nature methods

Analysis

https://doi.org/10.1038/s41592-024-02429-w

Benchmarking algorithms for single-cell multi-omics prediction and integration

Received: 14 April 2023

Accepted: 19 August 2024

Published online: 25 September 2024

Check for updates

Yinlei Hu[1,2,3,9], Siyuan Wan[1,2,4,9], Yuanhanyu Luo[5,6,9], Yuanzhe Li[1,2,4], Tong Wu[6,7], Wentao Deng[1,2], Chen Jiang[1,2], Shan Jiang[6], Yueping Zhang[4], Nianping Liu[8], Zongcheng Yang[1], Falai Chen[3,4] ✉, Bin Li[5,6] ✉ & Kun Qu[1,2,4,8] ✉

The development of single-cell multi-omics technology has greatly enhanced our understanding of biology, and in parallel, numerous algorithms have been proposed to predict the protein abundance and/or chromatin accessibility of cells from single-cell transcriptomic information and to integrate various types of single-cell multi-omics data. However, few studies have systematically compared and evaluated the performance of these algorithms. Here, we present a benchmark study of 14 protein abundance/chromatin accessibility prediction algorithms and 18 single-cell multi-omics integration algorithms using 47 single-cell multi-omics datasets. Our benchmark study showed overall totalVI and scArches outperformed the other algorithms for predicting protein abundance, and LS_Lab was the top-performing algorithm for the prediction of chromatin accessibility in most cases. Seurat, MOJITOO and scAI emerge as leading algorithms for vertical integration, whereas totalVI and UINMF excel beyond their counterparts in both horizontal and mosaic integration scenarios. Additionally, we provide a pipeline to assist researchers in selecting the optimal multi-omics prediction and integration algorithm.

In recent years, several single-cell multi-omics technologies have been developed, enabling the detection of multiple genomic information within a single cell. For instance, CITE-seq[1] and REAP-seq[2] can simultaneously detect RNA expression and surface protein abundance for a single cell; SHARE-seq[3], SNARE-seq[4] and 10x Multiome (https://www.10xgenomics.com/products/) can obtain RNA expression and chromatin accessibility information from the same cell. These technologies have enabled biologists to integrate multiple genomic information to study cellular function and development[5,6], unravel complex gene regulation mechanisms at the single-cell level[7,8] and predict cellular fate more precisely[9–11]. Multi-omics technologies have been applied to investigations of various tissues and organs, such as blood[2], skin[3] and brain[4,12]; however, most single-cell studies measure only the transcriptomic information of samples[13]. Owing to higher experimental costs and/or technical challenges, the usage of single-cell multi-omics technologies is currently not as extensive as that of single-cell single-omics technologies[13–16].

Moreover, current single-cell proteomic approaches face considerable challenges, notably, low throughput and substantial experimental costs. For instance, next-generation sequencing (NGS)-based methods like CITE-seq[1], scCUT&Tag[17] and ASAP-seq[18] typically capture less than 10% of the proteome expressed by any given cell. Although

[1]Department of Oncology, The First Affiliated Hospital of USTC, School of Basic Medical Sciences, Division of Life Sciences and Medicine, University of Science and Technology of China, Hefei, China. [2]Institute of Artificial Intelligence, Hefei Comprehensive National Science Center, Hefei, China. [3]School of Mathematical Science, University of Science and Technology of China, Hefei, China. [4]School of Artificial Intelligence and Data Science, University of Science and Technology of China, Hefei, China. [5]Tsinghua Institute of Multidisciplinary Biomedical Research, Tsinghua University, Beijing, China. [6]National Institute of Biological Sciences, Beijing, China. [7]College of Life Sciences, Beijing Normal University, Beijing, China. [8]School of Biomedical Engineering, Suzhou Institute for Advanced Research, University of Science and Technology of China, Suzhou, China. [9]These authors contributed equally: Yinlei Hu, Siyuan Wan, Yuanhanyu Luo. ✉e-mail: chenfl@ustc.edu.cn; libin@nibs.ac.cn; qukun@ustc.edu.cn

mass spectrometry-based techniques theoretically have the capacity to identify upwards of 8,000 proteins, they are prone to experimental artifacts[15]. These challenges highlight the essential need for predictive models capable of inferring comprehensive multi-omics information.

To tackle this issue, one strategy is to use a single-cell integration algorithm, such as Seurat[19] and LIGER[20], to map a single-cell RNA-sequencing (scRNA-seq) dataset onto a single-cell multi-omics dataset obtained from the same tissue or organ. Subsequently, one can identify the *k*-nearest-neighbor (KNN) cells for a given cell across various datasets in the mapping space and utilize the neighboring cells from multi-omics data to predict the protein abundance or chromatin accessibility of cells in scRNA-seq data. Another strategy is training a machine learning model with a multi-omics dataset and using the model to predict protein abundance and/or chromatin accessibility from scRNA-seq data, such as totalVI[6], scArches[21], LS_Lab[16], MultiVI[22], Guanlab-dengkw[16], sciPENN[23], BABEL[24], scVAEIT[25], cTP-net[13], scMOG[26], scMoGNN[27] and CMAE[28].

Another pivotal challenge in single-cell multi-omics data analysis is the development of robust and efficient computational strategies for data integration[29], which encompass a broad collection of algorithms tailored to specific tasks, including the association of different omics modalities, termed vertical integration, with algorithms such as Seurat, MOJITOO[30], totalVI, Multigrate[31], SCOIT[32], CiteFuse[33], DeepMAPS[34], scArches, scVAEIT, scAI[35], MultiVI, MOFA+[36], scMVP[37], MIRA[38] and Schema[39]. Others facilitate batch correction across multi-omics datasets, known as horizontal integration, exemplified by tools like totalVI, scArches, MultiVI, UINMF[40], MOFA+, Multigrate, scVAEIT, MIRA[38] and scMoMaT[41]. Additionally, some aim to integrate single-cell datasets sharing at least one type of omics information, a process known as mosaic integration, with algorithms including totalVI, scArches, scVAEIT, Multigrate, MultiVI, UINMF, scMoMaT and StabMap[42].

These single-cell multi-omics prediction and integration algorithms have substantially deepened our understanding of various biological and pathological processes[5,6,13,18,22–24,26,38,40,42–46]. However, there are few comprehensive studies that compare the performance of these algorithms. Here, we propose a framework to systematically benchmark the performance of 14 multi-omics prediction algorithms in predicting chromatin accessibility or protein abundance of cells from scRNA-seq data using 36 multi-omics datasets and 6 evaluation metrics (Fig. 1 and Supplementary Tables 1–3). We also performed a systematical benchmark on 18 multi-omics integration algorithms designed for vertical, horizontal or mosaic integration using 35 datasets (with cell-type labels) and ten evaluation metrics (Fig. 1 and Supplementary Tables 4–9). Additionally, we compared the performance of these algorithms using different batches of data for training and prediction and measured the computational resources consumed by each algorithm. Our benchmarking pipeline is available at https://github.com/QuKunLab/MultiomeBenchmarking/.

Results

Benchmarking framework for multi-omics prediction algorithms

In the benchmark study, we compared the performance of 14 algorithms in predicting multi-omics information. Eleven of the fourteen algorithms, including totalVI, scArches, Guanlab-dengkw, sciPENN, scMoGNN, Seurat, BABEL, scVAEIT, cTP-net, CMAE and LIGER, can be used to predict the protein abundance of cells from scRNA-seq data (Fig. 1a). We collected 25 single-cell RNA + protein datasets generated by using the CITE-seq, REAP-seq and DOGMA-seq technologies as the ground truths to evaluate the performance of the 11 protein prediction algorithms (Fig. 1b and Supplementary Tables 1 and 2). Notably, DOGMA-seq technology can simultaneously capture RNA expression, protein abundance and chromatin accessibility information for each cell. Nine of the fourteen algorithms can predict the chromatin accessibility of cells from scRNA-seq data, including LS_Lab, MultiVI, scVAEIT, LIGER, Seurat, BABEL, scMOG, scMoGNN and CMAE. To compare the performance of the 9 chromatin prediction algorithms, we collected 12 single-cell RNA + ATAC datasets obtained by SNARE-seq, SHARE-seq, ISSAAC-seq, 10x Multiome and DOGMA-seq technologies (Fig. 1c and Supplementary Tables 1 and 2).

Before processing the datasets with the 14 algorithms, we removed low-quality cells and rarely captured transcripts/proteins/chromatin fragments from each dataset according to the criteria of the data source papers (Supplementary Table 3). We then compared the performance of these algorithms in two scenarios: using one dataset (intra-dataset) and using two datasets from the same organ/tissue (inter-dataset). In the intra-dataset scenario, we randomly split the cells in the dataset into an 80% training set and a 20% test set, with an equal probability for each cell to be assigned to either set. In the inter-dataset scenario, we used one dataset as the training set and the RNA expression matrix of another dataset as the test set. We trained each algorithm with the training set and used each trained algorithm and the transcriptomics information of each test set to predict the protein abundances or chromatin accessibility of the cells in the test set.

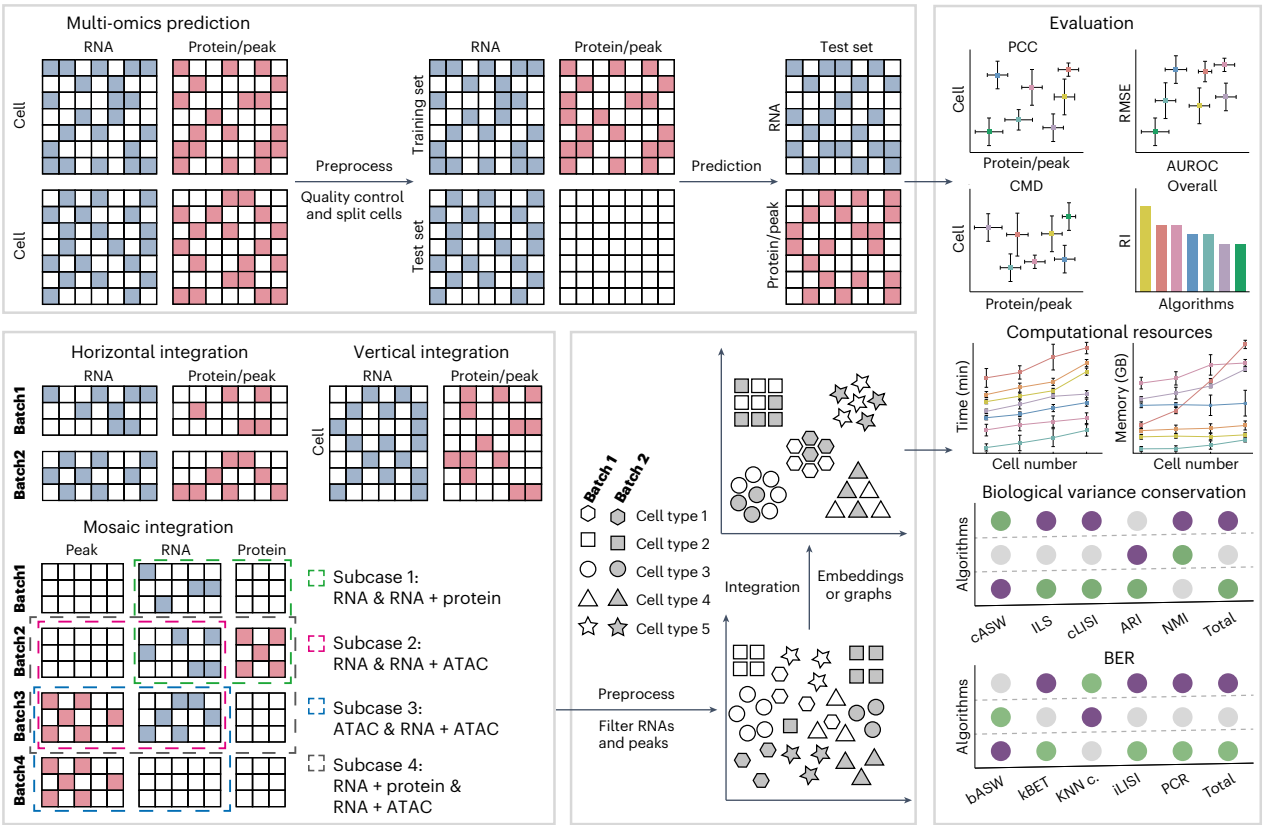
To evaluate the performance of each algorithm, we calculated the cell–cell and protein–protein Pearson correlation coefficients (PCCs) between the predicted and reference matrices for the single-cell RNA + protein datasets and the cell–cell and peak–peak PCCs for the single-cell RNA + ATAC datasets, where higher PCC values indicate higher prediction accuracy. In data from the assay for transposase-accessible chromatin with sequencing (ATAC-seq), 'peaks' refer to accessible DNA fragments. Given the binary nature of the reference matrices, we used the area under the receiver operating characteristic (AUROC) to evaluate the accuracy of the chromatin accessibility prediction algorithms, with a higher AUROC indicating superior performance. To assess the error of each algorithm, we computed two cell–cell correlation matrices of the test set, one from protein abundance or chromatin accessibility predicted by the algorithm and another from the reference data. Then, we used the difference between the two correlation matrices (that is, correlation matrix distance, CMD) as a representation of the prediction error[47–49]. We also calculated the CMDs of the protein–protein or peak–peak correlation matrices to represent the errors of these algorithms in predicting patterns of protein abundance or DNA accessibility. We also utilized the root mean square error (RMSE) to quantify the deviation between the predicted and reference values.

In addition, we defined a rank index (RI) to comprehensively consider each algorithm under six metrics (that is, cell–cell PCC, protein–protein PCC, cell–cell CMD, protein–protein CMD, AUROC and RMSE) and two scenarios (that is, intra-dataset and inter-dataset). An algorithm will be assigned a score of 1 if its PCC/AUROC value is higher than the median PCC/AUROC value of all algorithms or if its

Fig. 1 | Workflow and multi-omics datasets for benchmarking. **a**, Benchmarking workflow for 14 multi-omics prediction algorithms and 18 multi-omics integration algorithms, including 11 algorithms that can predict protein abundance, 9 algorithms that can predict chromatin accessibility, 15 algorithms for vertical integration, 9 algorithms for horizontal integration and 8 algorithms for mosaic integration. We adopted 6 and 10 metrics for the multi-omics prediction and integration algorithms, respectively, to evaluate the performance of these algorithms, and we also assessed their robustness and consumed computational resources. KNN c., KNN connectivity. **b**, 28 single-cell datasets that contain RNA expression and protein abundance information for each cell. **c**, 14 single-cell datasets that contain RNA expression and chromatin accessibility information for each cell. **d**, Five single-cell datasets that contain either RNA expression or chromatin accessibility information for each cell. These datasets were used for the assessment of mosaic integration algorithms. CBMCs, cord blood-derived mast cells; HSPCs, hematopoietic stem and progenitor cells.

https://doi.org/10.1038/s41592-024-02429-w

a



b

Dataset ID	Technique	Type/tissue	Species	Batch	Cells	RNA	Sparsity of RNA	Protein	Sparsity of protein
Dataset 1–2	CITE-seq	BMMCs	Human	14	96,847	15,481	0.93	139	0.10
Dataset 3		Brain immune cells	Human	11	85,000	22,950	0.92	16	0.21
Dataset 4		CBMCs	Human	1	8,613	36,280	0.96	13	0.00
Dataset 5		Glioblastomas	Human	3	21,589	14,131	0.83	268	0.48
Dataset 6		Glioblastomas	Mouse	2	24,559	12,411	0.81	174	0.72
Dataset 7		HSPCs	Mouse	2	10,473	18,162	0.81	38	0.89
Dataset 8		MALT tumor	Human	1	8,412	33,555	0.96	17	0.00
Dataset 9–10		Murine splenic myeloid cells	Mouse	2	27,693	30,456	0.92	11	0.00
Dataset 11		Naive brains	Mouse	1	13,052	19,848	0.88	34	0.31
Dataset 12–19		PBMCs	Human	40	684,251	31,400	0.95	278	0.09
Dataset 20		Peripheral memory T cells	Human	46	500,089	33,538	0.96	31	0.30
Dataset 21–22		Spleen and lymph nodes	Mouse	4	32,648	13,553	0.89	211	0.10
Dataset 23–24	REAP-seq	PBMCs	Human	2	7,488	32,738	0.98	48	0.61
Dataset 25–26	DOGMA-seq	PBMCs	Human	2	13,383	36,495	0.94	210	0.46
Dataset 27	TEA-seq	PBMCs	Human	1	25,517	17,882	0.94	46	0.19
Dataset 28	inCITE-seq	Hippocampus	Mouse	2	19,279	12,526	0.98	4	0.03

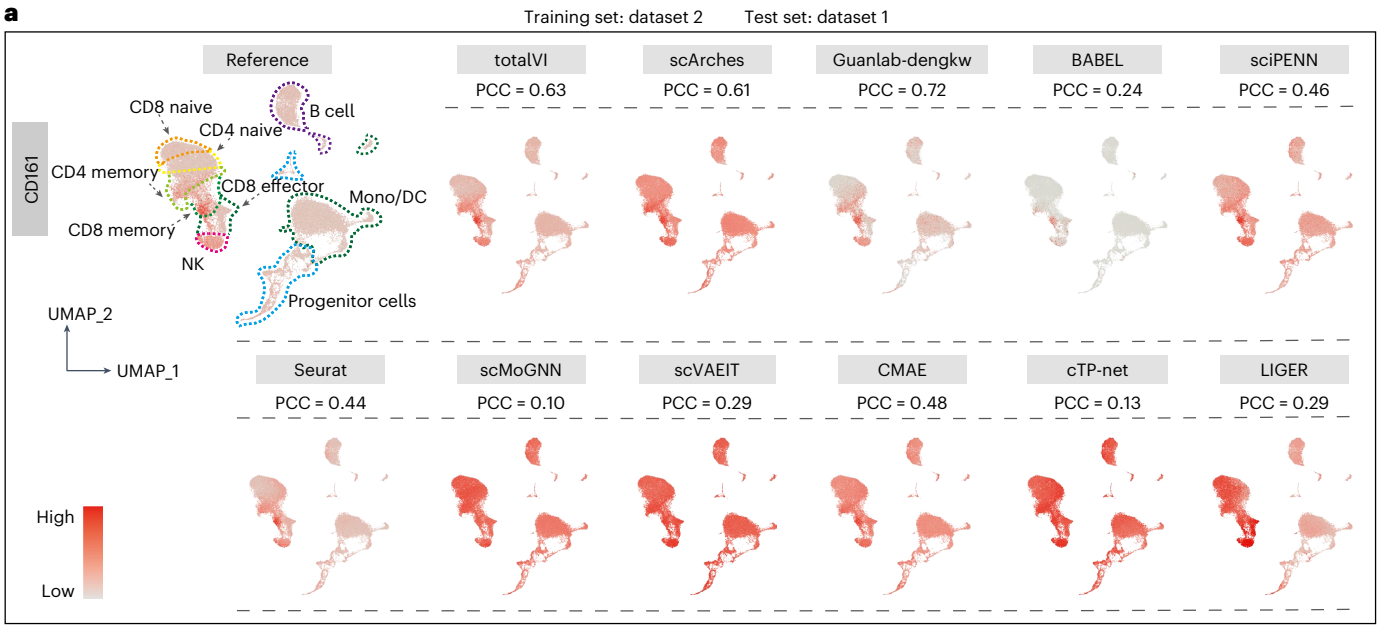
c

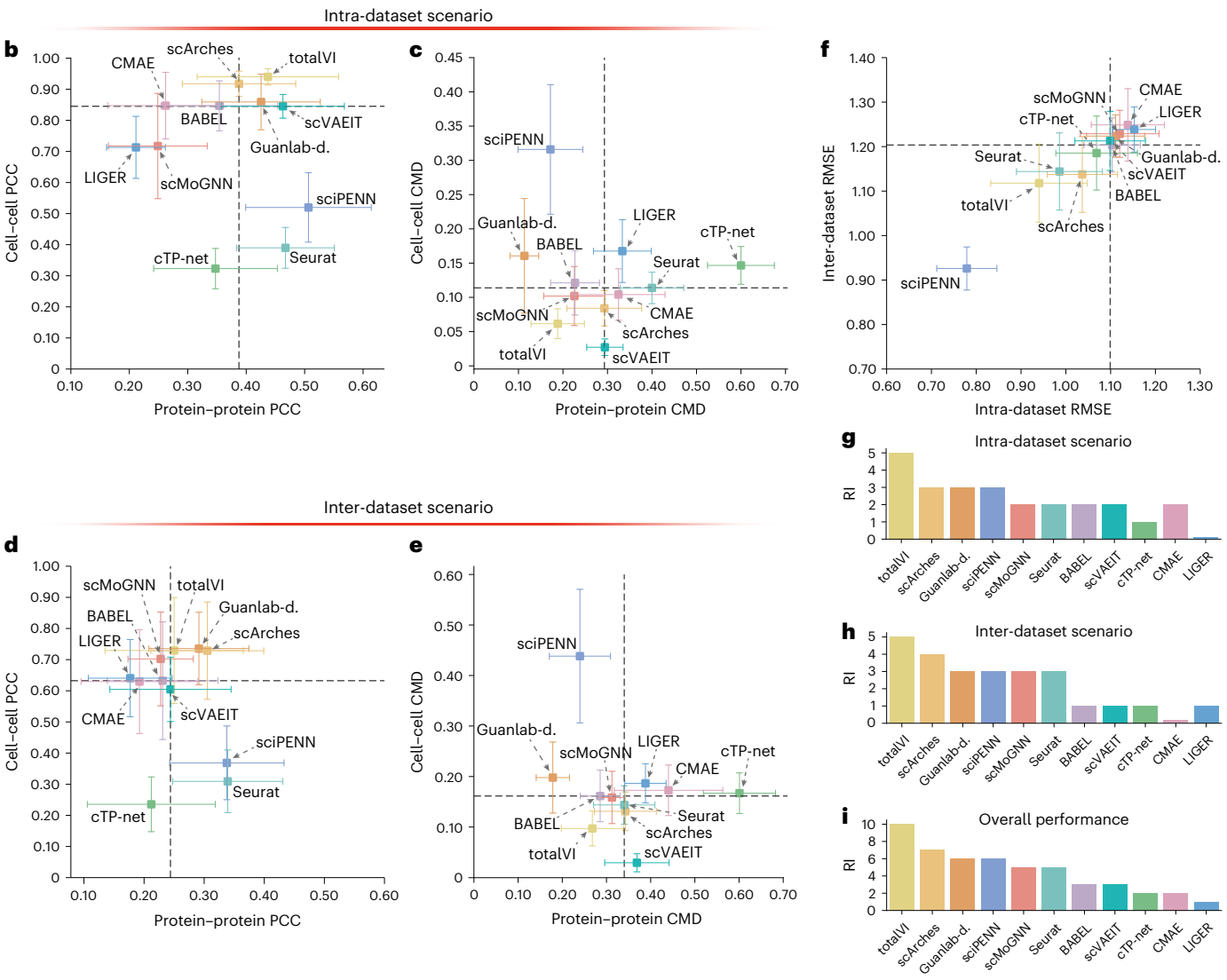
Dataset ID	Technique	Type/tissue	Species	Batch	Cells	RNA	Sparsity of RNA	Peak	Sparsity of peak
Dataset 29	SHARE-seq	Skin	Mouse	1	34,774	23,296	0.97	338,300	0.99
Dataset 30		Adult brain	Mouse	1	2,344	10,203	0.91	7,622	0.997
Dataset 31	SNARE-seq	Adult brain	Mouse	1	8,055	12,775	0.97	90,358	0.99
Dataset 32	ISSAAC-seq	Adult brain	Mouse	1	10,361	15,342	0.94	169,134	0.95
Dataset 33	10x Multiome	Adult brain	Human	1	2,855	16,910	0.91	130,862	0.92
Dataset 34–37		PBMCs	Human	4	5,812	13,052	0.92	95,729	0.94
Dataset 38		Retina	Mouse	1	9,383	6,275	0.94	59,353	0.95
Dataset 39		BMMCs	Human	13	69,249	13,431	0.94	103,375	0.97
Dataset 40–41	DOGMA-seq	PBMCs	Human	2	13,383	28,310	0.93	68,825	0.92
Dataset 42	TEA-seq	PBMCs	Human	1	25,517	17,882	0.94	128,853	0.98

d

Dataset ID	Technique	Type/tissue	Species	Batch	Cells	RNA	Sparsity of RNA	Peak	Sparsity of peak
Dataset 43	scRNA-seq	Spleen cell	Mouse	13	26,843	8,795	0.89	/	/
Dataset 44		Retina	Mouse	2	19,089	15,276	0.85	/	/
Dataset 45		Adult brain	Mouse	1	6,361	32,877	0.83	/	/
Dataset 46		HSPCs	Mouse	2	5,579	16,045	0.82	/	/
Dataset 47	scATAC-seq	Retina	Mouse	3	25,938	/	/	283,817	0.99

https://doi.org/10.1038/s41592-024-02429-w





https://doi.org/10.1038/s41592-024-02429-w

Fig. 2 | Performance of 11 algorithms in predicting protein abundance from RNA expression. a, CD161 abundances in the reference data of dataset 1 (CITE-seq; BMMCs) and the results predicted by the 11 algorithms. **b**,**c**, Average PCC (**b**) and CMD (**c**) values between the reference data and the predicted results for the intra-dataset scenario, that is, the training and test sets are from the same datasets. The *x* and *y* axes are the cell–cell and protein–protein PCC/CMD values, respectively, and the dashed lines are the medians of all algorithms' results. Error bars indicate the s.d. of 23 datasets. Data are presented as mean values ± 0.5 times the s.d. **d**,**e**, Same as **b** and **c**, but the results were predicted for the inter-dataset scenario, that is, the training and test sets are from different datasets. Error bars indicate the s.d. of 10 datasets. **f**, Average RMSE values between the reference data and the predicted results for the intra-dataset scenario (*x* axes) and inter-dataset scenario (*y* axes). Error bars indicate the s.d. of 23 datasets (*x* axes) or 10 datasets (*y* axes). Data are presented as mean values ± 0.5 times the s.d. **g**,**h**, RI values of 11 algorithms in the intra-dataset (**g**) and inter-dataset (**h**) scenarios. **i**, The overall performance of 11 algorithms in both intra-dataset and inter-dataset scenarios. DC, dendritic cell; NK, natural killer.

CMD/RMSE value is lower than the median CMD/RMSE (Methods). The scores for the six metrics and two scenarios were then aggregated to calculate the RI value, providing an evaluation of the algorithm's overall performance.

Performance of protein abundance prediction algorithms

First, we compared the performance of 11 algorithms for predicting protein abundance using transcriptomic information, with dataset 2 (CITE-seq, human, bone marrow mononuclear cells, BMMCs) and dataset 1 (CITE-seq, human, BMMCs) used as training and test sets, respectively. In the reference data of dataset 1, the CD161 protein is highly expressed in natural killer (NK) cells and $CD8^+$ memory T cells[50,51], and the CD4 protein is highly expressed in $CD4^+$ T cells[52]. Among the 11 algorithms evaluated, Guanlab-dengkw performed best in predicting the pattern of CD161 abundance across different types of cells (PCC = 0.72), followed by totalVI (PCC = 0.63) and scArches (PCC = 0.61; Fig. 2a). For the abundance of CD4, scArches (PCC = 0.81) and sciPENN (PCC = 0.80) generated results with the highest PCC values, followed by cTP-net, Seurat and totalVI (PCC = 0.76, 0.74 and 0.70; Supplementary Fig. 1).

We then applied the 11 protein abundance prediction algorithms to 23 single-cell RNA + protein datasets to assess their performance in the intra-dataset scenario. For the prediction results on these datasets, totalVI had the highest average cell–cell PCC (0.94), while sciPENN had the highest average protein–protein PCC (0.51). When considering both the cell–cell PCC and protein–protein PCC of each algorithm, the PCC values of totalVI and Guanlab-dengkw were both higher than the medians of all 11 algorithms (Fig. 2b). Additionally, we calculated the cell–cell and protein–protein CMD values between the reference data and the results predicted by each algorithm for the 23 datasets. We found that totalVI and scMoGNN had average cell–cell CMDs (0.06 and 0.10, respectively) and protein–protein CMDs (0.19 and 0.23, respectively) lower than the medians of all the algorithms, demonstrating their superior performance (Fig. 2c).

To assess the accuracy of each prediction algorithm in the inter-dataset scenario, we adopted ten pairs of single-cell RNA + protein datasets, where the two datasets in each pair were obtained from the same tissue/organ under different conditions. We utilized all proteins from the training data to train the algorithms and evaluated the performance of these algorithms using proteins present in both the training and test sets. totalVI, scArches and Guanlab-dengkw had average cell–cell and protein–protein PCC values higher than the medians of all 11 algorithms (Fig. 2d), while the average cell–cell and protein–protein CMD values of totalVI and scMoGNN were lower than the medians of all these algorithms (Fig. 2e). The RMSE values of totalVI (0.94/1.12), scArches (1.04/1.14), sciPENN (0.78/0.93), Seurat (0.99/1.14) and cTP-net (1.07/1.19) were lower than the medians of all the algorithms in both the intra-dataset and inter-dataset scenarios (Fig. 2f). We then computed the RI values for all the algorithms across the intra-dataset and inter-dataset results and evaluated their overall performance (Fig. 2g–i). Notably, totalVI and scArches had RI values of 10 and 7, respectively, which exceed those of sciPENN (6), Guanlab-dengkw (6), Seurat (5), scMoGNN (5), Babel (3), scVAEIT (3), CMAE (2), cTP-net (2) and LIGER (1; Fig. 2i).

After analyzing the prediction results for each dataset, we observed substantial variation in protein–protein PCCs across different proteins (Supplementary Figs. 2–7). For example, the PCC values of the CD19 protein predicted by the 11 algorithms were >0.7, but the PCC values of the CD223 protein were all <0.4 (Supplementary Fig. 8). To investigate the underlying reasons for this phenomenon, we divided the proteins in each dataset into two categories: (1) RNA-correlated (RC) proteins, for which the PCC between an RC protein abundance and an RNA expression level was ≥0.5 in the original single-cell RNA + protein dataset; and (2) RNA-uncorrelated (RU) proteins, for which the PCC between an RU protein abundance and any RNA expression level was <0.5. When evaluating the performance of the 11 algorithms in predicting the two categories of proteins, we found that the accuracies of all these algorithms for RC proteins (median protein–protein PCC > 0.55, median protein–protein CMD < 0.18, median RMSE < 1.07; Extended Data Fig. 1) were higher than those for the RU proteins (median protein–protein PCC < 0.26, median protein–protein CMD > 0.27, median RMSE > 1.16; Extended Data Fig. 2). These findings suggest that the prediction of RU protein abundance remains a challenging task.

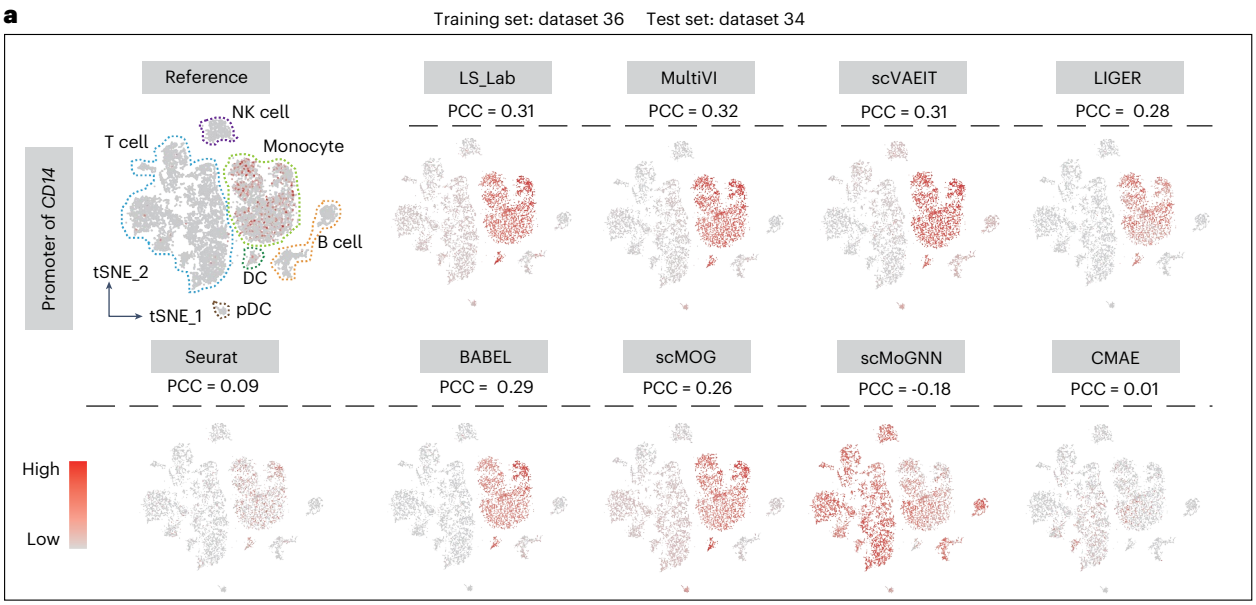
To delve deeper into their prediction stability, we calculated the differences in the upper and lower quartiles (DUL) of protein–protein PCCs for each algorithm across 33 datasets, covering both inter-dataset and intra-dataset scenarios (Methods). This analysis revealed considerable variability in the algorithms' prediction performance (median DUL ranging from 0.27 to 0.48), although some datasets showed more stability than others (Supplementary Fig. 9). Furthermore, we examined the correlation between the DUL values and five crucial attributes of these datasets (Methods). Our analysis revealed a notable correlation between the DUL values and the sparsity of the protein abundance matrix in three of the top four algorithms (totalVI, scArches and sciPENN; PCCs > 0.36, *P* values < 0.05; Supplementary Fig. 10). These findings suggest that the sparsity of the protein abundance matrix is one of the main factors impacting the predictive stability of these algorithms.

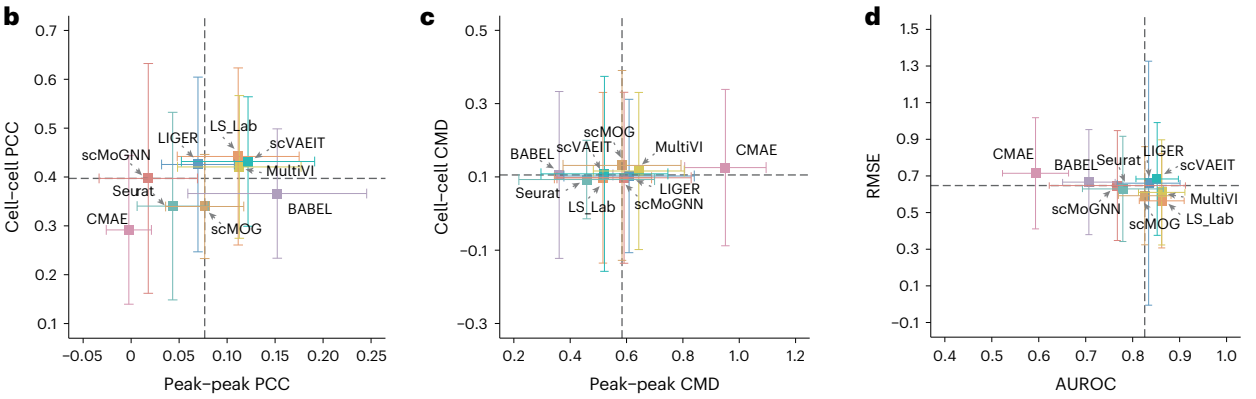
Performance of chromatin accessibility prediction algorithms

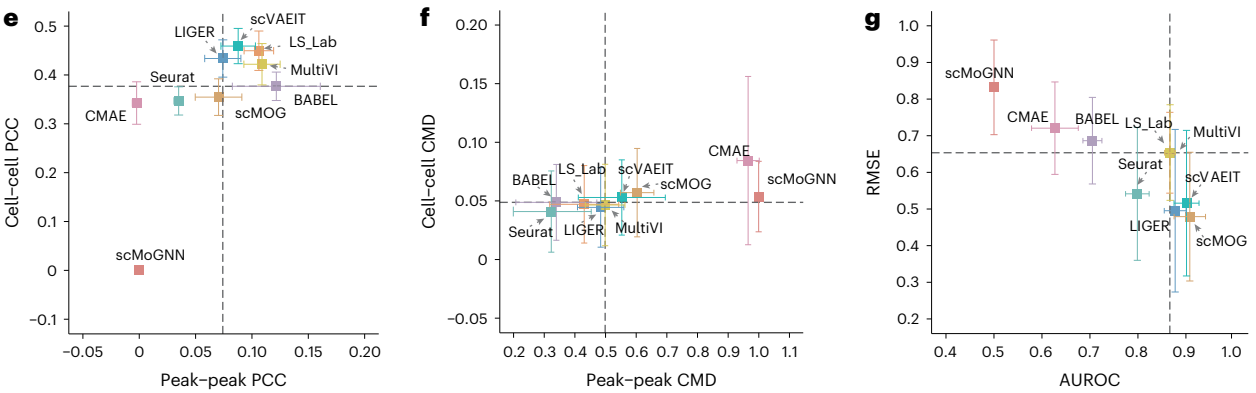
To compare the performance of the nine algorithms, including LS_Lab, scVAEIT, LIGER, MultiVI, Seurat, scMOG, BABEL, scMoGNN and CMAE, in

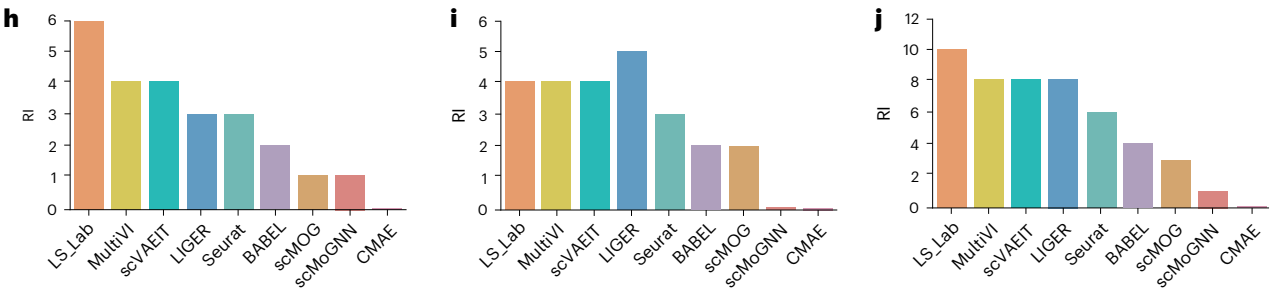
Fig. 3 | Performance of nine algorithms in predicting chromatin accessibility information from RNA expression. a, The chromatin accessibility of the promoter region of the *CD14* gene in the reference data of dataset 34 (10x Multiome, Human, PBMCs) and the results predicted by each algorithm. pDC, plasmacytoid dendritic cell. **b**–**d**, Average PCC (**b**), CMD (**c**) and RMSE and AUROC (**d**) values between the reference data and the predicted results for the intra-dataset scenario, that is, the training and test sets are from the same datasets. The *x* and *y* axes in **a** and **b** are the cell–cell and peak–peak PCC/CMD values, respectively, and the dashed lines are the medians of all algorithms' results. The *x* and *y* axes in **c** are the AUROC and RMSE, respectively, and the dashed lines are the medians of all algorithms' results. Error bars indicate the s.d. of 11 datasets. Data are presented as mean values ± 0.5 times the s.d. **e**–**g**, Same as **b**–**d**, but the results were predicted for the inter-dataset scenario, that is, the training and test sets are from different datasets. Error bars indicate the s.d. of eight datasets. **h**,**i**, RI values of nine algorithms in the intra-dataset (**h**) and inter-dataset (**i**) scenarios. **j**, The overall performance of seven algorithms in both intra-dataset and inter-dataset scenarios.

https://doi.org/10.1038/s41592-024-02429-w









https://doi.org/10.1038/s41592-024-02429-w

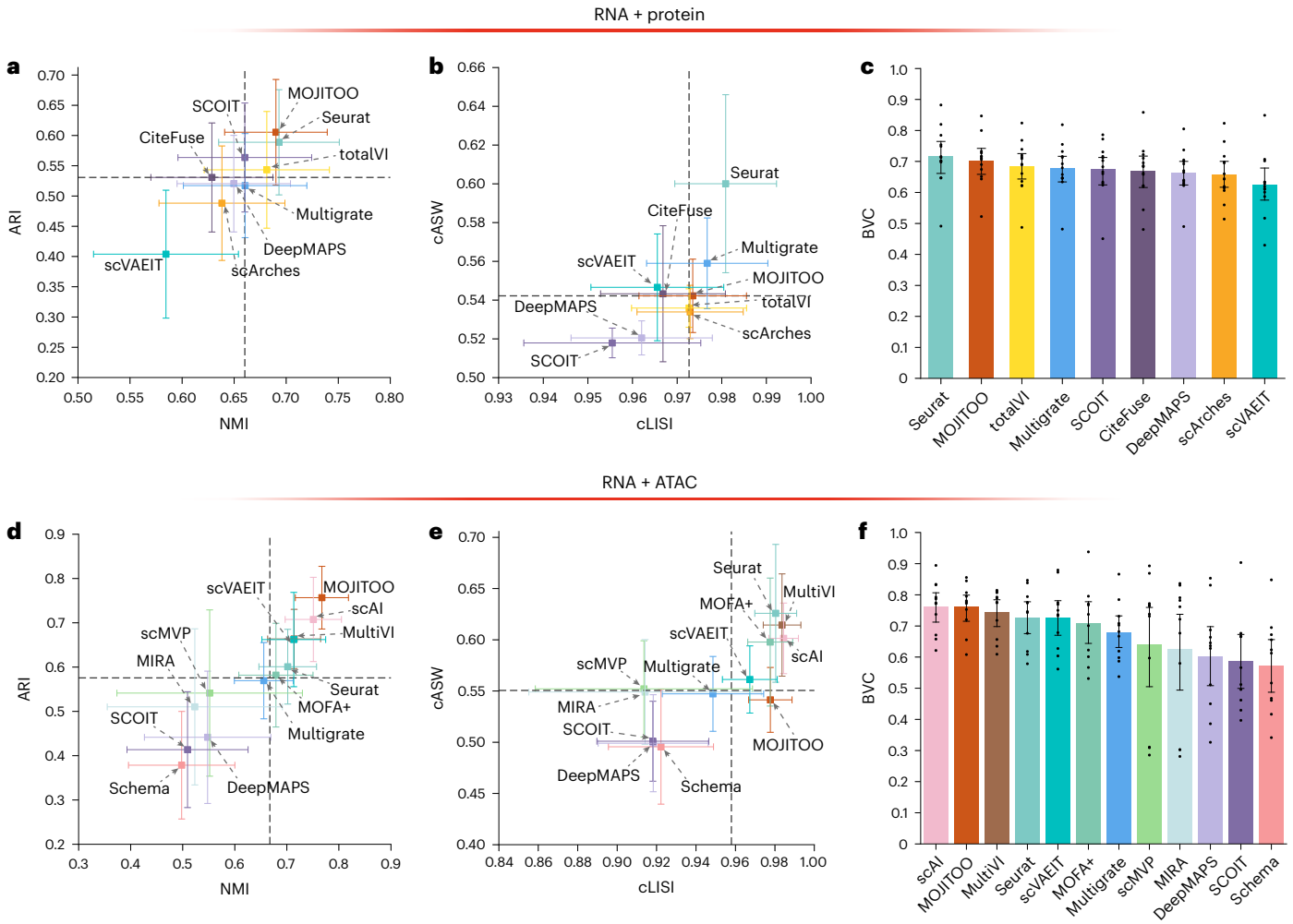


Fig. 4 | Benchmarking results for vertical integration. a, Average ARI and NMI values of nine vertical integration algorithms integrating RNA expression and protein abundance across 13 single-cell RNA + protein datasets. The *x* and *y* axes are average NMI and ARI values, respectively, and the dashed lines are the medians of all algorithms' results. Error bars indicate the s.d. of 13 datasets. Data are presented as mean value ± 0.5 times the s.d. **b**, Same as **a**, but the results were evaluated by the average cLISI and cASW values. **c**, Bar plots illustrate the overall performance of these algorithms, as evaluated by BVC scores across 13 RNA + protein datasets. Data are presented as mean values and 95% confidence intervals; $N = 13$ datasets. Each dot represents the BVC score of an algorithm on a dataset. **d**–**f**, Same as **a**–**c**, but the results were generated by 12 algorithms integrating RNA expression and chromatin information across 11 RNA + ATAC datasets. Data are presented as mean values and 95% confidence intervals; $N = 11$ datasets.

predicting chromatin accessibility, we first trained them with dataset 36 (10x Multiome, human; peripheral blood mononuclear cells (PBMCs)) and then used them to predict ATAC-seq peaks for dataset 34 (10x Multiome, human, PBMCs). In the reference data of dataset 34, DNA fragments in the promoter regions of the genes *CD14* and *MS4A1* are highly accessible in monocyte/dendritic cells and B cells, respectively[53,54]. Among the nine algorithms, LS_Lab (PCC = 0.31/0.60), MultiVI (PCC = 0.32/0.67), scVAEIT (PCC = 0.31/0.58), LIGER (PCC = 0.28/0.65) and BABEL (PCC = 0.29/0.62) also predicted that the promoter regions of the two genes are highly accessible in the corresponding cell types (Fig. 3a and Supplementary Fig. 11).

We then applied the nine algorithms to 11 single-cell RNA + ATAC datasets to test their performance in the intra-dataset scenario. LS_Lab, MultiVI and scVAEIT had average cell–cell and peak–peak PCC values higher than the medians of all the algorithms (Fig. 3b), while the average cell–cell and peak–peak CMD values of LS_Lab and Seurat were lower than the medians of the nine algorithms (Fig. 3c,i). In terms of average RMSE and AUROC values, LS_Lab and MultiVI also outperformed the other algorithms (Fig. 3d). However, none of these algorithms has an average peak–peak PCC value higher than 0.16 (Supplementary Figs. 12 and 13), implying that the predicted chromatin accessibility across cells differs from the reference data. One possible reason for this difference is that the scATAC-seq matrices in the reference data are usually very sparse, but the predicted results of some algorithms can be dense. For example, the sparsity of the reference scATAC-seq matrix of dataset 35 is 0.93, but the results predicted by Seurat, LIGER and LS_Lab for this dataset have sparsity values of 0.70, 0.36 and 0.001, respectively.

Considering the sparsity of the reference data, we used the definitions of domains of regulatory chromatin (DORCs)[3] from the scATAC-seq analytical toolkit Signac[55] to evaluate the ability of the nine algorithms to predict chromatin accessibility patterns. For DORCs, we combined peaks that had accessibility patterns similar to the expression patterns of the nearby gene[3]. This scheme can generate low-sparsity DORC matrices, thereby reducing the impact of missing values in scATAC-seq data. The DORC–DORC PCC values of the nine algorithms were higher than their peak–peak PCC values, indicating that they have better prediction performance in DORCs (Extended Data Fig. 3). Moreover, among the nine algorithms, LS_Lab and MultiVI were still the top-performing algorithms for predicting the accessibility of DORCs (Extended Data Fig. 3).

To further explore the impact of matrix sparsity on the accuracy of chromatin accessibility predictions, we implemented a KNN-smoothing

https://doi.org/10.1038/s41592-024-02429-w

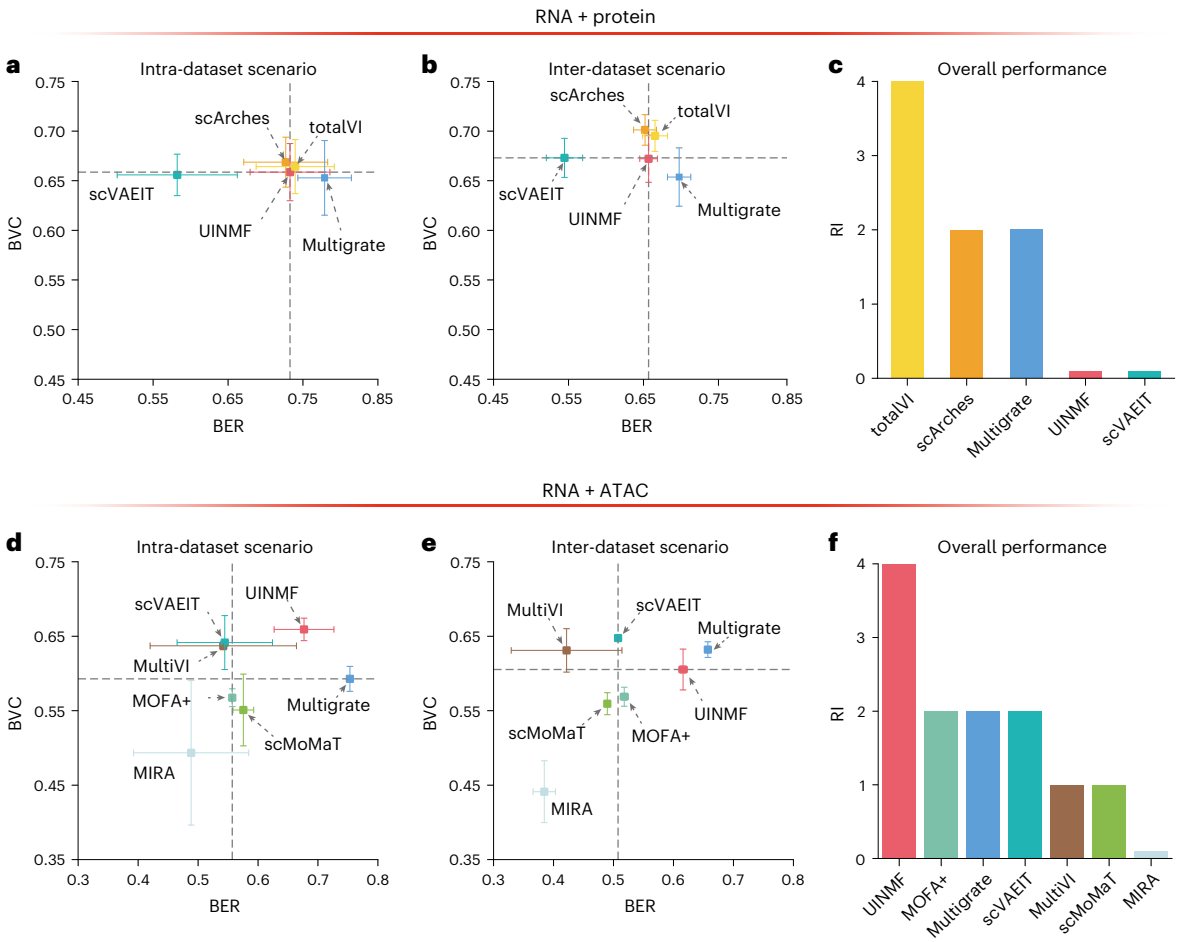


Fig. 5 | Benchmarking results for horizontal integration. a, Average BVC and BER scores of five horizontal integration algorithms for the intra-dataset scenario, that is, integrating batches from one dataset. The *x* and *y* axes are the average BVC (*x* axes) and BER (*y* axes) scores, respectively, and the dashed lines are the medians of all algorithms' results. Error bars indicate the s.d. of eight data groups. Data are presented as mean values ± 0.5 times the s.d. **b**, Same as **a**, but the results were generated by seven data groups in the inter-dataset scenario, that is, integrating batches from multiple datasets. Error bars indicate the s.d. of seven data groups. **c**, RI values for the five algorithms, derived from the average BVC and BER scores in both intra-dataset and inter-dataset scenarios. **d**–**f**, Same as **a**–**c**, but the plots present the results for seven horizontal integration algorithms, focusing on the integration of multiple RNA + ATAC batches. These results were generated from two data groups in the intra-dataset scenario (**d**) and two data groups in the inter-dataset scenario (**e**). The error bars of some algorithms (<0.015), including UINMF, MOFA+, Multigrate, scVAEIT, MultiVI and scMoMaT in **d** and **e**, are too small to be shown.

method (as utilized in MultiVI[22]) to reduce the noise in the raw count matrices of our test datasets (Methods). The introduction of this KNN-smoothing method resulted in considerable improvements in the predictive accuracy of eight of nine benchmarked algorithms, particularly the top-performing algorithms LS_Lab and MultiVI, whose peak–peak PCCs substantially increased from 0.11 and 0.11 to 0.53 and 0.49, respectively (Extended Data Fig. 4).

In addition, we assessed the performance of the nine algorithms when trained on one dataset and tested on another dataset, that is, the inter-dataset scenario. Both datasets were derived from the same tissue/organ but under different conditions. By comparing the PCC, CMD, RMSE and AUROC values of these algorithms, we found that LS_Lab, MultiVI, scVAEIT and LIGER outperformed the other algorithms in the inter-dataset scenario (Fig. 3e–g,i and Supplementary Figs. 14 and 15). Moreover, the RI values of the nine algorithms were calculated and ranked from high to low: LS_Lab (8), MultiVI (10), scVAEIT (9), LIGER (6), Seurat (4), BABEL (5), scMOG (4), scMoGNN (0) and CMAE (0; Fig. 3h–j), which also demonstrated the superiority of LS_Lab in predicting chromatin accessibility.

Considering that tissue-specific *cis*-elements are the subjects of interest for many studies[56,57], we also evaluated the performance of these algorithms in predicting transcription factor binding sites (TFBSs) and enhancers. Specifically, we downloaded the reference information of TFBSs/enhancers from EnhancerDB[58], and utilized Signac[55] to identify TFBSs/enhancers from the predicted chromatin accessibility of each dataset. By comparing the reference and predicted TFBSs/enhancers, we found that LS_Lab, MultiVI and scVAEIT outperformed the other algorithms in predicting tissue-specific *cis*-elements for both intra-dataset and inter-dataset scenarios (Supplementary Figs. 16–19).

Furthermore, we explored how the number of cells and RNAs in the training dataset influences the performance of the benchmarked algorithms (Methods). We found that including more cells from the same batch in the training dataset improved the predictive accuracy of the algorithms for both protein abundance and chromatin accessibility (Supplementary Figs. 20 and 21). However, including more RNAs in the training dataset helps to improve the accuracy of predicting protein abundance but not chromatin accessibility (Supplementary Figs. 22–25).

Performance of vertical integration algorithms

We first assessed 9 of 15 vertical integration algorithms capable of integrating RNA expression and protein abundance, including Seurat, MOJITOO, totalVI, Multigrate, SCOIT, scArches, scVAEIT, CiteFuse and

DeepMAPS (Supplementary Table 4). This evaluation was conducted using 13 single-cell RNA + protein datasets (Fig. 1 and Supplementary Table 7). We used four metrics—adjusted Rand index (ARI), normalized mutual information (NMI), average silhouette width of cell types (cASW) and cell-type separation local inverse Simpson's index (cLISI)—to assess each algorithm's performance on cell clustering and preservation of biological variation. The overall performance was gauged using the average of these metrics, termed the biological variation conservation (BVC) score.

Seurat and MOJITOO exhibited outstanding results, with the highest average ARI (0.59/0.61) and NMI (0.69/0.69), and Seurat was also leading in cASW (0.60) and cLISI (0.98; Fig. 4a,b and Supplementary Figs. 26 and 27). Their average BVC scores were 0.72 and 0.70, respectively, surpassing totalVI (0.68), Multigrate (0.68), SCIOIT (0.67), CiteFuse (0.67), DeepMaps (0.66), scArches (0.66) and scVAEIT (0.63; Fig. 4c).

Subsequently, we used the same metrics to evaluate 12 algorithms for integrating RNA expression and chromatin accessibility, including scAI, MOJITOO, MultiVI, Seurat, scVAEIT, MOFA+, Multigrate, scMVP, MIRA, DeepMAPS, SCOIT and Schema (Supplementary Table 4). This evaluation was conducted using 11 single-cell RNA + ATAC datasets (Fig. 1b, Supplementary Table 7 and Supplementary Figs. 28 and 29). MOJITOO displayed the highest average ARI (0.76) and NMI (0.77; Fig. 4d), while Seurat and scAI excelled in cASW (0.63) and cLISI (0.98) values, respectively (Fig. 4e). These findings highlight MOJITOO's proficiency in cell clustering, as well as the effectiveness of Seurat and MultiVI in representing cell–cell similarity. Notably, scAI was the only algorithm that ranked in the top three for all four metrics (Fig. 4d,e). Furthermore, the average BVC scores revealed scAI, MOJITOO and MultiVI as the top performers in integrating RNA expression and chromatin accessibility (Fig. 4f).

Performance of horizontal integration algorithms

Horizontal integration algorithms effectively remove batch effects and conserve biological variation in multi-omics data. We evaluated five horizontal integration algorithms—totalVI, scArches, Multigrate, UINMF and scVAEIT—that integrate single-cell RNA + protein datasets (Fig. 1a and Supplementary Table 5). These algorithms were assessed using the BVC score for cell clustering and biological variation preservation (Methods) and the batch effect removal (BER) score. The BER score is calculated as the average of five metrics including the KNN batch effect test (kBET), KNN graph connectivity, average silhouette width of batches (bASW), LISI of batch mixing (iLISI) and principal component regression (PCR). The overall performance was gauged using the RI values.

In the intra-dataset scenario, using eight single-cell RNA + protein data groups (Supplementary Table 8 and Supplementary Figs. 30 and 31), totalVI and scArches demonstrated higher average BVC scores compared to the median of all the algorithms (Fig. 5a). totalVI and Multigrate exhibited higher average BER scores than the median (Fig. 5a). For the inter-dataset scenario, using seven data groups constructed from nine datasets (Supplementary Table 8 and Supplementary Figs. 32 and 33), totalVI and scArches again surpassed the median BVC scores, and totalVI and Multigrate excelled in BER scores (Fig. 5b). Together with the RI values (Fig. 5c), totalVI was found to outperform the others in both scenarios.

Seven horizontal integration algorithms, including UINMF, MOFA+, Multigrate, scVAEIT, MultiVI, scMoMaT and MIRA, were designed for single-cell RNA + ATAC data integration. We then tested their performance using four data groups in two intra-dataset and inter-dataset scenarios (Fig. 5d,e and Supplementary Fig. 34 and Supplementary Tables 5 and 8). While scVAEIT showed the highest average BVC score and Multigrate had the highest average BER score, UINMF was the only algorithm with both BVC and BER scores above the median and achieved the highest RI value, indicating its superior performance (Fig. 5f).

Performance of mosaic integration algorithms

We evaluated the efficacy of eight mosaic integration algorithms, including totalVI, scArches, Multigrate, scMoMaT, scVAEIT, StabMap, MultiVI and UINMF, across 55 paired datasets in four distinct subcases (Fig. 1a Supplementary Tables 6 and 9). These algorithms are designed to integrate multi-omics datasets sharing at least one type of omics information (Methods).

In subcase 1, focusing on integrating scRNA-seq data with single-cell RNA + protein data, we compared seven algorithms using 8 paired datasets to assess their performance in intra-dataset scenarios and 11 paired datasets for inter-dataset scenarios (Supplementary Tables 6 and 9 and Supplementary Figs. 35–39). Throughout these evaluations, totalVI and scArches stood out, consistently delivering superior BVC and BER scores compared to their counterparts (Fig. 6a,b). Moreover, these two algorithms also achieved the top two RI values, outperforming the others in this subcase (Fig. 6c).

In subcase 2, which concentrates on integrating scRNA-seq data with single-cell RNA + ATAC data, we tested six algorithms across 13 paired datasets (Supplementary Tables 6 and 9). These datasets included 4 paired datasets for intra-dataset scenarios and 9 for inter-dataset scenarios (Supplementary Figs. 40–43). In this analysis, UINMF and MultiVI emerged as the top performers, showing the highest average scores in both the BVC and BER scores across both the intra-dataset and inter-dataset scenarios (Fig. 6d,e). Additionally, these two algorithms achieved the highest RI values across 13 datasets, indicating their superior performance in this particular subcase (Fig. 6f).

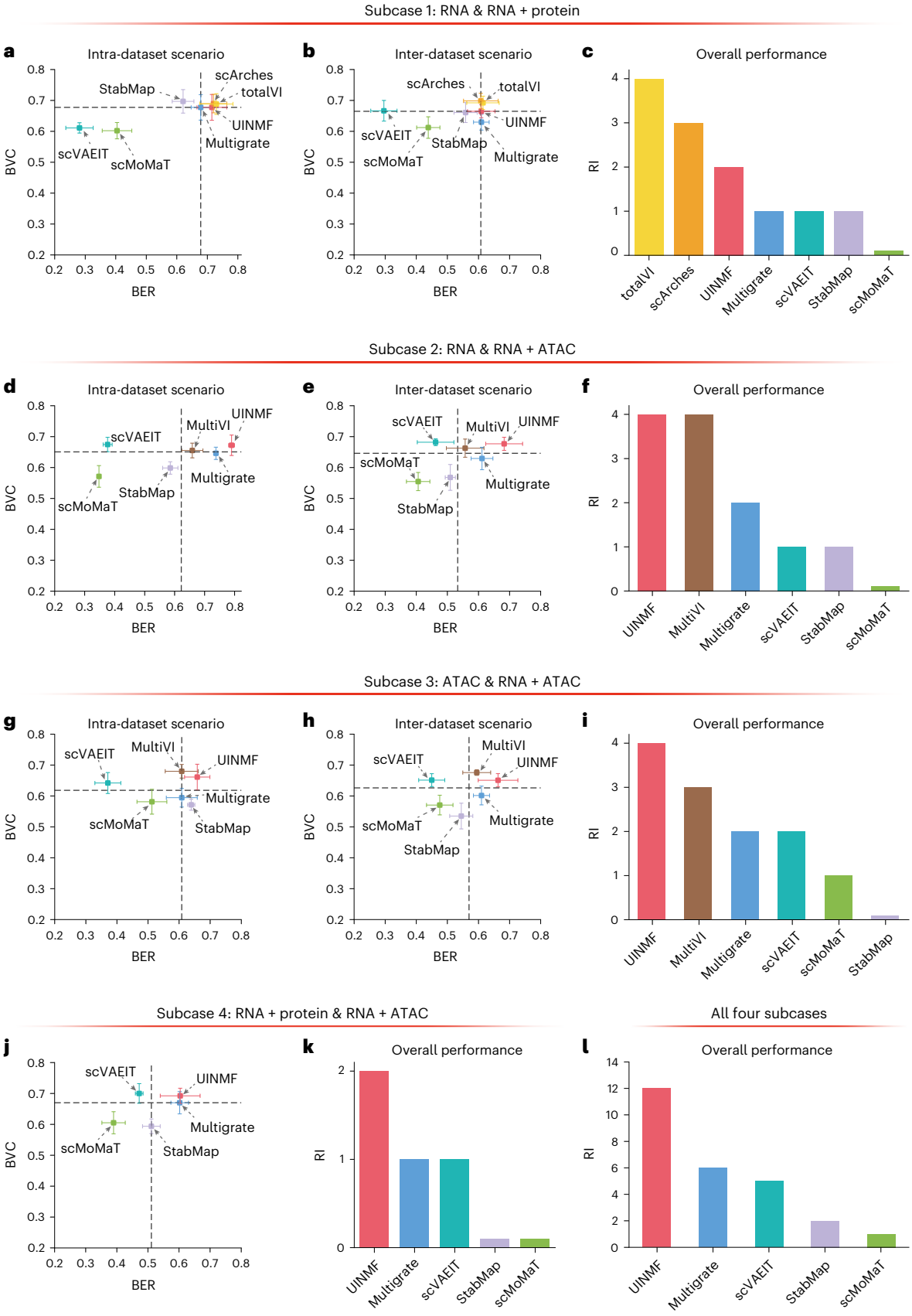
In subcase 3, which involves the integration of scATAC with single-cell RNA + ATAC data, we assessed six algorithms using a total of 12 paired datasets—4 for intra-dataset scenarios and 8 for inter-dataset scenarios (Supplementary Tables 6 and 9 and Supplementary Figs. 44–46). In this evaluation, MultiVI consistently demonstrated the highest BVC scores. Additionally, UINMF stood out by achieving the best BER scores among all the evaluated algorithms in both the intra-dataset and inter-dataset scenarios (Fig. 6g,h). Moreover, these two algorithms also attained the top two RI values across the datasets, underscoring their superior performance in this subcase (Fig. 6i).

In subcase 4, focusing on the integration of single-cell RNA + protein with RNA + ATAC data (Supplementary Table 6), we evaluated five algorithms using 11 paired datasets, all of which were specifically tailored to inter-dataset scenarios (Supplementary Table 9 and Supplementary Figs. 47–49). In this analysis, UINMF and scVAEIT emerged with the highest BVC scores. Concurrently, UINMF and Multigrate demonstrated the best performance in BER scores (Fig. 6j). These results collectively positioned UINMF as the most effective algorithm for this particular subcase (Fig. 6k).

Fig. 6 | Benchmarking results for mosaic integration. a,**b**, Scatterplots compare the average BVC and BER scores for seven mosaic integration algorithms for subcase 1 in the intra-dataset (**a**) and inter-dataset (**b**) scenarios. The dashed lines indicate the median results for all the algorithms. Error bars indicate the s.d. across all paired datasets in the intra-dataset (**a**) or inter-dataset (**b**) scenarios. These average and s.d. values were derived from 8 (**a**) or 11 (**b**) paired datasets. Data are presented as mean values ± 0.5 times the s.d. **c**, RI values of the seven algorithms were derived from the average BVC and BER in both intra-dataset and inter-dataset scenarios. **d**–**i**, Same as **a**–**c**, but these plots present the results of mosaic integration algorithms, focusing on subcase 2 (**d**–**f**) and subcase 3 (**g**–**i**). These average and s.d. values were derived from 4 (**d**), 9 (**e**), 4 (**g**) or 8 (**h**) paired datasets. **j**, Average BVC and BER scores of five mosaic integration algorithms focusing on subcase 4 across 11 paired datasets in the inter-dataset scenario. Error bars indicate the s.d. across 11 paired datasets. **k**, RI values of the five algorithms were derived from the average BVC and BER in **j**. **l**, The total RI values for the five algorithms encompass all four subcases. The error bars of MultiVI in **h** are too small to be shown.

https://doi.org/10.1038/s41592-024-02429-w

Among the nine algorithms assessed for mosaic integration, UINMF, StabMap, Multigrate, scMoMaT and scVAEIT were versatile enough to be applied across all four subcases (Supplementary Table 6). To provide a comprehensive performance comparison, we calculated the total RI values for each of these algorithms across all 55 datasets. When ranked based on these RI values, UINMF distinguished itself as the most effective algorithm among the five, highlighting UINMF's exceptional capability in mosaic integration across diverse datasets (Fig. 6l).



Computational resource evaluation

We undertook comprehensive computational resource evaluations for all the prediction and integration algorithms across different scenarios by downsampling the appropriate datasets (Supplementary Figs. 50 and 51). To minimize randomness, each sampling was repeated five times, and each test was performed twice. All the CPU-based algorithms were tested on an X64 CPU platform (2.5 GHz, 27.5 MB cache, 40 CPU cores) with 384 GB of DDR4 memory, while all the GPU-based algorithms were tested on an NVIDIA Tesla V100 (32 GB of memory) platform.

In our analysis, we observed that, of the top-performing algorithms in each scenario, totalVI, LS_Lab, Seurat, MultiVI and UINMF were able to complete prediction or integration tasks within 6 h using less than 256 GB of memory (Extended Data Figs. 5 and 6). However, scArches (predicting protein abundance) and scAI (vertical integration of RNA + ATAC data) encountered memory errors when tasked with datasets exceeding 500,000 and 20,000 cells, respectively. Overall, these evaluations provide insights into the computational demands and efficiency of these algorithms, highlighting their feasibility for various dataset sizes and computational platforms.

Discussion

In this study, we evaluated the performance of 14 algorithms for predicting protein abundance or chromatin accessibility from single-cell transcriptomics information. Although no single algorithm consistently outperformed the others in every metric and dataset (Supplementary Figs. 52 and 53 and Methods), we found overall totalVI and scArches were the best performers in predicting protein abundance, and LS_Lab was superior to the other algorithms in predicting chromatin accessibility in most cases. However, these algorithms encounter challenges in predicting the abundance of proteins that are not associated with any RNA expression (that is, RU proteins), which could be addressed by integrating additional omics information, such as metabolomic[59] and spatial omics[60,61] data, into prediction models. Additionally, the accessibility variation of DNA fragments predicted by these algorithms had low correlation with the reference scATAC-seq data, possibly due to the high sparsity of the reference data. Nonetheless, the predictive capability of these algorithms can be substantially improved when using the DORC × cell matrix and the denoised/smoothed peak × cell matrix as reference values in the test set. Notably, our comprehensive benchmarking not only identifies the most proficient algorithms but also sheds light on the extent to which complex chromatin–RNA–protein relationships can be predicted.

We also systematically evaluated the performance of 18 algorithms that can integrate single-cell multi-omics data across eight diverse scenarios. For vertical integration, Seurat and MOJITOO emerged as the top performers in merging RNA expression with protein abundance. scAI and MOJITOO distinguished themselves in integrating RNA expression with chromatin accessibility. Regarding horizontal integration, totalVI demonstrated its superiority in amalgamating various batches of single-cell RNA + protein data, while UINMF excelled in integrating batches of single-cell RNA + ATAC data. In the context of mosaic integration, totalVI came highly recommended for combining scRNA-seq with single-cell RNA + protein data, and UINMF was particularly effective for other mosaic integration scenarios.

In addition, we assessed the computational resources consumed by each algorithm. We also provide a pipeline (https://github.com/QuKunLab/MultiomeBenchmarking/) and a summary (Extended Data Figs. 7–10) to help researchers select the top-performing and most efficient algorithms for their multi-omics studies. Concerning the selection of training sets, we recommend choosing a training set that has a large number of cells and a high degree of RNA overlap with the test set (only for protein abundance prediction algorithms) and consists of cell types that are as similar as possible to those in the test set, ideally originating from the same tissue or organ (Supplementary Figs. 20–25 and 54–58 and Methods).

Our evaluation also highlighted that machine learning algorithms based on singular value decomposition, such as Guanlab-dengkw and LS_Lab, performed prominently in predicting protein abundance or chromatin accessibility. Among deep learning algorithms, totalVI utilizing probabilistic models outperformed other algorithms. Considering that singular value decomposition and probabilistic models both have noise reduction effects on single-cell data[48,62], the usage of noise reduction models is probably the reason for the higher prediction accuracy of these algorithms. In single-cell multi-omics integration analyses, algorithms like totalVI and UINMF achieved superior performance by accounting for data-specific properties like noise, nonnegativity and low-rank characteristics, potentially underscoring the significance of these data attributes in enhancing the performance of these algorithms. We also found that the sparsity of these datasets somewhat affected the accuracy of these algorithms in predicting protein abundance or chromatin accessibility (Supplementary Fig. 59). A possible solution is to first apply an imputation algorithm, such as SAVER[48], WEDGE[49], totalVI[6] or scBasset[63], to predict the missing values in the original data and then use the imputed dataset for training and prediction.

In addition, many spatial genomics technologies, including 10x Visium, Stereo-seq[64] and MERFISH[65], only detect the distribution of transcripts in space[61]. A technical solution is to extend multi-omics technology to the field of spatial omics, such as the DBiT-seq and spatial-CITE-seq technologies developed by R. Fan's laboratory[66,67]. Another possible solution is to adopt a single-cell multi-omics dataset and a spatial transcriptomics dataset as training and test sets, respectively, and apply the best-performing algorithms to predict the distribution of protein abundance or chromatin accessibility in space. Therefore, this benchmark study may help researchers choose appropriate algorithms for the joint analysis of not only single-cell RNA-seq and multi-omics data but also spatial transcriptomics data and single-cell multi-omics data. However, it is important to note that spatial and non-spatial data may exhibit different distributions[20], which could affect the performance of the algorithms.

Algorithms leveraging large language models have been developed recently to predict missing modalities or integrate multiple modalities from extensive single-cell multi-omics datasets; however, due to current hardware limitations, our benchmark study does not encompass these prediction or integration algorithms based on large language models, such as Geneformer[68], scGPT[69] and scFoundation[70].

Online content

Any methods, additional references, Nature Portfolio reporting summaries, source data, extended data, supplementary information, acknowledgements, peer review information; details of author contributions and competing interests; and statements of data and code availability are available at https://doi.org/10.1038/s41592-024-02429-w.

References

1. Stoeckius, M. et al. Simultaneous epitope and transcriptome measurement in single cells. *Nat. Methods* **14**, 865–868 (2017).
2. Peterson, V. M. et al. Multiplexed quantification of proteins and transcripts in single cells. *Nat. Biotechnol.* **35**, 936–939 (2017).
3. Ma, S. et al. Chromatin potential identified by shared single-cell profiling of RNA and chromatin. *Cell* **183**, 1103–1116.e20 (2020).
4. Chen, S., Lake, B. B. & Zhang, K. High-throughput sequencing of the transcriptome and chromatin accessibility in the same cell. *Nat. Biotechnol.* **37**, 1452–1457 (2019).
5. Hao, Y. et al. Integrated analysis of multimodal single-cell data. *Cell* **184**, 3573–3587.e29 (2021).
6. Gayoso, A. et al. Joint probabilistic modeling of single-cell multi-omic data with totalVI. *Nat. Methods* **18**, 272–282 (2021).
7. Zhang, L., Zhang, J. & Nie, Q. DIRECT-NET: an efficient method to discover *cis*-regulatory elements and construct regulatory networks from single-cell multiomics data. *Sci. Adv.* **8**, eabl7393 (2022).

8. Kartha, V. K. et al. Functional inference of gene regulation using single-cell multi-omics. *Cell Genom.* **2**, 100166 (2022).
9. Li, C., Virgilio, M. C., Collins, K. L. & Welch, J. D. Multi-omic single-cell velocity models epigenome–transcriptome interactions and improves cell fate prediction. *Nat. Biotechnol.* **41**, 387–398 (2023).
10. Gorin, G., Svensson, V. & Pachter, L. Protein velocity and acceleration from single-cell multiomics experiments. *Genome Biol.* **21**, 39 (2020).
11. La Manno, G. et al. RNA velocity of single cells. *Nature* **560**, 494–498 (2018).
12. Xu, W. et al. ISSAAC-seq enables sensitive and flexible multimodal profiling of chromatin accessibility and gene expression in single cells. *Nat. Methods* **19**, 1243–1249 (2022).
13. Zhou, Z., Ye, C., Wang, J. & Zhang, N. R. Surface protein imputation from single cell transcriptomes by deep neural networks. *Nat. Commun.* **11**, 651 (2020).
14. Bennett, H. M., Stephenson, W., Rose, C. M. & Darmanis, S. Single-cell proteomics enabled by next-generation sequencing or mass spectrometry. *Nat. Methods* **20**, 363–374 (2023).
15. Gatto, L. et al. Initial recommendations for performing, benchmarking and reporting single-cell proteomics experiments. *Nat. Methods* **20**, 375–386 (2023).
16. Lance, C. et al. Multimodal single cell data integration challenge: results and lessons learned. In *Proc. NeurIPS 2021 Competitions and Demonstrations Track* (eds. Kiela, D. et al.) 162–176 (PMLR, 2022).
17. Bartosovic, M., Kabbe, M. & Castelo-Branco, G. Single-cell CUT&Tag profiles histone modifications and transcription factors in complex tissues. *Nat. Biotechnol.* **39**, 825–835 (2021).
18. Mimitou, E. P. et al. Scalable, multimodal profiling of chromatin accessibility, gene expression and protein levels in single cells. *Nat. Biotechnol.* **39**, 1246–1258 (2021).
19. Stuart, T. et al. Comprehensive integration of single-cell data. *Cell* **177**, 1888–1902.e21 (2019).
20. Welch, J. D. et al. Single-cell multi-omic integration compares and contrasts features of brain cell identity. *Cell* **177**, 1873–887.e17 (2019).
21. Lotfollahi, M. et al. Mapping single-cell data to reference atlases by transfer learning. *Nat. Biotechnol.* **40**, 121–130 (2022).
22. Ashuach, T. et al. MultiVI: deep generative model for the integration of multimodal data. *Nat. Methods* **20**, 1222–1231 (2023).
23. Lakkis, J. 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.* **4**, 940–952 (2022).
24. Wu, K. E., Yost, K. E., Chang, H. Y. & Zou, J. BABEL enables cross-modality translation between multiomic profiles at single-cell resolution. *Proc. Natl Acad. Sci. USA* **118**, e2023070118 (2021).
25. Du, J.-H., Cai, Z. & Roeder, K. Robust probabilistic modeling for single-cell multimodal mosaic integration and imputation via scVAEIT. *Proc. Natl Acad. Sci. USA* **119**, e2214414119 (2022).
26. Lan, M., Zhang, S. & Gao, L. Efficient generation of paired single-cell multiomics profiles by deep learning. *Adv. Sci* **10**, 2301169 (2023).
27. Wen, H. et al. *Proc. 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining* (Association for Computing Machinery, 2022).
28. Yang, K. D. et al. Multi-domain translation between single-cell imaging and sequencing data using autoencoders. *Nat. Commun.* **12**, 31 (2021).
29. Baysoy, A., Bai, Z., Satija, R. & Fan, R. The technological landscape and applications of single-cell multi-omics. *Nat. Rev. Mol. Cell Biol.* **24**, 695–713 (2023).
30. Cheng, M., Li, Z. & Costa, I. G. MOJITOO: a fast and universal method for integration of multimodal single-cell data. *Bioinformatics* **38**, i282–i289 (2022).
31. Lotfollahi, M., Litinetskaya, A. & Theis, F. J. Multigrate: single-cell multi-omic data integration. Preprint at *bioRxiv* https://doi.org/10.1101/2022.03.16.484643 (2022).
32. Wang, R. H., Wang, J. & Li, S. C. Probabilistic tensor decomposition extracts better latent embeddings from single-cell multiomic data. *Nucleic Acids Res.* **51**, e81 (2023).
33. Kim, H. J., Lin, Y., Geddes, T. A., Yang, J. Y. H. & Yang, P. CiteFuse enables multi-modal analysis of CITE-seq data. *Bioinformatics* **36**, 4137–4143 (2020).
34. Ma, A. et al. Single-cell biological network inference using a heterogeneous graph transformer. *Nat. Commun.* **14**, 964 (2023).
35. Jin, S., Zhang, L. & Nie, Q. scAI: an unsupervised approach for the integrative analysis of parallel single-cell transcriptomic and epigenomic profiles. *Genome Biol.* **21**, 25 (2020).
36. Argelaguet, R. et al. MOFA+: a statistical framework for comprehensive integration of multi-modal single-cell data. *Genome Biol.* **21**, 111 (2020).
37. Li, G. et al. A deep generative model for multi-view profiling of single-cell RNA-seq and ATAC-seq data. *Genome Biol.* **23**, 20 (2022).
38. Lynch, A. W. et al. MIRA: joint regulatory modeling of multimodal expression and chromatin accessibility in single cells. *Nat. Methods* **19**, 1097–1108 (2022).
39. Singh, R., Hie, B. L., Narayan, A. & Berger, B. Schema: metric learning enables interpretable synthesis of heterogeneous single-cell modalities. *Genome Biol.* **22**, 131 (2021).
40. Kriebel, A. R. & Welch, J. D. UINMF performs mosaic integration of single-cell multi-omic datasets using nonnegative matrix factorization. *Nat. Commun.* **13**, 780 (2022).
41. Zhang, Z. et al. scMoMaT jointly performs single cell mosaic integration and multi-modal bio-marker detection. *Nat. Commun.* **14**, 384 (2023).
42. Ghazanfar, S., Guibentif, C. & Marioni, J. C. Stabilized mosaic single-cell data integration using unshared features. *Nat. Biotechnol.* **42**, 284–292 (2024).
43. De Biasi, S. et al. Circulating mucosal-associated invariant T cells identify patients responding to anti-PD-1 therapy. *Nat. Commun.* **12**, 1669 (2021).
44. Heumos, L. et al. Best practices for single-cell analysis across modalities. *Nat. Rev. Genet.* **24**, 550–572 (2023).
45. Miao, Z., Humphreys, B. D., McMahon, A. P. & Kim, J. Multi-omics integration in the age of million single-cell data. *Nat. Rev. Nephrol.* **17**, 710–724 (2021).
46. Argelaguet, R., Cuomo, A. S. E., Stegle, O. & Marioni, J. C. Computational principles and challenges in single-cell data integration. *Nat. Biotechnol.* **39**, 1202–1215 (2021).
47. Wang, J. et al. Data denoising with transfer learning in single-cell transcriptomics. *Nat. Methods* **16**, 875–878 (2019).
48. Huang, M. et al. SAVER: gene expression recovery for single-cell RNA sequencing. *Nat. Methods* **15**, 539–542 (2018).
49. Hu, Y. et al. WEDGE: imputation of gene expression values from single-cell RNA-seq datasets using biased matrix decomposition. *Brief. Bioinform.* **22**, bbab085 (2021).
50. Truong, K.-L. et al. Killer-like receptors and GPR56 progressive expression defines cytokine production of human $CD4^+$ memory T cells. *Nat. Commun.* **10**, 2263 (2019).
51. Fergusson, J. R. et al. $CD161^{int}CD8^+$ T cells: a novel population of highly functional, memory CD8+ T cells enriched within the gut. *Mucosal Immunol.* **9**, 401–413 (2016).
52. Kung, P. C., Goldstein, G., Reinherz, E. L. & Schlossman, S. F. Monoclonal antibodies defining distinctive human T cell surface antigens. *Science* **206**, 347–349 (1979).

53. Liang, Y. & Tedder, T. F. Identification of a CD20-, FcεRIβ-, and HTm4-Related gene family: sixteen new MS4A family members expressed in human and mouse. *Genomics* **72**, 119–127 (2001).
54. Ziegler-Heitbrock, H. W. L. & Ulevitch, R. J. CD14: cell surface receptor and differentiation marker. *Immunol. Today* **14**, 121–125 (1993).
55. Stuart, T., Srivastava, A., Madad, S., Lareau, C. A. & Satija, R. Single-cell chromatin state analysis with Signac. *Nat. Methods* **18**, 1333–1341 (2021).
56. Spitz, F. & Furlong, E. E. M. Transcription factors: from enhancer binding to developmental control. *Nat. Rev. Genet.* **13**, 613–626 (2012).
57. Gertz, J. et al. Distinct properties of cell-type-specific and shared transcription factor binding sites. *Mol. Cell* **52**, 25–36 (2013).
58. Kang, R. et al. EnhancerDB: a resource of transcriptional regulation in the context of enhancers. *Database* **2019**, bay141 (2019).
59. Buergel, T. et al. Metabolomic profiles predict individual multidisease outcomes. *Nat. Med.* **28**, 2309–2320 (2022).
60. Lewis, S. M. et al. Spatial omics and multiplexed imaging to explore cancer biology. *Nat. Methods* **18**, 997–1012 (2021).
61. Li, B. et al. Benchmarking spatial and single-cell transcriptomics integration methods for transcript distribution prediction and cell type deconvolution. *Nat. Methods* **19**, 662–670 (2022).
62. Linderman, G. C. et al. Zero-preserving imputation of single-cell RNA-seq data. *Nat. Commun.* **13**, 192 (2022).
63. Yuan, H. & Kelley, D. R. scBasset: sequence-based modeling of single-cell ATAC-seq using convolutional neural networks. *Nat. Methods* **19**, 1088–1096 (2022).
64. Chen, A. et al. Spatiotemporal transcriptomic atlas of mouse organogenesis using DNA nanoball-patterned arrays. *Cell* **185**, 1777–1792.e21 (2022).
65. Chen, K. H., Boettiger, A. N., Moffitt, J. R., Wang, S. & Zhuang, X. Spatially resolved, highly multiplexed RNA profiling in single cells. *Science* **348**, aaa6090 (2015).
66. Su, G. et al. Spatial multi-omics sequencing for fixed tissue via DBiT-seq. *STAR Protoc.* **2**, 100532 (2021).
67. Liu, Y. et al. High-plex protein and whole transcriptome co-mapping at cellular resolution with spatial CITE-seq. *Nat. Biotechnol.* **41**, 1405–1409 (2023).
68. Theodoris, C. V. et al. Transfer learning enables predictions in network biology. *Nature* **618**, 616–624 (2023).
69. Cui, H. et al. scGPT: toward building a foundation model for single-cell multi-omics using generative AI. *Nat. Methods* **21**, 1470–1480 (2024).
70. Hao, M. et al. Large-scale foundation model on single-cell transcriptomics. *Nat. Methods* **21**, 1481–1491 (2024).

Methods

Data collection and preprocessing

We collected a diverse set of single-cell multi-omics sequencing data, including 25 single-cell RNA + protein datasets, 12 single-cell RNA + ATAC datasets, and 3 single-cell ATAC + RNA + protein datasets. These datasets were generated using various sequencing technologies, including CITE-seq, REAP-seq, SHARE-seq, SNARE-seq, 10x Multiome, TEA-seq, inCITE-seq and DOGMA-seq. To ensure data quality, we utilized Seurat's quality-control filters ('CreateSeuratObject()') with parameters described in the original papers to eliminate low-quality cells, RNAs and/or DNA fragments (that is, 'peaks' in scATAC-seq data). See Supplementary Tables 2 and 3 and 'Data availability' for more details on these datasets.

Peak calling for scATAC-seq data

For intra-dataset scenarios, we used different peak calling methods based on the dataset's origin. For SNARE-seq datasets, cisTopic was used with default parameters for peak calling[4]. For datasets generated by 10x Chromium-based sequencing technologies, such as ISSAAC-seq, DOGMA-seq, TEA-seq and 10x Multiome, Cell Ranger was used for peak calling[12,18,71]. For SHARE-seq technology datasets, MACS2 software was used with specific parameters '--nomodel --nolambda --keep-dup --call-summits'[3].

For inter-dataset scenarios, given that peaks are independently identified in each dataset, achieving perfect overlap is unlikely. To address this, we utilized the Signac tutorial (https://stuartlab.org/signac/) to merge peaks and create a unified peak set for multi-dataset analysis. Specifically, we used the merge() function with its default parameters (that is, 'merge.data = TRUE, merge.dr = NULL, project = 'SeuratProject'), ensuring that peaks from multiple datasets were appropriately merged for subsequent analysis.

We also recognized that peaks in regions other than promoters, enhancers and gene bodies might have biological importance[36,40]. Hence, we included all peaks that passed the quality control in both the training and test datasets in our analysis.

Cell-type annotation

We assessed the performance of 18 multi-omics integration algorithms using 35 datasets from our collection of 47 datasets, each annotated with cell-type information as indicated in the corresponding data source papers (Supplementary Tables 7–9). In the inter-dataset scenario, we merged various sub-cell types from different datasets to standardize the labeling of inconsistent sub-cell types. For instance, in a data group comprising dataset 34 and dataset 24, where all B cells in dataset 24 were categorized into a single cell type, we amalgamated memory B and naive B cells from dataset 34 into one unified B cell category. The revised cell-type labels, accommodating these adjustments, can be accessed at our GitHub repository via https://github.com/QuKunLab/MultiomeBenchmarking/. It is important to note that for benchmarking the performance of vertical integration algorithms, we selected the batch with the highest cell count in each dataset (Supplementary Table 7).

Parameter settings for algorithms

In our study, we evaluated the performance of 14 algorithms designed for predicting protein abundance and/or chromatin accessibility, as well as 18 algorithms specialized in single-cell multi-omics integration. Notably, five of these algorithms are versatile, and capable of handling both prediction and integration tasks. Consequently, our benchmark study encompasses a total of 27 distinct algorithms. The parameters used for each of these algorithms were determined as follows:

1. totalVI: We followed the instructions provided on the totalVI website via https://docs.scvi-tools.org/en/stable/tutorials/notebooks/multimodal/cite_scrna_integration_w_totalVI.html to perform our analysis. We set the parameters for the analyses as 'latent_distribution = normal and n_layers_decoder = 2'.
2. scArches: We followed the guidelines provided on the scArches website: https://scarches.readthedocs.io/en/latest/totalvi_surgery_pipeline.html. The model was trained with parameters 'epochs = 200, plan_kwargs = dict, weight_decay = 0.0'.
3. Guanlab-dengkw: We followed the tutorial provided on the Guanlab-dengkw GitHub repository: https://github.com/openproblems-bio/neurips2021_multimodal_topmethods/tree/main/src/predict_modality/methods/Guanlab-dengkw/.
4. sciPENN: We followed the guidelines provided on the sciPENN GitHub repository: https://github.com/jlakkis/sciPENN_codes/blob/master/Experiments/pbmc_to_malt%20sciPENN.ipynb/. We used the parameters 'n_epochs = 10,000, ES_max = 12, decay_max = 6, decay_step = 0.1, lr = 10^{-3}'.
5. BABEL: We followed the tutorial provided on the DANCE toolkit GitHub repository via https://github.com/OmicsML/dance/tree/main/examples/multi_modality/predict_modality/babel.py/. The predicted protein/ATAC modality was obtained using the 'predict' function in the 'BabelWrapper' class.
6. scMoGNN: We followed the tutorial provided on the DANCE toolkit GitHub repository via https://github.com/OmicsML/dance/tree/main/examples/multi_modality/predict_modality/scmogcn.py. The predicted protein/ATAC modality was obtained using the 'predict' function in the 'ScMoGCNWrapper' class.
7. CMAE: We followed the tutorial provided on the DANCE toolkit GitHub repository via https://github.com/OmicsML/dance/tree/main/examples/multi_modality/predict_modality/cmae.py/. The predicted protein/ATAC modality was obtained using the 'predict' function in the 'CMAE' class.
8. LS_Lab: We followed the tutorial provided on the neurips2021_multimodal_topmethods GitHub repository: https://github.com/openproblems-bio/neurips2021_multimodal_topmethods/blob/main/src/predict_modality/methods/LS_lab/run/script.py/.
9. cTP-net: We followed the instructions provided on the cTP-net GitHub repository via https://github.com/zhouzilu/cTPnet/. We set 'n_batches = 32 and max_epochs = 4'.
10. MultiVI: We followed the guidelines provided on the MultiVI website at https://docs.scvi-tools.org/en/stable/tutorials/notebooks/multimodal/MultiVI_tutorial.html. We used the 'get_accessibility_estimates' function of MultiVI to predict chromatin accessibility from single-cell transcriptomics information.
11. Seurat: To predict the protein abundance and chromatin accessibility, we followed the guidelines on the Seurat website at https://satijalab.org/seurat/archive/v3.2/integration.html. We set the parameter reduction = 'cca', and used the 'TransferData' function of Seurat. To perform vertical integration, we followed the tutorial provided on the Seurat website via https://satijalab.org/seurat/articles/weighted_nearest_neighbor_analysis/.
12. LIGER: We followed the tutorial provided on the LIGER GitHub repository at https://github.com/welch-lab/liger/. We used the 'imputeKNN' function of LIGER to predict protein abundance and chromatin accessibility, with the parameter 'norm = FALSE, scale = FALSE'.
13. scVAEIT: We adopted the guidelines outlined on the scVAEIT GitHub repository at https://github.com/jaydu1/scVAEIT/. The predictions of protein abundance and chromatin accessibility were derived using the 'get_recon' function of scVAEIT.
14. scMOG: We conducted the prediction of chromatin accessibility following the instructions provided on the scMOG GitHub repository via https://github.com/GaoLabXDU/scMOG/. The parameters were set to the default values, that is, 'hidden = 16, lr = 0.001'.

15. MOJITOO: We conducted the vertical integration of single-cell multi-omics data following the tutorial available on the MOJITOO GitHub repository via https://github.com/CostaLab/MOJITOO/. The 'mojitoo' function was utilized with all parameters set to the default values. For dimensionality reduction, principal component analysis was applied to the RNA expression data, and Latent Semantic Indexing was used for processing the chromatin accessibility data.
16. Multigrate: We integrated single-cell multi-omics data using the Multigrate module of scArches, following the instructions on https://docs.scarches.org/en/latest/multigrate.html. Specifically, we used the 'nb' loss function for RNA data, and the 'mse' function for ATAC and protein data.
17. SCOIT: We conducted the vertical integration following the tutorial provided on the SCOIT GitHub repository via https://github.com/deepomicslab/SCOIT/.
18. CiteFuse: We performed the vertical integration of the single-cell RNA expression and protein abundance data following the document on the CiteFuse GitHub repository at https://sydneybiox.github.io/CiteFuse/articles/CiteFuse.html.
19. DeepMAPS: We conducted the vertical integration following the procedures outlined in the tutorial on the DeepMAPS GitHub repository at https://github.com/OSU-BMBL/deepmaps/blob/master/scRNA_scATAC_analyses_tutorial.html. The heterogeneous graph transformation was performed by using the 'run_HGT' function with default parameters, that is, 'lr = 0.2, epoch = 30, n_hid = 128, n_heads = 16'.
20. scAI: We integrated the single-cell RNA expression and chromatin accessibility information following the document on the scAI GitHub repository via https://htmlpreview.github.io/?https://github.com/sqjin/scAI/blob/master/examples/walkthrough_Kidneydataset.html.
21. MOFA+: We conducted the integration of single-cell RNA expression and chromatin accessibility information in accordance with the guidelines provided on the MOFA's website at https://muon-tutorials.readthedocs.io/en/latest/single-cell-rna-atac/pbmc10k/3-Multimodal-Omics-Data-Integration.html.
22. scMVP: We performed the vertical integration of single-cell RNA expression and chromatin accessibility information in line with the tutorial on the scMVP GitHub repository via https://github.com/bm2-lab/scMVP/blob/master/demos/scMVP_tutorial.ipynb/. Specifically, for the 'MultiTrainer' function, we set train_size = 0.9 and frequency = 5.
23. Schema: We integrated the RNA expression and chromatin accessibility information in single-cell RNA + ATAC data following the instruction on the Schema GitHub repository via https://schema-multimodal.readthedocs.io/en/latest/overview.html#quick-start/.
24. MIRA: We performed the horizontal integration of single-cell RNA + ATAC data following the guidelines provided in the MIRA tutorial, available at https://mira-multiome.readthedocs.io/en/latest/tutorials.html. Specifically, we utilized the 'mira.topics.gradient_tune' function and the 'get_learning_rate_bounds' function to determine the optimal number of topics and the range of learning rates, respectively.
25. scMoMaT: We integrated single-cell multi-omics data using the tutorial of scMoMaT via https://github.com/PeterZZQ/scMoMaT/blob/main/demo_scmomat.ipynb/. For the mosaic integration of multi-omics data, we used the 'calc_pseudo_count.R' function to obtain the pseudo-scRNA-seq count.
26. UINMF: We followed the tutorial outlined in the UINMF GitHub repository tutorial, available at https://github.com/welch-lab/liger/. We adopted the 'optimizeALS' function of UINMF to obtain the low-dimensional embedding space from the raw multi-omics data, and then used the 'quantile_norm' function to normalize the cell embedding data.
27. StabMap: We followed the guidelines provided on the GitHub repository of StabMap via https://marionilab.github.io/StabMap/articles/stabMap_PBMC_Multiome.html.

Benchmark metrics

1. The PCC is defined by equation (1):

$$\mathrm{PCC}(\mathbf{x}, \mathbf{y}) = \frac{\sum_1^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_1^n (x_i - \bar{x})^2}\sqrt{(y_i - \bar{y})^2}} \quad (1)$$

When calculating cell–cell PCC, x_i and y_i represent the abundance of protein (or chromatin accessibility of peak i) in cell x and y, respectively. $\bar{x}$ and $\bar{y}$ represent the average values of $\{x_i\}$ and $\{y_i\}$, respectively. For the calculation of protein–protein (or peak–peak PCC), x_i and y_i are the protein abundance (or chromatin accessibility) of cell i for the protein x and y (or peak x and y), respectively.

2. CMD was usually used to measure the difference between two correlation matrices $\boldsymbol{R_1}$ and $\boldsymbol{R_2}$, and a lower CMD value indicates a better result[47–49]. The CMD is defined according to equation (2):

$$d(\boldsymbol{R}_1, \boldsymbol{R}_2) = 1 - \frac{\mathrm{trace}(\boldsymbol{R}_1\boldsymbol{R}_2)}{\|\boldsymbol{R}_1\|_F \|\boldsymbol{R}_2\|_F} \quad (2)$$

where $\mathrm{trace}(\boldsymbol{R}_1\boldsymbol{R}_2)$ represents the trace of matrix $\boldsymbol{R}_1 \times \boldsymbol{R}_2$ and $\|\cdot\|_F$ is the Frobenius norm of a matrix. In this study, each element in the correlation matrices $\boldsymbol{R}$ is the PCC value between two cells or two proteins/peaks.

3. The RMSE was used to quantify the difference between the predicted values ($\boldsymbol{X}$) and true values ($\hat{\boldsymbol{X}}$)[6,23]. To ensure comparability, both the predicted and true values were normalized using the same method and rescaled using z scores. The RMSE is mathematically defined as $\mathrm{RMSE} = \|\boldsymbol{X} - \hat{\boldsymbol{X}}\|_F$, where $\|\cdot\|_F$ is the Frobenius norm of a matrix.
4. The AUROC[72] is a key metric used to evaluate an algorithm's capacity to differentiate between binary categories (that is, 1 and 0). We used AUROC values to gauge the effectiveness of the algorithms in predicting chromatin accessibility. The range of AUROC values is from 0 to 1, where a value of 1 signifies perfect prediction accuracy. Conversely, an AUROC value of 0.5 indicates a model performing at par with random guessing, suggesting the absence of meaningful discriminative power.
5. To assess the concordance between known cell-type labels and the cell clusters identified by the Leiden algorithm, we utilized the ARI[73] and NMI[74] as our primary metrics. To avoid any potential biases arising from the resolution parameter inherent in the Leiden algorithm, we conducted Leiden clustering on cells at a range of resolutions, spanning from 0.1 to 2.0 in increments of 0.1, using the integrated results from each algorithm. The performance of each algorithm was then assessed based on the highest ARI and NMI values achieved across these varying resolutions.
6. The average silhouette width (ASW)[75] metric was utilized to gauge the precision of cell–cell distances calculated by each integration algorithm. ASW is an indicator of how well a cell matches with others in its cluster (intra-cluster similarity) and how distinct it is from cells in the closest different cluster (inter-cluster dissimilarity). The silhouette width for each cell was computed using the formula $\frac{b-a}{\max(a,b)}$, where a represents the average intra-cluster distance, and b represents the average nearest-cluster distance. Averaging all the silhouette widths of a set of cells yields the ASW, which ranges between −1 and 1. In our

analysis, we leveraged ASW scores based on cell-type labels (cASW) and batch labels (bASW) to evaluate each algorithm's effectiveness in conserving biological variation and removing batch effects, respectively. A higher cASW value signifies improved accuracy in cell-type separation, whereas a lower bASW indicates more effective correction of batch effects. To ensure a standardized evaluation, we transformed the cASW and bASW values using linear transformations (in line with the method used in ref. 76) so that higher values consistently indicate superior performance for both the cASW and bASW.

7. The LISI[77] was utilized to assess the results of our integration algorithms in terms of cell-type separation (denoted as cLISI) and batch mixing (referred to as iLISI). In the context of our study, a lower cLISI value is indicative of more effective cell-type separation, signifying enhanced conservation of biological variation. Conversely, a higher iLISI value reflects better integration of different batches, pointing to more successful removal of batch effects. To maintain consistency in our evaluation criteria and ensure that higher values of both iLISI and cLISI represent improved performance, we applied linear transformations to these values[76].
8. The KNN graph connectivity (KNN connectivity)[78] metric was used to evaluate the connectivity between cells within each cell type in a KNN graph. This metric is calculated by equation (3):

$$\text{KNN connectivity} = \frac{1}{N}\sum_{i=1}^{N}\frac{\max(m_i)}{n_i} \quad (3)$$

where N is the total number of cell types, n_i is the cell number of cell type i, and $\max(m_i)$ is the cell number of the largest connected subgroup of cell type i in the KNN graph.

9. We used PCR[78] to measure the BER for multi-omics integration algorithms. The PCR was defined according to equation (4):

$$\text{PCR} = \sum_{i=1}^{n}\text{var}(A|\text{PC}_i)R^2(\text{PC}_i|\text{batch}) \quad (4)$$

where A can be the RNA expression matrix, the chromatin accessibility matrix, the protein abundance matrix or the low-dimensional embedding matrix generated by an integration algorithm. PC_i is the i-th principal component of A, $\text{var}(A|\text{PC}_i)$ is the variance of A on PC_i, and R^2 (PC_i, |, batch)signifies the squared correlation coefficient between PC_i and the batch labels of cells.

10. We used the kBET[78] as a metric to quantify the extent of BER by each integration algorithm. kBET assesses the similarity between two key components: the composition of batch labels among the k-nearest-neighbors of a cell ($\boldsymbol{C}_{\text{KNN}}$) and the overall batch labels composition across all cells ($\boldsymbol{C}_{\text{batch}}$). Ideally, in a scenario where the batch effect has been effectively eliminated, $\boldsymbol{C}_{\text{KNN}}$ should be equal to $\boldsymbol{C}_{\text{batch}}$, resulting in a kBET value of 1. We calculated the kBET value for each algorithm's integration results using scIB[76] with default settings.
11. The isolated label score (ILS)[76] was utilized to assess the effectiveness of horizontal or mosaic integration algorithms in embedding cell connectivity graphs into a low-dimensional space and isolating specific cell types that are present in only a subset of data batches. Specifically, for any given cell type i that occurs in k_i batches, the ILS is determined by averaging the ASW values for cell types that are present in k_{min} batches. Here k_{min} represents the smallest number among all k_i values.
12. Ranking index (RI) was used to gauge the overall performance of each algorithm. The RI value of algorithm i is defined according to equation (5):

$$\text{RI}_i = \sum_j B(v_{ij}) \quad (5)$$

where $B()$ is the Heaviside function, that is, $B(x) = \begin{cases} 0, x < 0 \\ 1, x \geq 0 \end{cases}$, and $B(v_{ij})$ represents whether algorithm i is one of the top-performing algorithms when using metric j for comparison. For metrics where a lower value signifies better performance, such as CMD and RMSE, we defined $v_{ij} = \text{median}(Y_{:,j}) - Y_{ij}$. In contrast, for metrics where a higher value signifies better performance, such as PCC and AUROC, we defined $v_{ij} = Y_{ij} - \text{median}(Y_{:,j})$. Here, Y_{ij} refers to the value of the j-th metric for the i-th algorithm, and $Y_{:,j}$ represents the array of values for the j-th metric across all algorithms being evaluated.

13. The dataset-specific rank index quantifies an algorithm's relative performance within a specific dataset, based on the number of metrics for which it ranks in the top 50%. This method is a nuanced variation of the RI, which is computed across all datasets. For example, if the algorithm totalVI ranks among the top 50% for three specific metrics (such as cell–cell PCC, protein–protein PCC, RMSE) in dataset A, its dataset-specific rank index for that dataset would be assigned as 3.
14. The BVC score is a crucial metric for each integration algorithm's performance on a given dataset. It is calculated as the mean of several metrics: ARI, NMI, cASW, cLISI and ILS. The BVC score is instrumental in evaluating how well an algorithm preserves biological variation across datasets. However, in the context of vertical integration algorithms, where only a single batch is involved, the ILS is not applicable. Therefore, for these algorithms, the BVC score is derived only from the ARI, NMI, cASW and cLISI metrics, and these metrics are normalized using the scaling method outlined by scIB[76].
15. The BER score for each integration algorithm on a given dataset is the average of five key metrics: kBET, KNN connectivity, bASW, iLISI and PCR. The BER score assesses an algorithm's ability to effectively mitigate batch effects. Similarly to the BVC score, the values of kBET, KNN connectivity, bASW, iLISI and PCR were rescaled for BER score calculation, following the transformation methodology of scIB[76].

Average values and standard deviations

In our comparison of single-cell multi-omics prediction algorithms, we calculate the average and standard deviation for each metric, including cell–cell PCC, protein–protein/peak–peak PCC, RMSE and AUROC, across all datasets. Specifically focusing on cell–cell PCC, we denoted the reference data as $\boldsymbol{A}$ and the prediction results from a given algorithm as $\boldsymbol{B}$ for the i-th test dataset. In matrices $\boldsymbol{A}$ and $\boldsymbol{B}$, rows represent cells, and columns represent proteins or peaks. For the j-th cell in the i-th dataset, we compute the PCC value between the respective rows in $\boldsymbol{A}$ and $\boldsymbol{B}$ ($\boldsymbol{A}(j,:)$ and $\boldsymbol{B}(j,:)$), which we denote as PCC_j. The performance of the algorithm on the i-th dataset is then represented by the median of these PCC values (PCC_1, PCC_2, ..., PCC_n), labeled as m_i, where n denotes the total number of cells in the i-th test dataset. Lastly, the overall performance of the algorithm across all test datasets, in terms of cell–cell PCC, is determined by the average (μ) and standard deviation (σ) of these medians (m_1, m_2, ..., m_q), where q signifies the total number of test datasets. Similarly, we calculated the median PCC of all proteins/peaks in a dataset for each algorithm, and used the average and standard deviation of these medians across all datasets to plot the average protein–protein/peak–peak PCC and error bar, respectively. In addition, we used a similar strategy to calculate the average values and standard deviations of RMSE and AUROC for each algorithm.

For the benchmark of single-cell multi-omics integration algorithms, we adopted the same strategy as above to calculate the average values and standard deviations of ten metrics[61] for each algorithm, including kBET, KNN connectivity, bASW, iLISI, PCR, cLISI, ARI, NMI, cASW and ILS.

KNN smoothing of scATAC-seq data

To reduce the sparsity of the scATAC-seq data, we implemented a KNN-smoothing method, as utilized in MultiVI[22], to diminish the noise

in the raw count matrices of our test datasets. Notably, we applied the KNN-smoothing method only to the test datasets, while the training sets remained unaltered. Specifically, we first utilized latent semantic indexing on the scATAC-seq data to obtain a cellular embedding in a 30-dimensional space. Once the 30-dimensional embedding was established, we then identified the 50 nearest neighbors for each cell within this reduced space. The chromatin accessibility for each cell in the KNN-smoothed scATAC-seq data was then determined by calculating the mean accessibility of its 50 nearest neighbors.

Stability analysis for protein abundance prediction algorithms

We evaluated the stability of the protein abundance prediction algorithm by calculating the difference between the upper and lower quartiles (DUL) of the protein–protein PCCs for each algorithm across all 33 test datasets in both inter-dataset and intra-dataset scenarios. For example, if the upper and lower quartiles of the protein–protein PCCs for totalVI on dataset A were x and y, respectively, the DUL for that dataset would be calculated as x–y.

The impact of cell number on the performance of prediction algorithms

To examine the effect of varying the number of cells in the training set (training cells) on the accuracy of predicting protein abundance from RNA expression, we set aside 2,000 cells from dataset 2 (CITE-seq, human, BMMCs, 134 proteins) as the test set and created training sets with varying cell counts ranging from 1,000 to 64,000 from the same dataset. To ensure the reliability of our findings, this sampling process was repeated five times for each cell count in the training set. A similar procedure was conducted using dataset 12 (CITE-seq, human, PBMCs, 228 proteins). Moreover, we applied a comparable approach to assess the impact of the number of cells in the training set on the performance of algorithms predicting chromatin accessibility via RNA expression. This assessment was performed using dataset 39 (10x Multiome, human, BMMCs) and dataset 34 (10x Multiome, human, PBMCs).

The impact of RNA number on the performance of prediction algorithms

To examine the effect of the number of RNAs in the training set on the performance of the benchmarked algorithms, we undertook the following approach:

1. **Dataset preparation**: From a given dataset, we randomly selected 10,000 cells as the training set and 2,000 cells as the test set. For the training set, we sampled 1,000, 2,000, 4,000 and 8,000 RNAs to construct different versions of the training data.
2. **RNA sampling strategies**: We used two distinct methods for selecting training RNAs:
   - The highly variable gene (HVG) scheme involved selecting the top N (N = 1,000, 2,000, 4,000 and 8,000) highly variable RNAs.
   - For the random scheme, we randomly sampled N (N = 1,000, 2,000, 4,000 and 8,000) RNAs from among all the detected RNAs. This random sampling procedure was repeated five times for each RNA count.

Impact of overlapping cell types between the training and test datasets on prediction performance

We assessed how the proportion of overlapping cell types between the training and test datasets affects prediction performance. Our method involved randomly sampling N cells from a given dataset to form a test dataset containing k cell types and then creating training datasets (TD_1, TD_2,…, TD_k), each comprising $5N$ cells. Each TD_i ($i = 1 \sim k$) includes i cell types, with an equal number of cells for each type, and shared i cell types with the test set. This process was repeated five times for robustness. Notably, the number of cells was identical for all cell types in the test dataset. By using this method, we evaluated protein abundance prediction algorithms using dataset 12 (CITE-seq, human, PBMC, 228 proteins) and the chromatin accessibility prediction algorithms with dataset 39 (10x Multiome, human, BMMCs).

Scalability analysis

The scalability analysis was conducted on a supercomputing platform consisting of a CPU computer cluster with two Intel Xeon Scale 6248 CPUs (2.5 GHz, 27.5 MB cache and 40 CPU cores), 384 GB of DDR4 memory (2,933 MHz) and 2 TB of AEP memory, as well as a GPU computer cluster with two Intel Xeon Scale 6248 CPUs, 384 GB of DDR4 memory and an NVIDIA Tesla V100 graphics card (32 GB of memory and 5,120 CUDA cores).

In the evaluation of computer resources consumed by each algorithm for predicting protein abundance, we used a human peripheral memory T cell dataset from the Gene Expression Omnibus (GEO; GSE158769) and filtered out genes that were detected in fewer than ten cells. Next, we used the Scanpy function 'sc.pp.highly_variable_genes' with the parameters min_mean = 0.0125, max_mean = 3 and min_disp = 0.5 to select the top 3,000 HVGs in the RNA expression matrix. To simulate data of different scales, we randomly sampled 1,000, 5,000, 10,000, 50,000, 100,000 and 500,000 cells from this dataset as the training set and sampled 20% cells from the same dataset as the test set. To minimize randomness, each sampling was repeated five times and each test was performed twice.

When assessing computer resources consumed by each algorithm for predicting chromatin accessibility, we downloaded a mouse skin dataset from the GEO database (GSE140203) and filtered out genes that were detected in fewer than ten cells. Then, we used the 'sc.pp.highly_variable_genes' function in Scanpy with the parameters min_mean = 0.0125, max_mean = 3 and min_disp = 0.5 to select 4,211 HVGs using transcriptomics information. To simulate data with different numbers of cells, we randomly sampled 3,500, 7,000, 14,000 and 28,000 cells from the mouse skin dataset as the training set and sampled 20% of cells from the dataset to build a test set. We also repeated the sampling process five times for each cell number and performed two tests for each dataset.

To evaluate the computational resource consumption of nine vertical integration algorithms for integrating RNA expression and protein abundance within a single-cell RNA + protein dataset, we constructed various datasets with different cell counts—5,000, 10,000, 20,000, 40,000, and 80,000—by randomly sampling cells from dataset 2 (CITE-seq, human, BMMCs, 66,175 cells, 134 proteins). Each sampled dataset contained 3,000 HVGs and 134 proteins. To minimize randomness, each sampling was repeated five times and each test was performed twice. Similarly, we used dataset 39 (10x Multiome, human, BMMCs, 69,249 cells) to evaluate the computational resources required by 12 vertical integration algorithms for integrating RNA expression and chromatin accessibility. For each sampled dataset, we used 3,000 HVGs and 30,000 peaks.

To assess the computational resources consumed by five horizontal integration algorithms in integrating single-cell RNA + protein data, we created five single-cell RNA + protein data groups by randomly sampling cells from dataset 20 (CITE-seq, human, peripheral memory T cells, 500,089 cells, 31 proteins). Each data group comprises 3,000 genes, 31 proteins and 2,500 to 40,000 cells. We repeated the sampling process five times for each cell number and performed two tests for each data group. Similarly, we sampled cells from dataset 39 (10x Multiome, human, BMMCs, 69,249 cells) to evaluate the computational resources consumed by seven horizontal integration algorithms for integrating single-cell RNA + ATAC data. Each data group sampled from dataset 39 included 3,000 RNAs and 30,000 peaks.

To assess the computational resources consumed by seven mosaic integration algorithms in integrating scRNA-seq data with single-cell

RNA + protein data (subcase 1), we sampled 2,500 to 40,000 cells from dataset 12 (CITE-seq, human, PBMCs, 161,764 cells, 228 proteins) as single-cell RNA + protein datasets, and 2,500 to 40,000 cells from dataset 13 (CITE-seq, human, PBMCs, 49,147 cells, 54 proteins) to construct scRNA-seq datasets. The single-cell RNA + protein datasets included 3,000 RNAs and 228 proteins, and the scRNA-seq datasets included 3,000 RNAs. To evaluate the computational resources of the six mosaic integration algorithms in integrating scRNA-seq/scATAC-seq data and single-cell RNA + ATAC data (subcase 2/3), we randomly sampled 1,000 to 8,000 cells from dataset 39 (10x Multiome, human, BMMCs, 69,249 cells) as single-cell RNA + ATAC datasets, and 2,000 to 16,000 cells from the same dataset as scRNA-seq/scATAC-seq datasets. Each dataset contained 3,000 RNAs and 30,000 peaks. To assess the computational costs of five mosaic integration algorithms when integrating single-cell RNA + protein data with single-cell RNA + ATAC data (subcase 4), we sampled 3,000 to 24,000 cells from dataset 2 (CITE-seq, human, BMMCs, 66,175 cells, 134 proteins) as single-cell RNA + protein datasets, and 2,000 to 16,000 cells from dataset 39 (10x Multiome, human, BMMCs, 69,249 cells) as single-cell RNA + ATAC datasets. Each single-cell RNA + protein dataset has 3,000 RNAs and 134 proteins, and each single-cell RNA + ATAC data group has 3,000 RNAs and 30,000 peaks. To minimize randomness, each sampling was repeated five times and each test was performed twice.

Reporting summary

Further information on research design is available in the Nature Portfolio Reporting Summary linked to this article.

Data availability

A summary of the multi-omics datasets used in the benchmark study, including the sequencing technologies and the websites where the raw data are available as follows: dataset 1 (human BMMCs): CITE-seq, GSE128639 (ref. 5); dataset 2 (human BMMCs): CITE-seq, GSE194122 (ref. 79); dataset 3 (human brain immune cells): CITE-seq, GSE201048 (ref. 80); dataset 4 (human CBMCs): CITE-seq, GSE100866 (ref. 1); dataset 5 (human glioblastomas): CITE-seq, GSM4972212 (ref. 81); dataset 6 (mouse glioblastomas): CITE-seq, GSE163120 (ref. 81); dataset 7 (mouse HSPCs): CITE-seq, GSE175702 (ref. 82); dataset 8 (human MALT tumor): CITE-seq, https://support.10xgenomics.com/single-cell-gene-expression/datasets/3.0.0/malt_10k_protein_v3; dataset 9–10 (mouse murine splenic myeloid cells): CITE-seq, GSE149544 (ref. 83); dataset 11 (mouse naive brains): CITE-seq, GSE148127 (ref. 84); dataset 12–13 (human PBMCs): CITE-seq, GSE164378 (ref. 5); dataset 14–15 (human PBMCs): CITE-seq, https://zenodo.org/record/6348128#.Y5f40LJBzDU (ref. 30); dataset 21–22 (mouse spleen and lymph nodes): CITE-seq, GSE150599 (ref. 6); dataset 23–24 (human PBMCs): REAP-seq, GSE100501 (ref. 2); dataset 25–26 and dataset 40–41 (human PBMCs): DOGMA-seq, GSE156478 (ref. 18); datasets 27 and 42 (human PBMCs): TEA-seq, GSE158013 (ref. 71); dataset 28 (human PBMCs): inCITE-seq, GSE163480 (ref. 85); dataset 29 (skin of mouse): SHARE-seq, GSE140203 (ref. 3); dataset 30 (adult brain of mouse): SHARE-seq, GSE140203 (ref. 3); dataset 31 (adult brain of mouse): SNARE-seq, GSE126074 (ref. 4); dataset 32 (adult brain of mouse): ISSAAC-seq, https://www.ebi.ac.uk/biostudies/arrayexpress/studies/E-MTAB-11264/ (ref. 12); dataset 33 (adult brain of mouse): 10x Multiome, https://www.10xgenomics.com/resources/datasets/frozen-human-healthy-brain-tissue-3-k-1-standard-2-0-0/; dataset 34 (10,000 PBMCs with granulocytes removed): 10x Multiome, https://www.10xgenomics.com/ resources/datasets/pbmc-from-a- healthy-donor-granulocytes-removed-through-cell-sorting-10-k-1-standard-2-0-0/; dataset 35 (3,000 PBMCs with granulocytes removed): 10x Multiome, https://www.10xgenomics.com/resources/ datasets/pbmc-from-a-healthy-donor-granulocytes-removed-through-cell-sorting-3-k- 1-standard-2-0-0/; dataset 36 (10,000 PBMCs): 10x Multiome, https://www.10xgenomics.com/resources/datasets/pbmc- from-a-healthy-donor-no-cell- sorting-10-k-1-standard-2-0-0/; dataset 37 (3,000 PBMCs): 10x Multiome, https://www.10xgenomics.com/resources/ datasets/pbmc-from-a-healthy-donor-no- cell-sorting-3-k-1-standard-2-0-0/; dataset 38 (mouse retina): 10x Multiome, GSE201402 (ref. 86); dataset 39 (human BMMCs): 10x Multiome, GSE194122 (ref. 79); dataset 43 (mouse spleen): scRNA-seq, GSE132901 (ref. 87); dataset 44 (mouse retain): scRNA-seq, GSE181251 (ref. 88); dataset 45 (mouse adult brain): scRNA-seq, GSE246147 (ref. 89); dataset 46 (mouse HSPCs): scRNA-seq, GSE175702 (ref. 82); dataset 47 (mouse retain): scATAC-seq, GSE181251 (ref. 88). Source data are provided with this paper.

Code availability

We have uploaded the codes and scripts used for the benchmark study and figure plotting to a GitHub website, which can be accessed at https://github.com/QuKunLab/MultiomeBenchmarking/. Code is also available in the Zenodo repository via https://doi.org/10.5281/zenodo.10540843 (ref. 90).

References

71. Swanson, E. et al. Simultaneous trimodal single-cell measurement of transcripts, epitopes, and chromatin accessibility using TEA-seq. *eLife* **10**, e63632 (2021).
72. Hand, D. J. & Till, R. J. A simple generalisation of the area under the ROC curve for multiple class classification problems. *Mach. Learn.* **45**, 171–186 (2001).
73. Hubert, L. & Arabie, P. Comparing partitions. *J. Classif.* **2**, 193–218 (1985).
74. Strehl, A. & Ghosh, J. Cluster ensembles–a knowledge reuse framework for combining multiple partitions. *J. Mach. Learn. Res.* **3**, 583–617 (2002).
75. Rousseeuw, P. J. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *J. Comput. Appl. Math.* **20**, 53–65 (1987).
76. Luecken, M. D. et al. Benchmarking atlas-level data integration in single-cell genomics. *Nat. Methods* **19**, 41–50 (2022).
77. Korsunsky, I. et al. Fast, sensitive and accurate integration of single-cell data with Harmony. *Nat. Methods* **16**, 1289–1296 (2019).
78. Büttner, M., Miao, Z., Wolf, F. A., Teichmann, S. A. & Theis, F. J. A test metric for assessing single-cell RNA-seq batch correction. *Nat. Methods* **16**, 43–49 (2019).
79. Luecken, M. D. et al. A sandbox for prediction and integration of DNA, RNA, and proteins in single cells. In *Proc. Neural Information Processing Systems Track on Datasets and Benchmarks* (eds. Vanschoren, J. & Yeung, S.) 13 (NeurIPS, 2021).
80. Kumar, P. et al. Single-cell transcriptomics and surface epitope detection in human brain epileptic lesions identifies pro-inflammatory signaling. *Nat. Neurosci.* **25**, 956–966 (2022).
81. Pombo Antunes, A. R. et al. Single-cell profiling of myeloid cells in glioblastoma across species and disease stage reveals macrophage competition and specialization. *Nat. Neurosci.* **24**, 595–610 (2021).
82. Konturek-Ciesla, A. et al. Temporal multimodal single-cell profiling of native hematopoiesis illuminates altered differentiation trajectories with age. *Cell Rep.* **42**, 112304 (2023).
83. Lukowski, S. W. et al. Absence of Batf3 reveals a new dimension of cell state heterogeneity within conventional dendritic cells. *iScience* **24**, 102402 (2021).
84. Golomb, S. M. et al. Multi-modal single-cell analysis reveals brain immune landscape plasticity during aging and gut microbiota dysbiosis. *Cell Rep.* **33**, 108438 (2020).
85. Chung, H. et al. Joint single-cell measurements of nuclear proteins and RNA in vivo. *Nat. Methods* **18**, 1204–1212 (2021).
86. Dou, J. et al. Bi-order multimodal integration of single-cell data. *Genome Biol.* **23**, 112 (2022).

87. Kimmel, J. C. et al. Murine single-cell RNA-seq reveals cell-identity-and tissue-specific trajectories of aging. *Genome Res.* **29**, 2088–2103 (2019).
88. Lyu, P. et al. Gene regulatory networks controlling temporal patterning, neurogenesis, and cell-fate specification in mammalian retina. *Cell Rep.* **37**, 109994 (2021).
89. Sun, W. et al. Spatial transcriptomics reveal neuron–astrocyte synergy in long-term memory. *Nature* **627**, 374–381 (2024).
90. Hu, Y. et al. Benchmarking algorithms for single-cell multi-omics prediction and integration. *Zenodo* https://doi.org/10.5281/zenodo.10540843 (2024).

Acknowledgements

This work was supported by the National Natural Science Foundation of China grants (T2125012 to K.Q.), the National Key R&D Program of China (2020YFA0112200 and 2022YFA1303200 to K.Q.), the National Natural Science Foundation of China grants (32170668 to B.L.; 12371383 and 61972368 to F.C.), CAS Project for Young Scientists in Basic Research YSBR-005 (to K.Q.), Anhui Province Science and Technology Key Program (202003a07020021 to K.Q.), the Fundamental Research Funds for the Central Universities (YD2070002019, WK9110000141 and WK2070000158 to K.Q.; WK0010000085 to Y.H.), Anhui Provincial Natural Science Foundation (2308085QA07 to Y.H.) and China Postdoctoral Science Foundation (2023M733383 to Y.H.). We thank the USTC supercomputing center and the School of Life Science Bioinformatics Center for providing computing resources for this project.

Author contributions

K.Q., B.L. and F.C. conceived the project. Y.H., S.W. and Y. Luo designed the framework and performed data analysis with help from T.W., S.J., Y.Z., N.L. and Z.Y. Y. Li, W.D. and C.J. contributed in the revision. B.L., Y.H. and K.Q. wrote the manuscript with input from all authors. K.Q. supervised the entire project. All authors read and approved the final manuscript.

Competing interests

The authors declare no competing interests.

Additional information

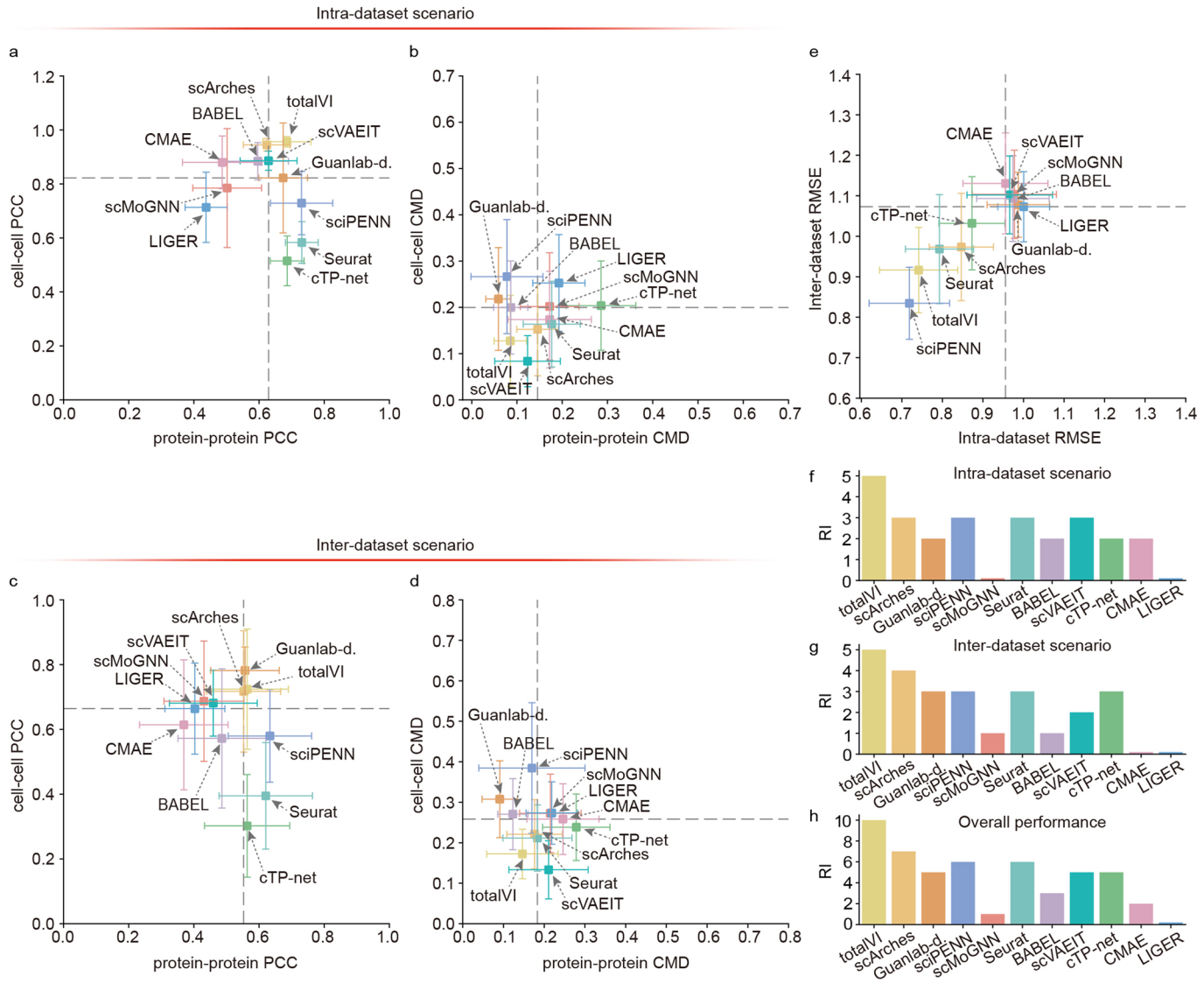
Extended data is available for this paper at https://doi.org/10.1038/s41592-024-02429-w.

Supplementary information The online version contains supplementary material available at https://doi.org/10.1038/s41592-024-02429-w.

Correspondence and requests for materials should be addressed to Falai Chen, Bin Li or Kun Qu.

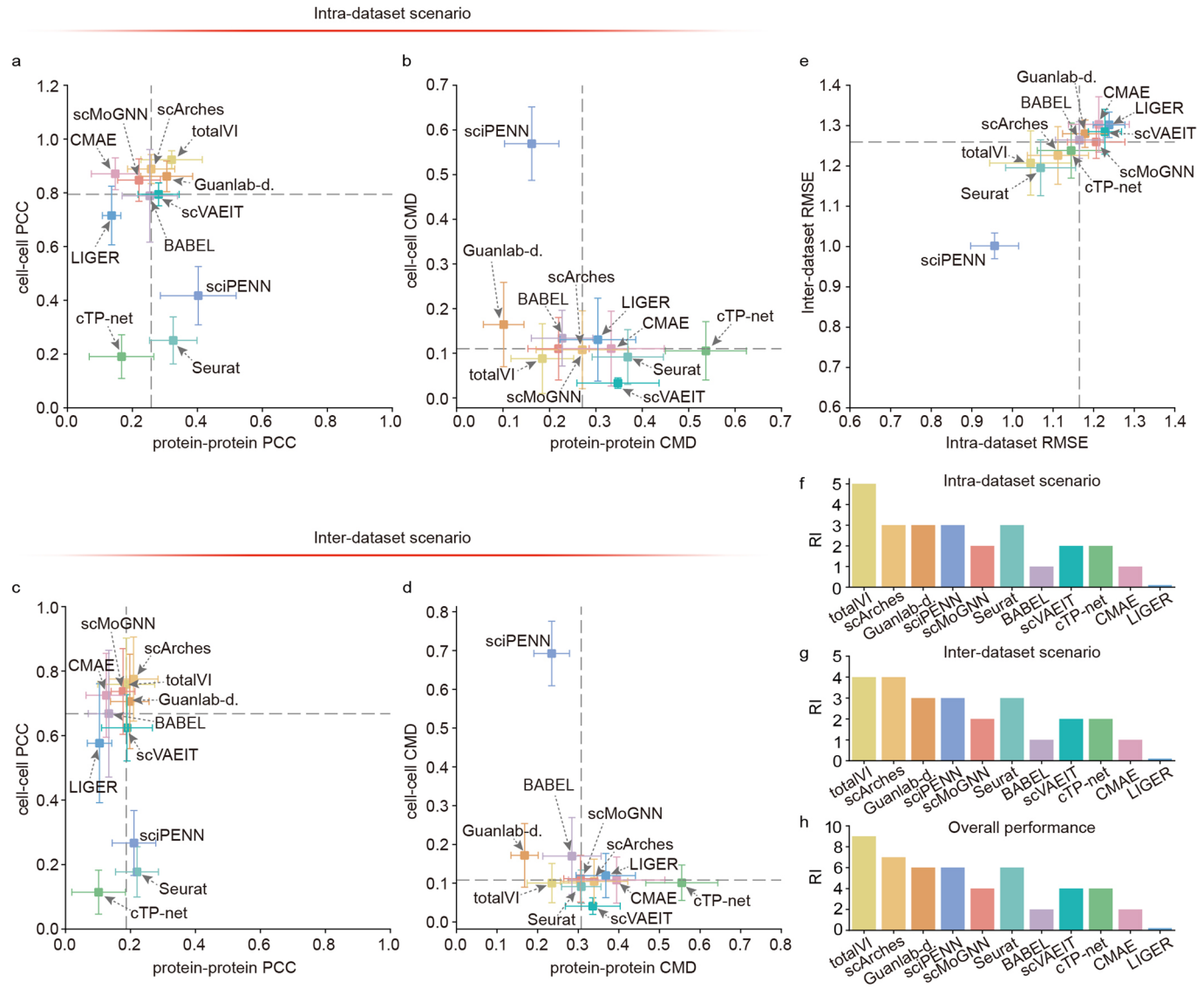
Peer review information *Nature Methods* thanks Jinmiao Chen and the other, anonymous, reviewer(s) for their contribution to the peer review of this work. Peer reviewer reports are available. Primary Handling Editors: Hui Hua and Lin Tang, in collaboration with the *Nature Methods* team.

https://doi.org/10.1038/s41592-024-02429-w



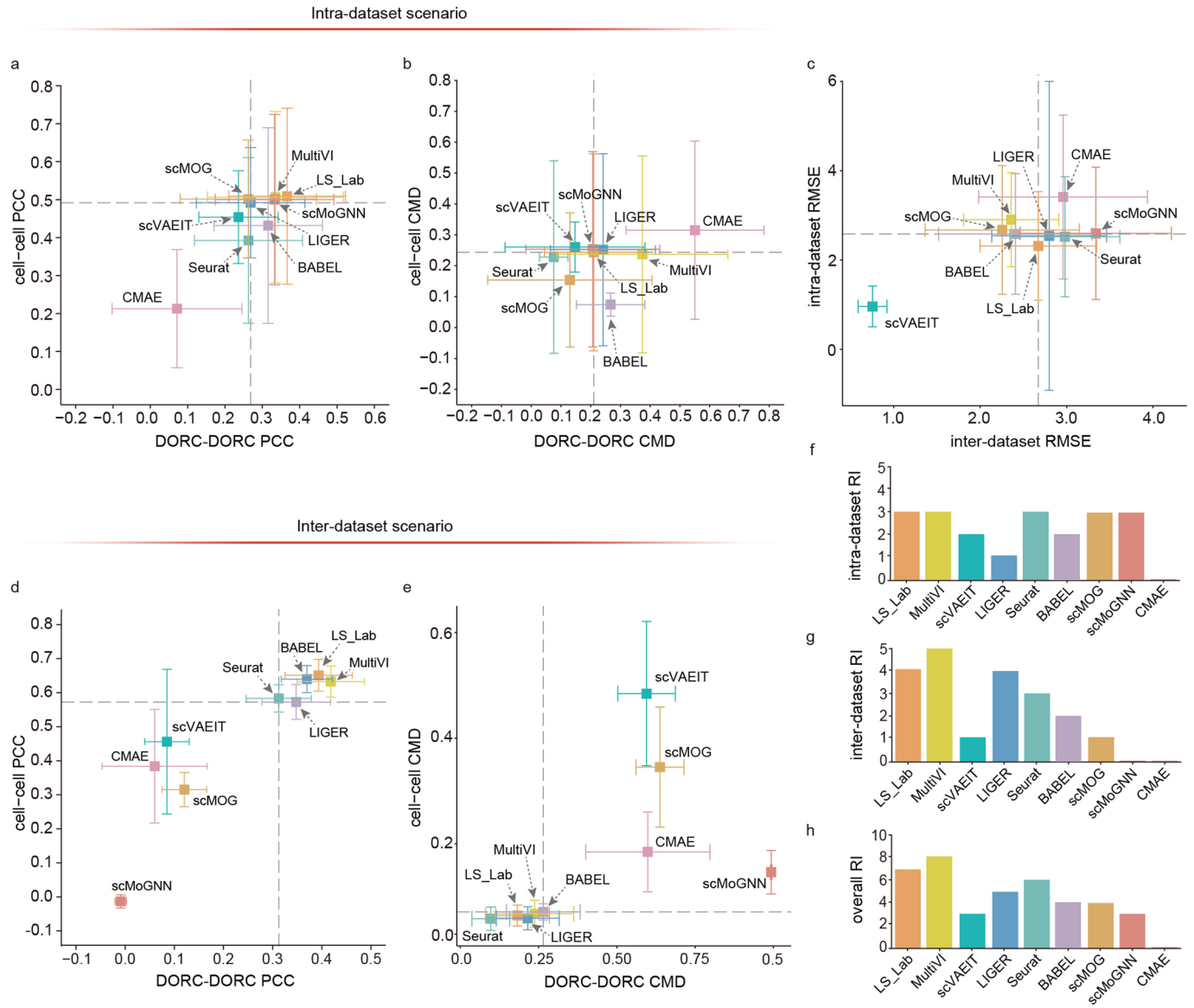
Extended Data Fig. 1 | Performance of eleven algorithms in predicting RC protein abundance from RNA expression. a, **b**, Average PCC (**a**) and CMD (**b**) values between the reference and predicted RC protein expression for the intra-dataset scenario, that is, the training and test sets are from the same datasets. The X and Y axes are the cell–cell and protein–protein PCC/CMD, respectively, and the dashed lines are the medians of all algorithms' results. Error bar: standard deviation of 23 datasets. Data are presented as mean values +/- 0.5xSD. **c**, **d**, Same as (**a**) and (**b**), but the results were predicted for the inter-dataset scenario, that is, the training and test sets are from different datasets. Error bar: standard deviation of 10 datasets. **e**, Average RMSE values between the reference data and the predicted results for the intra-dataset scenario (X axes) and inter-dataset scenario (Y axes). Error bars: standard deviation of 23 datasets (X axes) or 10 datasets (Y axes). Data are presented as mean values +/− 0.5xSD. **f**, **g**, Rank index (RI) values of eleven algorithms in the intra-dataset (**f**) and inter-dataset (**g**) scenarios. **h**, The overall performance of eleven algorithms in both intra-dataset and inter-dataset scenarios. Source data for this figure are provided.

https://doi.org/10.1038/s41592-024-02429-w



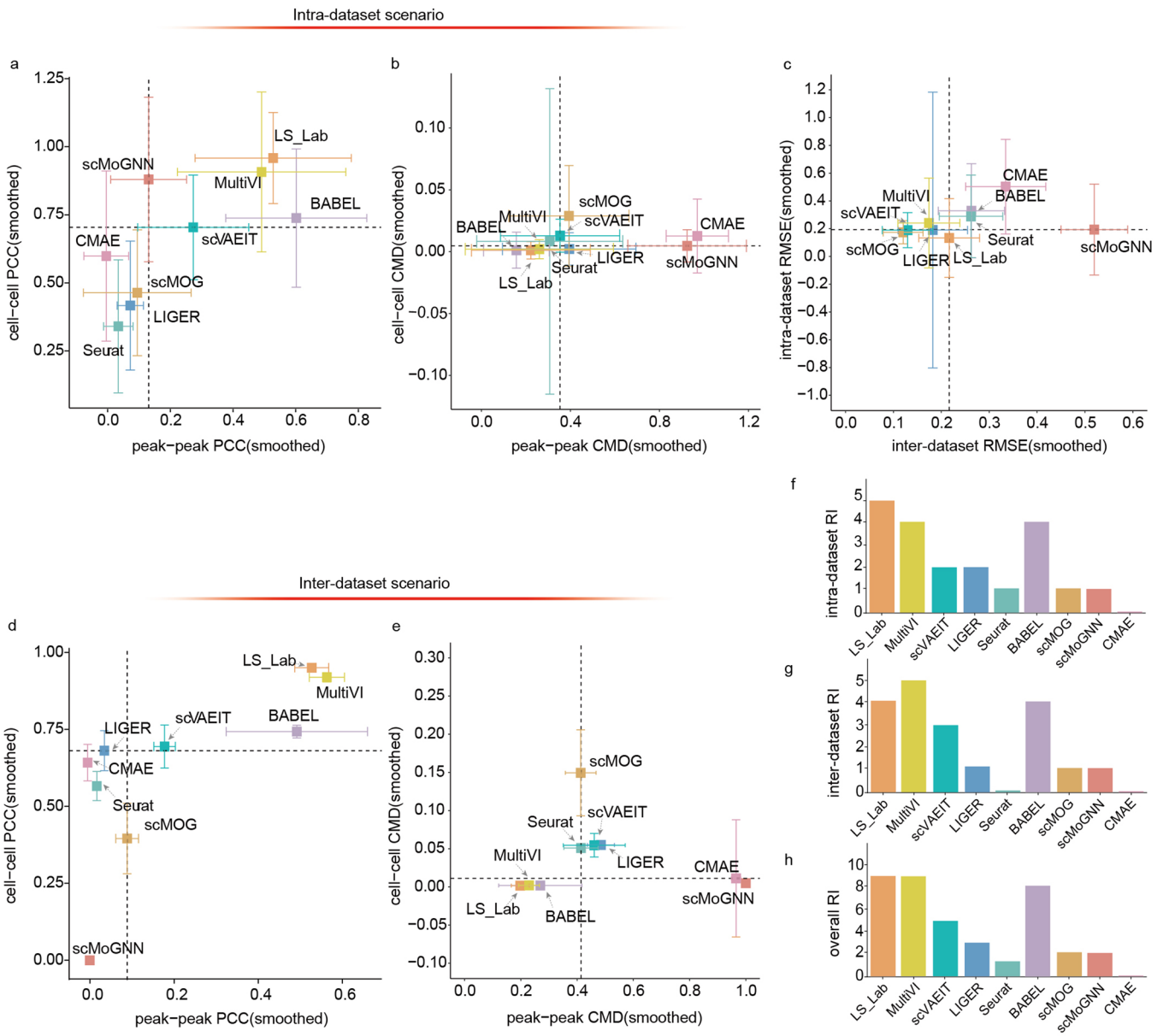
Extended Data Fig. 2 | Performance of eleven algorithms in predicting RU protein abundance from RNA expression. a, **b**, Average PCC (**a**) and CMD (**b**) values between the reference and predicted RU protein abundance for the intra-dataset scenario, that is, the training and test sets are from the same datasets. The X and Y axes are the cell–cell and protein–protein PCC/CMD, respectively, and the dashed lines are the medians of all algorithms' results. Error bar: standard deviation of 23 datasets. Data are presented as mean values +/− 0.5xSD. **c**, **d**, Same as (**a**) and (**b**), but the results were predicted for the inter-dataset scenario, that is, the training and test sets are from different datasets. Error bar: standard deviation of 10 datasets. **e**, Average RMSE values between the reference data and the predicted results for the intra-dataset scenario (X axes) and inter-dataset scenario (Y axes). Error bars: standard deviation of 23 datasets (X axes) or 10 datasets (Y axes). Data are presented as mean values +/− 0.5xSD. **f**, **g**, Rank index (RI) values of seven algorithms in the intra-dataset (**f**) and inter-dataset (**g**) scenarios. **h**, The overall performance of seven algorithms in both intra-dataset and inter-dataset scenarios. Source data for this figure are provided.

https://doi.org/10.1038/s41592-024-02429-w



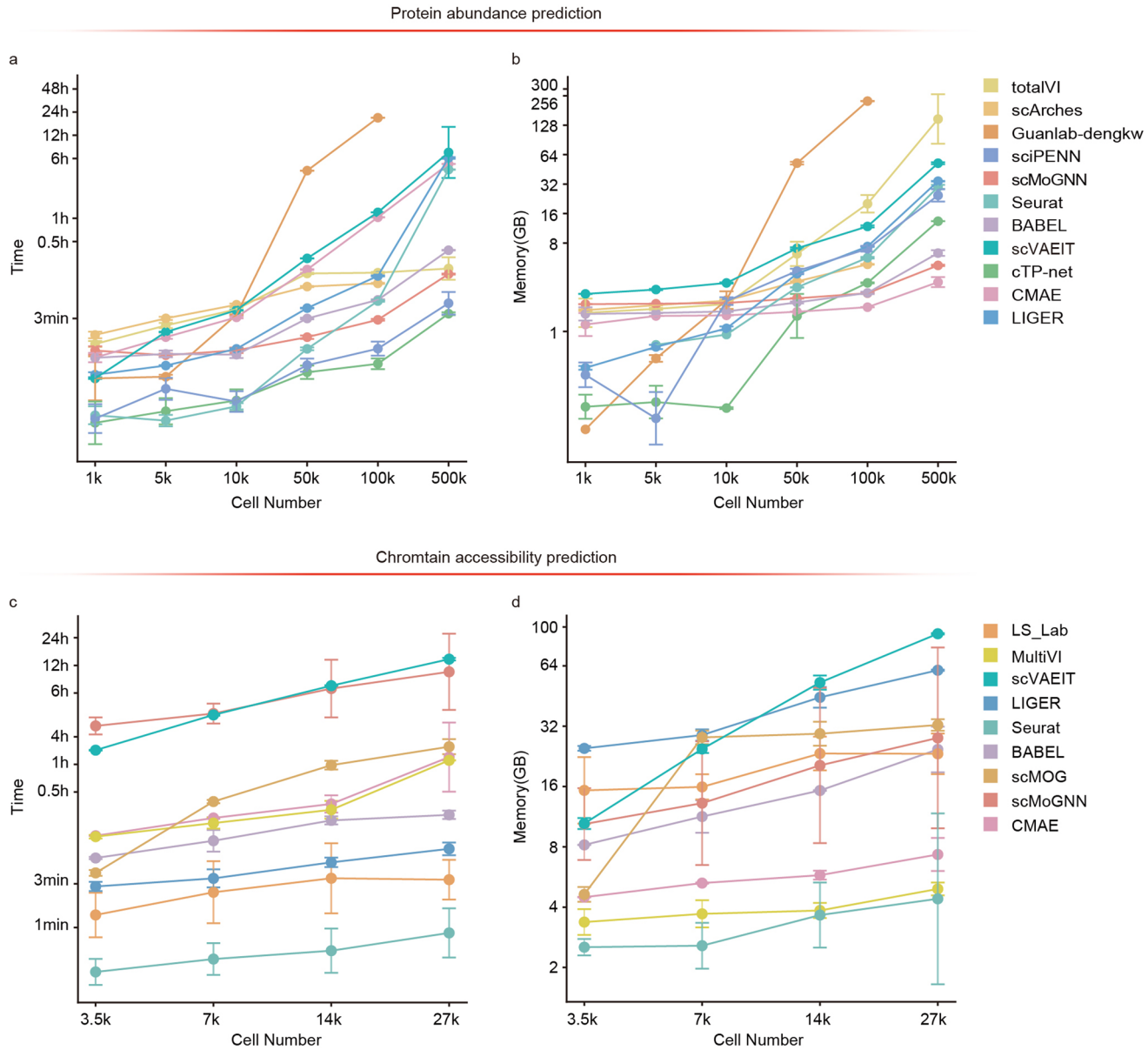
Extended Data Fig. 3 | Performance of nine chromatin accessibility prediction algorithms when converting peaks to DORCs. a, b, Average PCC (**b**) and CMD (**c**) values between the reference data and the predicted results for the intra-dataset scenario, that is, the training and test sets are from the same datasets. The X and Y axes are the cell–cell and DORC-DORC PCC/CMD axes, respectively, and the dashed lines are the medians of all algorithms' results. Error bar: standard deviation of 11 datasets. Data are presented as mean values +/− 0.5xSD. **c**, Average RMSE values between the reference data and the predicted results for the intra-dataset scenario (X axes) and inter-dataset scenario (Y axes). Error bar: standard deviation of 11 datasets (X axes) or 8 datasets (Y axes). Data are presented as mean values +/− 0.5xSD. **d, e**, Same as (**a**) and (**b**), but the results were predicted for the inter-dataset scenario, that is, the training and test sets are from different datasets. Error bar: standard deviation of 8 datasets. **f**, **g**, Rank index (RI) values of nine algorithms in the intra-dataset (**e**) and inter-dataset (**f**) scenarios. **h**, The overall performance of nine algorithms in both intra-dataset and inter-dataset scenarios. Source data for this figure are provided.

https://doi.org/10.1038/s41592-024-02429-w



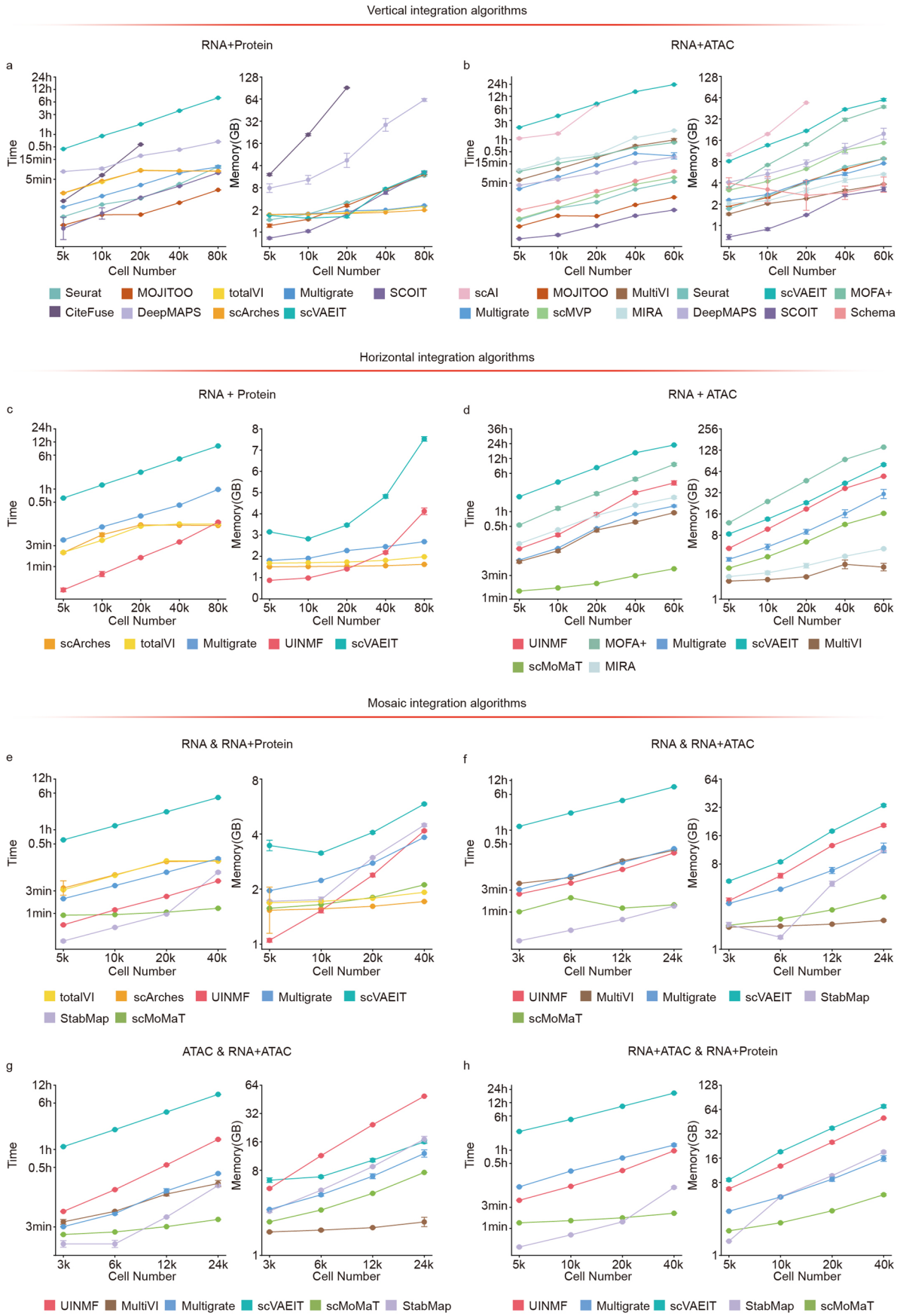
Extended Data Fig. 4 | Performance of nine chromatin accessibility prediction algorithms when using smoothed ATAC-seq matrix. a, b, Average PCC (**b**) and CMD (**c**) values between the KNN-smoothing reference data and the predicted results for the intra-dataset scenario, that is, the training and test sets are from the same datasets. The X and Y axes are the cell–cell and peak-peak PCC/CMD, respectively, and the dashed lines are the medians of all the algorithm results. Error bar: standard deviation of 11 datasets. Data are presented as mean values +/− 0.5xSD. **c**, Average RMSE values between the KNN-smoothing reference data and the predicted results for the intra-dataset scenario (X axes) and inter-dataset scenario (Y axes). Error bar: standard deviation of 11 datasets (X axes) or 8 datasets (Y axes). Data are presented as mean values +/− 0.5xSD. **d, e**, Same as (**a**) and (**b**), but the results were predicted for the inter-dataset scenario, that is, the training and test sets are from different datasets. Error bar: standard deviation of 8 datasets. **f, g**, Rank index (RI) values of nine algorithms in the intra-dataset (**f**) and inter-dataset (**g**) scenarios. **h**, The overall performance of nine algorithms in both intra-dataset and inter-dataset scenarios. Source data for this figure are provided.

https://doi.org/10.1038/s41592-024-02429-w



Extended Data Fig. 5 | Computational resources consumed by the fourteen multi-omics prediction algorithms. a, **b**, The computational time and memory cost of eleven algorithms for predicting protein abundance in datasets with different numbers of cells. Guanlab-dengkw and scArches reported memory errors and stopped when processing the dataset with 500k cells. Error bar: standard deviation of 5 down-samplings and 2 tests. Data are presented as mean values +/− 0.5xSD. **c**, **d**, The computer time and memory cost of nine algorithms for predicting chromatin accessibility in datasets with different numbers of cells. Error bar: standard deviation of 5 down-samplings and 2 tests. Data are presented as mean values +/− 0.5xSD. Source data for this figure are provided.

https://doi.org/10.1038/s41592-024-02429-w

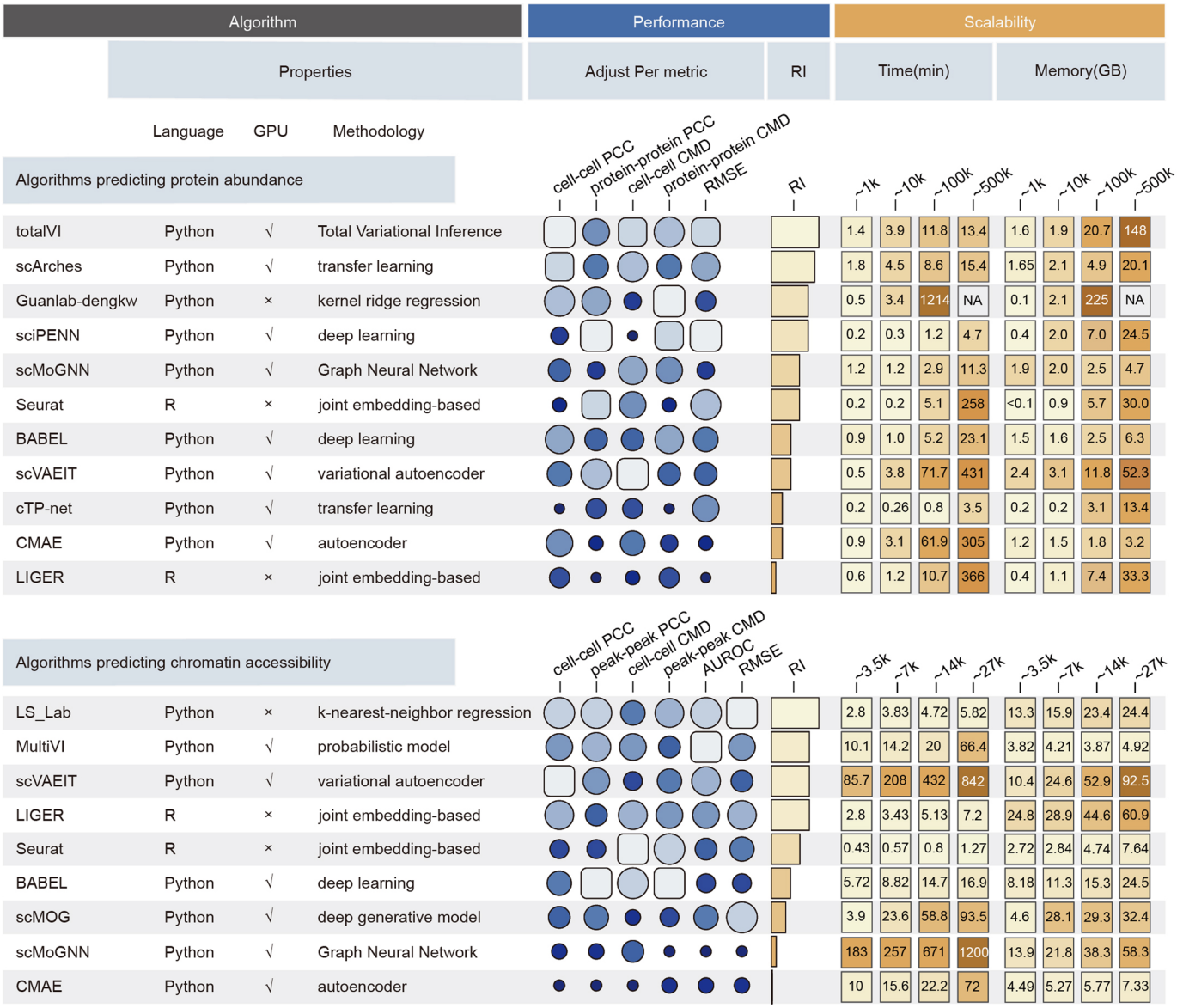


Extended Data Fig. 6 | See next page for caption.

https://doi.org/10.1038/s41592-024-02429-w

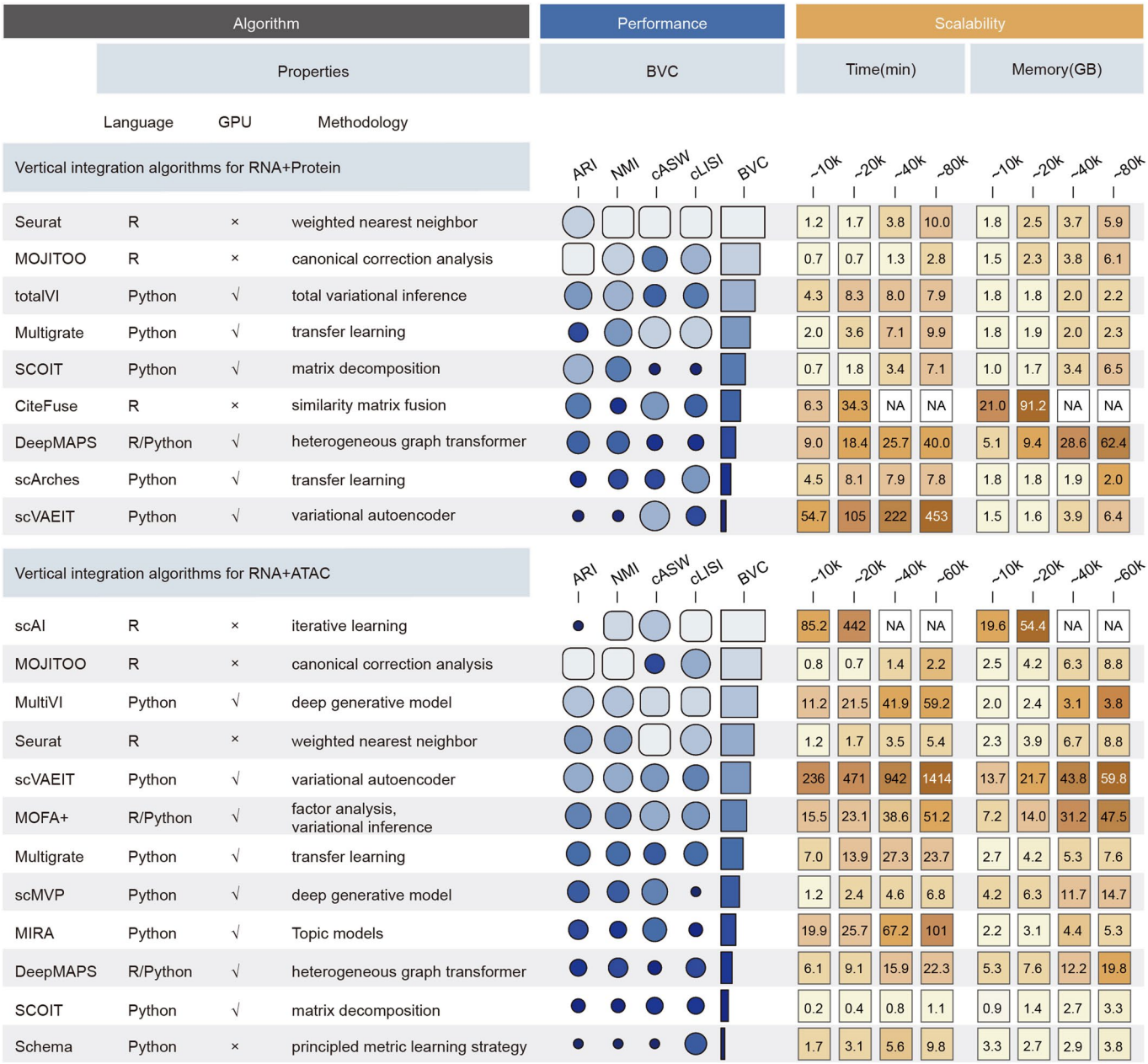
Extended Data Fig. 6 | Computational resources consumed by eighteen single-cell multi-omics integration algorithms. a, Computer time and memory used by nine vertical integration algorithms when integrating RNA expression and protein abundance for datasets with different numbers of cells. CiteFuse reported memory errors and stopped when processing datasets with over 20k cells. Error bar: standard deviation of 5 down-samplings and 2 tests. Data are presented as mean values +/− 0.5xSD. **b**, Same as (**a**), but the results were generated by twelve vertical integration algorithms when integrating RNA expression and chromatin accessibility. scAI reported memory errors and stopped when processing datasets with over 20k cells. **c**, Computer time and memory cost of five horizontal integration algorithms when integrating single-cell RNA+Protein data for datasets with different numbers of cells. Error bar: standard deviation of 5 down-samplings and 2 tests. Data are presented as mean values +/− 0.5xSD. **d**, Same as (**c**), but the results were generated by seven horizontal integration algorithms when integrating single-cell RNA + ATAC data. **e**, Computer time and memory cost of seven mosaic integration algorithms when integrating scRNA-seq and single-cell RNA+Protein data for datasets with different numbers of cells. Error bar: standard deviation of 5 down-samplings and 2 tests. Data are presented as mean values +/− 0.5xSD. **f**-**h**, Same as (**e**), but the results were generated by mosaic integration algorithms when integrating scRNA-seq data and single-cell RNA + ATAC data (**b**), integrating scATAC-seq data and single-cell RNA + ATAC data (**c**), and integrating single-cell RNA+Protein data and single-cell RNA + ATAC data (**d**). Source data for this figure are provided.

https://doi.org/10.1038/s41592-024-02429-w



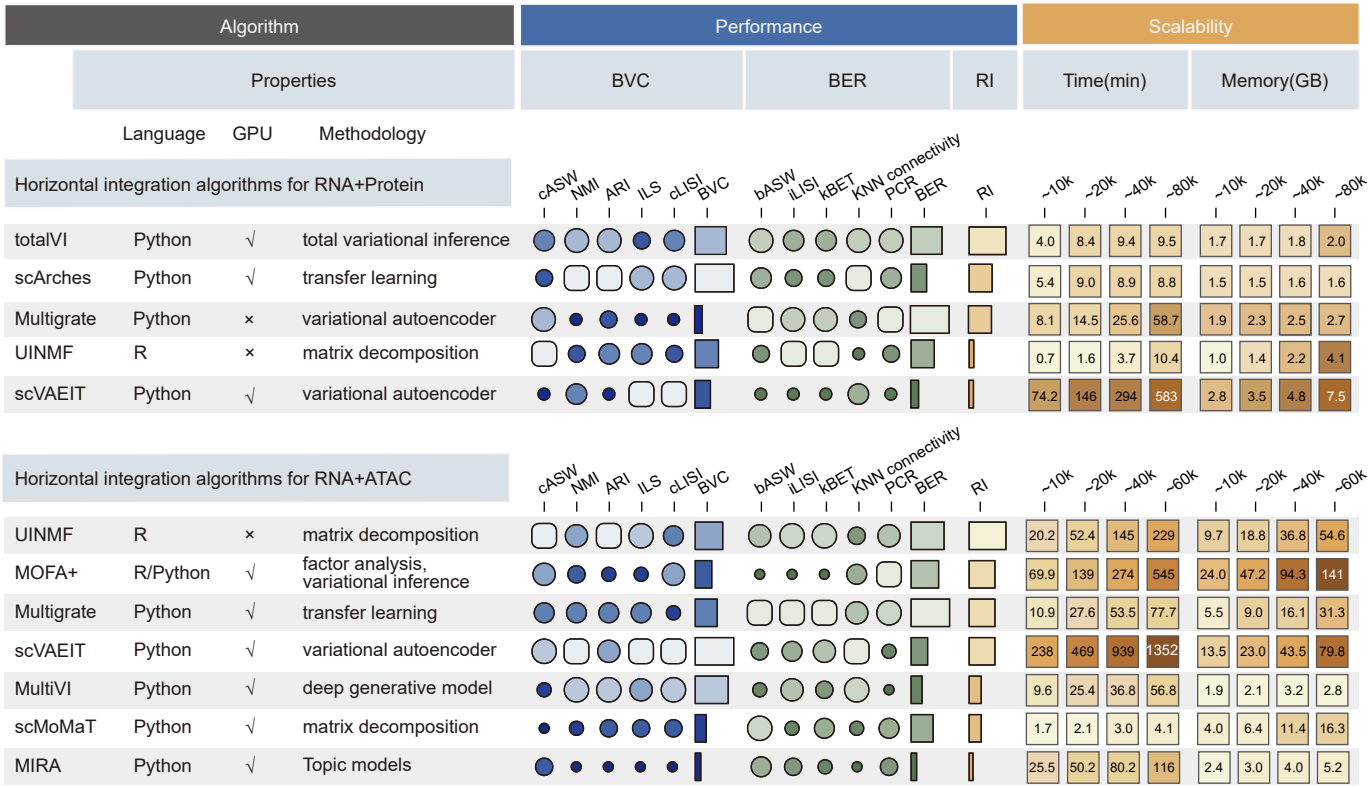
Extended Data Fig. 7 | Summary of the performance of the fourteen multi-omics prediction algorithms. The figure shows: (i) the properties of these algorithms, including the programming languages, methodologies, and GPU acceleration requirements. (ii) the overall performance of these algorithms, evaluated by six metrics in both the inter- and intra-scenarios. A lighter color (and/or a larger dot) indicates better performance for a given metrics. (iii) the computer time and memory consumed by these algorithms for different sizes of datasets; 'NA' indicates a memory error or invalid result. Source data for this figure are provided.

https://doi.org/10.1038/s41592-024-02429-w



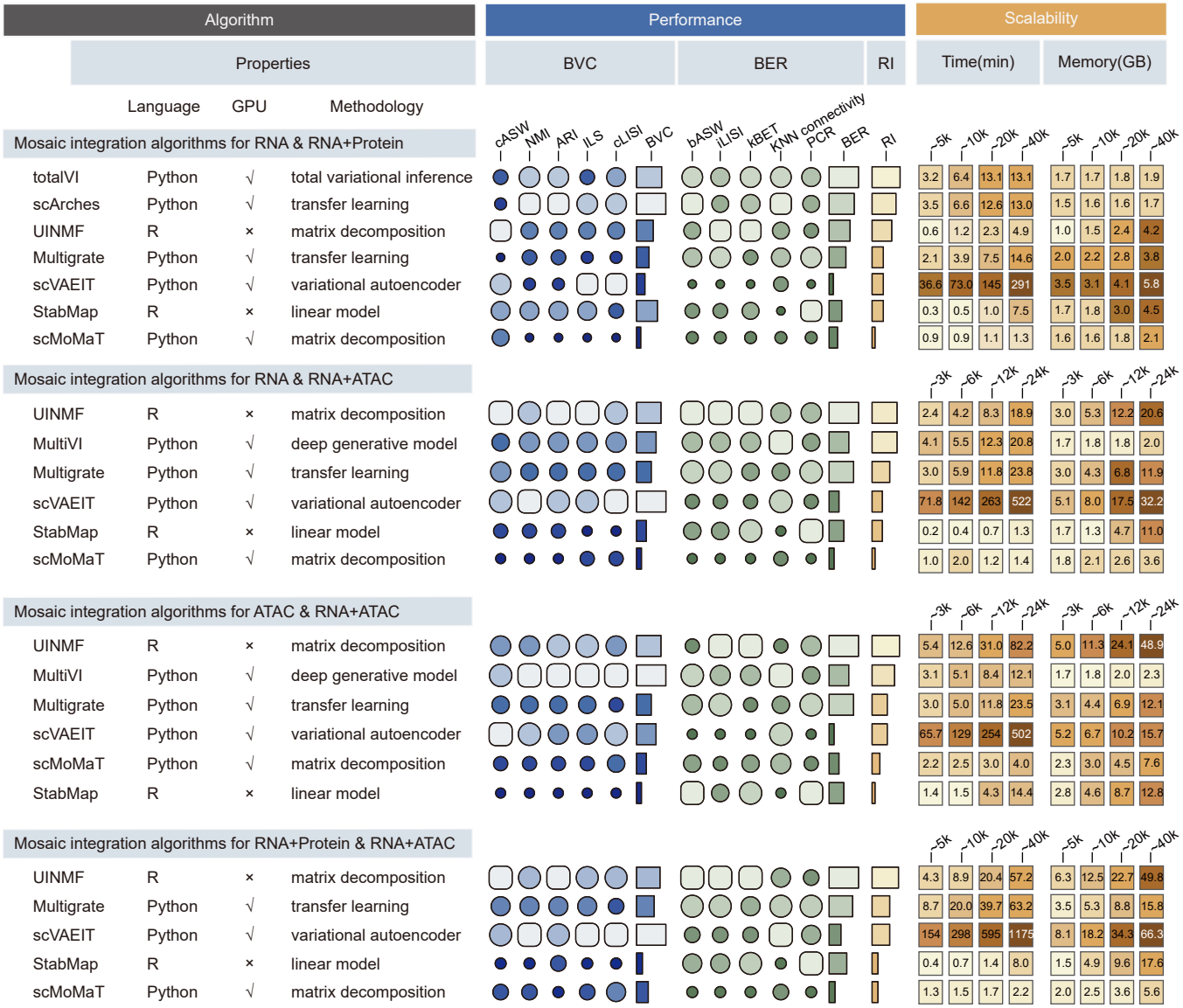
Extended Data Fig. 8 | Summary of the performance of the fifteen vertical integration algorithms. The figure shows: (i) the properties of these algorithms, including the programming languages, methodologies, and GPU acceleration requirements; (ii) the overall performance of these algorithms, evaluated by four metrics. (iii) the computer time and memory consumed by these algorithms for different sizes of datasets; 'NA' indicates a memory error or invalid result. Source data for this figure are provided.

https://doi.org/10.1038/s41592-024-02429-w



Extended Data Fig. 9 | Summary of the performance of nine horizontal integration algorithms. The figure shows: (i) the properties of these algorithms, including the programming languages, methodologies, and GPU acceleration requirements; (ii) the overall performance of these algorithms, evaluated by ten metrics in both the inter- and intra-scenarios. (iii) the computer time and memory consumed by these algorithms for different sizes of datasets; 'NA' indicates a memory error or invalid result. Source data for this figure are provided.

https://doi.org/10.1038/s41592-024-02429-w



Extended Data Fig. 10 | Summary of the performance of eight mosaic integration algorithms. The figure shows: (i) the properties of these algorithms, including the programming languages, methodologies, and GPU acceleration requirements; (ii) the overall performance of these algorithms, evaluated by ten metrics in both the inter- and intra-scenarios. (iii) the computer time and memory consumed by these algorithms for different sizes of datasets; 'NA' indicates a memory error or invalid result. Source data for this figure are provided.

nature portfolio

Corresponding author(s): Kun Qu

Last updated by author(s): Jun 21, 2024

Reporting Summary

Nature Portfolio wishes to improve the reproducibility of the work that we publish. This form provides structure for consistency and transparency in reporting. For further information on Nature Portfolio policies, see our Editorial Policies and the Editorial Policy Checklist.

Statistics

For all statistical analyses, confirm that the following items are present in the figure legend, table legend, main text, or Methods section.

n/a	Confirmed
☐	☒ The exact sample size (*n*) for each experimental group/condition, given as a discrete number and unit of measurement
☐	☒ A statement on whether measurements were taken from distinct samples or whether the same sample was measured repeatedly
☐	☒ The statistical test(s) used AND whether they are one- or two-sided *Only common tests should be described solely by name; describe more complex techniques in the Methods section.*
☐	☒ A description of all covariates tested
☐	☒ A description of any assumptions or corrections, such as tests of normality and adjustment for multiple comparisons
☐	☒ A full description of the statistical parameters including central tendency (e.g. means) or other basic estimates (e.g. regression coefficient) AND variation (e.g. standard deviation) or associated estimates of uncertainty (e.g. confidence intervals)
☐	☒ For null hypothesis testing, the test statistic (e.g. *F*, *t*, *r*) with confidence intervals, effect sizes, degrees of freedom and *P* value noted *Give P values as exact values whenever suitable.*
☒	☐ For Bayesian analysis, information on the choice of priors and Markov chain Monte Carlo settings
☒	☐ For hierarchical and complex designs, identification of the appropriate level for tests and full reporting of outcomes
☐	☒ Estimates of effect sizes (e.g. Cohen's *d*, Pearson's *r*), indicating how they were calculated

Our web collection on statistics for biologists contains articles on many of the points above.

Software and code

Policy information about availability of computer code

Data collection: No software was used for data collection.

Data analysis

We compared the performance of 12 multi-omics prediction algorithms: BABEL(integrated in DANCE v1.0.0rc0), CMAE(integrated in DANCE v1.0.0rc0), MultiVi(v0.19.0), LS_Lab (updated Feb 7, 2022), scMoGNN(integrated in DANCE v1.0.0rc0), Seurat(v4.3.0.1), LIGER(v1.0.1), totalVI(v0.19.0), scArches(v0.5.9), cTP-net(v0.2.4), sciPENN(v0.9.6), scVAEIT(v0.0.0), scMOG(updated May 17, 2023), and Guanlab-dengkw (updated Mar 16, 2022).

We also compared the performance of 18 multi-omics integration algorithms: totalVI(v0.19.0), scAI(v1.0.0), MOFA+(v0.7.0), Seurat(v4.3.0.1), scArches(v0.5.9), CiteFuse(v1.10.0), SCOIT(v0.1.2.1), MOJITOO(v1.0), MultiVI(v0.19.0), DeepMAPS(v1.16.0), Schema(v0.1.5.5), MIRA(v2.1.0), scMVP(v0.0.1), scVAEIT(v0.0.0), Multigrate(v0.0.2), UINMF(v1.0.1), scMoMaT (v0.2.2), StabMap (v0.1.8).

For peak calling, cisTopic (v0.1) was employed for SNARE-seq datasets; Cell Ranger ATAC (v2.0.0) was used for ISSAAC-seq datasets; Cell Ranger (v3.0) and Cell Ranger ARC (v1.0.0) were used for DOGMA-seq datasets; Cell Ranger ARC (v1.0.0) was used for TEA-seq datasets; Cell Ranger ARC (v2.0.0) was used for 10x Multiome datasets; and MACS2 (v2.1.2) was used for SHARE-seq technology datasets.

We utilized Signac (v1.2.0) to calculate the DORC, merge peaks, and create a unified peak set for multi-dataset analysis. Scanpy (v1.9.4) was used to preprocess the data, and scIB (v1.1.4) was employed to calculate all metrics for evaluating the multi-omics integration algorithms.

The code used in this paper is available at https://github.com/QuKunLab/MultiomeBenchmarking.
The files in prediction_env and integration_env provided the software dependencies along with their version numbers.

For manuscripts utilizing custom algorithms or software that are central to the research but not yet described in published literature, software must be made available to editors and reviewers. We strongly encourage code deposition in a community repository (e.g. GitHub). See the Nature Portfolio guidelines for submitting code & software for further information.

Data

Policy information about availability of data

All manuscripts must include a data availability statement. This statement should provide the following information, where applicable:

- Accession codes, unique identifiers, or web links for publicly available datasets
- A description of any restrictions on data availability
- For clinical datasets or third party data, please ensure that the statement adheres to our policy

Dataset 1: CITE-seq, available at https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE128639.
Dataset 2: CITE-seq, available at https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE194122.
Dataset 3: CITE-seq, available at https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE201048.
Dataset 4: CITE-seq, available at https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE100866.
Dataset 5: CITE-seq, available at https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSM4972212.
Dataset 6: CITE-seq, available at https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE163120.
Dataset 7: CITE-seq, available at https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE175702.
Dataset 8: CITE-seq, available at https://support.10xgenomics.com/single-cell-gene-expression/datasets/3.0.0/malt_10k_protein_v3
Dataset 9-10: CITE-seq, available at https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE149544.
Dataset 11: CITE-seq, available at https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE148127.
Dataset 12-13: CITE-seq, available athttps://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE164378.
Dataset 14-15: CITE-seq, available at https://zenodo.org/record/6348128#.Y5f40LJBzDU.
Dataset 21-22: CITE-seq, available at https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE150599.
Dataset 23-24: REAP-seq, available at https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE100501.
Dataset 25-26 and Dataset 40-41: DOGMA-seq, available at https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE156478.
Dataset 27 and Dataset 42: TEA-seq, available at https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE158013.
Dataset 28: inCITE-seq, available at https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE163480.
Dataset 29-30: SHARE-seq, available at https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE140203.
Dataset 31: SNARE-seq, available at https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE126074.
Dataset 32: ISSAAC-seq, available at https://www.ebi.ac.uk/biostudies/arrayexpress/studies/E-MTAB-11264.
Dataset 33: 10x Multiome, available at https://www.10xgenomics.com/resources/datasets/frozen-human-healthy-brain-tissue-3-k-1-standard-2-0-0.
Dataset 34: 10x Multiome, available at https://www.10xgenomics.com/resources/datasets/pbmc-from-a-healthy-donor-granulocytes-removed-through-cell-sorting-10-k-1-standard-2-0-0.
Dataset 35: 10x Multiome, available at https://www.10xgenomics.com/resources/datasets/pbmc-from-a-healthy-donor-granulocytes-removed-through-cell-sorting-3-k-1-standard-2-0-0.
Dataset 36: 10x Multiome, available at https://www.10xgenomics.com/resources/datasets/pbmc-from-a-healthy-donor-no-cell-sorting-10-k-1-standard-2-0-0.
Dataset 37: 10x Multiome, available at https://www.10xgenomics.com/resources/datasets/pbmc-from-a-healthy-donor-no-cell-sorting-3-k-1-standard-2-0-0.
Dataset 38: 10x Multiome, available at https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE201402.
Dataset 39: 10x Multiome, available at https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE194122.
Dataset 43: scRNA-seq, available at https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE132901.
Dataset 44: scRNA-seq, available at https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE181251.
Dataset 45: scRNA-seq, available at https://0-www-ncbi-nlm-nih-gov.brum.beds.ac.uk/geo/query/acc.cgi?acc=GSE246147.
Dataset 46: scRNA-seq, available at https://0-www-ncbi-nlm-nih-gov.brum.beds.ac.uk/geo/query/acc.cgi?acc=GSE175702.
Dataset 47: scATAC-seq, available at https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE181251.

The raw data of these datasets are also available at : https://mailustceducn-my.sharepoint.com/:f:/g/personal/hyl2016_mail_ustc_edu_cn/EgYFP7tITKBBuAhkdtrIOg4B1Eyo-_iBx1VKBWSK0r-9rA?e=gmhocx. A summary of these datasets is given in Supplementary Table 2-3.

Human research participants

Policy information about studies involving human research participants and Sex and Gender in Research.

Reporting on sex and gender	Not applicable
Population characteristics	Not applicable
Recruitment	Not applicable
Ethics oversight	Not applicable

Note that full information on the approval of the study protocol must also be provided in the manuscript.

Field-specific reporting

Please select the one below that is the best fit for your research. If you are not sure, read the appropriate sections before making your selection.

☒ Life sciences ☐ Behavioural & social sciences ☐ Ecological, evolutionary & environmental sciences

For a reference copy of the document with all sections, see nature.com/documents/nr-reporting-summary-flat.pdf

Life sciences study design

All studies must disclose on these points even when the disclosure is negative.

Sample size	We used 47 single-cell multiomics datasets from published studies, including 25 RNA+Protein, 11 RNA+ATAC, 3 RNA+ATAC+Protein, 4 scRNA-seq, and 1 scATAC-seq datasets. All the data used are from public domains, and the sample size is determined in the original publication. The RNA+Protein datasets were generated using CITE-seq, inCITE-seq and REAP-seq, the RNA+ATAC datasets were produced using 10X Multiomics, SHARE-seq, SNARE-seq, and ISSAAC-seq, and the RNA+ATAC+Protein datasets were obtained via TEA-seq and DOGMA-seq. The details of these datasets are listed as follows: Dataset 1 has 30672 cells, 17009 RNAs, and 25 Proteins; Dataset 2 has 66175 cells, 13953 RNAs, and 134 Proteins; Dataset 3 has 85000 cells, 22950 RNAs, and 16 Proteins; Dataset 4 has 8613 cells, 36280 RNAs, and 13 Proteins; Dataset 5 has 21589 cells, 14131 RNAs, and 268 Proteins; Dataset 6 has 24559 cells, 12411 RNAs, and 174 Proteins; Dataset 7 has 10473 cells, 18162 RNAs, and 38 Proteins; Dataset 8 has 8412 cells, 33555 RNAs, and 17 Proteins; Dataset 9 has 12224 cells, 16549 RNAs, and 11 Proteins; Dataset 10 has 12119 cells, 15989 RNAs, and 11 Proteins; Dataset 11 has 13052 cells, 19848 RNAs, and 34 Proteins; Dataset 12 has 161764 cells, 33538 RNAs, and 228 Proteins; Dataset 13 has 49147 cells, 33538 RNAs, and 54 Proteins; Dataset 14 has 7108 cells, 33514 RNAs, and 52 Proteins; Dataset 15 has 3362 cells, 33514 RNAs, and 52 Proteins; Dataset 16 has 7865 cells, 33555 RNAs, and 17 Proteins; Dataset 17 has 355433 cells, 15354 RNAs, and 38 Proteins; Dataset 18 has 78531 cells, 26078 RNAs, and 21 Proteins; Dataset 19 has 21043 cells, 26078 RNAs, and 21 Proteins; Dataset 20 has 500089 cells, 33538 RNAs, and 31 Proteins; Dataset 21 has 16828 cells, 13553 RNAs, and 112 Proteins; Dataset 22 has 15820 cells, 13553 RNAs, and 209 Proteins; Dataset 23 has 3158 cells, 32738 RNAs, and 48 Proteins; Dataset 24 has 4330 cells, 32738 RNAs, and 48 Proteins; Dataset 25 has 7624 cells, 36495 RNAs, and 210 Proteins; Dataset 26 has 6139 cells, 36495 RNAs, and 210 Proteins; Dataset 27 has 25517 cells, 17882 RNAs, and 46 Proteins; Dataset 28 has 19279 cells, 12526 RNAs, and 4 Proteins; Dataset 29 has 34774 cells, 23296 RNAs, 338300 peaks; Dataset 30 has 2344 cells, 10203 RNAs, 7622 peaks; Dataset 31 has 8055 cells, 12775 RNAs, 90358 peaks; Dataset 32 has 10361 cells, 15342 RNAs, 169134 peaks; Dataset 33 has 2855 cells, 16910 RNAs, 130862 peaks; Dataset 34 has 10137 cells, 15408 RNAs, 139470 peaks; Dataset 35 has 2592 cells, 11755 RNAs, 80445 peaks; Dataset 36 has 8105 cells, 14184 RNAs, 102360 peaks; Dataset 37 has 2413 cells, 10860 RNAs, 60639 peaks; Dataset 38 has 9383 cells, 6275 RNAs, 59353 peaks;

Dataset 39 has 69249 cells, 13431 RNAs, 103375 peaks;
Dataset 40 has 7624 cells, 28310 RNAs, 68825 peaks;
Dataset 41 has 6139 cells, 28310 RNAs, 68825 peaks;
Dataset 42 has 25517 cells, 17882 RNAs, 128853 peaks;
Dataset 43 has 26843 cells, 8795 RNAs;
Dataset 44 has 19089 cells, 15276 RNAs;
Dataset 45 has 6361 cells, 32877 RNAs;
Dataset 46 has 5579 cells, 16045 RNAs;
Dataset 47 has 25938 cells, 283817 peaks.

Data exclusions

No data was excluded from the study. In the data preprocessing step, we utilized Seurat's quality control filters ('CreateSeuratObject()') with parameters described in the original papers to eliminate low-quality cells, genes, and/or DNA fragments (i.e., "peaks" in scATAC-seq data). See Supplemental Tables 2-3 for more details on these datasets.

Replication

To ensure the reproducibility of our experimental findings, we verified the performance of fourteen multi-omics prediction algorithms and eighteen integration algorithms across multiple datasets under two scenarios. In the test of computational resources, we repeated the sampling process five times for each cell count and conducted the test twice for each dataset to minimize randomness. When examining the effect of varying cell/RNA counts on the performance of prediction algorithms, we replicated this sampling process five times for each cell count/RNA count to ensure the reliability of our findings. Additionally, to assess the impact of the proportion of overlapping cell types between training and test datasets on prediction performance, we constructed five distinct groups of datasets, each with a varying proportion of overlapping cell types, to guarantee robustness.

Randomization

To evaluate the performance of multi-omics prediction algorithms, we adopted two scenarios. In the intra-dataset scenario, we randomly split the cells in the dataset into an 80% training set and a 20% test set, with an equal probability for each cell to be assigned to either set. In the inter-dataset scenario, we used one dataset as the training set and the RNA expression matrix of another dataset as the test set. When examining the effect of varying cell counts, RNA counts, and the proportion of overlapping cell types between training and test datasets on the performance of prediction algorithms, each cell, RNA, or cell type is equally likely to be selected.

Blinding

In the comparison of prediction algorithms, all algorithms were blinded to the ground truth of protein abundance or chromatin accessibility information in the test set. The outputs from these algorithms were then compared to the ground truth available in the respective datasets. Similarly, in the comparison of integration algorithms, all algorithms were blinded to the reference cell type labels. The results generated by these algorithms were then compared to the reference cell type labels available in the respective datasets.

Reporting for specific materials, systems and methods

We require information from authors about some types of materials, experimental systems and methods used in many studies. Here, indicate whether each material, system or method listed is relevant to your study. If you are not sure if a list item applies to your research, read the appropriate section before selecting a response.

Materials & experimental systems

n/a	Involved in the study
☒	☐ Antibodies
☒	☐ Eukaryotic cell lines
☒	☐ Palaeontology and archaeology
☒	☐ Animals and other organisms
☒	☐ Clinical data
☒	☐ Dual use research of concern

Methods

n/a	Involved in the study
☒	☐ ChIP-seq
☒	☐ Flow cytometry
☒	☐ MRI-based neuroimaging