this study, we present an extensive benchmark of twelve state-of-the-art imputation methods using eleven CITE-seq and REAP-seq datasets across six distinct benchmark scenarios. We employ various accuracy measures to quantitatively evaluate the predicted values at both the protein and cell levels. Additionally, we assess the methods' sensitivity to training data size, robustness across experiments, and efficiency in terms of time and memory. We also assess their popularity based on the number of stars on their official GitHub repositories and evaluate their user-friendliness in terms of installation, code, and tutorial. Our findings indicate that Seurat-based methods, particularly Seurat v4 (PCA) and Seurat v3 (PCA), demonstrate superior accuracy and robustness across diverse experiments, showing

relative insensitivity to training data size. They are also highly efficient in terms of memory usage, widely popular with numerous stars on their GitHub repository, and provide high-quality installation guides, codes, and tutorials. However, they exhibit longer running times compared to some deep learning-based methods, which highlights scalability concerns and underscores the necessity for future enhancements to manage larger datasets effectively. Additionally, we offer a decision-tree-style guidance scheme that intuitively presents the recommended methods for specific scenarios based on benchmark evaluation results, facilitating more efficient selection of the most appropriate methods.

Results

Overview of the benchmark scheme

The overall pipeline of this benchmark study is illustrated in Fig. 1. In each experiment, we use one CITE-seq or REAP-seq dataset containing paired transcriptomic and proteomic data as the training data. For the test data, we mask the proteomic data from another CITE-seq or REAP-seq dataset, retaining only the transcriptomic data to simulate scRNA-seq data, and then use various imputation methods to predict the corresponding proteomic data (Fig. 1a).

To comprehensively evaluate the performance of these imputation methods, our benchmark includes twelve state-of-the-art methods: four Seurat-based methods (Seurat v3 (CCA), Seurat v3 (PCA), Seurat v4 (CCA), and Seurat v4 (PCA)), cTP-net, sciPENN, scMOG, scMoGNN, TotalVI, Babel, moETM, and scMM. These methods are categorized based on their imputation strategies (Fig. 1b): imputing by mutual nearest neighbors, imputing by learning a mapping between transcriptomic and proteomic data using deep learning, and imputing by learning a joint latent representation using an encoder-decoder framework. To test the generalizability and robustness of these imputation methods, we use eleven datasets and conduct experiments under six distinct benchmark scenarios (Fig. 1b and Additional file 1: Tables S1, S2): (1) Random holdout: A dataset is randomly divided into training and test sets to address the case without technical or biological differences; (2) Different training data sizes: Evaluating performance with varying training data sizes to understand how training data size influences each method; (3) Different samples: Considering the scenario where the training and test datasets come from different samples; (4) Different tissues: Testing each method's generalizability when predicting protein expression for cells from tissues different from those used in the training set; (5) Different clinical states: Assessing each method's ability to transfer between datasets with biological variations; (6) Different protocols: Investigating performance when training and test datasets are derived from different sequencing protocols.

After generating imputation values using different methods (Fig. 1c), we design a comprehensive framework to evaluate their performance (Fig. 1d). First, we evaluate the accuracy of methods using Pearson correlation coefficient (PCC) and root mean square error (RMSE). To provide an overall performance metric, we also introduce an average rank score (ARS) that combines the rank score values of methods based on PCC and RMSE. A higher ARS value indicates better accuracy performance across all metrics in the experiment. Second, we assess how the methods' accuracy performance changes with varying training data sizes by running the methods on training sets of different sample sizes. This analysis helps to understand how the amount of training data influences the

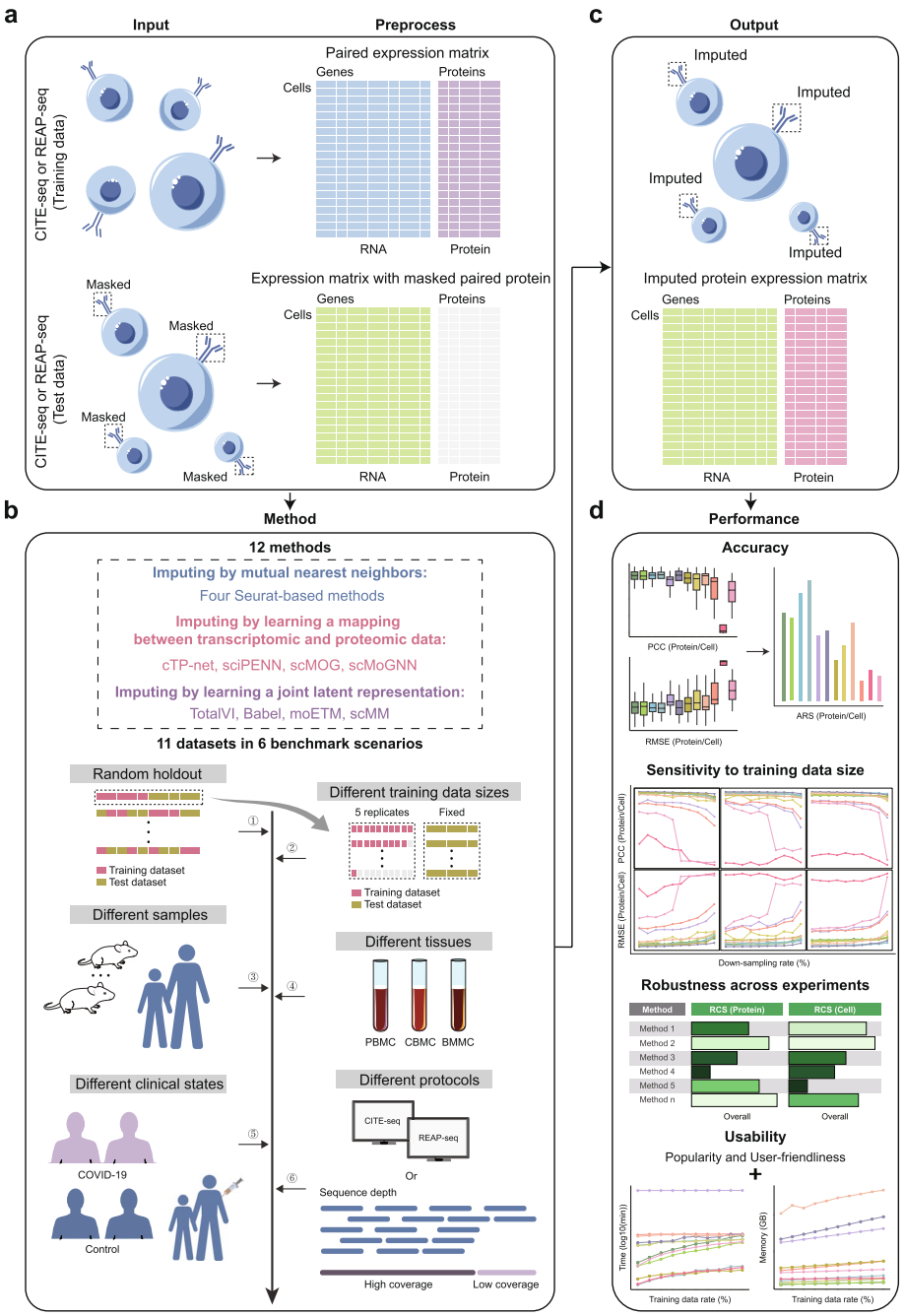

Fig. 1 Overview of the benchmark framework. **a** The raw CITE-seq or REAP-seq datasets are preprocessed, with one dataset containing paired gene and protein expression matrices used as the training data, and the other dataset having its protein expression matrix masked, with only the gene expression matrix used to simulate scRNA-seq data, which serves as the test data. **b** Twelve state-of-the-art methods, categorized according to their imputation strategies, are evaluated using eleven datasets across six benchmark scenarios. **c** After training on the training data, the imputed protein expression matrix for the test data is generated through inference. **d** The benchmark results are assessed based on accuracy, sensitivity to training data size, robustness across experiments, and usability in terms of popularity, user-friendliness, running time, and memory usage

methods' accuracy performance. Third, we evaluate the robustness of methods across experiments by introducing a robustness composite score (RCS). This metric considers both the mean and standard deviation of the ARS values across different experiments. We primarily evaluate experiments demonstrating technical and biological differences that closely resemble those conducted in real-world applications. These experiments stem from scenarios involving different samples, tissues, clinical states, and protocols. A high RCS value indicates that a method not only performs well on average but also maintains consistent performance across all experiments with technical and biological differences. Accurate protein abundances across cells are crucial for tasks such as differential expression analysis and omics feature correlation analysis, while accurate protein abundances in individual cells are essential for tasks like cell clustering analysis and cell trajectory inference. Therefore, we assess the methods at both the protein and cell levels for the above evaluations to accommodate the varying requirements of different downstream tasks. Finally, we compare the methods in terms of usability metrics, including popularity (measured by the number of stars on their official GitHub repositories), user-friendliness (measured by the quality of installation procedures, codes, and tutorials), running time, and memory usage.

Scenario 1: evaluating accuracy performance over random holdout

To evaluate the performance of different imputation methods, we begin with a straightforward scenario where the training and test datasets are randomly divided from the same dataset. We utilize three widely referenced datasets: CITE-PBMC-Stoeckius [5], CITE-CBMC-Stoeckius [5], and CITE-BMMC-Stuart [12], which have been extensively used in previous studies assessing surface protein expression imputation methods [12, 14, 19, 21]. For each dataset, we randomly split the cells into two groups: a training dataset comprising half of the cells and a test dataset with the remaining half. The training dataset is used to train the models, and the test dataset is used to evaluate their performance. To account for variability in the dataset split, we repeat the experiment five times and present the results of each repetition using boxplots. Finally, in this scenario, we conduct a total of 15 experiments, consisting of three datasets, with five repeated experiments for each dataset.

Figure 2a shows the median PCC of each method across proteins or cells in each replicate experiment, while the corresponding results evaluated using RMSE are presented in Additional file 2: Fig. S1. Most methods exhibit stable performance across different replicates, except for moETM, which appears sensitive to the split between training and test datasets. Notably, moETM demonstrates superior and stable performance with the CITE-CBMC-Stoeckius dataset but exhibits considerable performance fluctuations with the other two datasets, suggesting that its performance may heavily depend on the underlying dataset. The performance of each method also varies across datasets and evaluation metrics, with no clear overall winner. To summarize these results, we calculate the average of the 15 ARS values (from three datasets, five repetitions) at both the protein and cell levels. We find that cTP-net outperforms other methods at the protein level while achieving moderate performance at the cell level (Fig. 2b, c). Unlike cTP-net, which shows a preference for the protein level, Seurat v4 (PCA), Seurat v4 (CCA), and

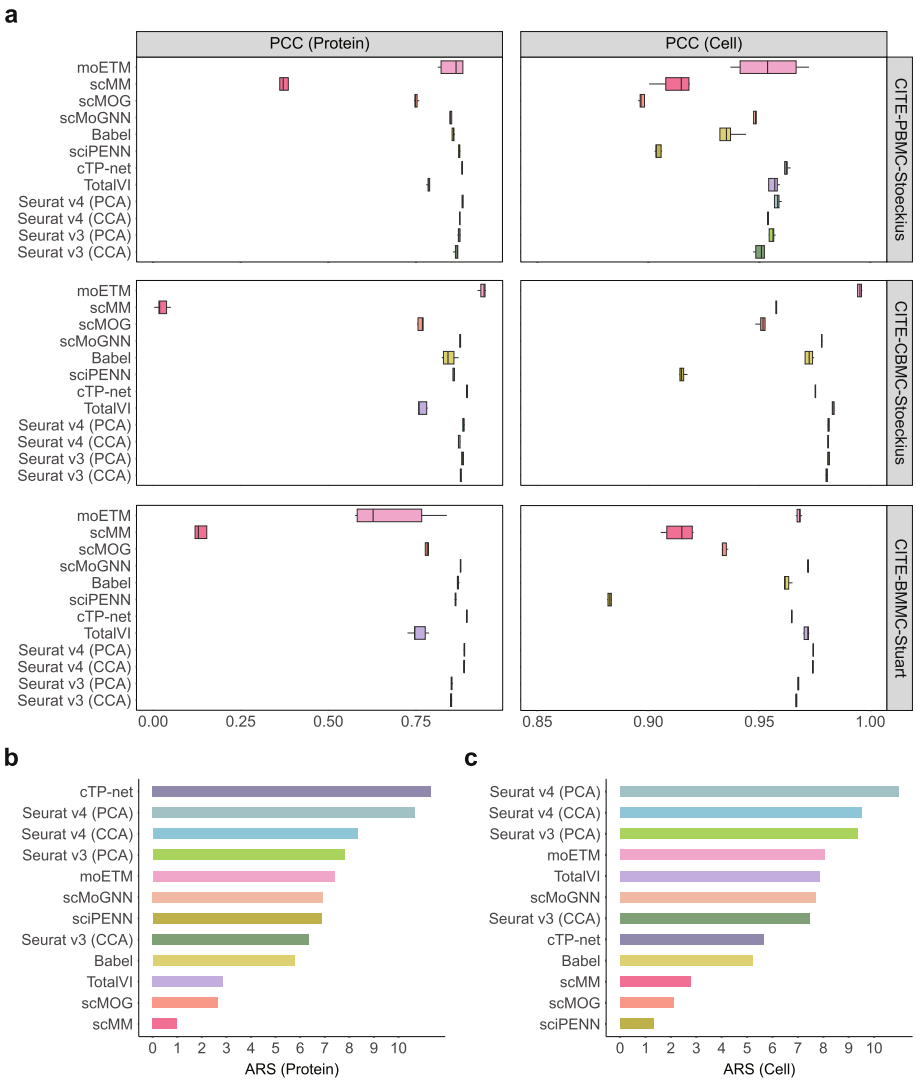

Fig. 2 Comparison of PCC among methods over random holdout. **a** Boxplots for PCC of each method in experiments with the CITE-PBMC-Stoeckius, CITE-CBMC-Stoeckius, and CITE-BMMC-Stuart datasets. The boxplots display the median PCC of each method across proteins or cells in each replicate experiment. Center line: median; box limits: upper and lower quartiles; whiskers: 1.5 times interquartile range. **b**, **c** Barplots for ARS value of each method at the protein (**b**) and cell (**c**) levels. Data are presented as the average of the 15 ARS values (from three datasets, five repetitions) across all experiments in this scenario. Methods are ordered according to their performance

Seurat v3 (PCA) demonstrate competitive performance at both the protein and cell levels (Fig. 2b, c).

Scenario 2: evaluating accuracy performance over different training data sizes

We investigate the impact of training data size variations on the accuracy performance of imputation methods. Using the CITE-PBMC-Stoeckius, CITE-CBMC-Stoeckius, and CITE-BMMC-Stuart datasets, we first randomly split each dataset into training and test sets, following scenario 1. Subsequently, we down-sample the training dataset by removing cells at intervals of 10% from 0 to 90%, while keeping the test dataset constant. To address variability, we conduct five replicate experiments for

each dataset. In total, we conduct 150 experiments in this scenario, using three datasets and performing five repeated experiments for each dataset across ten different down-sampling rates.

Under each down-sampling rate, we first calculate the median PCC and RMSE across proteins or cells for each experiment, and then take the median of these values across five replicate experiments to obtain a robust performance measure, whose trends across different down-sampling rates are illustrated in Fig. 3a and Additional file 2: Fig. S2. As expected, imputation performance generally decreases as the training dataset size is reduced. Notably, methods such as Seurat v3 (CCA), Seurat v4 (CCA), and Seurat v4 (PCA) show relative insensitivity to training data size

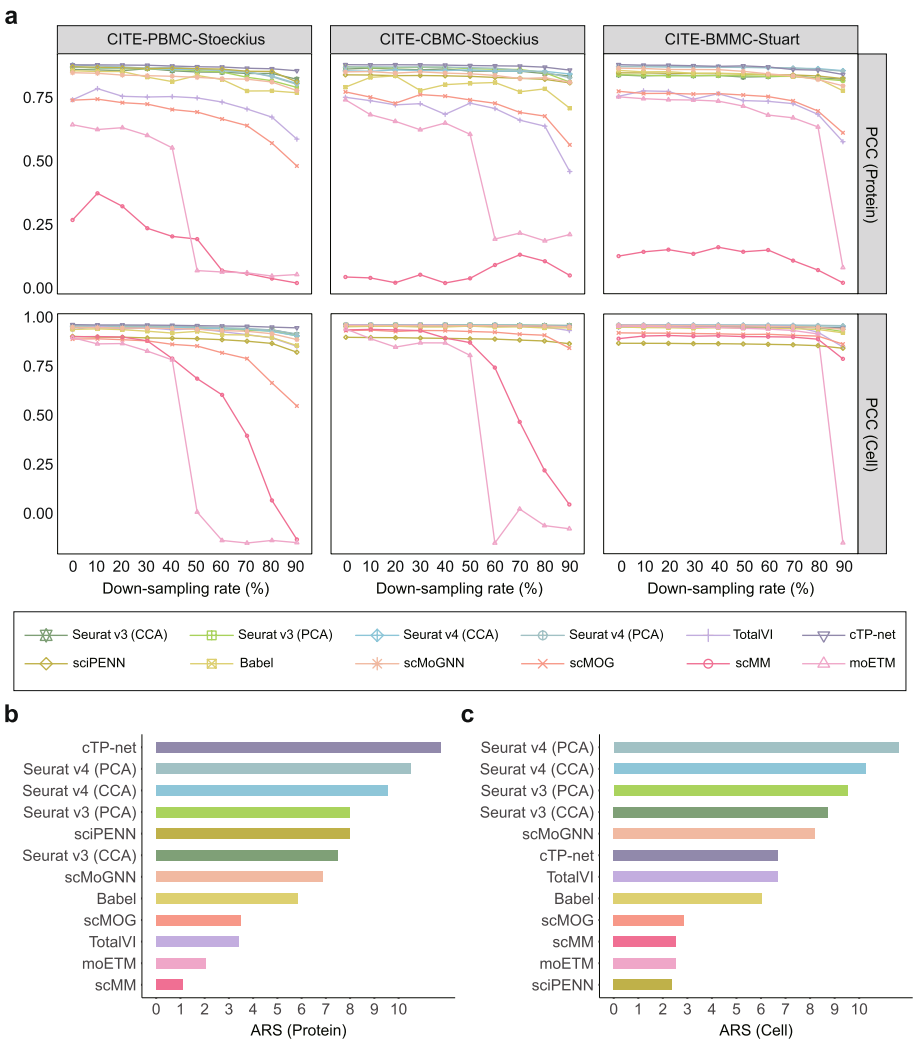

Fig. 3 Comparison of PCC among methods over different training data sizes. **a** Line plots for PCC of each method as the down-sampling rate of the training dataset changes. The horizontal axis represents the down-sampling rate of the training dataset. The vertical axis represents the median PCC performance at the protein or cell level across five replicate experiments under each down-sampling rate. Each line represents a method. **b**, **c** Barplots for ARS value of each method at the protein (**b**) and cell (**c**) levels. Data are presented as the average of the 150 ARS values (from three datasets, five repetitions, and ten down-sampling rates) across all experiments in this scenario. Methods are ordered according to their performance

variations, maintaining robust performance. In contrast, deep learning-based methods like scMM, scMOG, and moETM, which perform poorly initially, are more sensitive to reductions in training data size. TotalVI also exhibits some sensitivity at the protein level. This sensitivity may be due to the larger training datasets required by deep learning models for optimal performance. To comprehensively rank the twelve imputation methods, we calculate the average of the 150 ARS values (from three datasets, five repetitions, and ten down-sampling rates) at both the protein and cell levels. cTP-net, Seurat v4 (PCA), and Seurat v4 (CCA) demonstrate the best performance across various down-sampling rates at the protein level (Fig. 3b). At the cell level, Seurat v4 (PCA), Seurat v4 (CCA), and Seurat v3 (PCA) outperform other methods (Fig. 3c).

Scenario 3: evaluating accuracy performance over different samples

In this scenario, we evaluate the performance of imputation methods when the training and test datasets originate from different samples, reflecting common real-world conditions. We use three datasets: CITE-PBMC-Li [26, 27], CITE-SLN111-Gayoso [18], and CITE-SLN208-Gayoso [18]. The CITE-PBMC-Li dataset includes data from eight volunteers measured before and after HIV vaccination. To eliminate potential batch differences from biological variation, we use only pre-vaccination data. The volunteers are randomly assigned to two non-overlapping groups: group 1, consisting of four volunteers, and group 2, comprising the remaining four. We conduct two complementary experiments, alternating between using one group as the training set and the other as the test set. To account for randomness, we repeat the group assignments five times and conduct the experiments for each random division. The CITE-SLN111-Gayoso and CITE-SLN208-Gayoso datasets contain data from the spleen and lymph node tissues of two mice. For each dataset, we perform two complementary experiments, alternating between using one mouse as the training set and the other as the test set. In total, 14 experiments are conducted in this scenario. For the CITE-PBMC-Li dataset, two complementary experiments are performed with five repetitions, while for the CITE-SLN111-Gayoso and CITE-SLN208-Gayoso datasets, two complementary experiments are conducted for each.

A comparison of the evaluation results from experiments involving different datasets reveals significant differences. In experiments involving the CITE-PBMC-Li dataset, moETM consistently achieves the best performance in protein-level evaluation metrics (Fig. 4a and Additional file 2: Fig. S3a). However, no single method consistently outperforms others at the cell level, with TotalVI, Seurat v3 (PCA), and scMoGNN each demonstrating their respective strengths (Fig. 4a and Additional file 2: Fig. S3a). Boxplots in Additional file 2: Fig. S4 are based on the median evaluation metric value across proteins or cells of each repetition, showing the performance of each method across different random divisions. We observe that most methods exhibit relatively stable performance, with the aforementioned methods consistently maintaining their respective advantages. In the CITE-SLN111-Gayoso dataset, TotalVI and Seurat-based methods excel at the protein and cell levels, respectively (Fig. 4b and Additional file 2: Fig. S3b). In the CITE-SLN208-Gayoso dataset, TotalVI leads at both the protein level and for PCC at the cell level (Fig. 4c and Additional

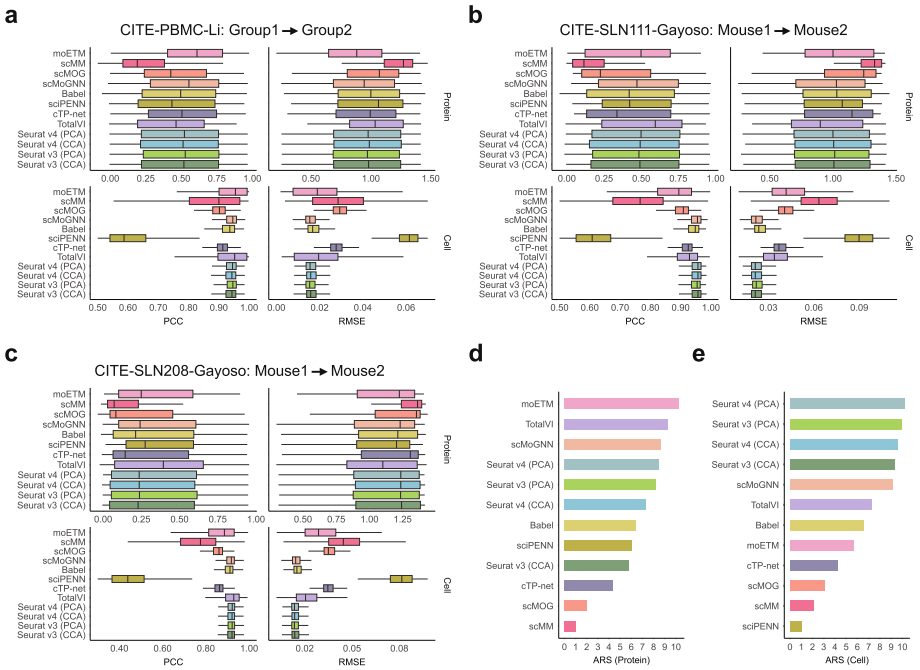

Fig. 4 Comparison of quantitative evaluation metrics among methods over different samples. **a**–**c** Boxplots for PCC or RMSE of each method in the CITE-PBMC-Li: Group1→Group2 (**a**), CITE-SLN111-Gayoso: Mouse1→Mouse2 (**b**), and CITE-SLN208-Gayoso: Mouse1→Mouse2 experiments (**c**). The boxplots display values calculated at the protein or cell level. In (**a**), the results represent all five repetitions of the experiment, with all values at the protein or cell level across these repetitions included in the boxplot, where each point indicates the evaluation metric value of a protein or cell from the five repetitions. Center line: median; box limits: upper and lower quartiles; whiskers: 1.5 times interquartile range. **d**, **e** Barplots for ARS value of each method at the protein (**d**) and cell (**e**) levels. Data are presented as the average of the six ARS values (from three datasets, two complementary experiments per dataset) across all experiments in this scenario. Methods are ordered according to their performance

file 2: Fig. S3c). To summarize, we evaluate the methods' performance at the protein and cell levels by averaging the six ARS values (from three datasets, two complementary experiments per dataset). To account for the potential impact of varying numbers of experiments across datasets on the overall results, the ARS for the CITE-PBMC-Li: Group1→Group2 and CITE-PBMC-Li: Group2→Group1 experiments are calculated using the median evaluation metric values across five repetitions. moETM, TotalVI, and scMoGNN show superior performance at the protein level (Fig. 4d). Seurat-based methods consistently demonstrate superior performance when focusing on the accuracy of protein abundances at the cell level (Fig. 4e).

Scenario 4: evaluating accuracy performance over different tissues

We assess the performance of the methods when the training and test datasets are derived from different tissues. We utilize three datasets: CITE-BMMC-Stuart (bone marrow mononuclear cells), CITE-CBMC-Stoeckius (cord blood mononuclear cells), and CITE-PBMC-Stoeckius (peripheral blood mononuclear cells), each representing cells from distinct but related blood sources [28, 29]. Each of these datasets is paired with one another, resulting in six experiments where each dataset is alternately used as the training and test dataset.

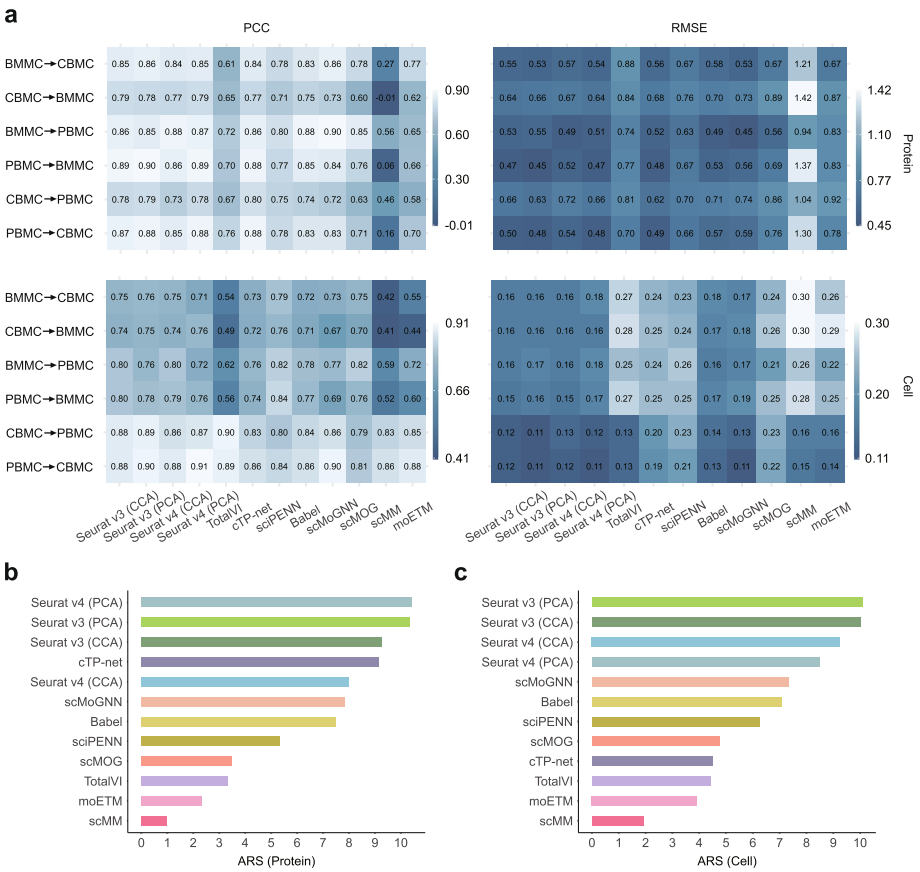

Fig. 5 Comparison of quantitative evaluation metrics among methods over different tissues. **a** Heatmaps for PCC or RMSE of each method in each experiment of this scenario. The horizontal axis represents each method and the vertical axis represents each experiment. The values are the median across all proteins or cells. **b**, **c** Barplots for ARS value of each method at the protein (**b**) and cell (**c**) levels. Data are presented as the average of the six ARS values across all experiments in this scenario. These methods are ordered according to their performance

Summarizing the results of these six experiments (Fig. 5a), we observe variability in benchmark results across different assessment metrics. Specifically, for metrics at the protein level, Seurat-based methods generally lead in performance except in the BMMC→PBMC and CBMC→PBMC experiments, where scMoGNN and cTP-net outperform other methods, respectively. For PCC at the cell level, sciPENN shows superior performance, except in the CBMC→PBMC and PBMC→CBMC experiments, where TotalVI and Seurat v4 (PCA) perform best, respectively. Seurat-based methods consistently demonstrate superior performance in RMSE at the cell level across all experiments. An interesting observation is that protein-level metrics are more sensitive to the direction of data migration. The leading methods achieve higher PCC values and lower RMSE values in the BMMC→CBMC and PBMC→CBMC experiments compared to their respective complementary experiments. To summarize the results across all six experiments using average ARS values, Seurat v4 (PCA), Seurat v3 (PCA), and Seurat v3 (CCA) exhibit superior performance for protein-level metrics (Fig. 5b). Seurat v3 (PCA), Seurat v3 (CCA), and Seurat v4 (CCA) lead in performance for cell-level metrics (Fig. 5c).

Scenario 5: evaluating accuracy performance over different clinical states

In this scenario, we assess the ability of the methods to transfer between datasets with biological variations. We use three datasets: CITE-PBMC-Haniffa [30], CITE-PBMC-Sanger [31], and CITE-PBMC-Li. The first two datasets are related to COVID-19, while the last one pertains to human immunodeficiency virus (HIV). The CITE-PBMC-Haniffa dataset includes data from volunteers with varying illness severity, healthy volunteers, and patients with severe non-COVID-19 respiratory illnesses. We design two experiments: one using data from healthy volunteers to infer data from critical patients, and another using data from non-COVID-19 acute respiratory disease patients to infer data from asymptomatic individuals. For benchmarking, we randomly select five samples each from the healthy volunteer and critical patient groups due to their large data size. To minimize the influence of randomness on the benchmark results, we perform five repetitions of the experiment. The CITE-PBMC-Sanger dataset categorizes patients by treatment severity. We first use data from asymptomatic patients not requiring oxygen therapy as the training dataset and data from symptomatic patients not requiring oxygen therapy as the test dataset. Next, we use data from symptomatic patients not requiring oxygen therapy as the training dataset and data from symptomatic patients requiring extracorporeal membrane oxygenation (ECMO) therapy as the test dataset. The CITE-PBMC-Li dataset includes data from eight volunteers before and after HIV vaccination. We design two experiments: one using pre-vaccination data (Day0) as the training set and data from the third day post-vaccination (Day3) as the test set, and the other using Day0 data as the training set and data from the seventh day post-vaccination (Day7) as the test set. In the CITE-PBMC-Li: Day0→Day3 experiment, we randomly select data from four volunteers before vaccination as the training set, and use data from the remaining four volunteers collected on the third day post-vaccination as the test set. The same experimental setup is also applied in the CITE-PBMC-Li: Day0→Day7 experiment. To reduce the impact of randomness in training and test set partitioning on the benchmark results, we perform five repetitions for each experiment. In total, 18 experiments are conducted. Among these, the experiments involving CITE-PBMC-Haniffa: Healthy→Critical, CITE-PBMC-Li: Day0→Day3, and CITE-PBMC-Li: Day0→Day7 are each repeated five times to account for sampling randomness.

Benchmark results for protein-level metrics indicate that moETM consistently achieves superior performance across all experiments (Fig. 6a–c and Additional file 2: Fig. S5). Notably, in the four COVID-19 experiments, moETM significantly surpasses other methods, while in the remaining two experiments, scMoGNN demonstrates performance comparable to moETM. This trend remains consistent across repeated experiments (Additional file 2: Fig. S6). In this scenario, characterized by significant technical differences and biological variations, cTP-net's performance decreases significantly compared to scenarios 1 and 2 (Figs. 2, 3, 6 and Additional file 2: Figs. S1, S2, S5), highlighting its limitations in handling batch differences without correction. For cell-level metrics, the results vary across experiments (Fig. 6a–c and Additional file 2: Fig. S5). No single method achieves the best performance in all experiments, and the rankings of methods vary considerably. Finally, we employ the ARS to assess the overall performance of these methods in this scenario. To mitigate the impact of varying numbers of experiments on the evaluation results, for experiments with repetitions, the ARS is

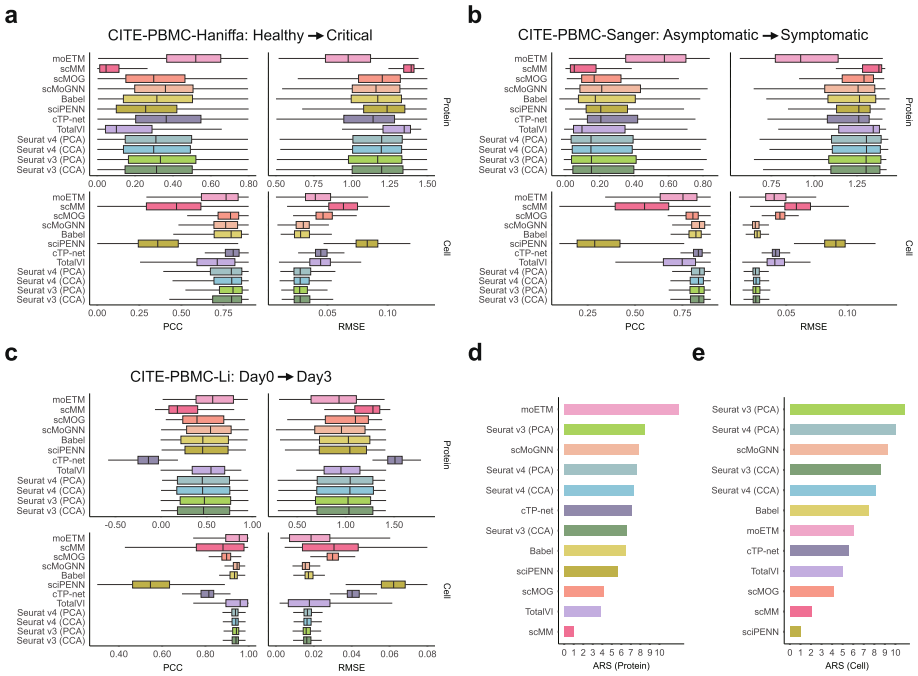

Fig. 6 Comparison of quantitative evaluation metrics among methods over different clinical states. **a**–**c** Boxplots for PCC or RMSE of each method in the CITE-PBMC-Haniffa: Healthy→Critical (**a**), CITE-PBMC-Sanger: Asymptomatic→Symptomatic (**b**), and CITE-PBMC-Li: Day0→Day3 experiments (**c**). The boxplots display values calculated at the protein or cell level. In (**a** and **c**), the results represent all five repetitions of the experiments, with all values at the protein or cell level across these repetitions included in the boxplot, where each point indicates the evaluation metric value of a protein or cell from the five repetitions. Center line: median; box limits: upper and lower quartiles; whiskers: 1.5 times interquartile range. **d**, **e** Barplots for ARS value of each method at the protein (**d**) and cell (**e**) levels. Data are presented as the average of the six ARS values across all experiments in this scenario. These methods are ordered according to their performance

calculated based on the median evaluation metric values across five repetitions. Overall, the top three methods by ARS at the protein level are moETM, Seurat v3 (PCA), and scMoGNN (Fig. 6d). At the cell level, the top three methods are Seurat v3 (PCA), Seurat v4 (PCA), and scMoGNN (Fig. 6e).

Scenario 6: evaluating accuracy performance over different protocols

We delve deeper into the performance of imputation methods in the scenario where training and test datasets originate from different sequencing protocols. Four datasets are utilized: CITE-PBMC10K-10X [32], CITE-PBMC5K-10X [33], CITE-PBMC-Stoeckius, and REAP-PBMC-Peterson [6]. The primary distinction between the first two datasets lies in their sequencing depths [18]. For each pair of datasets, two experiments are conducted, alternating between using one dataset as the training dataset and the other as the test dataset. The latter two datasets differ in sequencing technologies. We also perform two experiments using these latter two datasets. Thus, a total of four experiments are conducted in this scenario.

Upon summarizing the results of these experiments (Fig. 7a), we observe that Seurat-based methods consistently exhibit superior generalization capabilities across all experiments. Their performance remains among the top regardless of the evaluation metrics

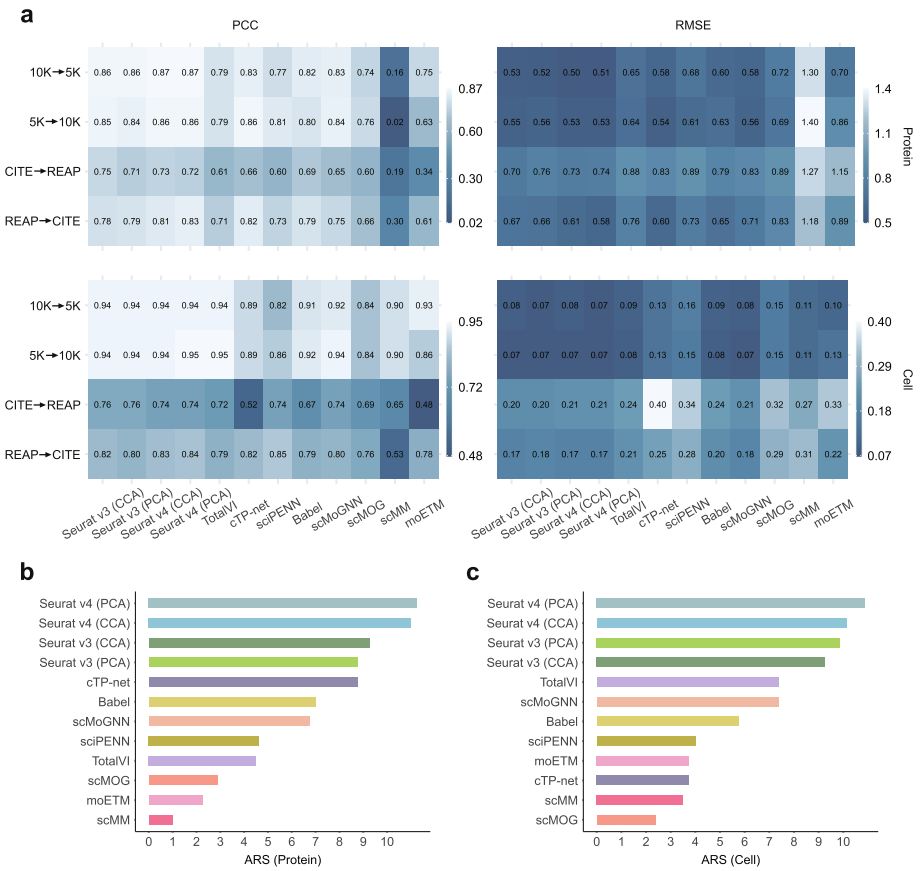

Fig. 7 Comparison of quantitative evaluation metrics among methods over different protocols. **a** Heatmaps for PCC or RMSE of each method in each experiment of this scenario. The horizontal axis represents each method and the vertical axis represents each experiment. The values are the median across all proteins or cells. **b**, **c** Barplots for ARS value of each method at the protein (**b**) and cell (**c**) levels. Data are presented as the average of the four ARS values across all experiments in this scenario. These methods are ordered according to their performance

employed. Seurat v4 generally outperforms Seurat v3, except in the CITE→REAP experiment. Notably, comparing the outcomes of experiments with reciprocal training and test datasets unveils an intriguing finding: leveraging the REAP-PBMC-Peterson dataset as the training dataset yields superior imputation performance compared to using the CITE-PBMC-Stoeckius dataset. Based on the average ARS values across all four experiments, Seurat v4 (PCA), Seurat v4 (CCA), and Seurat v3 (CCA) emerge as the top performers for protein-level metrics (Fig. 7b). Conversely, for cell-level metrics, the leading methods are Seurat v4 (PCA), Seurat v4 (CCA), and Seurat v3 (PCA) (Fig. 7c).

Evaluating usability in terms of time and memory

We evaluate the usability of different imputation methods in terms of time and memory. Using a computational platform with a 16,896 KB L3 Cache, 48 CPU cores, and an NVIDIA Tesla V100 GPU, we conduct experiments on the CITE-BMMC-Stuart dataset. Following the settings from scenario 2, we use various training data rates (from 10 to 100% in 10% intervals), where the training data rate is equivalent to 1 minus the downsampling rate in scenario 2. To reduce biases caused by fluctuations in the experimental

environment and enhance the reliability and robustness of the evaluation results, we perform five repeated experiments for each training data rate.

From the running time trends shown in Fig. 8a, which is based on the medians of the repeated experiments, and the specific recorded values presented in Additional file 1: Table S3, several patterns emerge. cTP-net requires significantly more time than the other methods, often exceeding 11 h, mainly due to its data denoising process with SAVER-X [34]. Other methods can be grouped into three categories based on their running times. TotalVI and scMOG have longer but relatively stable running times across different training data rates. In contrast, sciPENN, Babel, and moETM are the most time-efficient methods, completing tasks in under a minute. While their running times slightly increase with higher training data rates, they remain significantly faster than the other methods. The remaining methods show a clear increase in running time as the training data rate rises. Notably, Seurat v4 is slower than Seurat v3 at lower training data rates, likely due to its more complex preprocessing. However, as the training data rate increases, Seurat v3 becomes slower than Seurat v4, indicating greater sensitivity to training dataset size. Moreover, CCA is slower than PCA within Seurat. Additional file 1: Table S3 presents the detailed running times for each method across repeated experiments. Although variability is observed in some repetitions, the fluctuations remain consistently within a reasonable range.

Regarding memory usage, as shown in Fig. 8b, which is based on the medians of the repeated experiments, and Additional file 1: Table S4, the methods can be divided into three groups. At higher training data rates, both scMOG and scMoGNN exceed 20 GB in memory usage, significantly surpassing the other methods, with scMoGNN showing a more pronounced increase compared to scMOG. The excessive memory usage of scMOG and scMoGNN may be attributed to the pretraining mechanism and the incorporation of graph structures, respectively. cTP-net uses between 10 and 20 GB, with

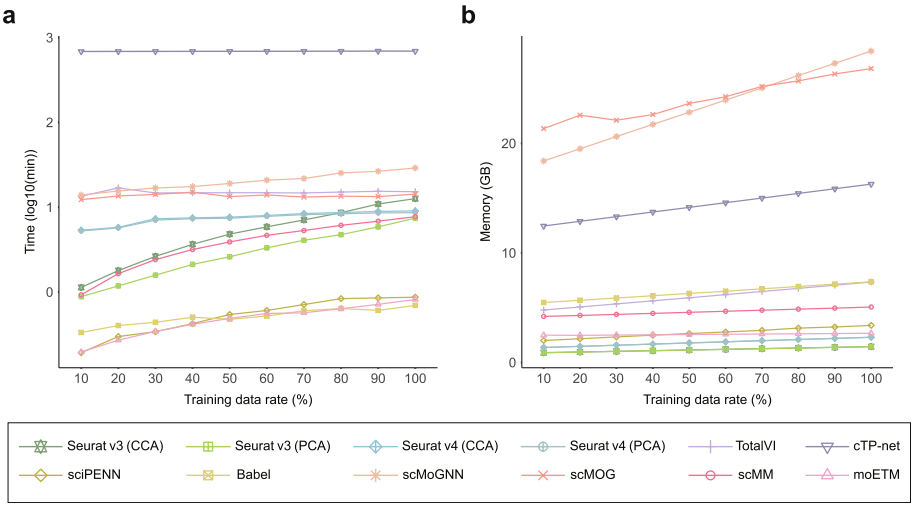

Fig. 8 Comparison of running time and memory usage among methods. **a**, **b** Line plots for running time (**a**) and memory usage (**b**) as the training data rate changes. The horizontal axis represents the training data rate. The vertical axis shows the running time in $\log_{10}$ minutes (**a**) or memory usage in GB (**b**). The values for running time and memory usage are based on the median of the five repetitions. Each line represents a method

usage increasing as the training data rate rises, likely due to data denoising. The remaining methods use less than 10 GB, with minor variations. Within Seurat, memory usage does not depend on the dimensionality reduction method but is slightly higher for Seurat v4 than Seurat v3. Additional file 1: Table S4 records the detailed memory usage for each method across repeated experiments. The results show that memory usage exhibits less fluctuation than running time across repetitions.

Overall summary of benchmark results

We summarize the performance of these methods across four primary dimensions: accuracy, sensitivity to training data size, robustness across experiments, and usability. The accuracy of each scenario is defined as the mean average rank score (ARS) values of different experiments within that scenario, while the overall accuracy is evaluated by the mean ARS values across all scenarios. Sensitivity to training data size is assessed using two metrics: rank score of increments of accuracy performance, which quantifies the variability of methods with changes in training data size, and average-increment composite score (AICS), which considers both the average performance of methods and their variability to training data size to reflect the effectiveness of models. This evaluation is conducted in scenario 2. Robustness across experiments is evaluated by the robustness composite score (RCS), which is calculated based on the ARS values from all experiments with technical and biological differences, indicating the stability and competitiveness of accuracy across these real-world-like experiments. These experiments are conducted on the scenarios of different samples, tissues, clinical states, and protocols, resembling experiments in real-world application scenarios. Both accuracy, sensitivity to training data size, and robustness across experiments are examined at both the protein and cell levels. Usability encompasses time, memory, and quality. For time and memory, we calculate both the mean and increment relative to the training data size using the results recorded in Fig. 8. These metrics provide insights into the efficiency of the methods and their variability to training data size, respectively. Quality is measured through popularity and user-friendliness. The popularity is represented by the number of stars on each method's official GitHub repository (last updated on 15 December 2024). We score the user-friendliness of methods based on three aspects: installation, code, and tutorial. Each method starts with 5 points in each aspect, with points deducted for any identified issues. The user-friendliness score for each method is then calculated by summing the points across all three aspects. The overall benchmark results are summarized in Fig. 9 and the accuracy evaluation results for specific scenarios are shown in Additional file 2: Fig. S7. Based on the results of our study, we draw several findings.

In terms of accuracy, we observe that at the protein level, benchmark results vary across scenarios. Notably, cTP-net tends to show superior performance primarily in scenarios without batch differences (Additional file 2: Fig. S7b, left), likely because it transfers networks learned in the training dataset to the test dataset without performing batch correction. Conversely, moETM and scMoGNN perform well in scenarios with batch differences (Additional file 2: Fig. S7b, left), highlighting the strengths of joint representations and graph neural networks in handling such complexities. Seurat-based methods consistently are the top three methods in all scenarios except for different samples (Additional file 2: Fig. S7b, left), with Seurat v4 (PCA) leading overall (Fig. 9b, left).

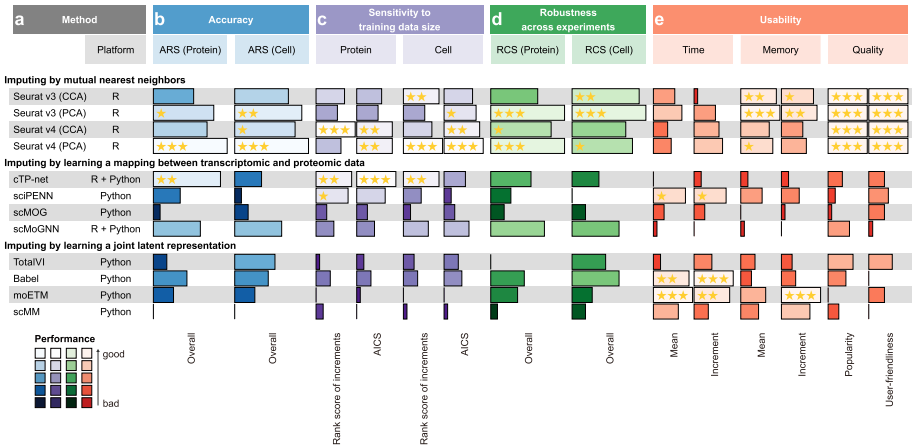

Fig. 9 Overall summary of benchmark results for the methods. **a** Names of the twelve methods and their primary programming languages. Methods are categorized into three main classes based on their imputation strategies. **b** Evaluation of accuracy using mean ARS values across different scenarios. Methods are compared based on the mean ARS values, with better performance indicating more accurate imputation performance. Longer rectangular bars and lighter colors denote better performance. **c** Sensitivity analysis of accuracy performance to training data size using the rank score of increments and AICS. Methods are compared based on these criteria, with better performance indicating less variability to training data size or greater effectiveness. Longer rectangular bars and lighter colors denote better performance. **d** Robustness assessment across experiments with technical and biological differences using RCS. Methods are compared based on the RCS values, with better performance indicating more robust imputation performance. Longer rectangular bars and lighter colors denote better performance. **e** Usability evaluation in terms of time, memory, and quality. Time and memory are assessed using data from Fig. 8, with better performance indicating greater efficiency or less variability to training data size. Quality is evaluated based on popularity and user-friendliness, with better performance indicating more popular or user-friendly methods. Longer rectangular bars and lighter colors denote better performance. In each evaluation, the top three methods are marked with stars: three stars for the best-performing method, two stars for the second-best, and one star for the third

At the cell level, Seurat-based methods consistently show superior performance (Fig. 9b, right), utilizing mutual nearest neighbor cells to achieve accurate protein abundances in individual cells. Among these methods, PCA-based dimensionality reduction yields better results than CCA (Fig. 9b, right). Notably, in scenarios with biological variation embedded in batch differences, such as different clinical states, scMoGNN performs comparably to Seurat-based methods (Additional file 2: Fig. S7b, right), underscoring the advantages of higher-order topological relationships in complex batch differences.

In terms of sensitivity to training data size, we find that at the protein level, cTP-net, Seurat v4 (PCA), and Seurat v4 (CCA) are the most effective (Fig. 9c, left). In Seurat-based methods, PCA-based dimensionality reduction exhibits greater variability to training data size compared to CCA (Fig. 9c, left). At the cell level, the most effective methods are Seurat v4 (PCA), Seurat v4 (CCA), and Seurat v3 (PCA) (Fig. 9c, right). Among these, Seurat v4 (PCA) consistently demonstrates excellent performance across various training dataset sizes (Fig. 9c, right). In contrast, the performance of the remaining two methods exhibits relatively greater variability to training data size (Fig. 9c, right). Further analysis of the AICS evaluation results under varying ω_{ai} settings indicates that the results remain relatively stable when ω_{ai} exceeds 0.5, especially for the top-performing methods (Additional file 1: Tables S5, S6). The aforementioned evaluation results can assist users in considering the training data size when selecting methods.

In experiments with technical and biological differences, at the protein level, methods such as Seurat v4 (PCA) and Seurat v3 (PCA), which achieve excellent accuracy, also tend to be relatively robust (Fig. 9d, left and Additional file 2: Fig. S7b, left). However, exceptions exist, such as moETM, which exhibits high accuracy only in the scenarios of different samples and clinical states, resulting in less robust performance across all scenarios (Fig. 9d, left and Additional file 2: Fig. S7b, left). At the cell level, Seurat v3 (PCA), Seurat v3 (CCA), and Seurat v4 (PCA) outperform other methods and also consistently demonstrate superior accuracy across most scenarios (Fig. 9d, right and Additional file 2: Fig. S7b, right). Notably, while Seurat v4 (CCA) slightly outperforms Seurat v3 (CCA) in accuracy evaluations, it is less competitive in robustness assessments (Fig. 9d, right and Additional file 2: Fig. S7b, right). Further analysis of the RCS evaluation results under different ω_{ms} settings reveals that when ω_{ms} is greater than 0.5, the RCS evaluation results remain relatively stable, particularly for the top-performing methods (Additional file 1: Tables S7, S8). The robustness assessment results in experiments closely resembling real-world scenarios can serve as a supplementary guide for users when selecting methods for specific scenarios.

Regarding usability, we first evaluate efficiency based on running time and memory usage. We find that cTP-net and scMoGNN, despite high accuracy, are less efficient in terms of time and memory (Fig. 9e, left and middle and Additional file 1: Tables S3, S4). Conversely, among the methods with relatively excellent accuracy performance, moETM is the most time-efficient and exhibits the least variability to training data size (Fig. 9e, left and Additional file 1: Table S3). Seurat-based methods are the most memory-efficient and show the less variability to training data size (Fig. 9e, middle and Additional file 1: Table S4). However, they have longer running times compared to some deep learning-based methods, and the running time increases significantly with the growth of the training data size. Regarding popularity, Seurat-based methods dominate, likely due to Seurat's multifunctional suite for single-cell data analyses (Fig. 9e, right and Additional file 1: Table S9). In terms of user-friendliness, the Seurat-based methods are also leading, followed by TotalVI and sciPENN (Fig. 9e, right, Additional file 1: Table S10). These three methods consistently achieve high scores across the aspects of installation, code, and tutorial, whereas other methods exhibit more issues in one or more of these aspects.

Upon comprehensive evaluation, Seurat-based methods, particularly Seurat v4 (PCA) and Seurat v3 (PCA), emerge as the most favorable options, demonstrating superior accuracy and robustness across diverse experiments, and showing relative insensitivity to training data size. Their ability to handle various sources of single-cell data effectively, while maintaining memory efficiency and user-friendly features, makes them top choices for the surface protein expression imputation task. However, they exhibit longer running times compared to some deep learning-based methods, highlighting scalability concerns and underscoring the necessity for future enhancements to effectively manage larger datasets.

Decision-tree-style guidance scheme for method selection

Furthermore, we provide users scenario-specific method recommendations in the form of a decision tree (Fig. 10). This concise and intuitive scheme is designed to help users in identifying the most suitable methods for each specific scenario. Each branch of the decision tree

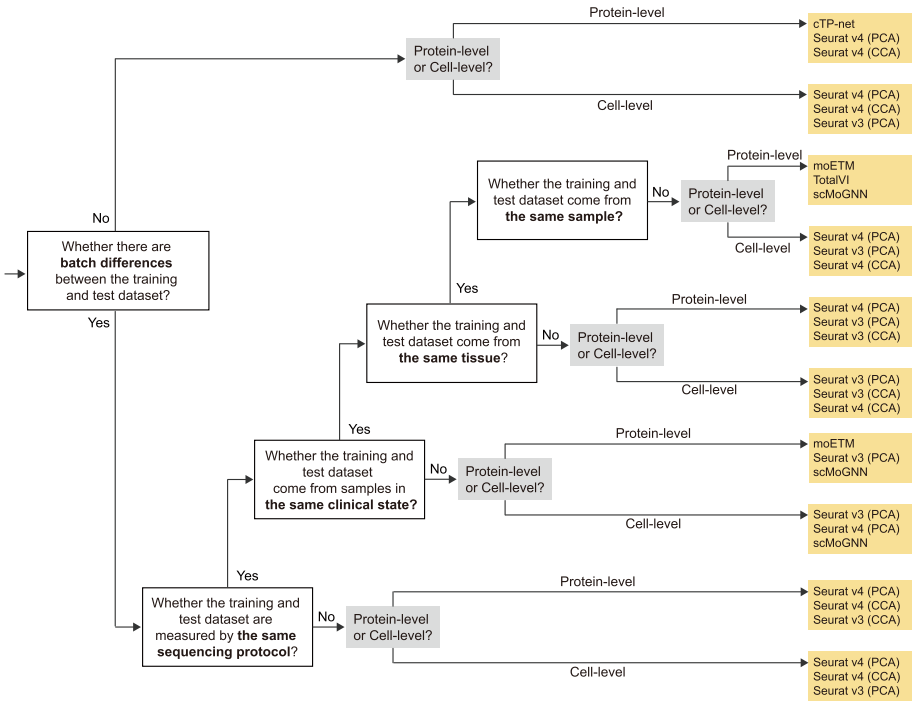

Fig. 10 Decision-tree-style guidance scheme for method selection in each scenario. Each branch represents a distinct experimental scenario in our study and is further divided into protein-level and cell-level analyses. For both levels, we recommend the top three performing methods for each scenario based on the evaluation results of ARS, providing tailored, scenario-specific guidance for method selection. From top to bottom, the results correspond to scenario 1, scenario 3, scenario 4, scenario 5, and scenario 6, respectively

represents a distinct experimental scenario evaluated in our study. For each scenario, we recommend three methods based on ARS evaluation results for both the protein and cell levels (as described in Additional file 2: Fig. S7), catering to diverse downstream experimental needs.

As shown in our overall evaluation results (Fig. 9), Seurat v4 (PCA) and Seurat v3 (PCA) are the recommended methods in most scenarios. However, exceptions exist in certain cases, highlighting that some methods perform better in specific scenarios, thus expanding the range of choices available to users. For example, when prioritizing protein-level accuracy, cTP-net is the most recommended method in scenario without batch differences. In scenario with different samples, moETM, TotalVI, and scMoGNN are recommended, while in scenario with different clinical states, moETM and scMoGNN are similarly preferred. When prioritizing cell-level accuracy, we also recommend scMoGNN in scenario involving different clinical states. In addition to the scenario-based method selection guidance scheme, we also provide a summary table in Additional file 1: Table S11, outlining the imputation strategy, strengths, weaknesses, and recommended application scenarios of each method, to help users better understand the differences between the methods.

Discussion

The emergence of CITE-seq and REAP-seq technologies has revolutionized our understanding of cellular heterogeneity by enabling simultaneous profiling of gene expression and surface protein expression at the single-cell level. However, widespread adoption of

these technologies is hampered by technical challenges and high costs, leading to the limited availability of publicly accessible datasets for studying complex tissues. Leveraging machine learning methods to impute surface proteomic data from transcriptomic data presents a promising solution to this challenge, enabling the acquisition of paired multimodal datasets for comprehensive analysis. Despite the development of various computational methods for surface protein data imputation, a comprehensive evaluation of their performance remains elusive. In this benchmark study, we bridge this gap by assessing twelve state-of-the-art imputation methods across accuracy, sensitivity to training data size, robustness across experiments, and usability.

Our findings unveil several key insights. Seurat-based methods, particularly Seurat v4 (PCA) and Seurat v3 (PCA), consistently exhibit competitive performance at both protein and cell levels (Fig. 9b and Additional file 2: Fig. S7b). In contrast, while other methods may excel at one level, their performance tends to falter at the other, with varying outcomes across different scenarios (Additional file 2: Fig. S7b). Sensitivity analysis reveals that Seurat-based methods are relatively insensitive to variations in training data size (Figs. 2, 9c), whereas other deep learning-based methods, such as scMM, scMOG, TotalVI, and moETM, display higher sensitivity to reductions in training data size (Figs. 2, 9c). Additionally, Seurat-based methods, particularly Seurat v4 (PCA) and Seurat v3 (PCA), demonstrate robustness across different experiments with technical and biological differences (Fig. 9d). Furthermore, efficiency analysis highlights moETM and Seurat-based methods as the most efficient and least variable options for time and memory, respectively, among the methods with relatively excellent accuracy performance (Figs. 8, 9e and Additional file 1: Tables S3, S4), making them appealing choices for practical applications. Overall, our findings underscore the exceptional performance of Seurat-based methods, particularly Seurat v4 (PCA) and Seurat v3 (PCA), across multiple metrics, coupled with their popularity and user-friendly features.

While the results presented in this study are based on datasets with available surface protein ground truth for performance evaluation, we also conduct exploratory analyses on scenarios lacking ground truth. In the absence of ground truth, evaluating the validity of the imputed protein expression presents a challenge. To address this, we examine whether the clustering structure of cells is preserved between the transcriptomic and imputed proteomic data. In extensive experiments conducted without ground truth, we evaluate the consistency between the clustering derived from imputed proteomic data and transcriptomic data using the Adjusted Rand Index (ARI) (see Additional file 2: Supplementary note 1 for details). The findings reveal that Seurat-based methods consistently achieve high clustering concordance across the majority of datasets, while other methods exhibit greater variability in performance, indicating a lack of stability (Additional file 2: Figs. S8–S22). In the absence of surface protein ground truth, the validation results are consistent with those from the previous benchmark results, further underscoring the effectiveness of Seurat-based methods.

However, we also note that Seurat-based methods, particularly those relying on Seurat v4, tend to exhibit longer running times compared to some deep learning-based methods, such as moETM (Figs. 8b, 9e, left and Additional file 1: Table S3). Furthermore, their running time increment relative to training data size is also comparatively larger (Figs. 8b, 9e, left and Additional file 1: Table S3), indicating potential scalability

challenges with larger datasets. As datasets continue to grow exponentially, reaching sizes of millions or even larger, the feasibility of using Seurat-based methods may become limited. Therefore, there is an urgent need to enhance these methods to effectively handle large datasets [35]. Additionally, the relatively less competitive performance of deep learning-based methods may partly result from insufficiently large training datasets. Addressing this limitation could involve developing more efficient and effective deep learning-based methods through pretraining and fine-tuning. For instance, pretraining on large-scale scRNA-seq data using self-supervised learning, followed by fine-tuning using paired data generated from CITE-seq and REAP-seq, could be a viable approach. One potential avenue is to adapt large language models like scGPT [36] and Geneformer [37], pretrained on extensive scRNA-seq data, to predict surface protein expression based on gene expression data.

Conclusions

In this study, we comprehensively evaluate twelve state-of-the-art imputation methods for surface protein expression, emphasizing accuracy, sensitivity to training data size, robustness across experiments, and usability. Seurat-based methods, particularly Seurat v4 (PCA) and Seurat v3 (PCA), stand out as the best performers, demonstrating competitive accuracy and robustness across experiments, and showing relative insensitivity to training dataset size, with memory-efficient and user-friendly features. However, these methods exhibit longer running times compared to certain deep learning-based approaches, highlighting scalability concerns and underscoring the necessity for future enhancements to manage larger datasets effectively.

Methods

Dataset collection and quality control

In this study, we employ eleven publicly available datasets for our benchmark analysis, each meticulously selected from reputable sources to ensure reliability and relevance. In addition, we select transcriptomic data of human peripheral blood mononuclear cells generated by seven different single-cell and single-nucleus RNA-sequencing (scRNA-seq and snRNA-seq) technologies from a systematic study to evaluate the imputation performance of methods in the absence of surface protein ground truth [38] (see Additional file 2: Supplementary note 2 for details about the datasets). The datasets are named following a standardized convention that includes the sequencing technologies, tissues, and authors involved. These datasets encompass CITE-PBMC-Stoeckius [5], CITE-CBMC-Stoeckius [5], CITE-BMMC-Stuart [12], CITE-PBMC-Li [26, 27], CITE-SLN111-Gayoso [18], CITE-SLN208-Gayoso [18], CITE-PBMC-Haniffa [30], CITE-PBMC-Sanger [31], CITE-PBMC10K-10X [32], CITE-PBMC5K-10X [33], REAP-PBMC-Peterson [6], CEL-PBMC-Ding [38], Drop-PBMC-Ding [38], inDrops-PBMC-Ding [38], SeqWell-PBMC-Ding [38], Smart-PBMC-Ding [38], 10xV2-PBMC-Ding [38], and 10xV3-PBMC-Ding [38].

For the CITE-PBMC-Stoeckius and CITE-CBMC-Stoeckius datasets, which are generated from species-mixing experiments, we isolate human cells by filtering the datasets to include only those with more than 90% of UMI counts mapped to human genes [5]. Subsequently, we remove low-quality genes (fewer than 10 counts across all cells)

and low-quality cells (fewer than 200 genes detected) [14]. These criteria are adopted from the original article and cTP-net [5, 14]. For the CITE-SLN111-Gayoso and CITE-SLN208-Gayoso datasets, which have isotype control antibodies and hashtag antibodies in their panels, we remove these antibodies in accordance with the original article [18]. Quality control procedures for the REAP-PBMC-Peterson dataset adhere to the criteria outlined in the original article and cTP-net [6, 14]. Initially, we filter out cells with high mitochondrial gene expression (more than 20% counts from mitochondrial genes) and fewer than 250 genes detected [6]. This is followed by the exclusion of low-quality genes (fewer than 10 counts across all cells) [14]. For scRNA-seq and snRNA-seq datasets, we filter out low-quality genes within each experimental batch (see Additional file 2: Supplementary note 2), defined as those with fewer than 5 counts across all cells in the CEL-PBMC-Ding, SeqWell-PBMC-Ding (Experiment2), and Smart-PBMC-Ding datasets, or fewer than 10 in other datasets. For the remaining datasets, we utilize preprocessed data provided directly by the authors, ensuring consistency and reliability in our analysis. Detailed summaries of the datasets after quality control are presented in Additional file 1: Table S1.

Method implementing details

Seurat [12, 13]. We follow the tutorial on https://satijalab.org/seurat/articles/multimodal_reference_mapping. This tutorial is based on Seurat v4, with the preprocessing part for gene expression data using the *SCTransform* function. We also conduct experiments using the preprocessing steps described in the Seurat v3 paper [12]. When performing dimensionality reduction of the gene expression data, both canonical correlation analysis (CCA) and principal component analysis (PCA) are recommended [12]. We consider these two cases when conducting our experiments. We set the *reduction* parameter to *cca* or *pcaproject* in the *FindTransferAnchors* function. We use the *TransferData* function to transfer the surface protein data from the training dataset to the test dataset. We use the default settings for all other parameters. These four different methods are named Seurat v3 (CCA), Seurat v3 (PCA), Seurat v4 (CCA), and Seurat v4 (PCA).

cTP-net [14]. cTP-net consists of two steps. First, it uses SAVER-X to denoise the raw gene expression data and then predicts surface protein expression using the proposed cTP-net model. We follow the guidelines on the GitHub repository of SAVER-X (https://github.com/jingshuw/SAVERX) for denoising the raw gene expression data [34]. After that, we use the code from https://github.com/zhouzilu/cTPnet/blob/master/extdata/training_05152020.py to learn the prediction model. We use the default settings for all parameters.

sciPENN [15]. We follow the tutorial provided on the GitHub repository of sciPENN: https://github.com/jlakkis/sciPENN. For experiments containing batch information within the training and test datasets, we pass the batch key information to the parameters *train_batchkeys* and *test_batchkey* of the *sciPENN_API*. We use the default settings for all other parameters.

scMOG [16]. We use the code available at https://github.com/GaoLabXDU/scMOG/blob/main/scMOG_code/bin/train_protein.py to train the model, and then utilize the code from https://github.com/GaoLabXDU/scMOG/blob/main/scMOG_

code/bin/predict-protein.py for imputing the test dataset. All parameters are set to their default values.

scMoGNN [17]. We follow the tutorial available at https://github.com/openproblems-bio/neurips2021-notebooks/blob/main/notebooks/templates/NeurIPS_CITE_GEX_analysis.ipynb to preprocess the data [39]. Subsequently, we utilize the code from https://github.com/OmicsML/dance/blob/main/examples/multi_modality/predict_modality/scmogcn.py [40] for imputing surface protein expression. When dealing with experiments containing batch information within the training and test datasets, we set the parameter *no_batch_features* to *False*. Otherwise, we set it to *True*. All other parameters are kept at their default settings.

TotalVI [18]. We follow the tutorial provided on the scvi-tools website: https://docs.scvi-tools.org/en/stable/tutorials/notebooks/multimodal/cite_scrna_integration_w_totalVI.html [41]. For experiments containing batch information within the training and test datasets, we pass the batch key information to the parameter *batch_key* in both the *sc.pp.highly_variable_genes* and *scvi.model.TOTALVI.setup_anndata* functions. Following the solution provided on https://github.com/scverse/scvi-tools/issues/1281, in some experiments conducted in scenario 2, we adjust the parameter *lr* to 4×10^{-4} in the *model.train* function. These experiments include replicate experiments 1, 3, and 4 under the down-sampling rate of 90%, replicate experiments 3, 4, and 5 under the down-sampling rate of 80%, replicate experiment 4 under the down-sampling rate of 50% in the CITE-BMMC-Stuart dataset, and all replicate experiments under the down-sampling rate of 0% in the CITE-PBMC-Stoeckius dataset. All other parameters are set to their default values.

Babel [19]. We follow the preprocessing steps in the original paper [19]. Subsequently, we follow the tutorial on https://github.com/OmicsML/dance-tutorials/blob/main/dance_tutorial.ipynb to learn the prediction model [40]. When the down-sampling rate of the CITE-PBMC-Stoeckius and CITE-CBMC-Stoeckius datasets is 90% in scenario 2, or when the training data rate of these two datasets is 10% in the "Evaluating usability in terms of time and memory" section, we adjust the parameter *batchsize* to 32. All other parameters are kept at their default settings.

moETM [20]. We utilize the code from https://github.com/manqizhou/moETM/blob/main/dataloader.py to preprocess the data. Subsequently, we use the code from https://github.com/manqizhou/moETM/blob/main/main_cross_prediction_rna_protein.py for imputations. For experiments containing batch information within the training and test datasets, we incorporate this batch key information as additional inputs. All other parameters are kept at their default settings.

scMM [21]. We implement scMM using the code from https://github.com/OmicsML/dance/blob/main/examples/multi_modality/predict_modality/scmm.py [40]. Following the solution provided at https://github.com/scverse/scanpy/issues/1504, when the down-sampling rate of the CITE-PBMC-Stoeckius datasets is 90% in scenario 2, or the training data rate of this dataset is 10% in the "Evaluating usability in terms of time and memory" section, we set the parameter *span* to 0.5 in the *sc.pp.highly_variable_genes* to select the highly variable genes. All other parameters are kept at their default settings.

Benchmark metrics

Metrics for evaluating accuracy of methods

We devise a comprehensive assessment framework to quantitatively evaluate the accuracy performance of methods, encompassing three pivotal metrics: Pearson correlation coefficient (PCC), root mean square error (RMSE), and average rank score (ARS).

PCC. PCC (Pearson correlation coefficient) gauges the degree of correlation between the predicted values and the ground truth. At the protein level, it is calculated as:

$$r_p = \frac{\sum_{i=1}^{N}\left(\hat{Y}_{ip} - \hat{\mu}_p\right)\left(Y_{ip} - \mu_p\right)}{\sqrt{\sum_{i=1}^{N}\left(\hat{Y}_{ip} - \hat{\mu}_p\right)^2} \cdot \sqrt{\sum_{i=1}^{N}\left(Y_{ip} - \mu_p\right)^2}} \tag{1}$$

where $\hat{Y}_{ip}$ and Y_{ip} represent the predicted and true expressions of protein p in cell i, respectively. Similarly, $\hat{\mu}_p$ and μ_p denote the mean predicted and true expressions across all cells for protein p respectively, with N denoting the total number of cells. Additionally, we evaluate the correlation at the cell level, denoted as r_i, which is calculated as:

$$r_i = \frac{\sum_{p=1}^{P}\left(\hat{Y}_{ip} - \hat{\mu}_i\right)\left(Y_{ip} - \mu_i\right)}{\sqrt{\sum_{p=1}^{P}\left(\hat{Y}_{ip} - \hat{\mu}_i\right)^2} \cdot \sqrt{\sum_{p=1}^{P}\left(Y_{ip} - \mu_i\right)^2}} \tag{2}$$

where $\hat{\mu}_i$ and μ_i represent the mean predicted and true expressions across all proteins for cell i respectively, and P represents the total number of proteins.

RMSE. RMSE (root mean square error) quantifies the absolute difference in numerical magnitude between the predicted values and the ground truth. At the protein level, we initially standardize the predicted and true expressions using Z-score transformation for comparability. RMSE for protein p is then defined as:

$$e_p = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(\hat{Y}'_{ip} - Y'_{ip}\right)^2} \tag{3}$$

where $\hat{Y}'_{ip}$ and Y'_{ip} represent the Z-score standardized predicted and true expressions of protein p in cell i, respectively. We also compute RMSE at the cell level after performing ℓ_2 normalization across proteins for each cell, which is defined as:

$$e_i = \sqrt{\frac{1}{P}\sum_{p=1}^{P}\left(\hat{Y}''_{ip} - Y''_{ip}\right)^2} \tag{4}$$

where $\hat{Y}''_{ip}$ and Y''_{ip} represent the ℓ_2 normalized predicted and true expressions of protein p in cell i, respectively.

ARS. We introduce ARS (average rank score) to conduct a comprehensive evaluation of methods, incorporating the aforementioned metrics. In each experiment, we calculate the four metrics for methods (PCC and RMSE values calculated respectively at the protein and cell levels), and rank the methods accordingly based on the median values of these metrics, where a method with better performance is assigned a higher rank score value. Given the

rank scores based on PCC (denoted as PCC_RS) and RMSE (denoted as RMSE_RS), we define the ARS as follows:

$$\mathrm{ARS}(\mathrm{PCC\_RS},\ \mathrm{RMSE\_RS}) = \frac{1}{2}(\mathrm{PCC\_RS} + \mathrm{RMSE\_RS}) \tag{5}$$

Specifically, based on the rank scores $\mathrm{PCC\_RS}_{\mathrm{protein}}$ and $\mathrm{RMSE\_RS}_{\mathrm{protein}}$ at the protein level, we can obtain the ARS at the protein level as follows:

$$\mathrm{ars}_{\mathrm{protein}} = \mathrm{ARS}\left(\mathrm{PCC\_RS}_{\mathrm{protein}}, \mathrm{RMSE\_RS}_{\mathrm{protein}}\right) \tag{6}$$

Similarly, we can obtain the ARS at the cell level as follows:

$$\mathrm{ars}_{\mathrm{cell}} = \mathrm{ARS}(\mathrm{PCC\_RS}_{\mathrm{cell}}, \mathrm{RMSE\_RS}_{\mathrm{cell}}) \tag{7}$$

where $\mathrm{PCC\_RS}_{\mathrm{cell}}$ and $\mathrm{RMSE\_RS}_{\mathrm{cell}}$ are the rank scores of methods for PCC and RMSE metrics at the cell level, respectively. A higher ARS value indicates better accuracy performance across all metrics in the experiment.

Metrics for evaluating the influences of training data size variations

In evaluating the influences of training data size variations on methods' accuracy performance, running time, and memory usage, we introduce the mean to evaluate methods in terms of average accuracy or efficiency, and the increment to assess methods in terms of variability. Additionally, in assessing the influences on methods' accuracy performance, i.e., the sensitivity of methods to training data size, we propose the average-increment composite score (AICS) as a comprehensive measure that considers both average accuracy and variability to reflect the effectiveness of methods.

Means of accuracy performance. We introduce means of accuracy performance to assess the average accuracy of methods across all training data sizes. In dataset d from scenario 2, for each down-sampling rate π (where π ranges from 0 to 90% in increments of 10%), $\mathrm{PCC}^{d}_{\mathrm{protein}}(\pi)$ and $\mathrm{RMSE}^{d}_{\mathrm{protein}}(\pi)$ represent the median PCC and RMSE values across five replicate experiments at the protein level, respectively. The means of accuracy performance based on PCC and RMSE are defined as:

$$\overline{\mathrm{PCC}}^{d}_{\mathrm{protein}} = \frac{1}{10}\sum_{\pi}\mathrm{PCC}^{d}_{\mathrm{protein}}(\pi) \tag{8}$$

$$\overline{\mathrm{RMSE}}^{d}_{\mathrm{protein}} = \frac{1}{10}\sum_{\pi}\mathrm{RMSE}^{d}_{\mathrm{protein}}(\pi) \tag{9}$$

Similarly, for the median PCC and RMSE values across five replicate experiments at the cell level, we can calculate the mean values in dataset d from scenario 2, denoted as:

$$\overline{\mathrm{PCC}}^{d}_{\mathrm{cell}} = \frac{1}{10}\sum_{\pi}\mathrm{PCC}^{d}_{\mathrm{cell}}(\pi) \tag{10}$$

$$\overline{\mathrm{RMSE}}^{d}_{\mathrm{cell}} = \frac{1}{10}\sum_{\pi}\mathrm{RMSE}^{d}_{\mathrm{cell}}(\pi) \tag{11}$$

A higher mean value based on PCC or a lower mean value based on RMSE indicates better performance in terms of PCC or RMSE across all training data sizes in dataset d.

Means of running time and memory usage. The means of running time ($\bar{T}$) and memory usage ($\bar{M}$) evaluate efficiency across all training data rates. For each rate θ (where θ is equivalent to 1 minus the down-sampling rate π in scenario 2, ranging from 10 to 100% in increments of 10%), $T(\theta)$ and $M(\theta)$ represent the running time and memory usage, respectively. The means of running time and memory usage are computed as:

$$\bar{T} = \frac{1}{10}\sum_{\theta} T(\theta) \tag{12}$$

$$\bar{M} = \frac{1}{10}\sum_{\theta} M(\theta) \tag{13}$$

A lower mean value indicates more efficiency in terms of time or memory.

Increments of accuracy performance. We introduce increments of accuracy performance to assess the variability of methods to training data size in terms of accuracy. In dataset d from scenario 2, $\Delta_{\mathrm{PCC}^d_{\mathrm{protein}}}$ and $\Delta_{\mathrm{RMSE}^d_{\mathrm{protein}}}$ represent the increments based on PCC and RMSE, respectively. They are defined as the sum of the absolute differences over all adjacent down-sampling rates:

$$\Delta_{\mathrm{PCC}^d_{\mathrm{protein}}} = \sum_{\pi'} \left|\mathrm{PCC}^d_{\mathrm{protein}}(\pi' - 10) - \mathrm{PCC}^d_{\mathrm{protein}}(\pi')\right| \tag{14}$$

$$\Delta_{\mathrm{RMSE}^d_{\mathrm{protein}}} = \sum_{\pi'} \left|\mathrm{RMSE}^d_{\mathrm{protein}}(\pi') - \mathrm{RMSE}^d_{\mathrm{protein}}(\pi' - 10)\right| \tag{15}$$

where π' and $\pi' - 10$ are the down-sampling rates, and $\pi' \in \{10\%, 20\%, \ldots, 90\%\}$. Similarly, we calculate the increment values at the cell level as:

$$\Delta_{\mathrm{PCC}^d_{\mathrm{cell}}} = \sum_{\pi'} \left|\mathrm{PCC}^d_{\mathrm{cell}}(\pi' - 10) - \mathrm{PCC}^d_{\mathrm{cell}}(\pi')\right| \tag{16}$$

$$\Delta_{\mathrm{RMSE}^d_{\mathrm{cell}}} = \sum_{\pi'} \left|\mathrm{RMSE}^d_{\mathrm{cell}}(\pi') - \mathrm{RMSE}^d_{\mathrm{cell}}(\pi' - 10)\right| \tag{17}$$

A lower increment value indicates less variability of accuracy performance in terms of PCC or RMSE to training data size in dataset d.

Increments of running time and memory usage. The increments of running time (Δ_{time}) and memory usage (Δ_{memory}) measure the variability of methods to training data rate in terms of time and memory. They are defined as the sum of the absolute differences over all adjacent training data rates:

$$\Delta_{\mathrm{time}} = \sum_{\theta'} \left|T(\theta') - T(\theta' - 10)\right| \tag{18}$$

$$\Delta_{\text{memory}} = \sum_{\theta'} \left| M(\theta') - M(\theta' - 10) \right| \tag{19}$$

where θ' and $\theta' - 10$ are the training data rates, and $\theta' \in \{20\%, 30\%, \ldots, 100\%\}$. A lower increment value indicates less variability to training data size in terms of time or memory.

Rank score of means of accuracy performance. To consolidate the means of accuracy performance based on PCC and RMSE, as well as the results for different datasets in scenario 2, we introduce the rank score of means of accuracy performance. Firstly, at the protein level, for the dataset d in scenario 2, we rank the methods accordingly based on the $\overline{\text{PCC}}^{d}_{\text{protein}}$ and $\overline{\text{RMSE}}^{d}_{\text{protein}}$, where a method with better performance is assigned a higher rank score value, denoted as $\text{PCC\_RS}^{\text{mean}}_{\text{protein}_d}$ and $\text{RMSE\_RS}^{\text{mean}}_{\text{protein}_d}$, respectively. Subsequently, we can obtain the ARS based on these rank scores. Next, we average the ARS values across all datasets in this scenario, denoted as $\overline{\text{ars}}^{\text{mean}}_{\text{protein}}$, which is defined as:

$$\overline{\text{ars}}^{\text{mean}}_{\text{protein}} = \frac{1}{|D|} \sum_{d} \text{ARS}\left(\text{PCC\_RS}^{\text{mean}}_{\text{protein}_d}, \text{RMSE\_RS}^{\text{mean}}_{\text{protein}_d}\right) \tag{20}$$

where d represents the datasets used in scenario 2: CITE-PBMC-Stoeckius, CITE-CBMC-Stoeckius, and CITE-BMMC-Stuart, and $|D|$ denotes the total number of datasets, equal to 3 here. Similarly, we calculate the mean of ARS values across all datasets in this scenario at the cell level:

$$\overline{\text{ars}}^{\text{mean}}_{\text{cell}} = \frac{1}{|D|} \sum_{d} \text{ARS}\left(\text{PCC\_RS}^{\text{mean}}_{\text{cell}\ d}, \text{RMSE\_RS}^{\text{mean}}_{\text{cell}\ d}\right) \tag{21}$$

where $\text{PCC\_RS}^{\text{mean}}_{\text{cell}\ d}$ and $\text{RMSE\_RS}^{\text{mean}}_{\text{cell}\ d}$ are the rank scores of $\overline{\text{PCC}}^{d}_{\text{cell}}$ and $\overline{\text{RMSE}}^{d}_{\text{cell}}$, respectively. Finally, we rank the methods accordingly based on the $\overline{\text{ars}}^{\text{mean}}_{\text{protein}}$ and $\overline{\text{ars}}^{\text{mean}}_{\text{cell}}$, where a method with higher ARS value is assigned a higher rank score value, to obtain the rank scores of means of accuracy performance, which are denoted as $\text{MEAN\_RS}_{\text{protein}}$ and $\text{MEAN\_RS}_{\text{cell}}$, respectively. A higher rank score of means value indicates better average accuracy performance across all training data sizes and datasets in scenario 2.

Rank score of increments of accuracy performance. Similarly, we introduce the rank score of increments of accuracy performance to consolidate the increments based on PCC and RMSE across different datasets in scenario 2. Firstly, at the protein level, for the dataset d in scenario 2, we rank the methods accordingly based on the $\Delta_{\text{PCC}^{d}_{\text{protein}}}$ and $\Delta_{\text{RMSE}^{d}_{\text{protein}}}$, where a method with lower increments is assigned a higher rank score value, denoted as $\text{PCC\_RS}^{\Delta}_{\text{protein}_d}$ and $\text{RMSE\_RS}^{\Delta}_{\text{protein}_d}$, respectively. Subsequently, we can obtain the ARS based on these rank scores. Next, we average the ARS values across all datasets in this scenario, denoted as $\overline{\text{ars}}^{\Delta}_{\text{protein}}$, which is defined as:

$$\overline{\text{ars}}^{\Delta}_{\text{protein}} = \frac{1}{|D|} \sum_{d} \text{ARS}\left(\text{PCC\_RS}^{\Delta}_{\text{protein}_d}, \text{RMSE\_RS}^{\Delta}_{\text{protein}_d}\right) \tag{22}$$

Similarly, we calculate the mean of ARS values across all datasets in this scenario at the cell level:

$$\overline{\text{ars}}^{\Delta}_{\text{cell}} = \frac{1}{|D|}\sum_{d} \text{ARS}\left(\text{PCC\_RS}^{\Delta}_{\text{cell}d}, \text{RMSE\_RS}^{\Delta}_{\text{cell}d}\right) \tag{23}$$

where $\text{PCC\_RS}^{\Delta}_{\text{cell}d}$ and $\text{RMSE\_RS}^{\Delta}_{\text{cell}d}$ are the rank scores of $\Delta_{\text{PCC}^{d}_{\text{cell}}}$ and $\Delta_{\text{RMSE}^{d}_{\text{cell}}}$, respectively. Finally, we rank the methods accordingly based on the $\overline{\text{ars}}^{\Delta}_{\text{protein}}$ and $\overline{\text{ars}}^{\Delta}_{\text{cell}}$, where a method with higher ARS value is assigned a higher rank score value, to obtain the rank scores of increments of accuracy performance, which are denoted as $\Delta\_\text{RS}_{\text{protein}}$ and $\Delta\_\text{RS}_{\text{cell}}$, respectively. A higher rank score of increments value indicates less variability to training data size in terms of accuracy over all datasets in scenario 2.

AICS. To comprehensively assess the sensitivity of methods to training data size, we introduce AICS (average-increment composite score). This metric evaluates sensitivity by not only focusing on the variability of accuracy performance, but also considering the average accuracy performance, and is defined as the weighted sum of the rank scores of means and increments of accuracy performance:

$$\text{AICS}_{\text{protein}} = \omega_{\text{ai}}\text{MEAN\_RS}_{\text{protein}} + (1-\omega_{\text{ai}})\Delta\_\text{RS}_{\text{protein}} \tag{24}$$

$$\text{AICS}_{\text{cell}} = \omega_{\text{ai}}\text{MEAN\_RS}_{\text{cell}} + (1-\omega_{\text{ai}})\Delta\_\text{RS}_{\text{cell}} \tag{25}$$

where $\text{MEAN\_RS}_{\text{protein}}$ and $\Delta\_\text{RS}_{\text{protein}}$ are the rank scores of means and increments of accuracy performance at the protein level, respectively. $\text{MEAN\_RS}_{\text{cell}}$ and $\Delta\_\text{RS}_{\text{cell}}$ are the rank scores of means and increments of accuracy performance at the cell level, respectively. ω_{ai} is a weight to balance the rank scores of means and increments values, and is recommended to be greater than 0.5, with a default setting of 0.8 (see Additional file 1: Tables S5, S6 for evaluation results under different ω_{ai} settings ranging from 0 to 1 in steps of 0.1). A higher AICS value indicates more effectiveness across all training data sizes and datasets in scenario 2.

Metrics for evaluating robustness of methods

The robustness composite score (RCS) is employed to assess the robustness of methods' accuracy across experiments with technical and biological differences, which is calculated based on the ARS values from all such experiments, thereby indicating the robustness of accuracy under real-world-like conditions.

RCS. We introduce RCS (robustness composite score) to evaluate the robustness of ARS values of methods across different experiments with technical and biological differences. We calculate the mean and standard deviation of ARS values of methods across all these experiments and rank them accordingly. A method with a higher mean value or lower standard deviation value is assigned a higher rank score value. At the protein level, RCS is defined as:

$$\text{RCS}_{\text{protein}} = \omega_{\text{ms}}\text{ARS\_RS}^{\text{mean}}_{\text{protein}} + (1-\omega_{\text{ms}})\text{ARS\_RS}^{\text{std}}_{\text{protein}} \tag{26}$$

where $\text{ARS\_RS}^{\text{mean}}_{\text{protein}}$ and $\text{ARS\_RS}^{\text{std}}_{\text{protein}}$ denote the rank scores for the mean and standard deviation at the protein level, respectively. Similarly, we can calculate RCS at the cell level:

$$\mathrm{RCS}_{\mathrm{cell}} = \omega_{\mathrm{ms}}\mathrm{ARS\_RS}_{\mathrm{cell}}^{\mathrm{mean}} + (1-\omega_{\mathrm{ms}})\mathrm{ARS\_RS}_{\mathrm{cell}}^{\mathrm{std}} \quad (27)$$

where $\mathrm{ARS\_RS}_{\mathrm{cell}}^{\mathrm{mean}}$ and $\mathrm{ARS\_RS}_{\mathrm{cell}}^{\mathrm{std}}$ denote the rank scores for the mean and standard deviation at the cell level, respectively. ω_{ms} is a weight to balance the rank scores of mean and standard deviation values, and is recommended to be greater than 0.5, with a default setting of 0.8 (see Additional file 1: Tables S7, S8 for evaluation results under different ω_{ms} settings ranging from 0 to 1 in steps of 0.1). Note that, based on the definition of RCS, the robustness in this study is a comprehensive concept that considers both the stability and competitiveness of the methods. A higher RCS value indicates more robustness across different experiments with technical and biological differences.

Supplementary information

The online version contains supplementary material available at https://doi.org/10.1186/s13059-025-03514-9.

Additional file 1: Supplementary tables S1-S11.

Additional file 2: Supplementary notes 1-2 and Supplementary figures S1-S22.

Acknowledgements
Not applicable.

Peer review information
Andrew Cosgrove was the primary editor of this article and managed its editorial process and peer review in collaboration with the rest of the editorial team. The peer-review history is available in the online version of this article.

Authors' contributions
X.F.Z. conceives the study. C.Y.L. and Y.J.H. prepare all datasets and conduct all the experiments. All authors write the manuscript. X.F.Z. supervises the whole project. All authors read and approved the final manuscript.

Funding
This work was supported by the National Natural Science Foundation of China [12271198, 11871026, and 62377019] and Fundamental Research Funds for the Central Universities [CCNU24AI001, CCNU24JC004, and CCNU24JCPT027].

Data availability
The datasets analyzed during the current study are all publicly available. CITE-PBMC-Stoeckius and CITE-CBMC-Stoeckius datasets are both obtained from GSE100866 [42]. CITE-BMMC-Stuart dataset is obtained from GSE128639 [43]. CITE-PBMC-Li dataset is obtained from https://doi.org/10.5281/zenodo.7779017 [44]. CITE-SLN111-Gayoso and CITE-SLN208-Gayoso datasets are obtained from https://github.com/YosefLab/totalVI_reproducibility/blob/master/data/spleen_lymph_111.h5ad and https://github.com/YosefLab/totalVI_reproducibility/blob/master/data/spleen_lymph_206.h5ad, respectively [45]. CITE-PBMC-Haniffa and CITE-PBMC-Sanger datases are obtained from https://upenn.app.box.com/s/1p1f1gblge3rqgk97ztr4daagt4fsue5/file/854676700495 and https://upenn.app.box.com/s/1p1f1gblge3rqgk97ztr4daagt4fsue5/file/854919546303, respectively [46]. CITE-PBMC10K-10X and CITE-PBMC5K-10X datases are obtained from https://github.com/YosefLab/totalVI_reproducibility/blob/master/data/pbmc_10k_protein_v3.h5ad and https://github.com/YosefLab/totalVI_reproducibility/blob/master/data/pbmc_5k_protein_v3.h5ad, respectively [45]. REAP-PBMC-Stoeckius dataset is obtained from GSE100501 [47]. All scRNA-seq and snRNA-seq datasets are obtained from https://singlecell.broadinstitute.org/single_cell/study/SCP424/single-cell-comparison-pbmc-data?cluster=Harmony%20TSNE&spatialGroups=-- &annotation=Method--group--study&subsample=all#study-summary [48]. A summary of the data after quality control is shown in Additional file 1: Table S1. All datasets used in the manuscript have also been curated and uploaded to a public repository in Zenodo with a DOI assignment (DOI: https://doi.org/10.5281/zenodo.12725699) [49]. The datasets are stored according to the experiments. Each experiment folder contains datasets that have undergone quality control and an intersection of genes and proteins, making them directly usable as input data for the experiments. The scripts for executing each imputation method, including running the model and evaluating its performance, are available on GitHub (https://github.com/Zhangxf-ccnu/scProtein) under the MIT license [50] and Zenodo (https://doi.org/10.5281/zenodo.12725699) under the Creative Commons Attribution 4.0 International license (CC BY 4.0) [49].

Declarations

Ethics approval and consent to participate
Not applicable.

Consent for publication
Not applicable.

Competing interests
The authors declare no competing interests.

Received: 1 August 2024 Accepted: 24 February 2025
Published online: 04 March 2025

References

1. Vandereyken K, Sifrim A, Thienpont B, Voet T. Methods and applications for single-cell and spatial multi-omics. Nat Rev Genet. 2023;24:494–515.
2. Baysoy A, Bai Z, Satija R, Fan R. The technological landscape and applications of single-cell multi-omics. Nat Rev Mol Cell Biol. 2023;24:695–713.
3. Argelaguet R, Cuomo ASE, Stegle O, Marioni JC. Computational principles and challenges in single-cell data integration. Nat Biotechnol. 2021;39:1202–15.
4. Heumos L, Schaar AC, Lance C, Litinetskaya A, Drost F, Zappia L, et al. Best practices for single-cell analysis across modalities. Nat Rev Genet. 2023;24:550–72.
5. Stoeckius M, Hafemeister C, Stephenson W, Houck-Loomis B, Chattopadhyay PK, Swerdlow H, et al. Simultaneous epitope and transcriptome measurement in single cells. Nat Methods. 2017;14:865–8.
6. Peterson VM, Zhang KX, Kumar N, Wong J, Li L, Wilson DC, et al. Multiplexed quantification of proteins and transcripts in single cells. Nat Biotechnol. 2017;35:936–9.
7. Todorovic V. Single-cell RNA-seq-now with protein. Nat Methods. 2017;14:1028–9.
8. Haque A, Engel J, Teichmann SA, Lönnberg T. A practical guide to single-cell RNA-sequencing for biomedical research and clinical applications. Genome Med. 2017;9:75.
9. Ma A, McDermaid A, Xu J, Chang Y, Ma Q. Integrative methods and practical challenges for single-cell multi-omics. Trends Biotechnol. 2020;38:1007–22.
10. Liu Y, Beyer A, Aebersold R. On the dependency of cellular protein levels on mRNA abundance. Cell. 2016;165:535–50.
11. Abreu RDS, Penalva LO, Marcotte EM, Vogel C. Global signatures of protein and mRNA expression levels. Mol BioSyst. 2009;5:1512–26.
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. Hao Y, Hao S, Andersen-Nissen E, Mauck WM, Zheng S, Butler A, et al. Integrated analysis of multimodal single-cell data. Cell. 2021;184:3573-87.e29.
14. Zhou Z, Ye C, Wang J, Zhang NR. Surface protein imputation from single cell transcriptomes by deep neural networks. Nat Commun. 2020;11:651.
15. Lakkis J, Schroeder A, Su K, Lee MY, Bashore AC, Reilly MP, et al. A multi-use deep learning method for CITE-seq and single-cell RNA-seq data integration with cell surface protein prediction and imputation. Nat Mach Intell. 2022;4:940–52.
16. Lan M, Zhang S, Gao L. Efficient generation of paired single-cell multiomics profiles by deep learning. Adv Sci. 2023;10:2301169.
17. Wen H, Ding J, Jin W, Wang Y, Xie Y, Tang J. Graph neural networks for multimodal single-cell data integration. In Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining. New York: Association for Computing Machinery; 2022. p. 4153–63.
18. Gayoso A, Steier Z, Lopez R, Regier J, Nazor KL, Streets A, et al. Joint probabilistic modeling of single-cell multi-omic data with totalVI. Nat Methods. 2021;18:272–82.
19. Wu KE, Yost KE, Chang HY, Zou J. BABEL enables cross-modality translation between multiomic profiles at single-cell resolution. Proc Natl Acad Sci. 2021;118:e2023070118.
20. Zhou M, Zhang H, Bai Z, Mann-Krzisnik D, Wang F, Li Y. Single-cell multi-omics topic embedding reveals cell-type-specific and COVID-19 severity-related immune signatures. Cell Rep Methods. 2023;3:100563.
21. Minoura K, Abe K, Nam H, Nishikawa H, Shimamura T. A mixture-of-experts deep generative model for integrated analysis of single-cell multiomics data. Cell Rep Methods. 2021;1:100071.
22. Zhao BS, Roundtree IA, He C. Post-transcriptional gene regulation by mRNA modifications. Nat Rev Mol Cell Biol. 2016;18:31–42.
23. Jackson RJ, Hellen CUT, Pestova TV. The mechanism of eukaryotic translation initiation and principles of its regulation. Nat Rev Mol Cell Biol. 2010;11:113–27.
24. Mowen KA, David M. Unconventional post-translational modifications in immunological signaling. Nat Immunol. 2014;15:512–20.
25. Derrigo M, Cestelli A, Savettieri G, Liegro ID. RNA-protein interactions in the control of stability and localization of messenger RNA. Int J Mol Med. 2014;5:111–34.
26. Li SS, Kochar NK, Elizaga M, Hay CM, Wilson GJ, Cohen KW, et al. DNA priming increases frequency of T-cell responses to a vesicular stomatitis virus HIV vaccine with specific enhancement of $CD8^+$ T-cell responses by interleukin-12 plasmid DNA. Clin Vaccine Immunol. 2017;24:e00263-17.
27. Elizaga M, Li SS, Kochar NK, Wilson GJ, Allen MA, Tieu HVN, et al. Safety and tolerability of HIV-1 multiantigen pDNA vaccine given with IL-12 plasmid DNA via electroporation, boosted with a recombinant vesicular stomatitis virus HIV Gag vaccine in healthy volunteers in a randomized, controlled clinical trial. PLoS One. 2018;13:e0202753.
28. Lu L, Shen R, Broxmeyer HE. Stem cells from bone marrow, umbilical cord blood and peripheral blood for clinical application: current status and future application. Crit Rev Oncol Hematol. 1996;22:61–78.
29. Wu AG, Michejda M, Mazumder A, Meehan KR, Menendez FA, Tchabo JG, et al. Analysis and characterization of hematopoietic progenitor cells from fetal bone marrow, adult bone marrow, peripheral blood, and cord blood. Pediatr Res. 1999;46:163–9.
30. Stephenson E, Reynolds G, Botting RA, Calero-Nieto FJ, Morgan MD, Tuong ZK, et al. Single-cell multi-omics analysis of the immune response in COVID-19. Nat Med. 2021;27:904–16.

31. Ballestar E, Farber DL, Glover S, Horwitz B, Meyer K, Nikolić M, et al. Single cell profiling of COVID-19 patients: an international data resource from multiple tissues; 2020. Preprint at https://doi.org/10.1101/2020.11.20.20227355.
32. 10X Genomics. 10k PBMCs from a healthy donor-gene expression and cell surface protein. https://www.10xgenomics.com/datasets/10-k-pbm-cs-from-a-healthy-donor-gene-expression-and-cell-surface-protein-3-standard-3-0-0. Accessed 21 June 2023.
33. 10X Genomics. 5k peripheral blood mononuclear cells (PBMCs) from a healthy donor with cell surface proteins (v3 chemistry). https://www.10xgenomics.com/datasets/5-k-peripheral-blood-mononuclear-cells-pbm-cs-from-a-healthy-donor-with-cell-surface-proteins-v-3-chemistry-3-1-standard-3-1-0. Accessed 21 June 2023.
34. Wang J, Agarwal D, Huang M, Hu G, Zhou Z, Ye C, et al. Data denoising with transfer learning in single-cell transcriptomics. Nat Methods. 2019;16:875–8.
35. He Z, Hu S, Chen Y, An S, Zhou J, Liu R, et al. Mosaic integration and knowledge transfer of single-cell multimodal data with MIDAS. Nat Biotechnol. 2024;42:1594–605.
36. Cui H, Wang C, Maan H, Pang K, Luo F, Duan N, et al. scGPT: toward building a foundation model for single-cell multi-omics using generative AI. Nat Methods. 2024;21:1470–80.
37. Theodoris CV, Xiao L, Chopra A, Chaffin MD, Al Sayed ZR, Hill MC, et al. Transfer learning enables predictions in network biology. Nature. 2023;618:616–24.
38. Ding J, Adiconis X, Simmons SK, Kowalczyk MS, Hession CC, Marjanovic ND, et al. Systematic comparison of single-cell and single-nucleus RNA-sequencing methods. Nat Biotechnol. 2020;38:737–46.
39. Luecken MD, Burkhardt DB, Cannoodt R, Lance C, Agrawal A, Aliee H, et al. A sandbox for prediction and integration of DNA, RNA, and proteins in single cells. In: Thirty-fifth conference on neural information processing systems datasets and benchmarks track (round 2). San Diego: Neural Information Processing Systems Foundation, Inc.; 2021.
40. Ding J, Liu R, Wen H, Tang W, Li Z, Venegas J, et al. DANCE: a deep learning library and benchmark platform for single-cell analysis. Genome Biol. 2024;25:72.
41. Gayoso A, Lopez R, Xing G, Boyeau P, Valiollah Pour Amiri V, Hong J, et al. A Python library for probabilistic analysis of single-cell omics data. Nat Biotechnol. 2022;40:163–6.
42. Stoeckius M. CITE-seq: large scale simultaneous measuremnt of epitopes and transcriptomes in single cells. Gene Expr Omnibus. https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE100866. Accessed 30 Jun 2023.
43. Butler A, Stuart T. Comprehensive integration of single-cell data. Gene Expr Omnibus. https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE128639. Accessed 30 Jun 2023.
44. Satija Lab. PBMC CITE-seq reference (1.0.0) [Data set]. Zenodo. https://doi.org/10.5281/zenodo.7779017. Accessed 16 Aug 2023.
45. Gayoso, A, Steier Z, Grisaitis WC. totalVI_reproducibility. Github Repository. https://github.com/YosefLab/totalVI_reproducibility. Accessed 23 Dec 2023.
46. Li M. sciPENN_Data. https://upenn.app.box.com/s/1p1f1gblge3rqgk97ztr4daagt4fsue5. Accessed 16 Aug 2023.
47. Peterson V, Zhang K. The dynamics of cellular response to therapeutic perturbation using multiplexed quantification of the proteome and transcriptome at single-cell resolution. Gene Expression Omnibus. https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE100501. Accessed 30 Jun 2023.
48. Levin J. Single cell comparison: PBMC data. https://singlecell.broadinstitute.org/single_cell/study/SCP424/single-cell-comparison-pbmc-data?cluster=Harmony%20TSNE&spatialGroups=-- &annotation=Method--group--study &subsample=all#study-summary. Accessed 6 Dec 2024.
49. Li CY, Hong YJ, Li B, Zhang XF. Benchmarking single-cell cross-omics imputation methods for surface protein expression. Zenodo. 2024. https://doi.org/10.5281/zenodo.12725699. Accessed 11 Jul 2024.
50. Li CY, Hong YJ, Li B, Zhang XF. Benchmarking single-cell cross-omics imputation methods for surface protein expression. Github. https://github.com/Zhangxf-ccnu/scProtein. Accessed 24 Jul 2024.

Publisher's Note