nature methods

Article

https://doi.org/10.1038/s41592-024-02214-9

scPROTEIN: a versatile deep graph contrastive learning framework for single-cell proteomics embedding

Received: 10 January 2023

Accepted: 16 February 2024

Published online: 19 March 2024

Check for updates

Wei Li[1,2,3], Fan Yang[2,3], Fang Wang[2], Yu Rong[2], Linjing Liu[2], Bingzhe Wu[2], Han Zhang[1] ✉ & Jianhua Yao[2] ✉

Single-cell proteomics sequencing technology sheds light on protein–protein interactions, posttranslational modifications and proteoform dynamics in the cell. However, the uncertainty estimation for peptide quantification, data missingness, batch effects and high noise hinder the analysis of single-cell proteomic data. It is important to solve this set of tangled problems together, but the existing methods tailored for single-cell transcriptomes cannot fully address this task. Here we propose a versatile framework designed for single-cell proteomics data analysis called scPROTEIN, which consists of peptide uncertainty estimation based on a multitask heteroscedastic regression model and cell embedding generation based on graph contrastive learning. scPROTEIN can estimate the uncertainty of peptide quantification, denoise protein data, remove batch effects and encode single-cell proteomic-specific embeddings in a unified framework. We demonstrate that scPROTEIN is efficient for cell clustering, batch correction, cell type annotation, clinical analysis and spatially resolved proteomic data exploration.

Until recently, the application of single-cell sequencing technologies mainly focused on the detection of single-cell transcriptome levels[1]; however, single-cell proteomic technologies are now also advancing remarkably[2–15]. While the messenger RNA levels of genes were previously considered to be proxies for protein levels[16,17], the transcriptional levels of genes and the contents of proteins that ultimately perform functions bear low consistency, indicating that the single-cell transcriptome alone is insufficient for deriving cellular protein levels[18,19]. Proteins are the ultimate executors of cellular functions and proteomics provides knowledge for deciphering cellular mechanisms through enzyme activity and posttranslational modifications[9]. Single-cell proteomics enables the identification of activated pathways in therapy-resistant cells and therefore provides biomarkers for cancer diagnosis and prognosis[20]. In recent years, improvements in the sensitivity of mass spectrometry (MS) and innovations in single-cell proteomic technologies have made it possible to quantify a range of proteins at the single-cell resolution[4,12].

Despite the above advantages, single-cell proteomic data have several problems, such as peptide quantification uncertainty, data missingness, batch effects and data noise due to the limitations of the current sample preparation, isotopic labeling methods and data acquisition[21,22]. First, batch effects hinder the analysis of single-cell proteomic data. These batch effects can be internal or external. Internal batch effects include liquid chromatography batch effects[21] and tandem mass tag batch effects[12]. External batch effects arise when integrating datasets from multiple data acquisitions with diverse sample preparation approaches (that is, triethylammonium bicarbonate, dimethyl sulfoxide or water solutions) and labeling strategies (that is, label-free or isotopic labeling). Second, not all constitutive peptide contents are delivered to the mass spectrometer due to the sample loss incurred during sample preparation, which is a crucial issue for single-cell proteomics. After the peptides are injected into the mass spectrometer, their signal intensities are influenced by their

[1]College of Artificial Intelligence, Nankai University, Tianjin, China. [2]AI Lab, Tencent, Shenzhen, China. [3]These authors contributed equally: Wei Li, Fan Yang. ✉e-mail: zhanghan@nankai.edu.cn; jianhuayao@tencent.com

ionization efficiency and peak selection of precursors for fragmentation in the data-dependent acquisition mode[21], potentially leading to noise or missingness in single-cell proteomic data. Third, for bottom-up single-cell proteomics, the core quantification lies at the peptide level. However, the existing single-cell proteomic data processing pipelines do not fully consider this hierarchical content. In practical application, the existing MS acquisition technologies exhibit imperfections in accurately quantifying peptide levels. Consequently, deriving protein contents without accounting for the inherent uncertainty of peptide measurements would yield inaccurate results[7]. To enhance precision in constructing protein abundance data, a recommended approach involves assigning uncertainty weights to peptides. This compensatory measure addresses the inherent inaccuracies associated with the quantification process.

The existing single-cell proteomic data processing methods are mainly migrated from pipelines for single-cell RNA sequencing (scRNA-seq) processing, such as routine imputation (that is, *k*-nearest neighbors (KNN) imputation), batch correction (ComBat) and normalization[21,23,24]. However, these pipelines cannot address the unique data analysis problems involving single-cell proteomic data[21]. First, KNN imputation can introduce large artifacts under severe batch effects in single-cell proteomic data. On the other hand, the existing batch correction method in the single-cell data processing pipeline, that is, ComBat, cannot well alleviate the data missingness problem alone without conducting imputation first. Hence, it would be problematic to conduct imputation and batch correction separately if the influence between them is ignored. Therefore, it is important to consider the influence and interaction between these problems, including simultaneous data denoising[21]. Second, the current batch correction methods in the existing data processing pipeline for single-cell proteomics require specific assumptions that hinder their generalization. For example, ComBat, a popular batch correction method that is adopted in the processing pipeline of the single-cell proteomics by MS (SCoPE2) dataset published by Specht et al.[25] (namely, SCoPE2_Specht; Supplementary Table 1), assumes that the same sets of cells are contained in two batches[21]. This might not be true when the cell types or cell states are unknown before conducting the experiment. Third, single-cell proteomics exhibits a hierarchical structure in which the proteins consist of peptides, and the protein contents are calculated based on the detected peptide signals. However, the existing analysis methods that have been applied to single-cell proteomic data cannot fully exploit this hierarchical information. Although more intricate peptide aggregation methods are available[26,27], they also lack the ability to estimate the uncertainty of peptide quantification.

In this Article, we developed a deep graph contrastive learning framework for single-cell proteomics embedding (scPROTEIN) to address the uncertainty of peptide quantification, data missingness, batch effects and high noise in a unified framework by providing versatile cell embeddings. First, for the datasets provided with raw peptide signal intensities, we proposed a multitask heteroscedastic regression model to estimate the uncertainty of peptide quantification and aggregated the peptide content to the protein level in an uncertainty-guided manner. Then, we built a graph structure to characterize the single-cell proteomic data, where the message passing process considering coexpression patterns helped alleviate the data missingness problem. A graph contrastive learning model with a designed alternating topology-attribute denoising module was developed, which could denoise the proteomic data and lead to an accurate representation. Furthermore, the discriminative property of contrastive learning and the denoising module could implicitly alleviate the batch effect together without knowing the prior knowledge of the dataset. Finally, the learned versatile cell embedding could be applied in various downstream tasks (that is, cell clustering, batch correction and cell type annotation). We also performed experiments on antibody-based single-cell proteomic data to evaluate the effectiveness of the clinical analysis process. Moreover, scPROTEIN could be easily generalized to learn from spatially resolved single-cell proteomic data.

Results

Overview of scPROTEIN

The overall framework of scPROTEIN is illustrated in Fig. 1. For datasets provided with raw peptide-level profiles, scPROTEIN starts from stage 1 (Fig. 1a) to learn the peptide uncertainty and determine the protein-level abundance in an uncertainty-guided manner. The protein-level data are then fed into stage 2 (Fig. 1b) and stage 3 (Fig. 1c) to learn the cell embeddings through graph contrastive learning together with a data denoising module. The learned cell embedding can then be applied in a variety of downstream tasks.

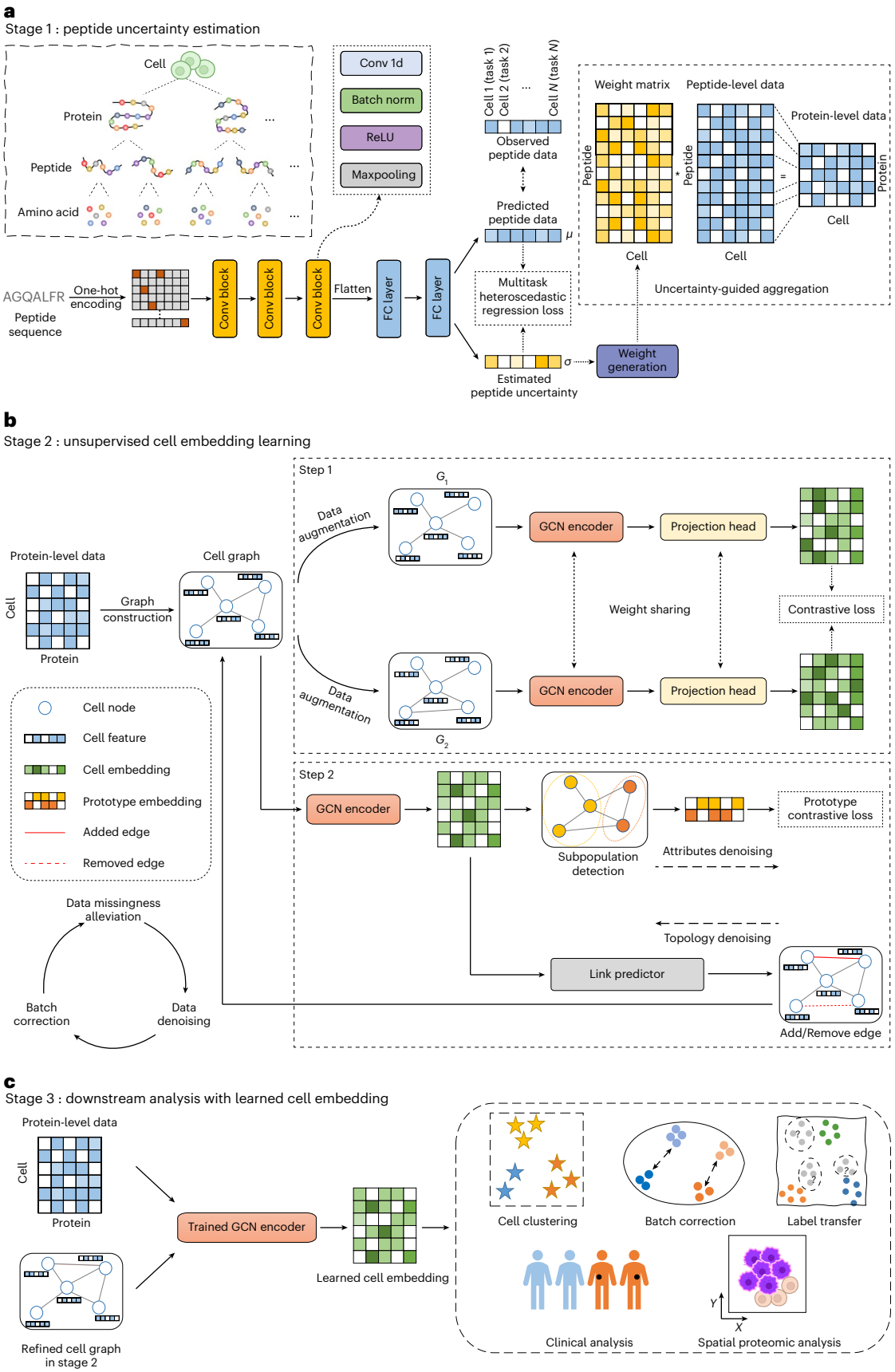
The first stage is to aggregate the peptide-level intensities to determine the protein-level content, and this step is guided by the uncertainty of peptide abundance quantification. We designed a multitask heteroscedastic regression model to estimate the uncertainty of each peptide signal in each cell, which can partially reflect the quality of the signal (that is, the amount of noise). The heteroscedastic regression model[28] assumes that the observation noise varies across different data samples, allowing us to estimate uncertainty levels varying across different peptides in the same cell. By considering the uncertainty estimation for all peptides in one cell as one specific task, our multitask learning method estimates all cells simultaneously. The combination of the heteroscedastic regression model and multitask learning enables the uncertainty of each peptide across different cells to be adaptively learned. As shown in Fig. 1a, the measured peptide quantities are used as the supervision signals. On the basis of the estimated uncertainty, we can then construct a weight for each peptide and perform uncertainty-guided aggregation over the peptide-level data to determine the protein-level abundance. In this way, a peptide signal with higher quality contributes more to its constitutive protein data to eliminate the influence of the inherent noise, and this process benefits the reliability and accuracy of the model. Then, the protein-level data can be used to learn comprehensive cell embeddings in stage 2. For datasets provided directly with their reconstructed protein-level profiles, scPROTEIN starts from stage 2.

The graph learning process within the neighborhood can help alleviate the data missingness problem. Specifically, each node in the graph has a specific semantic pattern that can be used to compensate for the other nodes during the message passing process. To this end, in the second stage, a cell graph is built, where each node represents a cell and the initial node feature represents the proteomic data within the cell. Notably, three characteristics are exhibited by our proposed graph model: (1) in the contrastive learning framework, two different

Fig. 1 | The architecture of scPROTEIN. a, A multitask heteroscedastic regression model for peptide uncertainty estimation. The hierarchical structure of cells, proteins, peptides and amino acids is shown in the left subdiagram. The uncertainty of peptide measurement is estimated in this stage and used to guide the aggregation of the peptide levels for the corresponding protein. The network takes multiple peptide sequences as inputs in each iteration with a minibatch training strategy. One peptide sequence is shown to illustrate the learning process. Conv 1d denotes a one-dimensional convolutional layer, and FC layer stands for a fully connected layer. Right: the aggregation process of using the learned uncertainty to obtain protein-level content. **b**, An unsupervised cell embedding model based on deep contrastive learning. A cell graph is constructed using single-cell proteomic data where cells with similar protein content patterns are connected. A contrastive learning scheme is employed to generate cell embeddings. Attribute denoising and topology denoising are alternated to further improve the embeddings. **c**, Inference used by the trained graph encoder to generate cell embeddings, which are then applied to various downstream tasks.

https://doi.org/10.1038/s41592-024-02214-9

perturbed views are generated, and the mutual information of the same node between these two views is maximized, which can lead to representations robust to noise; (2) in addition, when addressing data from different MS acquisitions, the overlapping proteins are utilized to build a shared cell graph. Therefore, the batch effect among different acquisitions can be implicitly alleviated by aligning the semantic



information of the same cell type through a contrastive loss; and (3) two denoising modules (attribute and topology modules) are alternated to mitigate the noise problem in the proteomic profile. During stage 2, the data missingness problem is alleviated, the batch effect is implicitly corrected and the noise in the given proteomic data can be largely removed.

In stage 3, the topology-refined cell graph obtained in stage 2 and the single-cell protein-level data are fed into the trained graph convolutional network (GCN) encoder from stage 2. Then, cell embeddings are learned, and these embeddings can be applied to various downstream tasks, including cell clustering, batch correction, cell type annotation and clinical analysis. Moreover, our proposed method can be extended to single-cell spatial proteomic data by constructing a cell graph based on spatial cell proximity and learn spatially informative embeddings. A detailed description of the scPROTEIN method is provided in Methods.

To comprehensively evaluate the performance of the proposed approach, we applied scPROTEIN on a range of single-cell proteomic datasets (detailed in Supplementary Table 1). We first showed the overall learning workflow from stage 1 to stage 3 by taking a representative single-cell proteomic dataset (SCoPE2_Specht)[25] as a specific example. Next, we qualitatively and quantitatively compared the cell clustering performance of scPROTEIN with that of other methods and presented the learned peptide uncertainty. We also compared stage 1 of scPROTEIN with other peptide aggregation methods. Then, we benchmarked five independent data integration tasks across six single-cell datasets with different cutting-edge MS acquisition techniques (SCoPE2, prioritized single-cell proteomics (pSCoPE), multiplexed data-independent acquisition (plexDIA), nanodroplet processing in one pot for trace samples (nanoPOTS) and nested nanoPOTS (N2)) and species (mouse and human cell lines). In addition, we investigated the application of scPROTEIN for the single-cell clinical proteomic data contained in the expanded clustered regularly interspaced short palindromic repeats-compatible cellular indexing of transcriptomes and epitopes by sequencing (ECCITE-seq) dataset[29], which is an antibody-based single-cell proteomic dataset collected from patient with cutaneous T cell lymphoma (CTCL). Furthermore, the Basel tissue microarray (BaselTMA) dataset[30], which consists of spatially resolved proteomic data from breast cancer tumor biopsy slices, was utilized to validate the scalability of scPROTEIN in single-cell spatial proteomics tasks. Finally, a systematic analysis of the sensitivity of the hyperparameters in scPROTEIN is presented in Extended Data Figs. 1 and 2.

Cell clustering and peptide uncertainty estimation

We first evaluated the cell embeddings learned by our proposed method in the cell clustering task and illustrated how scPROTEIN works from stage 1 to stage 3. We applied scPROTEIN on the SCoPE2_Specht dataset[25], which quantifies 3,042 proteins in 1,490 cells via the SCoPE2 technique. The existing single-cell proteomics data analysis pipeline[21] employs KNN-based imputation and ComBat-based batch correction (termed KNN–ComBat) for routine data preprocessing. Beyond that, due to the scarcity of single-cell proteomic computational methods, we also compared the performance of scPROTEIN with that of other five methods that are commonly applied to scRNA-seq data (MAGIC[31] and AutoClass[32] for data cleaning and Harmony[33], Scanorama[34] and Liger[35] for batch correction) to provide a more comprehensive evaluation. From the clustering results shown in Fig. 2a,b, we can see that our graph-based embedding approach achieved the best performance in terms of all the evaluation metrics on the SCoPE2_Specht dataset. For Scanorama and Liger, the biological differences between the two cell types were completely lost, resulting in poor adjusted rand index (ARI) (0.002 and 0.003) and normalized mutual information (NMI) (0.001 and 0.002) results. In addition, we illustrate a concrete example of the embedding learning process in Extended Data Fig. 3a.

Furthermore, the entire framework was tailored to analyze the hierarchical data structure of single-cell proteomics, where the abundance of a protein can be aggregated by the levels of its digested peptides. The estimated peptide-level uncertainty reflects the data noise varying across samples, as visualized in Fig. 2c. As the digested peptides obtained from different samples were pooled together after performing isotopic labeling and then detected via MS with consistent ionic behavior[4], the peptides from the same run of experiments displayed similar uncertainty patterns. We can observe four distinct heat map blocks along with their batch IDs in Fig. 2c. In the same proteins, the uncertainty measurements of different peptides varied due to their different ionization, co-isolation, fragmentation and sample preparation loss behaviors[22,36]. For instance, 'AYSSFGGGR_2' and 'DDFNSGFR_2' are two peptides that constitute the same protein, 'Q15056' (in the red box), but they exhibited very different uncertainty patterns across different batches and samples.

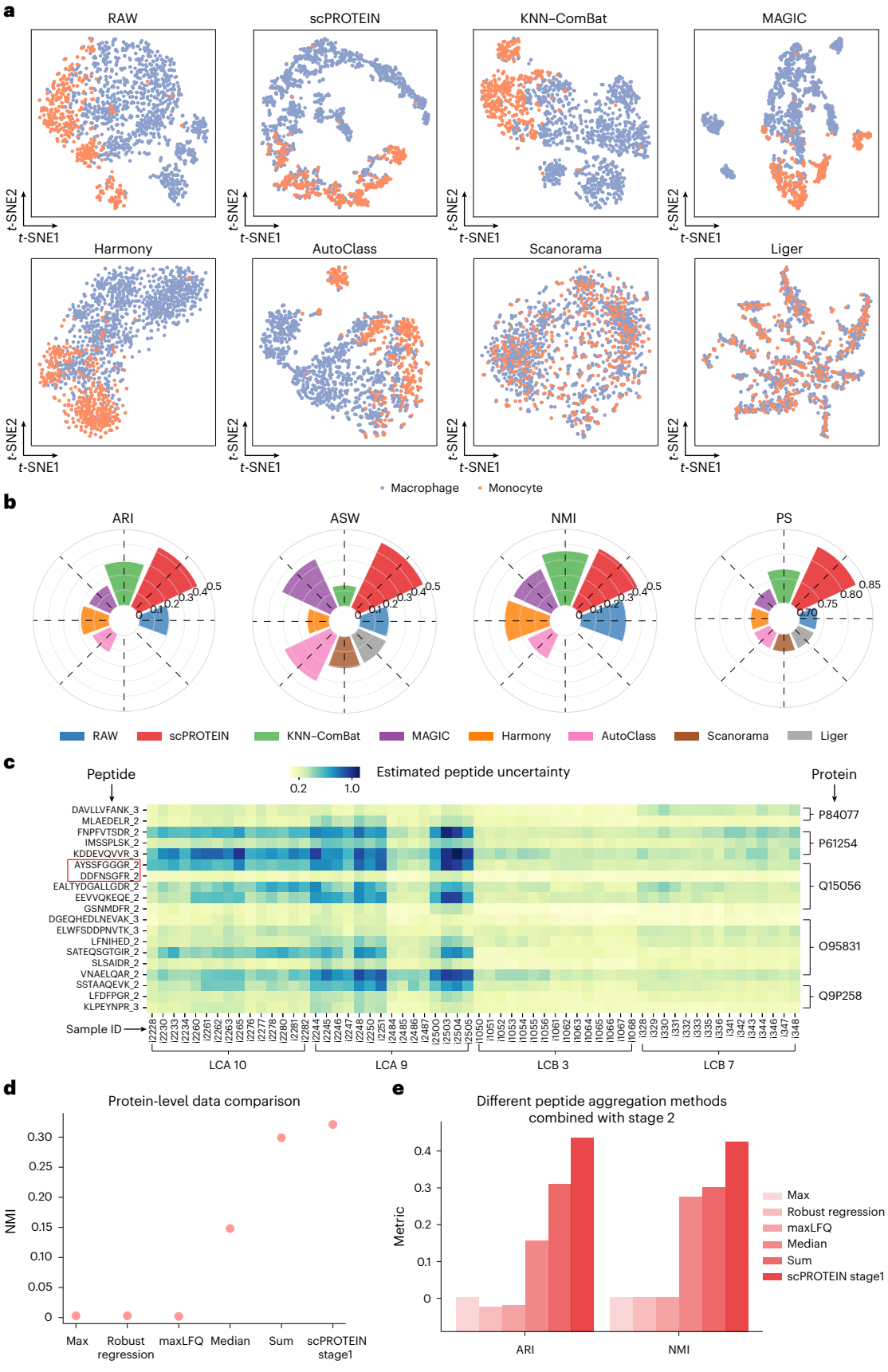
In addition, to validate the utility of stage 1 in scPROTEIN, we first directly compared the protein-level data obtained from stage 1 with those produced by other peptide aggregation methods (max, robust regression[26], MaxQuant's label-free quantitation algorithm (maxLFQ)[27], median and sum) in terms of cell clustering performance. Utilizing the protein data derived from stage 1 yielded the best performance (Fig. 2d and Supplementary Fig. 1), indicating its ability to generate more informative protein-level data in an uncertainty-guided manner. In addition, the ablation study shown in Fig. 2e displays the results obtained by our method on the SCoPE2_Specht dataset with and without performing uncertainty estimation in stage 1. For the scPROTEIN variant without uncertainty estimation, we utilized protein data from the max, robust regression, maxLFQ, median or sum aggregation method in combination with stage 2 to learn the cell representations. It can be seen that scPROTEIN performed better after executing the uncertainty adjustment, which validates the effectiveness of stage 1. We present the performance improvements achieved by stage 1 and stage 2 in Supplementary Fig. 2. Stage 1 of scPROTEIN effectively estimated the inherent noise in the peptide-level data, resulting in the generation of more informative protein-level data. Stage 2 mitigated the batch effects, data noise and data missingness problem, leading to more substantial improvements. Notably, the protein data obtained from stage 1 could also serve as inputs to enhance the performance of other competing methods, not only our stage 2. Supplementary Fig. 3 confirms the generalization ability of stage 1. To further demonstrate the superiority of scPROTEIN, we conducted a head-to-head comparison, consistently showing that scPROTEIN outperformed the other methods across all metrics (Supplementary Fig. 4).

Given the unavailability of ground-truth uncertainty information in the utilized peptide data, we conducted three distinct simulation experiments to provide an additional evaluation of the uncertainty

Fig. 2 | Cell clustering and peptide uncertainty estimation. a, *t*-SNE plots displaying the cell distributions produced with the embeddings of scPROTEIN and the comparison methods on the SCoPE2_Specht dataset. The plots are colored by cell type. Notably, *t*-SNE visualization, which is a nonlinear method, unavoidably has a certain degree of randomness. We show relatively consistent results over multiple repeated runs for analysis. **b**, Bar plots showing the ARI, ASW, NMI and PS values yielded by scPROTEIN and comparison methods when conducting cell clustering on the SCoPE2_Specht dataset. RAW denotes the metric evaluated on the raw protein levels. For each metric, a higher value (higher bar height) indicates better performance. **c**, A heat map showing the estimated uncertainties of each peptide signal across cells, colored by the estimated uncertainty. The batch information (four liquid chromatography (LC) batches) and protein information are shown below the heat map and on the right side of the heat map, respectively. **d**, Cell clustering performance achieved with protein-level data from various peptide aggregation methods, including max, robust regression, maxLFQ, median, sum and stage 1 of scPROTEIN. **e**, The performance of stage 2 attained using protein data from different peptide aggregation methods as inputs.

https://doi.org/10.1038/s41592-024-02214-9

estimation capacity of stage 1. Specifically, various levels of Gaussian noise were spiked into the original peptide-level data across three scenarios: (1) different cells (Supplementary Fig. 5), (2) different peptides (Supplementary Fig. 6) and (3) both different cells and different peptides (Supplementary Fig. 7). Our findings reveal that scPROTEIN effectively recovered the noise levels in a manner consistent with the noise



https://doi.org/10.1038/s41592-024-02214-9

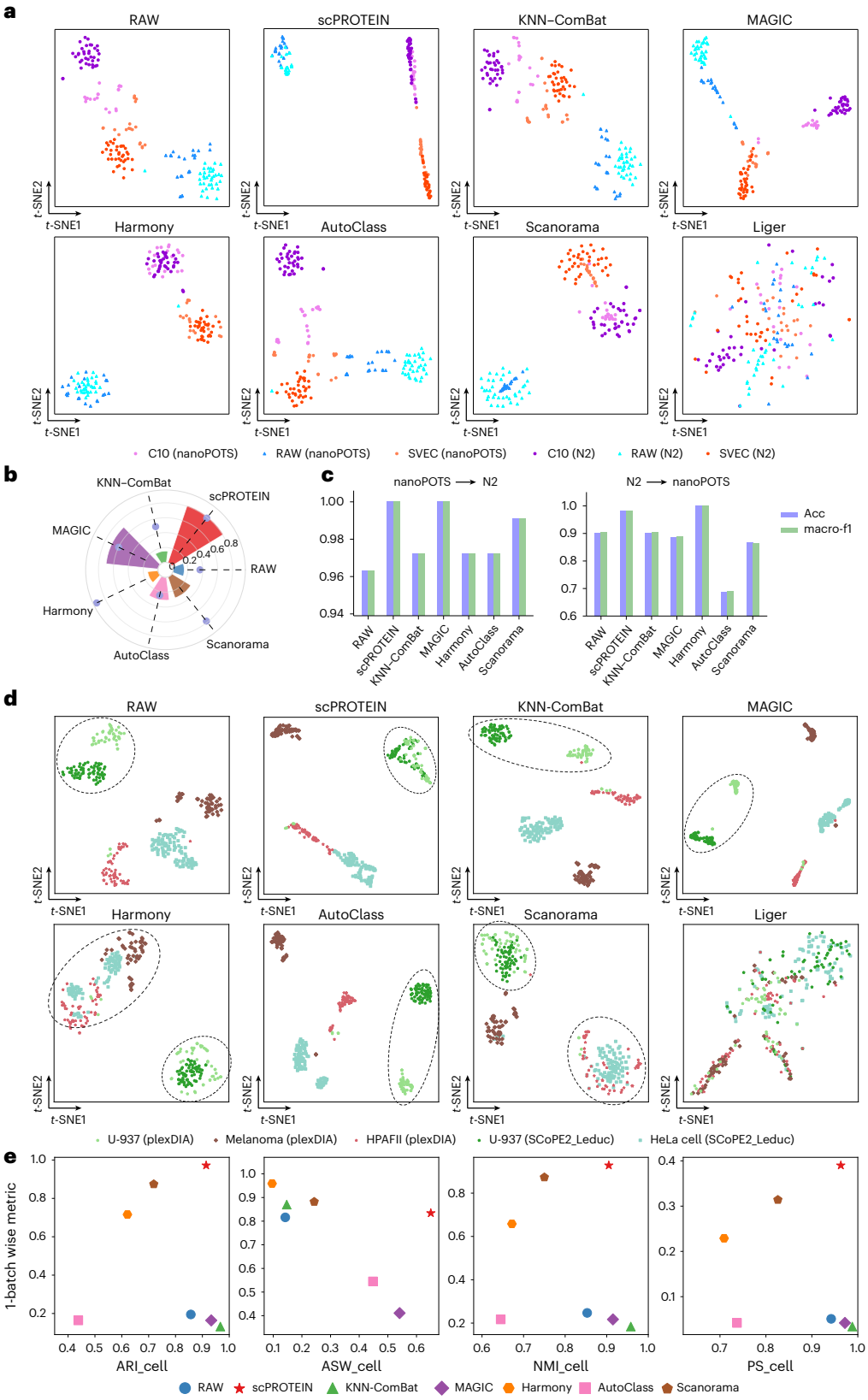


Fig. 3 | Data integration and label transfer. a, *t*-SNE plots showing the cells of the N2 and nanoPOTS datasets, colored by their data acquisitions and cell types. **b**, A bar plot showing the values of ASW_cell and 1-NMI_batch, in which the bar height denotes the value of ASW_cell with the cell labels as ground truth and the point height denotes the value of 1-NMI_batch with the batch labels as ground truth. **c**, Histograms showing the accuracy (acc) and macro-f1 score values produced by scPROTEIN and the comparison methods when transferring labels from nanoPOTS to N2 (left) and from N2 to nanoPOTS (right). **d**, *t*-SNE plots showing the cells of the SCoPE2_Leduc and plexDIA datasets, colored by different data acquisitions and cell types. U-937 is the shared cell type in the two datasets. **e**, ARI_cell, ASW_cell, NMI_cell and PS_cell results of scPROTEIN and the comparison methods with the cell type labels as ground truth (*x* axis) and the 1-metrics with batch labels as ground truth (*y* axis) on the SCoPE2_Leduc and plexDIA datasets.

variation trends observed across all three scenarios. Subsequently, we utilized the uncertainty estimated from Supplementary Fig. 7 (right) to generate protein-level data from the perturbed peptide data. We also replaced this process with the sum and median aggregation methods. To evaluate the impact of noise perturbations, we compared the performance changes induced before and after the introduction of noise for each peptide aggregation method (Supplementary Fig. 8). The observations reveal that both the median and sum aggregation methods were notably affected by noise perturbations, whereas the protein data derived from peptide uncertainty aggregation maintained relatively stable performance. Moreover, an additional analysis conducted using the true single-cell-derived proteomics (T-SCP) dataset[24], which includes cell cycle information (Supplementary Table 1), also affirmed the ability of scPROTEIN to capture subtle biological characteristics (Supplementary Fig. 9).

scPROTEIN enables data integration and label transfer

Due to systematic biases in the sample preparation, data acquisition and labeling strategies, unique batch effects are contained in single-cell proteomic data, which hinder the integration of data from different MS sequencing technologies. Therefore, we evaluated the robustness of scPROTEIN to batch effects in comparison with other methods in five independent experiments, where the batch effects induced by different single-cell proteomic platforms hampered the clustering with the original protein level. We first applied scPROTEIN to integrate two mouse cell datasets: N2 (ref. 12) (108 cells and 1,068 proteins) and nanoPOTS[11] (61 cells and 1,225 proteins). The overlapping proteins (762 proteins) were used as the raw protein profile to build a shared cell graph. In Fig. 3a, the visualization of the raw data shows a clear separation of the same cell type from both acquisitions. After processing them with scPROTEIN, the cells within each cell type were pulled closer, while those of different cell types remained properly separated. This indicates that scPROTEIN preserved the batch-invariant diversity property. Figure 3b shows the results of the quantitative analysis, where bars represent the average silhouette width (ASW) results for cell clustering. We can see that scPROTEIN achieved the best performance and MAGIC ranked second. The points represent 1-NMI_batch, for which Harmony achieved the best performance, and our method ranked second. Notably, compared with other batch correction methods such as Harmony and Scanorama, scPROTEIN aligned the semantic information of the same cell type through its contrastive loss, without needing to know the batch label of each cell. In addition, we show the comparison between the raw protein data and the embeddings learned by scPROTEIN in Extended Data Fig. 3b, which indicates that scPROTEIN could largely alleviate the batch effect.

We further performed label transfer to annotate the cell types across the N2 and nanoPOTS datasets, which contain the same set of cell types. The label transfer process was implemented by the KNN algorithm based on the batch correction results (Extended Data Fig. 3c). The achieved label transfer performance was highly dependent on the effectiveness of the batch corrections. Only when the batch effect was properly corrected and the diversity was properly preserved could cells of unknown types be moved closer to the correct cluster, thus enabling correct label transfer. As shown in Fig. 3c, left, when we took nanoPOTS as the reference set and N2 as the query set, the results indicated that scPROTEIN and MAGIC both correctly labeled all query cells (accuracy of 1.00 and macro-f1 of 1.00). However, the accuracy of MAGIC dropped to 0.885 when transferring labels from N2 to the nanoPOTS dataset. Our proposed method achieved an accuracy of 0.984 when transferring labels from N2 to the nanoPOTS dataset, showing stable performance.

In addition, we benchmarked the data integration performance achieved on the SCoPE2 dataset published by Leduc et al.[3] (namely, SCoPE2_Leduc, with 163 cells and 1,647 proteins) and the plexDIA[5] (164 cells and 1,242 proteins) dataset (Supplementary Table 1), where batch correction is more challenging since the two datasets have different sets of cell types. Utilizing the embedding process of scPROTEIN, U-937 cells from both MS sequencing acquisitions were closely clustered, while the other cell types remained separable (Fig. 3d). MAGIC also retained the cell diversity property, but it failed to correct the batch effect. With Harmony and Scanorama, although the U-937 cells from both acquisitions were well mixed, the biological differences among other cell types were simultaneously lost. Moreover, the quantitative benchmarking results in Fig. 3e indicate that scPROTEIN achieved balanced performance by drawing the same cell types closer while maintaining the diversity of the other cell types in this task, as measured by both cell type and batch-wise metrics. More experimental results are shown in Extended Data Fig. 4–6. Overall, scPROTEIN demonstrated promising performance in terms of removing the batch effect while retaining biological variability.

Application to clinical proteomic data

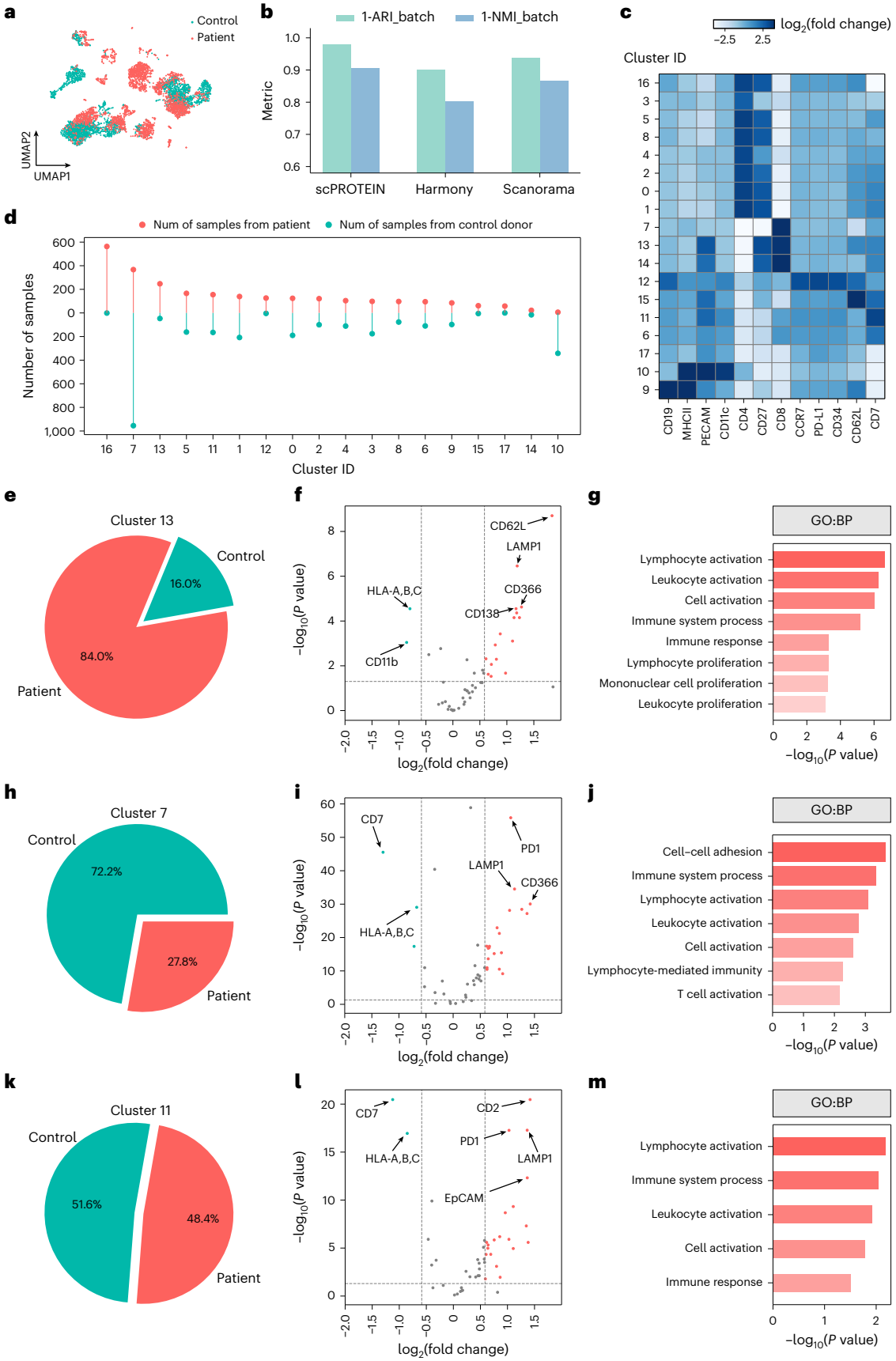
In this study, we briefly explored the application of our method to antibody-based single-cell proteomic data acquired from clinical tissues in the ECCITE-seq dataset[29], which quantifies 49 marker proteins across 6,500 cells from a healthy donor and 6,500 cells from a patient with CTCL. The Uniform Manifold Approximation and Projection (UMAP) of the raw data is shown in Fig. 4a, with cells from the healthy donor in green and those from the CTCL patient in red. We first integrated the data of two donors to remove the batch effect and examined the batch correction performance of scPROTEIN and the competing methods. As shown in Fig. 4b, scPROTEIN resulted in the highest 1-NMI_batch and 1-ARI_batch, indicating its batch effect removal ability.

After performing cell clustering and subsequently combining similar clusters ('Performance achieved on clinical single-cell proteomic data' section in Methods), the cells were grouped into 18 clusters with the embeddings learned by scPROTEIN (Extended Data Fig. 7a). The highly expressed proteins of each cluster are shown in Fig. 4c. In addition, from Fig. 4d, we can observe that the proportions of cells derived from the CTCL donor and from the healthy donor varied in different subgroups. We selected three representative clusters and conducted a

Fig. 4 | Application of scPROTEIN to clinical proteomic dataset. a, UMAP for raw data showing the cells from the healthy donor in green and from the CTCL patient in red. **b**, Bar plots showing the 1-metrics of scPROTEIN and the comparison methods (Harmony and Scanorama) with the batch labels as the ground truth. A higher value indicates better cross-donor batch correction performance. **c**, A heat map showing the levels of different protein markers across clusters. **d**, The numbers of cells acquired from the control donor in green and the CTCL donor in red for different clusters. **e**, The detailed ratio of the cluster 13 cells for the control donor and CTCL donor. **f**, A volcano plot showing the differentially expressed proteins found by contrasting the healthy cells and CTCL cells in cluster 13. The dots in red and green represent the identified upregulated and downregulated proteins of CTCL cells, respectively. We use a *P* value of 0.05 and fold change of 1.5 as thresholds. **g**, The top GO terms in the biological process (BP) for the identified upregulated proteins of the CTCL cells in cluster 13. The *P* values are computed using Fisher's one-tailed test and adjusted by the multiple-hypotheses testing method (g:SCS) of gProfiler. **h**, The detailed ratio of the cluster 7 cells for the control donor and CTCL donor. **i**, A volcano plot showing the differentially expressed proteins found by contrasting the healthy cells and CTCL cells in cluster 7. **j**, The top GO terms in the BP for the identified upregulated proteins of the CTCL cells in cluster 7. The *P* values are computed using Fisher's one-tailed test and adjusted by the multiple-hypotheses testing method (g:SCS) of gProfiler. **k**, The detailed ratio of the cluster 11 cells for the control donor and CTCL donor. **l**, A volcano plot showing the differentially expressed proteins found by contrasting the healthy cells and CTCL cells in cluster 11. **m**, The top GO terms in the BP for the identified upregulated proteins of the CTCL cells in cluster 11. The *P* values are computed using Fisher's one-tailed test and adjusted by the multiple-hypotheses testing method (g:SCS) of gProfiler.

https://doi.org/10.1038/s41592-024-02214-9

detailed analysis for the subgroups in Fig. 4e–m, and more results can be found in Extended Data Fig. 7b–g, in which the pie plots indicate the proportions of CTCL samples and healthy samples (Fig. 4e,h,k), the volcano plots depicts the differentially expressed proteins (Fig. 4f,i,l) found by contrasting the healthy samples and CTCL samples within each cluster and the enrichment analysis indicates the gene ontology



(GO) function of the identified upregulated proteins for the CTCL cells (Fig. 4g,j,m). We further tried to discover biomarker for the cancer cells. Programmed death 1 (PD1) was notably upregulated in the cells from CTCL patient in the results of our model (Fig. 4i). Increased PD1 levels have been detected in various immune cells acquired from CTCL patients and they function in the attenuation of the immune response and antitumor immunity during CTCL progression[37].

Application to spatial proteomic data

Spatial proteomic technologies are maturing and providing an increasing number of resources for tumor microenvironments. We applied our method to single-cell resolved spatial proteomic data derived from the BaselTMA dataset[30] by constructing a cell graph based on the spatial distance measure (Fig. 5a). As cells of the same cell types may have been located in close proximity within the tissue, our method could enhance the cell embeddings with the help of their spatial neighboring proteomes. In this way, we could fully utilize both the spatial information and the protein profile information of the spatial proteomic data. The advantage of spatial proteomics is that it can reveal spatial cell–cell interactions in tumor microenvironments. Therefore, we analyzed the tumor microenvironments with the embeddings learned by scPROTEIN. In addition, the constructed spatially informative cell graph can be naturally used to quantify the spatial heterogeneity degree, which was utilized to estimate the spatial heterogeneity of the tumor microenvironment. In particular, a metric shd was defined to quantify the spatial heterogeneity degree, and its detailed definition can be found in the 'Evaluation metrics' subsection in Methods. A high shd value represents a highly compartmentalized phenotype, and the tissue tended to be block like. In contrast, a low shd value denotes a high level of spatial mixing.

Here, the cell clusters identified by scPROTEIN from the tumor slices shown in Fig. 5b (top) indicated a highly compartmentalized phenotype and exhibited high shd scores (ranging from 0.880 to 0.908). In contrast, the spatial proteomic data derived from the normal slices retained relatively low shd scores (0.501 and 0.577), indicating high levels of intercell type mixing (Fig. 5c, left). The shd scores indicated two types of samples showing different spatial heterogeneity signatures ($P = 0.0015$, two-sided Mann–Whitney test). This observation was consistent with previous clinical study on breast tumors[30,38].

In addition, in Fig. 5b (bottom) and Fig. 5c (right), we depict the results obtained by directly clustering the raw protein data for tumor slices and normal slices, respectively. For two different types of slices, the results did not show any distinct signatures. We further show the density plot of the shd scores obtained across different slices for both scPROTEIN and the raw expression data in Fig. 5d. With scPROTEIN, the shd values for tumor slices and nontumor slices showed different ranges, while in the raw data, the ranges for both types largely overlapped. With the aid of both spatial location and protein profiling, scPROTEIN could learn cell representations that better reflected spatial heterogeneity, thus enabling the discrimination of compartmentalized tumor samples and nontumor samples. The raw protein data alone were not sufficient to distinguish such tumor/nontumor characteristics, and the shd ranges for both sample types greatly overlapped. These comparison results confirm the effectiveness of scPROTEIN in discriminating compartmentalized and highly mixed spatial phenotypes in spatially resolved single-cell proteomic data. More results are shown in Extended Data Fig. 8a,b.

Discussion

Although the widely used cellular indexing of transcriptomes and epitopes by sequencing technology facilitates the detection of cell surface proteins at scale, it suffers from a limitation regarding the numbers and types of detectable proteins. MS-based proteomics provides complementary advantages in terms of the detectable numbers and types of proteins (1,000–3,000 intracellular proteins), which are critical for analyzing cellular functions and disease progression. However, at the same time, it is important to simultaneously address the estimation of peptide uncertainty, data missingness, batch effects and data noise in MS-based single-cell proteomic data. Most of the existing methods only address one or two problems, but generally speaking, because these problems are often tangled with each other, the previously developed methods cannot be effectively used alone or simply combined.

To this end, we present a versatile deep graph contrastive learning model, scPROTEIN, that is tailored for single-cell proteomics embedding and solves the encountered data problems in a unified framework. First, with the uncertainty of the peptide signals estimated by a multitask heteroscedastic regression model, our method could aggregate peptide-level content to protein content. In this way, we could fully utilize and explore the available hierarchical information contained in single-cell proteomic data. Then, the cell graph constructed by the cellular expressions of proteomics displayed the cell similarity, and the data missingness problem could be implicitly alleviated by propagating information from neighboring cells. scPROTEIN was trained based on contrastive learning, which led to perturbation-resistant cell embeddings. To further obtain more robust representations, we designed an alternating topology-attribute denoising module. The noisy data could be largely cleaned and the initially constructed cell graph topology was enhanced. Moreover, contrastive learning is capable of pulling cells with similar patterns closer and pushing different cells apart, providing discrimination ability for batch correction as well as denoising. To our knowledge, this is the first study to establish cell embeddings for single-cell proteomic data to address all of the above problems, providing novel insight into exploring the representation of single-cell proteomic data in a data-driven manner.

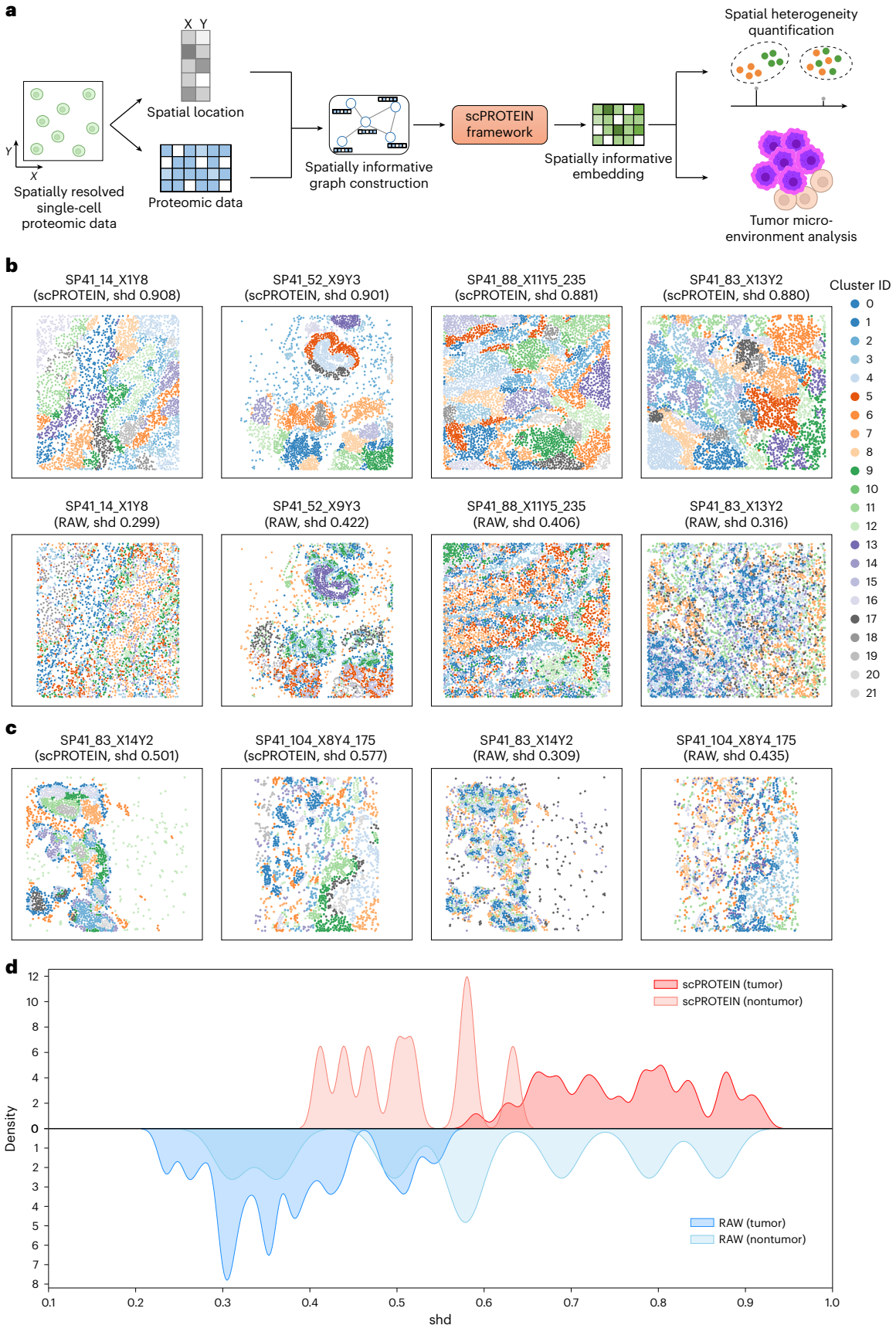
Extensive experiments proved the versatile applicability and superior performance of our method on both MS-based and antibody-based proteomics. Benefiting from the compact embeddings, scPROTEIN demonstrated promising performance in comparison with the existing single-cell proteomic data processing pipeline and other comparison methods in cell clustering, batch correction and cell type annotation task on many single-cell proteomic datasets. Moreover, scPROTEIN exhibited wide applicability to cases such as clinical analysis and single-cell spatially resolved proteomics data analysis. The application of our method to clinical single-cell proteomic data will be further investigated with the exploration of single-cell proteomics in clinical practice. In terms of generalizability, scPROTEIN could be extended to spatially resolved single-cell proteomic data. Building a cell graph based on cell proximity yields spatially informative embeddings and the constructed graph could be naturally used to quantify the degree of spatial heterogeneity. It can be foreseen that, with the

Fig. 5 | Application of scPROTEIN to spatial proteomic data. a, A diagram showing the application of scPROTEIN to spatial proteomic data. The spatial proteomic data are used to construct a spatially informative graph and then to infer spatially informative embeddings for tumor microenvironment analysis and spatial heterogeneity degree quantification. **b**, Visualizations of the learned spatially informative embedding clusters and the estimated spatial heterogeneity degrees within tumor samples (top). Representative samples with high shd values indicate highly compartmentalized phenotypes. Visualizations of the raw protein data clusters and the estimated spatial heterogeneity degrees within tumor samples (bottom). **c**, Visualizations of the learned spatially informative embeddings and the estimated spatial heterogeneity degrees within nontumor samples (left). Representative samples with low shd values indicate high levels of mixing between different cells. Visualizations of the raw protein data clusters and the estimated spatial heterogeneity degrees within nontumor samples (right). **d**, A density plot of the shd values for scPROTEIN and the raw protein data. The shd values yielded by scPROTEIN for tumor and nontumor slices exhibit different ranges, while for the raw expression data, the ranges are highly overlapped.

https://doi.org/10.1038/s41592-024-02214-9

rapid development and application of single-cell proteomic technologies, our method would play an increasingly important role in various single-cell proteomic data analysis scenarios.

Despite the above advantages, our investigation still possesses some limitations. First, some MS acquisition platforms provided raw profile data directly from the protein level, for which stage 1, tailored for



the hierarchical proteome structure, could not be employed. Second, since no ground truth is available for peptide uncertainty, the achieved peptide uncertainty estimation performance was indirectly evaluated by downstream tasks, and we endeavored to rectify this challenge with simulation experiments (Supplementary Figs. 5–7). By introducing an artificial intelligence (AI) peptide uncertainty estimation algorithm for the first time, we provided a preliminary solution to satisfy the urgent need of the single-cell proteomics community[7]. In the future, we expect more single-cell proteomics data with direct labels to be generated by the biological community, which can lay the foundation for the further development and validation of uncertainty estimation algorithms for MS results.

Regarding the benchmark experiments, it is important to highlight the scarcity of existing pipelines specifically designed for single-cell proteomics data. Consequently, few baseline methods were available for comparison, and we had to adapt some well-performed methods from the field of scRNA-seq. Since these methods were originally designed for scRNA-seq data, they may not be sufficient to address the tangled data problems in single-cell proteomics. We further enhanced these methods by the preceding imputation step to suit the unique characteristics of single-cell proteomics data and achieve better result, but scPROTEIN still obtained the best overall performance (Supplementary Fig. 10). In addition, due to the current scarcity of biologically meaningful labels for single-cell proteomics data, we set up an initial benchmark with widely recognized and representative cell clustering tasks (in the area of single-cell sequencing data embedding) to evaluate the performance of scPROTEIN. We believe that as single-cell proteomics sequencing technology advances and becomes widely adopted, wet laboratories will contribute more data with meaningful biological labels to the research community. This, in turn, will facilitate the establishment of more accurate, direct and convincing benchmarks, enabling subsequent single-cell proteomics data embedding algorithms to undergo more effective evaluations. Furthermore, with the rapid development of graph models, more informative graph construction methods will emerge in the future, which could further enrich our scPROTEIN model.

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/s41592-024-02214-9.

References

1. Svensson, V., Vento-Tormo, R. & Teichmann, S. A. Exponential scaling of single-cell RNA-seq in the past decade. *Nat. Protoc.* **13**, 599–604 (2018).
2. Slavov, N. Unpicking the proteome in single cells. *Science* **367**, 512–513 (2020).
3. Leduc, A., Huffman, R. G., Cantlon, J., Khan, S. & Slavov, N. Exploring functional protein covariation across single cells using nPOP. *Genome Biol.* **23**, 261 (2022).
4. Petelski, A. A. et al. Multiplexed single-cell proteomics using SCoPE2. *Nat. Protoc.* **16**, 5398–5425 (2021).
5. Derks, J. et al. Increasing the throughput of sensitive proteomics by plexDIA. *Nat. Biotechnol.* **41**, 50–59 (2023).
6. Doerr, A. Single-cell proteomics. *Nat. Methods* **16**, 20 (2019).
7. Marx, V. A dream of single-cell proteomics. *Nat. Methods* **16**, 809–812 (2019).
8. Perkel, J. M. Single-cell proteomics takes centre stage. *Nature* **597**, 580–582 (2021).
9. Schoof, E. M. et al. Quantitative single-cell proteomics as a tool to characterize cellular hierarchies. *Nat. Commun.* **12**, 3341 (2021).
10. Furtwängler, B. et al. Real-time search-assisted acquisition on a tribrid mass spectrometer improves coverage in multiplexed single-cell proteomics. *Mol. Cell. Proteomics* **21**, 100219 (2022).
11. Dou, M. et al. High-throughput single cell proteomics enabled by multiplex isobaric labeling in a nanodroplet sample preparation platform. *Anal. Chem.* **91**, 13119–13127 (2019).
12. Woo, J. et al. High-throughput and high-efficiency sample preparation for single-cell proteomics using a nested nanowell chip. *Nat. Commun.* **12**, 6246 (2021).
13. Gatto, L. et al. Initial recommendations for performing, benchmarking and reporting single-cell proteomics experiments. *Nat. Methods* **20**, 375–386 (2023).
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. Huffman, R. G. et al. Prioritized mass spectrometry increases the depth, sensitivity and data completeness of single-cell proteomics. *Nat. Methods* **20**, 714–722 (2023).
16. Khan, Z. et al. Primate transcript and protein expression levels evolve under compensatory selection pressures. *Science* **342**, 1100–1104 (2013).
17. Vogel, C. & Marcotte, E. M. Insights into the regulation of protein abundance from proteomic and transcriptomic analyses. *Nat. Rev. Genet.* **13**, 227–232 (2012).
18. Gygi, S. P., Rochon, Y., Franza, B. R. & Aebersold, R. Correlation between protein and mRNA abundance in yeast. *Mol. Cell Biol.* **19**, 1720–1730 (1999).
19. Marguerat, S. et al. Quantitative analysis of fission yeast transcriptomes and proteomes in proliferating and quiescent cells. *Cell* **151**, 671–683 (2012).
20. Irish, J. M., Kotecha, N. & Nolan, G. P. Mapping normal and cancer cell signalling networks: towards single-cell proteomics. *Nat. Rev. Cancer* **6**, 146–155 (2006).
21. Vanderaa, C. & Gatto, L. Replication of single-cell proteomics data reveals important computational challenges. *Expert Rev. Proteomics* **18**, 835–843 (2021).
22. Cheung, T. K. et al. Defining the carrier proteome limit for single-cell proteomics. *Nat. Methods* **18**, 76–83 (2020).
23. Mund, A. et al. Deep Visual Proteomics defines single-cell identity and heterogeneity. *Nat. Biotechnol.* **40**, 1231–1240 (2022).
24. Brunner, A.-D. et al. Ultra-high sensitivity mass spectrometry quantifies single-cell proteome changes upon perturbation. *Mol. Syst. Biol.* **18**, e10798 (2022).
25. Specht, H. et al. Single-cell proteomic and transcriptomic analysis of macrophage heterogeneity using SCoPE2. *Genome Biol.* **22**, 50 (2021).
26. Sticker, A., Goeminne, L., Martens, L. & Clement, L. Robust summarization and inference in proteome-wide label-free quantification. *Mol. Cell. Proteomics* **19**, 1209–1219 (2020).
27. Cox, J. et al. Accurate proteome-wide label-free quantification by delayed normalization and maximal peptide ratio extraction, termed MaxLFQ. *Mol. Cell. Proteomics* **13**, 2513–2526 (2014).
28. Kendall, A. & Gal, Y. What uncertainties do we need in Bayesian deep learning for computer vision? *Adv. Neural Inf. Process. Syst.* **30**, 5580–5590 (2017).
29. Mimitou, E. P. et al. Multiplexed detection of proteins, transcriptomes, clonotypes and CRISPR perturbations in single cells. *Nat. Methods* **16**, 409–412 (2019).
30. Jackson, H. W. et al. The single-cell pathology landscape of breast cancer. *Nature* **578**, 615–620 (2020).
31. van Dijk, D. et al. Recovering gene interactions from single-cell data using data diffusion. *Cell* **174**, 716–729 (2018).
32. Li, H., Brouwer, C. R. & Luo, W. A universal deep neural network for in-depth cleaning of single-cell RNA-seq data. *Nat. Commun.* **13**, 1901 (2022).

33. Korsunsky, I. et al. Fast, sensitive and accurate integration of single-cell data with Harmony. *Nat. Methods* **16**, 1289–1296 (2019).
34. Hie, B., Bryson, B. & Berger, B. Efficient integration of heterogeneous single-cell transcriptomes using Scanorama. *Nat. Biotechnol.* **37**, 685–691 (2019).
35. Welch, J. D. et al. Single-cell multi-omic integration compares and contrasts features of brain cell identity. *Cell* **177**, 1873–1887 (2019).
36. Boekweg, H. et al. Features of peptide fragmentation spectra in single-cell proteomics. *J. Proteome Res.* **21**, 182–188 (2022).
37. Samimi, S. et al. Increased programmed death-1 expression on CD4+ T cells in cutaneous T-cell lymphoma: implications for immune suppression. *Arch. Dermatol.* **146**, 1382–1388 (2010).
38. Keren, L. et al. A structured tumor-immune microenvironment in triple negative breast cancer revealed by multiplexed ion beam imaging. *Cell* **174**, 1373–1387 (2018).

Methods

The scPROTEIN model

Uncertainty estimation of peptide-level intensity. It has been shown that the detection of peptide sequence is stochastic, indicating that the detected signal may not be accurate[7]. Therefore, it is necessary to estimate the uncertainties of peptide-level signals and obtain more informative protein-level data.

To this end, we develop an uncertainty-aware framework in stage 1 to provide a peptide uncertainty estimation measure via the multitask heteroscedastic regression model, which is motivated by previous research[28]. We consider that the amino acid sequence can determine the peptide's hydrophobicity and ionization efficiency, which in turn affect the preparation, separation and detection processes used for single-cell proteomics samples[22]. From this perspective, the amino acid sequence may influence the accuracy of quantitative MS-based detection. As shown in Fig. 1a, the model takes the original peptide sequences as inputs and outputs the estimated uncertainty. We employ a convolutional neural network as the backbone framework.

Specifically, for each detected peptide sequence, we first use one-hot encoding to generate a matrix with a size of 20 ×peptide_length, where 20 is the total number of common amino acid types and peptide_length is the length of the input sequence. In addition, since the sequence lengths are variable, we perform a padding operation to align all the one-hot matrices to the maximum sequence length, which facilitates the subsequent encoding process.

The one-hot encoded matrices then go through three convolutional blocks, as shown in Fig. 1a, where each block consists of a one-dimensional convolutional layer, a batch normalization layer, a rectified linear unit (ReLU) activation function and a max pooling layer. The outputs are then flattened and passed through two fully connected layers. Finally, we design two independent dense layers to separately obtain the predicted abundance μ and estimated uncertainty σ, and the loss function used for training the whole peptide uncertainty (unc) estimation framework is as follows:

$$L_{\text{unc}} = \frac{1}{N \times P} \sum_{n=1}^{N} \sum_{p=1}^{P} \frac{1}{2\sigma_{pn}^2} \|\text{Pep}_{pn} - \mu_{pn}\|^2 + \frac{1}{2} \log \sigma_{pn}^2, \tag{1}$$

where μ_{pn} and Pep_{pn} represent the predicted and observed abundance of peptide sequence p in cell n, respectively, σ_{pn} represents the uncertainty of peptide sequence p in cell n, N denotes the number of cells and P is the number of peptide sequences. This loss function is composed of two terms, and each term involves σ_{pn}, which measures the data uncertainty and plays a role as a balancing factor. Specifically, to reduce the loss, the model cannot output an overly small σ_{pn} value since the term $\frac{1}{2\sigma_{pn}^2}$ would explode. On the other hand, the model cannot predict an extremely large σ_{pn} either since the second term would increase. Hence, the model outputs a reasonable uncertainty estimation value to balance both terms in the loss function. In addition, the $\| y_{pn} - \mu_{pn} \|^2$ term is the mean squared error loss between the predicted and observed abundance values. Only when this term is sufficiently small can scPROTEIN offer a small uncertainty prediction σ_{pn}, which indicates that the model is confident about the currently predicted peptide abundance value. Moreover, it is worth noting that stage 1 is primarily used to estimate the uncertainty at the peptide level rather than predicting the abundance of the peptides. The loss function used for the peptide abundance calculation is merely designed to assist in estimating the uncertainty of peptides.

Protein-level abundance aggregated from peptide uncertainty. Once the peptide uncertainty is estimated, we can compute the protein abundance level in an uncertainty-guided manner, as shown on the right side of Fig. 1a. A peptide signal with a higher uncertainty estimation value indicates a noisy signal and it should be assigned a smaller weight. In contrast, a peptide signal with lower uncertainty tends to be a highly confident signal and it should be given a larger weight. Therefore, we can obtain a protein-level abundance matrix X with each entry calculated as follows:

$$X_{fn} = \sum_{\text{pep}_p \in \text{Pro}_f} \text{Pep}_{pn} \cdot \frac{1}{\sigma_{pn}}, \tag{2}$$

where X_{fn} represents the aggregated abundance value of protein f in cell n and $\text{pep}_p \in \text{Pro}_f$ denotes peptide sequence p derived from protein f. The obtained protein-level data are then fed into stage 2 of the scPROTEIN framework as the initial feature matrix.

Graph construction. To make full use of the single-cell proteomic data, we convert the abundance data matrix into an undirected and unweighted cell-cell graph $G = (V, E, X)$. $V = \{v_1, v_2, \ldots, v_N\}$ is the set of cell nodes. $E = \{e_{ij}\} \subseteq V \times V$ represents the set of all edges. We take the protein-level data as the feature matrix $X = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\} \in \mathbb{R}^{N \times F}$, and F denotes the dimensionality of the feature vectors, which is also the number of proteins.

To obtain the cell graph topology, we first calculate a cellwise similarity matrix S via the Pearson correlation coefficient (PCC) based on the abundance feature vectors. S is formally stated as

$$S_{ij} = \text{PCC}(v_i, v_j) = \frac{\text{cov}(\mathbf{x}_i, \mathbf{x}_j)}{\sigma_{\mathbf{x}_i} \sigma_{\mathbf{x}_j}}, \tag{3}$$

where cov and σ denote the covariance and the standard deviation, respectively. Then, we set a threshold h to construct the cell graph, and the topological structure of cell graph G can be specified by a symmetric adjacency matrix $A \in \mathbb{R}^{N \times N}$:

$$A_{ij} = \begin{cases} 1, & \text{if } S_{ij} > h \\ 0, & \text{else} \end{cases} \tag{4}$$

The obtained binary adjacency matrix is then used to perform the following graph learning process. In addition, our scPROTEIN method can be easily extended to single-cell spatial proteomics. With the provided spatial location information, scPROTEIN can construct a cell graph based on spatial proximity. Specifically, the similarity matrix S is calculated on the basis of the Euclidean distance computed from the spatial coordinates as follows:

$$S_{ij} = \text{Euc\_dist}\left(\text{coordinate}_i, \text{coordinate}_j\right), \tag{5}$$

and we can similarly obtain the spatially informative cell graph topology by setting the threshold as mentioned above but utilizing $A_{ij} = 1$ when $S_{ij} < h$. Notably, our constructed spatially informative cell graph can subsequently be used to quantify the spatial heterogeneity degree.

As the protein profile data are noisy, the constructed cell graph cannot be completely accurate. Therefore, we apply two data augmentation strategies and a special denoising module to solve this problem and learn robust embeddings.

Graph contrastive learning for cell embedding generation. *Model architecture.* Since label information is always scarce and task specific, we propose to learn embeddings in an unsupervised manner to achieve better model generalizability. Motivated by a previous study[39], we propose a deep graph contrastive learning framework to learn comprehensive low-dimensional cell embeddings. This framework receives the cell graph G and feature matrix X as inputs. Specifically, to address the noise in single-cell proteomic data caused by the limitations of MS acquisition technology, we design a novel alternating topology-attribute denoising module, which yields more informative and noise-resistant cell embeddings. Overall, scPROTEIN contains four

major components, as shown in Fig. 1b: (1) a data augmentation module, (2) a GCN-based graph encoder, (3) a node-level graph contrastive learning module and (4) an alternating topology-attribute denoising module.

Data augmentation. Contrastive learning aims to learn the invariant representations between similar and dissimilar data pairs. To produce similar pairs, we employ data augmentation to generate different views of the input data. Here, we adopt two types of graph augmentation techniques: drop edge[40] and feature masking.

For drop edge, we randomly remove existing edges from E at a given ratio p_{de}. In particular, we sample an indicator matrix $R \in \mathbb{R}^{N\times N}$ to decide which edges will be removed. R_{ij} can be expressed as

$$R_{ij} \sim \text{Bernoulli}(1-p_{de}) \text{ if } A_{ij}=1 \text{ else } R_{ij}=0. \quad (6)$$

The perturbed adjacency matrix can be obtained by

$$\widetilde{A} = A \odot R, \quad (7)$$

where $\odot$ represents the Hadamard product operator.

For feature masking, an indicator vector $\mathbf{M} \in \mathbb{R}^{N\times 1}$ is generated with each entry $M_i \sim \text{Bernoulli}(1-p_{mf})$, where p_{mf} is a given feature masking probability. The masked feature matrix is expressed as

$$\widetilde{X} = [\mathbf{x}_1 \odot \mathbf{M}; \mathbf{x}_2 \odot \mathbf{M}; \ldots; \mathbf{x}_N \odot \mathbf{M}]^T. \quad (8)$$

$[\bullet;\bullet]$ represents the vector concatenation operator. In each training iteration, we employ data augmentation to generate two different but correlated graph views $G_1 = (V, \widetilde{E}_1, \widetilde{X}_1)$ and $G_2 = (V, \widetilde{E}_2, \widetilde{X}_2)$, as shown in Fig. 1b (step 1).

GCN-based graph encoder. After the augmented graph views G_1 and G_2 are prepared, we learn the cell embeddings based on node-level graph contrastive learning. Since GCN[41] provides a powerful learning model and is able to extract comprehensive embeddings for graph-structured data, we adopt it as the feature extractor to learn the latent pattern of the cell nodes. Concretely, the GCN performs a convolution operation on the graph and iteratively updates the node representations by passing messages among the neighborhoods. Taking the graph topology structure and the feature matrix as inputs, the low-dimensional node representations can be learned by a L-layer GCN as follows:

$$Z^{l+1} = \delta\left(\hat{D}^{-\frac{1}{2}}\hat{A}\hat{D}^{-\frac{1}{2}}Z^l W^l\right), \quad (9)$$

where $\hat{A} = A + I$ and I is the identity matrix. $\hat{D}$ is the degree matrix of $\hat{A}$ with $\hat{D}_{ii} = \sum_i \hat{A}_{ij}$. W^l is the layer-specific trainable weight parameter and Z^l represents the embedding matrix obtained by layer l. We use a parametric ReLU as the nonlinear activation function $\delta(\bullet)$. For the two augmented graph views, we adopt a weight-sharing GCN encoder to generate embedding matrices Z_1 and Z_2. We set the dimensionality of all GCN layers d and the dimensionality of the output embedding matrix is $\mathbb{R}^{N\times d}$.

Node-level graph contrastive learning. The obtained outputs Z_1 and Z_2 are then fed into a weight-sharing projection head g, which is used to project the embeddings from the two views into a common latent feature space in which the contrastive loss is constructed. g is implemented by a two-layer multilayer perceptron. By calculating the contrastive loss in the projected latent space, scPROTEIN can attain a better representation ability[42,43].

Then, we try to maximize the agreement between the representations of the two generated views for the same node. Taking node as an example, its embedding $\mathbf{z}_{1i}$ from view G_1 can be regarded as an anchor and the embedding $\mathbf{z}_{2i}$ from another view G_2 is treated as a positive sample. Naturally, the embeddings of other cell nodes are taken as negative samples. We want to maximize the agreement between the positive pairs (the anchor and its positive samples) and minimize the agreement between the negative pairs (the anchor and its negative samples). Here, we adopt the cosine similarity function $\cos(\bullet)$ to measure the similarity between samples, and we can arrive at the modified information noise-contrastive estimation (infoNCE) loss[44] for a positive pair $(\mathbf{z}_{1i}, \mathbf{z}_{2i})$ as follows:

$$l(\mathbf{z}_{1i}, \mathbf{z}_{2i}) = -\log \frac{e^{\theta(\mathbf{z}_{1i},\mathbf{z}_{2i})/\tau}}{e^{\theta(\mathbf{z}_{1i},\mathbf{z}_{2i})/\tau} + \sum_{j=1}^{N} \mathbb{1}_{[j\neq i]}\left(e^{\theta(\mathbf{z}_{1i},\mathbf{z}_{1j})/\tau} + e^{\theta(\mathbf{z}_{1i},\mathbf{z}_{2j})/\tau}\right)}, \quad (10)$$

where τ is the temperature parameter and $\mathbb{1}_{[j\neq i]}$ denotes the indicator function. $\theta(\mathbf{z}_{1i}, \mathbf{z}_{2i}) = \cos(g(\mathbf{z}_{1i}), g(\mathbf{z}_{2i}))$. $(\mathbf{z}_{1i}, \mathbf{z}_{1j})$ is the negative pair derived from the same augmented view and $(\mathbf{z}_{1i}, \mathbf{z}_{2j})$ is the negative pair obtained from different views. Since the two generated views can be switched, we can finally obtain the overall contrastive loss, which is expressed as

$$L_{\text{cont\_node}} = \frac{1}{2N}\sum_{i=1}^{N}[l(\mathbf{z}_{1i}, \mathbf{z}_{2i}) + l(\mathbf{z}_{2i}, \mathbf{z}_{1i})]. \quad (11)$$

By minimizing this node-level contrastive loss, positive node pairs are drawn closer and negative node pairs are pushed apart. The biological variability of each cell node can be effectively preserved and the batch effect can be implicitly corrected, which results in more robust embeddings.

Alternating topology-attribute denoising module. Compared with single-cell transcriptomic data, single-cell proteomic data face unique data noise. These data noises are generated during processes such as sample preparation (that is, sample losses and introduced contaminants) and MS-based quantification (that is, co-isolation and interference)[13]. Therefore, to obtain noise-resistant cell embeddings, we design an alternating topology-attribute denoising module, which can be decoupled into the following two major alternating steps.

Attribute denoising: to alleviate the noise problem in the single-cell proteomic profile, we develop an attribute denoising module based on prototype contrastive learning. As investigated in previous study[45], prototypes that are far away from the cluster boundaries are not vulnerable to noise compared with those near the boundary. On the basis of this observation, we propose exploiting the prototypes far from the boundary to update other noisy samples.

As shown in Fig. 1b (step 2), in each training epoch, we perform subpopulation detection via the k-means algorithm[46] based on the current learned embeddings with a predefined number of clusters K. After the k-means algorithm converges, we can assign each node its cluster label cluster_i. Then, we naturally take the K cluster centers $\{\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_K\}$ as the prototype representations since these centers are usually far away from the cluster boundary and more confident about their cluster labels. Formally, cluster center $\mathbf{c}_k$ is calculated by the average of the samples with cluster label k:

$$\mathbf{c}_k = \frac{1}{\text{Num}_k}\sum\nolimits_{\text{cluster}_i=k}\mathbf{z}_i, \quad (12)$$

where Num_k denotes the total number of samples with cluster label k and $\mathbf{z}_i$ is the learned cell embedding obtained for node i without data augmentation. This noise-resistant knowledge is then exploited to update the whole node representation matrix with the designed prototype contrastive loss as

$$L_{\text{cont\_proto}} = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{\mathbb{1}_{[\text{cluster}_i=k]}\exp(\mathbf{z}_i\cdot\mathbf{c}_k/\tau)}{\sum_{k=1}^{K}\exp(\mathbf{z}_i\cdot\mathbf{c}_k/\tau)}. \quad (13)$$

By minimizing this objective function, the similarity between each node and its cluster center is enlarged. Each prototype is used to enhance the information of its surrounding samples. Finally, we can obtain a denoised cell representation after completing an iterative training process. In addition, we further confirm that our scPROTEIN model is robust to the choice of the number of clusters K in Extended Data Fig. 2.

Topology denoising: to fix the existing inaccurate edges and further enhance the graph topology, we develop a topology denoising module. The graph learning process is highly dependent on the given graph structure. Hence, the noise that exists in the graph topology can degrade the model performance. The initial cell graph may contain some vital edges that have not been mined. On the other hand, some established edges may be pseudo edges, which might bring noisy information. With the network learning and attribute denoising processes, the characteristics of the nodes are continuously enhanced and cleaned and the semantic information of each cell node becomes clearer. Therefore, the edges in the network should be alternately updated accordingly.

To this end, we design a link predictor to alternately denoise the topology structure and, in turn, improve the quality of the learned embeddings. Considering the embedding matrix in iteration t as $Z^{(t)}$, we can obtain the corresponding similarity matrix $S_{ij}^{(t)}$ based on the pairwise PCC. Then, we choose the M edges with the highest probabilities in $S_{ij}^{(t)}$ as the set $\text{Edge\_add}^{(t)}$ and the M edges with the lowest probability scores as the set $\text{Edge\_remove}^{(t)}$. The current adjacency matrix $A_{ij}^{(t)}$ can be refined on the basis of these two sets. We suggest that edges with high probabilities should be added to enhance the graph representation ability of the model, while edges with low probabilities should be removed to eliminate the topological noise. Thus, we can obtain the updated adjacency matrix in iteration t as

$$A_{ij}^{(t)} = \begin{cases} 1 & \text{if } e_{ij} \in \text{Edge\_add}^{(t)} \\ 0 & \text{if } e_{ij} \in \text{Edge\_remove}^{(t)} \\ A_{ij}^{(t-1)} & \text{else} \end{cases} \quad (14)$$

As shown in Fig. 1b (step 2), the updated adjacency matrix passes through the next training iteration for data augmentation and attribute denoising. Then, the denoised node representations are, in turn, used to update the topological structure. Through such an alternating update mechanism, we largely relieve the impacts of the noise problem in single-cell proteomic data, and a more robust cell representation can be derived.

Overall loss function. The overall loss function of the scPROTEIN model consists of two parts, and it is expressed as follows:

$$L = L_{\text{cont\_node}} + \alpha L_{\text{cont\_proto}}, \quad (15)$$

where α is a balancing factor that is set to 0.05 in the experiments. We minimize this loss function to train the model parameters. Adaptive moment estimation[47] is used as the optimizer with a learning rate of 0.001.

Cell embedding generation. After the model converges, we can obtain the trained GCN encoder and a refined cell graph. As shown in Fig. 1c, the learned cell embeddings Z are generated on the refined graph topology in stage 2 with the informative trained GCN encoder. The obtained embeddings have largely mitigated this set of tangled problems, including data missingness, batch effects and data noise. Therefore, they can be used in a variety of applications, such as cell clustering, batch correction, label transfer, clinical analysis and spatial analysis. Notably, data augmentation is only used during the training process. During the inference process, none of the data augmentation techniques is exploited.

Datasets

Single-cell proteomic datasets. *SCoPE2_Specht*[25]. SCoPE2_Specht is a representative single-cell proteomic dataset that quantifies 3,042 proteins in 1,490 cells via SCoPE2. It contains two cell types: monocytes and macrophages. Notably, since monocyte cells can be differentiated into macrophage-like cells when polarizing cytokines are absent, the characteristics of these two cell types can be quite similar and result in a more difficult cell clustering task. We downloaded the data from ref. 48.

nanoPOTS[11]. The nanoPOTS dataset was prepared by nanoPOTS technology. It quantifies 1,225 proteins in 61 cells, which are composed of C10 cells, RAW cells and splenic vascular endothelial cells (SVECs). The data were downloaded from the MassIVE data repository with ID MSV000084110.

N2[12]. The N2 dataset, which was sampled from mouse blood, contains 108 cells and 1,068 proteins. This dataset contains three cell types (C10 cells, RAW cells and SVECs). We downloaded the data from the MassIVE data repository with ID MSV000086809.

SCoPE2_Leduc[3]. The SCoPE2_Leduc dataset was downloaded from ref. 49. It contains 163 cells and 1,647 proteins that were generated by SCoPE2. The SCoPE2_Leduc dataset contains two cell types: HeLa cells and U-937 cells.

plexDIA[5]. We downloaded the plexDIA dataset from ref. 50. It contains 1,242 different proteins are quantified for 164 cells and three cell types: melanoma cells, U-937 cells and pancreatic ductal adenocarcinoma (PDAC (HPAFII)) cells.

pSCoPE_Huffman[15]. We obtained the pSCoPE dataset published by Huffman et al. (namely pSCoPE_Huffman; Supplementary Table 1) from ref. 51 (derived from their original 'Benchmarking experiments: Fig. 1b,e data'). This dataset was sampled by pSCoPE and consists of 163 cells and 1,647 proteins. It contains three PDAC cell lines: CFPACI, HPAFII and BxPC3.

pSCoPE_Leduc[3]. The pSCoPE dataset of Leduc et al. (namely pSCoPE_Leduc; Supplementary Table 1) was downloaded from ref. 52. It was generated by the pSCoPE technique, and 2,844 different proteins are quantified for 1,543 cells. Melanoma cells and U-937 cells are the two cell types contained in the pSCoPE_Leduc dataset.

T-SCP[24]. The T-SCP dataset was downloaded from the PRIDE partner repository (accession no. PXD024043), comprising 225 HeLa cells and 1,810 proteins. It includes information on the cell cycle (G1, G1-S, G2 and G2-M).

Single-cell clinical proteomic dataset. *ECCITE-seq*[29]. The ECCITE-seq dataset was downloaded from the Gene Expression Omnibus (accession number GSE126310). It contains 49 surface protein markers that were detected via the cellular indexing of transcriptomes and epitopes by sequencing. The human peripheral blood mononuclear cells in ECCITE-seq were acquired from a healthy control donor or a patient with CTCL. After removing cell doublets, 2,767 cells from the control donor and 2,634 cells from the CTCL donor remained, which were then used for the exploration of downstream clinical analysis based on single-cell proteomic data.

Spatial proteomic dataset. *BaselTMA*[30]. The BaselTMA dataset was downloaded from Zenodo[53] and it includes 281 patients with breast cancer. The imaging mass cytometry technique was used to quantify 38 marker proteins and spatial tissue images at the same time. This resulted in each TMA being equipped with spatially resolved single-cell location information together with a protein quantification matrix.

Each tissue slice contains a 0.8 mm tumor core or a corresponding healthy breast tissue.

Data preprocessing

The single-cell proteomic data were processed according to the SCoPE2 pipeline[25]. The raw protein data were $\log_2$-transformed. In addition, for the ECCITE-seq dataset, we removed the cell doublets as the authors suggested with the scrublet Python package[54,55].

Baseline methods

The KNN–ComBat[21] pipeline contains two steps. The imputation step with KNN was implemented by the sklearn Python package[56] and ComBat was implemented for batch correction by the scanpy.pp.combat module via the SCANPY[57] package. MAGIC[31] was downloaded from ref. 58. Harmony[33] was implemented via the harmony-pytorch package[59]. The Scanorama[34] algorithm was adopted from the scanorama package[60]. We implemented AutoClass[32] from the public code[61]. For Liger[35], the rliger package was applied for benchmarking. The implementation details are illustrated in Supplementary Table 2.

Evaluation experiments

Cell clustering performance. We conducted a cell clustering experiment on SCoPE2_Specht, which is a representative single-cell proteomic dataset. We first exploited stage 1 of scPROTEIN to estimate the uncertainty of the peptide signals. The obtained uncertainty estimation matrix and the original peptide-level abundance data were then combined to produce protein-level data, which were then fed into stage 2 and stage 3 to learn informative cell embeddings. The embedding dimensionality was reduced by using the top 50 principal components selected by principal component analysis, and we then used the *t*-distributed stochastic neighbor embedding (*t*-SNE) algorithm to further reduce the dimensionality to two for two-dimensional visualization purposes, similar to Li et al.[32]. The principal component analysis and *t*-SNE functions were both implemented via the sklearn.manifold Python package. We utilized the *k*-means algorithm[46] to obtain the cell clusters.

Data integration performance. To evaluate the ability of scPROTEIN to address cross-cohort data with batch effects derived from various MS acquisitions and platforms, we conducted five different experiments at two levels. At the first level, we performed experiments on the N2 and nanoPOTS datasets with exactly the same set of cell types (Fig. 3a–c). These two datasets have three cell types: C10 cells, RAW cells and SVECs. At the second level, we tested the model's ability to reconcile the same cell type (cell line) from different acquisitions while maintaining the diversity of the different cell types. Specifically, we corrected the batch effects of SCoPE2_Leduc and plexDIA in Fig. 3d,e (merged U-937 cells), pSCoPE_Huffman and plexDIA in Extended Data Fig. 4 (merged PDAC (HPAFII) cells), pSCoPE_Leduc and plexDIA in Extended Data Fig. 5 (merged melanoma cells and U-937 cells) and pSCoPE_Leduc and SCoPE2_Leduc in Extended Data Fig. 6 (merged U-937 cells). In each data integration experiment and for all comparison methods, the shared proteins were exploited for analysis.

Cell type annotation performance. A cell type annotation (label transfer) task was conducted on the N2 and nanoPOTS datasets. After correcting the batch effect via scPROTEIN, the cell labels from one dataset could be naturally transferred to another dataset. The KNN algorithm was employed to infer the labels of the query set from the reference set in the learned latent space.

Application

Performance achieved on clinical single-cell proteomic data. The samples from the ECCITE-seq data were clustered by the Leiden algorithm (resolution of 0.6) based on the learned embeddings. This process was implemented via scanpy.tl.leiden. Next, we computed the PCCs among all clusters based on the average protein abundance values within each cluster by the dendrogram function in Scanpy. If the PCC between two clusters exceeded the threshold (0.99), we considered them highly similar and combined them accordingly. This process led to the clustering and subsequent combination of the cells into 18 subgroups. We could observe the different sample proportions and distinct protein signatures in various clusters using cell embedding. Differential expression protein analysis was carried out using the rank_gene_groups function in the SCANPY package with the statistical *t*-test method. The functional profiling of the GO for each subpopulation was performed by gProfiler[62,63].

Performance achieved on spatial single-cell proteomic data. The BaselTMA dataset was exploited to confirm the scalability of scPROTEIN for single-cell spatial proteomics. In each TMA slice, we constructed a cell graph, where the node attributes were the expression profiles and edges were established between cells based on their spatial cell locations. Then, spatially informative embeddings could be naturally obtained via the scPROTEIN model. The learned embeddings were then clustered via the Leiden algorithm (resolution of 0.6).

Evaluation metrics

The ARI[64], ASW[65], NMI[66] and purity score (PS[67]) were used for the cell clustering evaluation, and 1-batch-wise metrics (1-ASW_batch, 1-ARI_batch, 1-NMI_batch and 1-PS_batch) were used to measure the batch correction performance attained for the same cell type. We applied these metrics to the original feature space before performing dimensionality reduction. Accuracy and the macro-f1 score were applied in the cell type annotation experiments. For all metrics, a higher score represents better performance. These metrics are defined as follows:

ARI[64].

$$\mathrm{ARI} = \frac{\sum_{ij}\binom{n_{ij}}{2} - \left[\sum_i \binom{a_i}{2}\sum_j \binom{b_j}{2}\right]/\binom{n}{2}}{\frac{1}{2}\left[\sum_i \binom{a_i}{2} + \sum_j \binom{b_j}{2}\right] - \left[\sum_i \binom{a_i}{2}\sum_j \binom{b_j}{2}\right]/\binom{n}{2}}$$

where n_{ij} denotes the number of cells that are assigned to clusters i and j based on the true labels and the clustering labels, respectively, a_i is the number of cells from cluster i based on the true labels and b_j is the number of cells assigned to cluster j according to the clustering labels.

ASW[65].

$$\mathrm{ASW} = \frac{1}{N}\sum_i \frac{p(i) - q(i)}{\max\{q(i), p(i)\}}$$

where $q(i)$ represents the average distance from cell i to other cells in the same cluster, $p(i)$ is the average distance between cell i and other cells in the nearest cluster and N is the total number of cells.

NMI[66].

$$\mathrm{NMI} = \frac{\mathrm{MI}(\hat{Y}; Y)}{\sqrt{H(\hat{Y})H(Y)}}$$

$\mathrm{MI}(\hat{Y}; Y)$ denotes the mutual entropy between the predicted categorical distributions $\hat{Y}$ and the true categorical clustering distributions Y, and H is the Shannon entropy measure.

PS[67].

$$\mathrm{PS} = \frac{1}{N}\sum_i \max_j(n_{ij})$$

where n_{ij} denotes the number of cells assigned to cluster i, whose ground-truth label belongs to partition j. The PS quantifies the extent to which a cluster contains cells from only one partition.

Accuracy (Acc).

$$\mathrm{Acc} = \frac{1}{N}\sum_{i} I(\hat{Y}_i = Y_i)$$

$I(\cdot)$ denotes the indicator function.

Macro-f1. The macro-f1 score was calculated by the function f1_score with average of 'macro' in the scikit-learn package in Python.

shd. In addition, we defined a shd based on the constructed cell graph to quantify the spatial heterogeneity of the data. We first compared the cluster assignment of each cell with its spatially neighboring cells, which were identified by the constructed spatial informative graph. Then, the proportion of neighboring cells with the same cluster labels was used as the measure, which can be expressed as:

$$\mathrm{shd} = \frac{\sum_i \sum_{j\in \mathrm{Neigh}(i)} I(\mathrm{cluster}_i = \mathrm{cluster}_j)}{\sum_i \sum_{j\in \mathrm{Neigh}(i)} 1}$$

where $j \in \mathrm{Neigh}(i)$ *if* $e_{ij} \in E$ and $I(\cdot)$ denotes the indicator function. A high shd value denotes a highly compartmentalized phenotype, while a low shd value denotes a high level of intercell type mixing.

Hyperparameters

We set different hidden dimensions d according to the number of input proteins. The embedding dimensionality for the SCoPE2_Specht dataset learning was set to 400 (3,042 proteins). For the integration experiments conducted on pSCoPE_Huffman&plexDIA, pSCoPE_Leduc&plexDIA and pSCoPE_Leduc&SCoPE2_Leduc, since the numbers of input proteins were smaller (947, 1,075 and 1,000 input proteins, respectively), we set the dimensionality to 128. For the data integration experiments conducted on N2&nanoPOTS and SCoPE2_Leduc&plexDIA, we used 64 as the embedding dimensionality (762 and 682 proteins, respectively). For the ECCITE-seq and spatial proteomic datasets, since the numbers of proteins were limited (49 and 38), we used 32 as the hidden dimensionality. We also investigated the influence of the number of prototypes K, and the results are shown in Extended Data Fig. 2. We found that the scPROTEIN model is robust to K, and we used 2 as the default parameter for the SCoPE2_Specht dataset and 3 for all the data integration experiments. For the baseline methods, the hyperparameters are illustrated in Supplementary Table 2.

Statistical analysis

The two-sided Mann–Whitney test was utilized for the significance test and was implemented via the scipy.stats.mannwhitneyu package.

Reporting summary

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

Data availability

All data used in this study are publicly available, and their usages are fully illustrated in Methods. The SCoPE2_Specht[25] dataset was downloaded from ref. 48. The nanoPOTS dataset[11] was downloaded at MassIVE data repository with ID MSV000084110. The N2 dataset[12] was downloaded from MassIVE data repository with ID MSV000086809. The SCoPE2_Leduc dataset[3] was downloaded from ref. 49. The plexDIA dataset[5] was downloaded from ref. 50. The pSCoPE_Huffman dataset[15] was downloaded from ref. 51 (derived from their original 'Benchmarking experiments: Fig. 1b,e data'). The pSCoPE_Leduc dataset[3] was downloaded from ref. 52. The ECCITE-seq dataset[29] was downloaded from Gene Expression Omnibus with accession number GSE126310. The BaselTMA dataset[30] was downloaded from Zenodo[53]. The T-SCP dataset[24] was downloaded from the PRIDE partner repository (accession no. PXD024043). Source data are provided with this paper.

Code availability

The codes were implemented in Python and are released at GitHub (https://github.com/TencentAILabHealthcare/scPROTEIN) and Zenodo (https://doi.org/10.5281/zenodo.10547614)[68] with detailed instructions.

References

39. Zhu, Y. et al. Deep graph contrastive representation learning. in *ICML Workshop on Graph Representation Learning and Beyond* (2020).
40. Rong, Y., Huang, W., Xu, T. & Huang, J. DropEdge: towards deep graph convolutional networks on node classification. in *International Conference on Learning Representations* (2020).
41. Kipf, T. N. & Welling, M. Semi-supervised classification with graph convolutional networks. in *International Conference on Learning Representations* (2017).
42. Tschannen, M., Djolonga, J., Rubenstein, P. K., Gelly, S. & Lucic, M. On mutual information maximization for representation learning. in *International Conference on Learning Representations* (2019).
43. Chen, T., Kornblith, S., Norouzi, M. & Hinton, G. A simple framework for contrastive learning of visual representations. *PMLR* https://proceedings.mlr.press/v119/chen20j.html (2020).
44. van den Oord DeepMind, A., Li DeepMind, Y. & Vinyals DeepMind, O. Representation learning with contrastive predictive coding. Preprint at *arXiv* https://doi.org/10.48550/arxiv.1807.03748 (2018).
45. Wang, Y. & Yang, Y. Bayesian robust graph contrastive learning. Preprint at *arXiv* https://doi.org/10.48550/arxiv.2205.14109 (2022).
46. Ahmed, M., Seraj, R. & Islam, S. M. S. The *k*-means algorithm: a comprehensive survey and performance evaluation. *Electronics* **9**, 1295 (2020).
47. Kingma, D. & Ba, J. Adam: A method for stochastic optimization. in *International Conference on Learning Representations* (2015).
48. SCoPE2 data processed to ASCII text matrices. *slavovlab* https://scp.slavovlab.net/Specht_et_al_2019 (2019).
49. Raw data from experiments benchmarking nPOP. *slavovlab* https://scp.slavovlab.net/Leduc_et_al_2021 (2021).
50. plexDIA data organized by experiments. *slavovlab* https://scp.slavovlab.net/Derks_et_al_2022 (2022).
51. pSCoPE data processed to ASCII text matrices. *slavovlab* https://scp.slavovlab.net/Huffman_et_al_2022_v1 (2022).
52. Model systems: cell lines of monocytes (U937 cells) and melanoma cells (WM989-A6-G3). *slavovlab* https://scp.slavovlab.net/Leduc_et_al_2022 (2022).
53. The single-cell pathology landscape of breast cancer. *Zenodo* https://doi.org/10.5281/zenodo.3518284 (2019).
54. Wolock, S. L., Lopez, R. & Klein, A. M. Scrublet: computational identification of cell doublets in single-cell transcriptomic data. *Cell Syst.* **8**, 281–291 (2019).
55. scrublet. *GitHub* https://github.com/swolock/scrublet (2019).
56. scikit-learn. *scikit-learn* https://scikit-learn.org/stable/ (2011).
57. scanpy. *pypi* https://pypi.org/project/scanpy/ (2018).
58. MAGIC. *GitHub* https://github.com/KrishnaswamyLab/MAGIC (2018).
59. harmony-pytorch. *pypi* https://pypi.org/project/harmony-pytorch/ (2019).
60. scanorama. *pypi* https://pypi.org/project/scanorama/ (2019).
61. AutoClass. *GitHub* https://github.com/datapplab/AutoClass (2022).
62. Reimand, J. et al. g:Profiler—a web server for functional interpretation of gene lists. *Nucleic Acids Res.* **44**, W83–W89 (2016).

63. g:Profiler. *Bioinformatics, Algorithmics and Data Mining Group* https://biit.cs.ut.ee/gprofiler/gost (2016).
64. Hubert, L. & Arabie, P. Comparing partitions. *J. Classif.* **2**, 193–218 (1985).
65. Rousseeuw, P. J. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *J. Comput. Appl. Math.* **20**, 53–65 (1987).
66. Estévez, P. A., Tesmer, M., Perez, C. A. & Zurada, J. M. Normalized mutual information feature selection. *IEEE Trans. Neural Netw.* **20**, 189–201 (2009).
67. Mogotsi, I. C. & Christopher, D. in *Introduction to Information Retrieval* (eds Manning C. D. et al.) 192–195 (Cambridge Univ. Press, 2009).
68. Li, W. A versatile deep graph contrastive learning framework for single-cell proteomics embedding. *Zenodo* https://doi.org/10.5281/zenodo.10547614 (2024).

Acknowledgements

The authors thank R. Aebersold for his valuable suggestion regarding this work, P. Zhao for model development advice and S. Zhu for providing valuable knowledge in the field of MS. This work was supported by the National Natural Science Foundation of China (61973174 to H.Z. and 62373200 to H.Z.), the Key-Area Research and Development Program of Guangdong Province (2021B0101420005 to F.Y.) and the Young Elite Scientists Sponsorship Program by CAST (2023QNRC001 to F.Y.). The funders had no role in study design, data collection and analysis, decision to publish or preparation of the manuscript.

Author contributions

F.Y. and J.Y. conceived and designed the project. W.L. and H.Z. developed the method. W.L. performed the research and conducted the experiments under the supervision of F.Y., H.Z. and J.Y. W.L. and F.Y. analyzed the results. W.L. and F.Y. wrote the manuscript. W.L. finished the figures under the guidance of F.Y. and J.Y. F.W. helped polish the figures and manuscript. H.Z. and J.Y. revised the manuscript. Y.R. gave suggestions for building the graph model and improving the manuscript. L.L. helped with the revision and data analysis tasks. B.W. provided suggestions for utilizing trustworthy AI and improved the manuscript. All authors reviewed and approved the manuscript.

Competing interests

The authors declare no competing interests.

Additional information

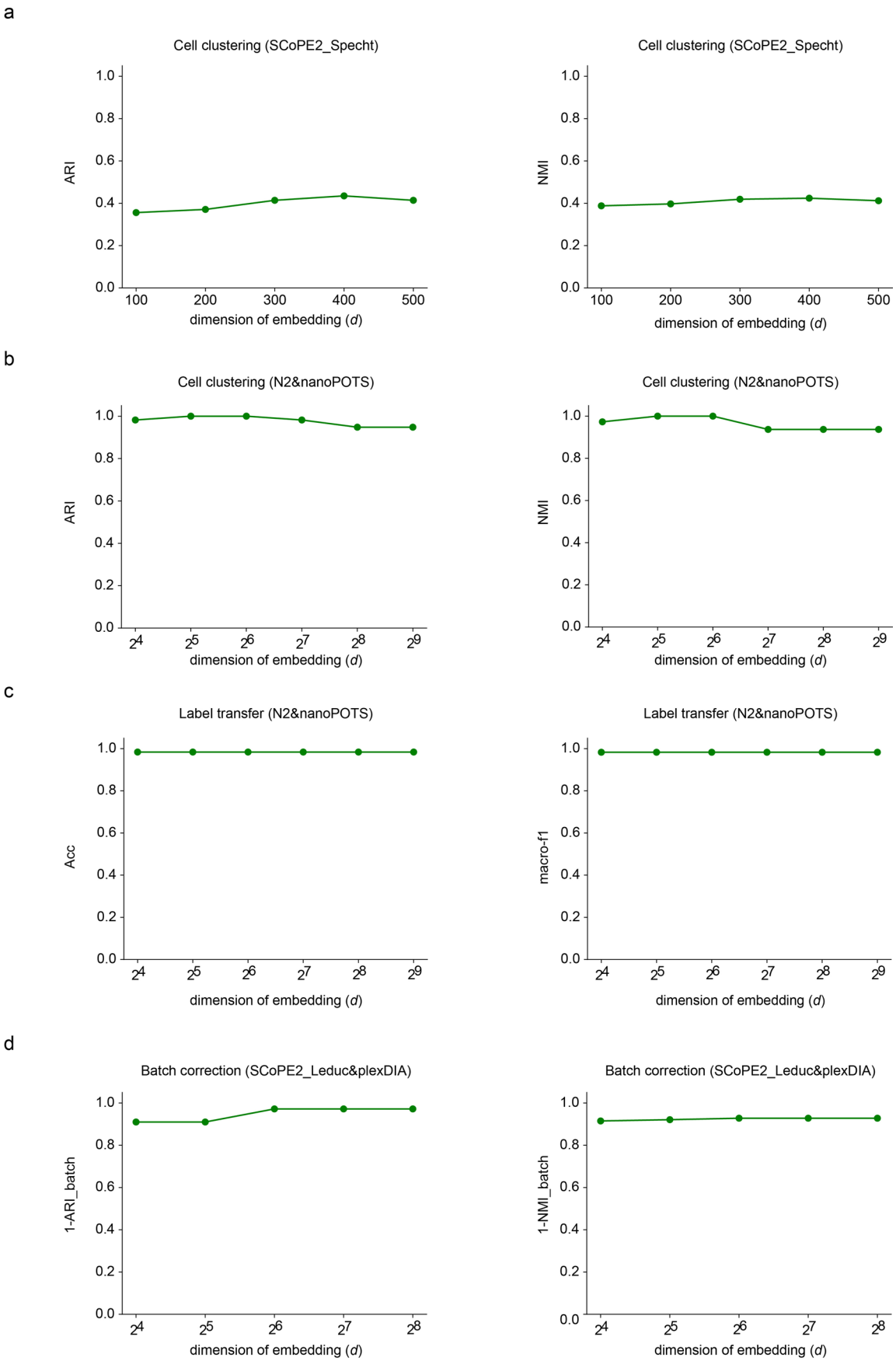
Extended data is available for this paper at https://doi.org/10.1038/s41592-024-02214-9.

Supplementary information The online version contains supplementary material available at https://doi.org/10.1038/s41592-024-02214-9.

Correspondence and requests for materials should be addressed to Han Zhang or Jianhua Yao.

Peer review information *Nature Methods* thanks the anonymous reviewers for their contribution to the peer review of this work. Primary Handling Editor: Arunima Singh, in collaboration with the *Nature Methods* team.

https://doi.org/10.1038/s41592-024-02214-9

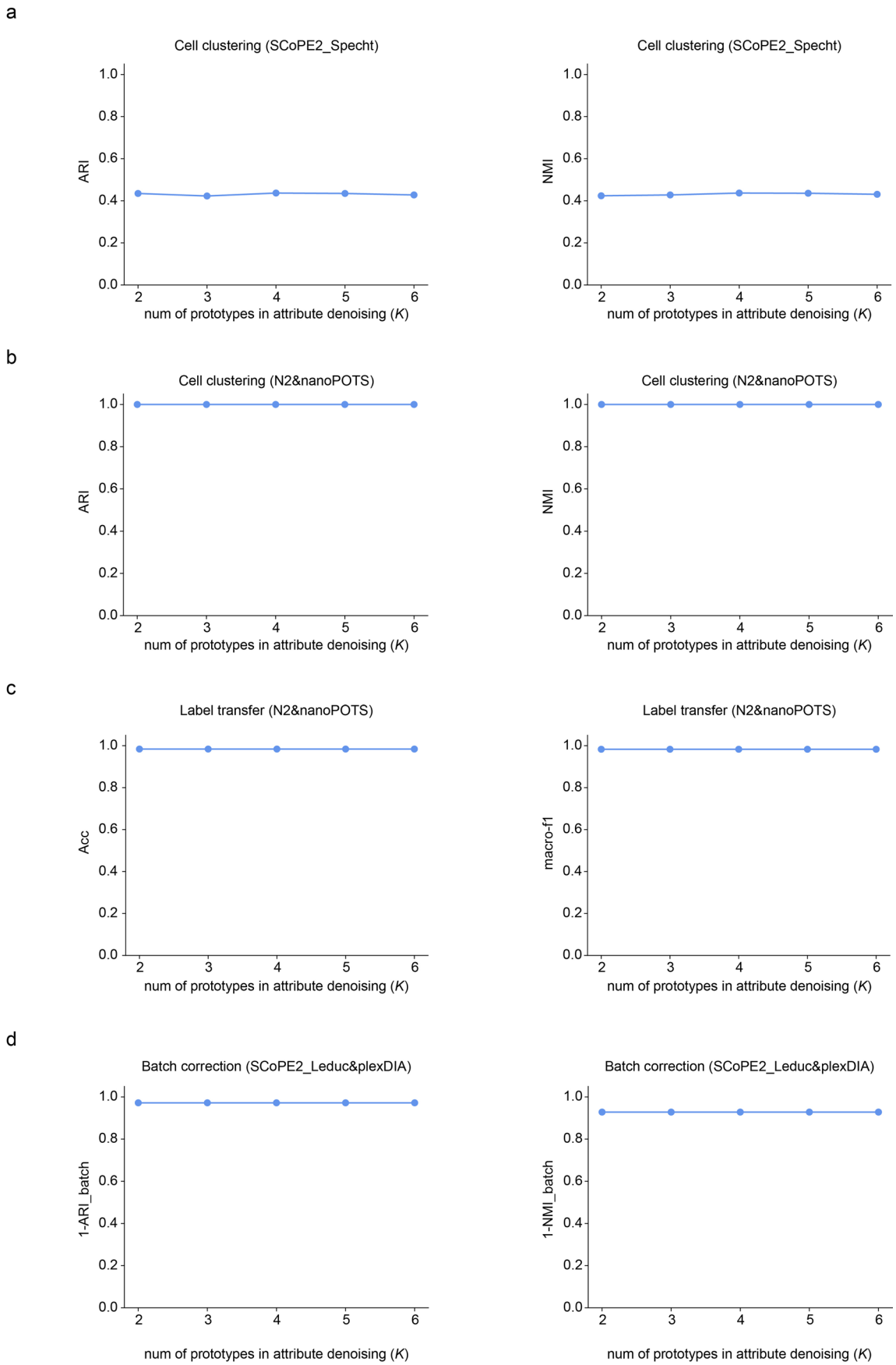


Extended Data Fig. 1 | See next page for caption.

https://doi.org/10.1038/s41592-024-02214-9

Extended Data Fig. 1 | Systematic analysis concerning the sensitivity of the hyperparameter *d*. **a**, Influence of the hyperparameter *d* on the ARI and NMI scores achieved in the clustering task using the SCoPE2_Specht dataset. *d* is the dimensionality of the learned latent embeddings. **b**, Influence of hyperparameter *d* on the ARI_cell and NMI_cell results obtained with the cell type labels as the ground truth in the data integration task using the N2 and nanoPOTS datasets. **c**, Influence of hyperparameter *d* on the accuracy and macro-f1 score values attained in the label transfer task (transferring from N2 to nanoPOTS). **d**, Influence of hyperparameter *d* on the 1-ARI_batch and 1-NMI_batch results obtained with the batch labels as the ground truth in the data integration task using the SCoPE2_Leduc and plexDIA datasets.

https://doi.org/10.1038/s41592-024-02214-9

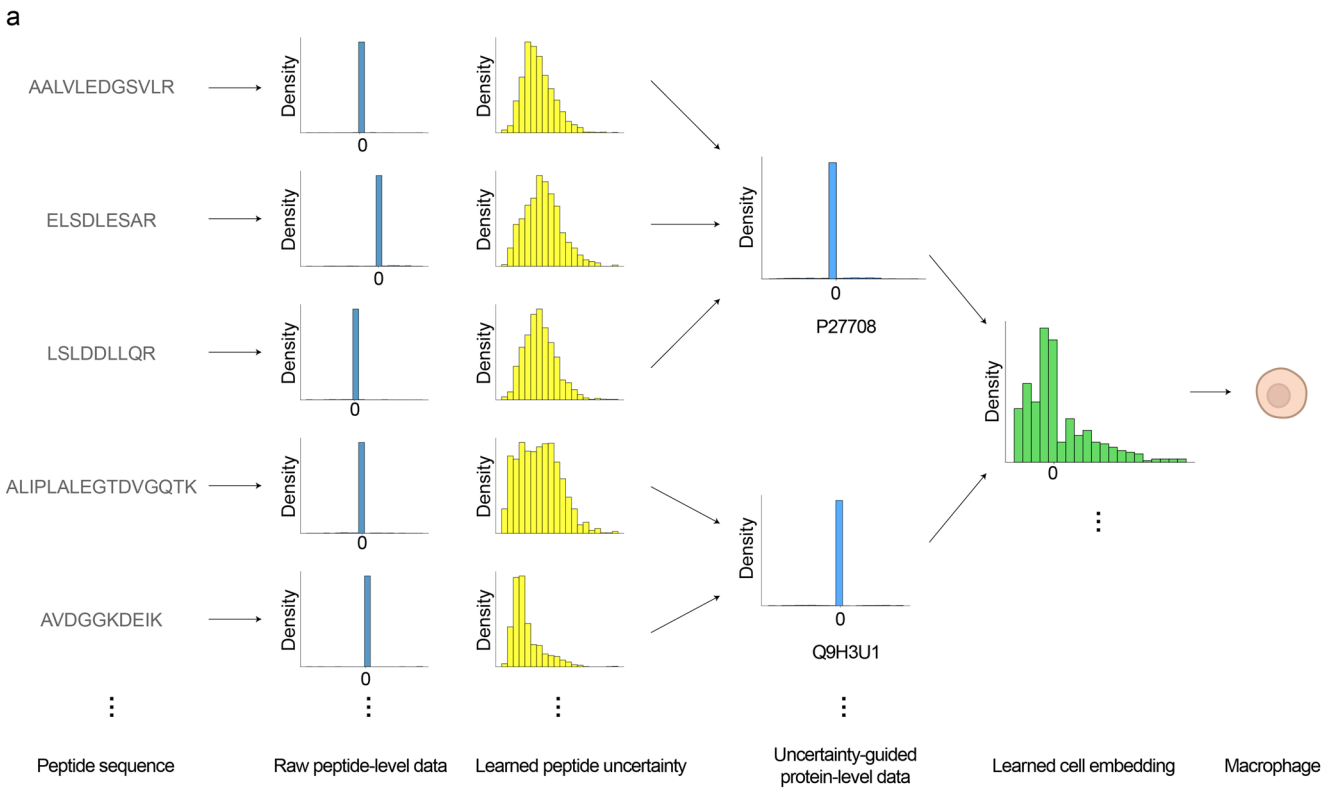


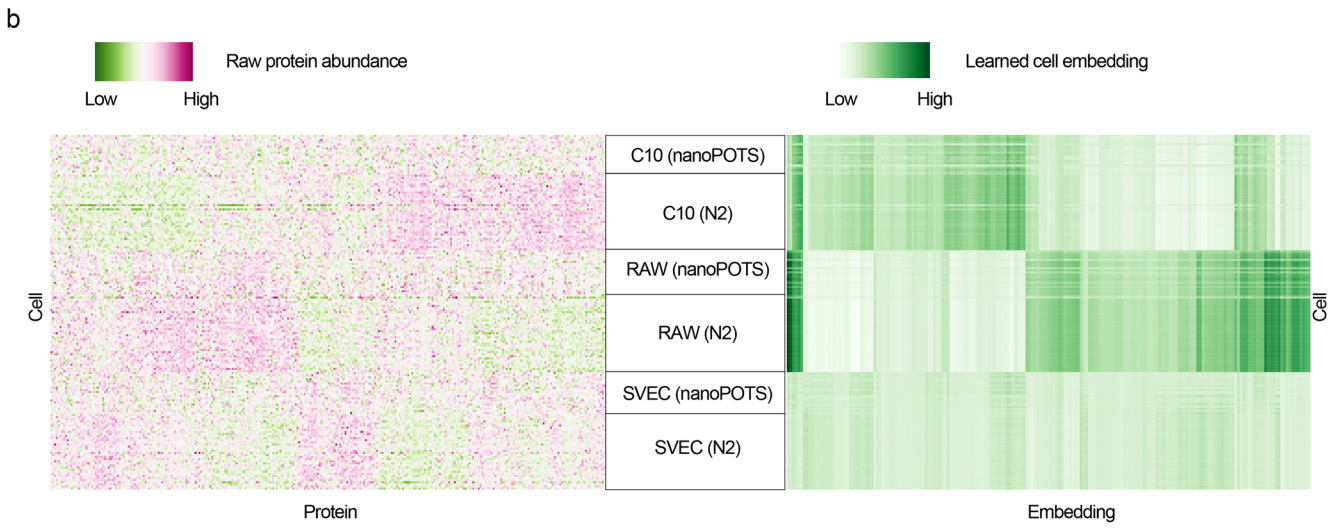
Extended Data Fig. 2 | See next page for caption.

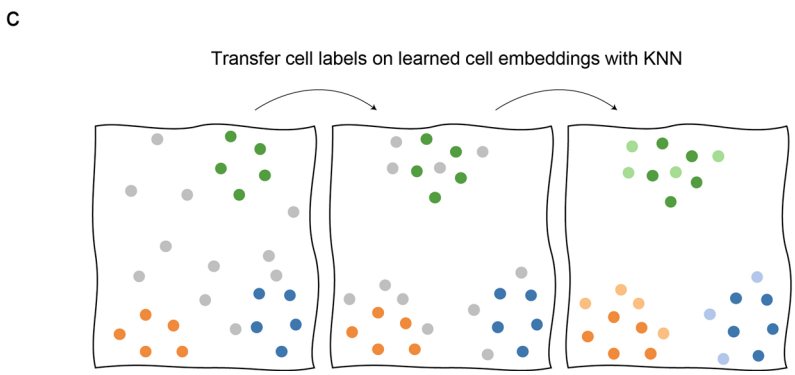
https://doi.org/10.1038/s41592-024-02214-9

Extended Data Fig. 2 | Systematic analysis concerning the sensitivity of the hyperparameter *K*. **a**, Influence of the hyperparameter *K* on the ARI and NMI scores achieved in the clustering task using the SCoPE2_Specht dataset. *K* is the number of prototypes in the attribute denoising process of *stage 2*. **b**, Influence of hyperparameter *K* on the ARI_cell and NMI_cell results obtained with the cell type labels as the ground truth in the data integration task using the N2 and nanoPOTS datasets. **c**, Influence of hyperparameter *K* on the accuracy and macro-f1 score values achieved in the label transfer task (transferring from N2 to nanoPOTS). **d**, Influence of hyperparameter *K* on the 1-ARI_batch and 1-NMI_batch results obtained with the batch labels as the ground truth in the data integration task using the SCoPE2_Leduc and plexDIA datasets.

https://doi.org/10.1038/s41592-024-02214-9





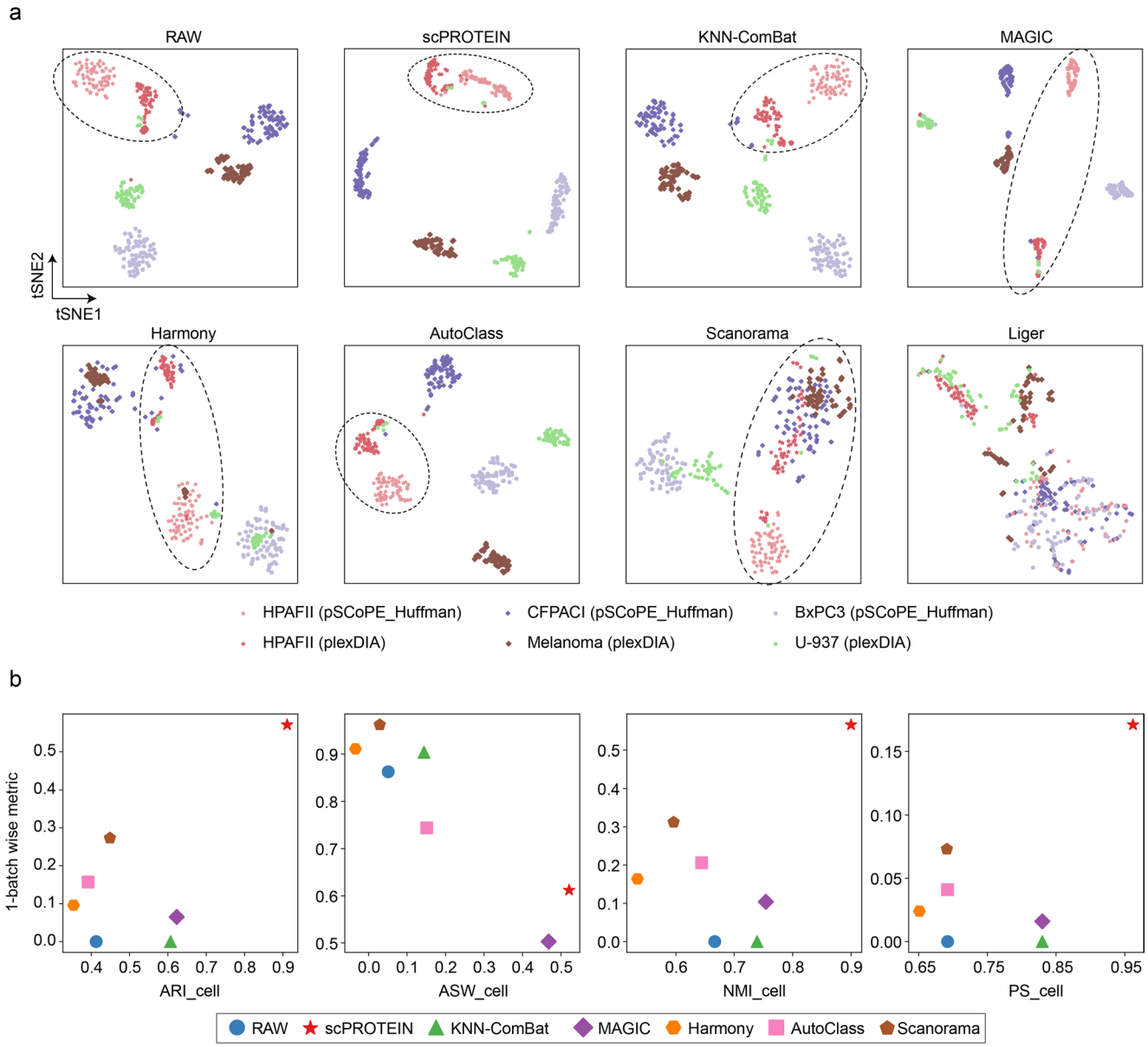


Extended Data Fig. 3 | See next page for caption.

https://doi.org/10.1038/s41592-024-02214-9

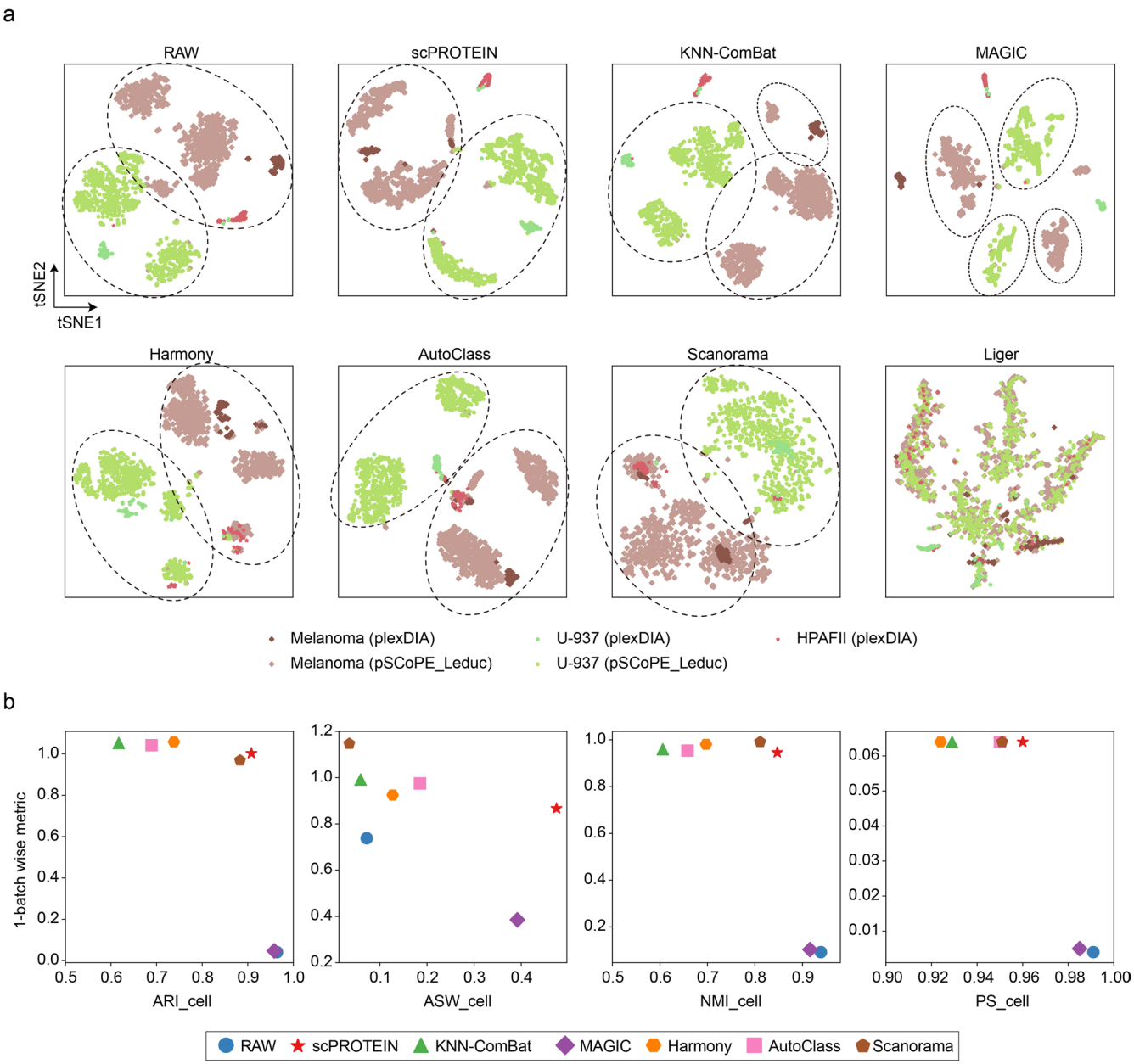
Extended Data Fig. 3 | Embedding visualizations produced for the SCoPE2_Specht, N2 and nanoPOTS datasets. a, The embedding learning process for the SCoPE2_Specht dataset. From left to right, we depict the learning process from the raw peptide level, the learned peptide uncertainty, the aggregated protein levels after executing uncertainty adjustments and the final learned cell embeddings. **b**, Visualization of parts of the raw protein profiles and the embeddings learned by scPROTEIN on the N2 and nanoPOTS datasets. In the left panel, we can observe that the batch effect is exhibited in the same cell type across the two datasets. In the right panel, scPROTEIN can greatly mitigate the batch effect, and the same cell type tends to show similar patterns. **c**, Diagram showing the label transfer process based on the learned embeddings. In the left panel, the gray dots represent cells with unknown labels from the query set, and the dots with other colors represent cells with known labels from the reference set. When the batch effect is effectively removed (middle panel), the gray cells can then be annotated accurately by KNN (right panel).

https://doi.org/10.1038/s41592-024-02214-9



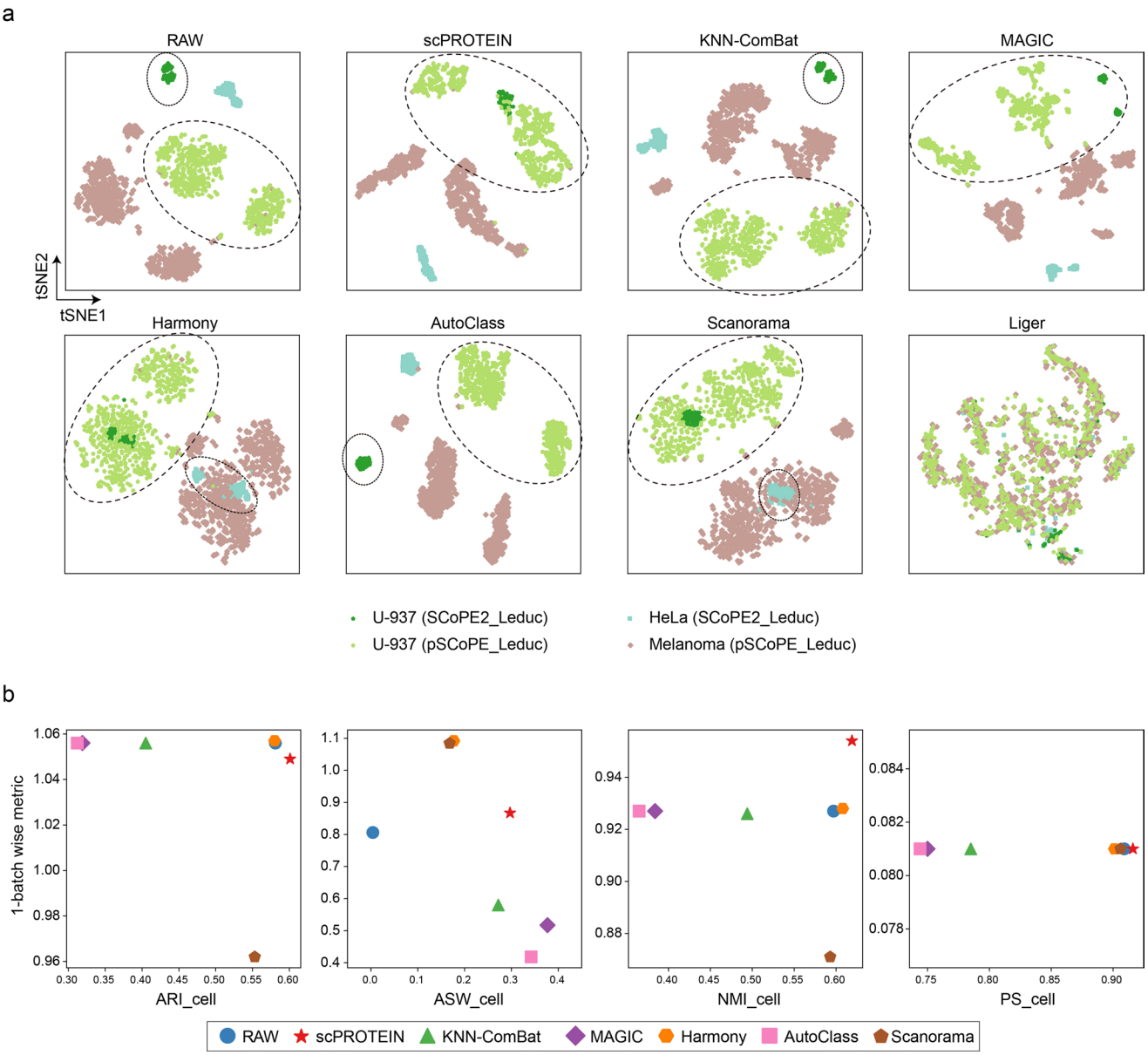
Extended Data Fig. 4 | Data integration results obtained on the pSCoPE_Huffman and plexDIA datasets. a, t-SNE plots showing the cells of the pSCoPE_Huffman and plexDIA datasets, colored by their data acquisitions and cell lines. HPAFII is the shared cell line between the two datasets. **b**, ARI_cell, ASW_cell, NMI_cell, and PS_cell results produced by scPROTEIN and the comparison methods with the cell type labels as the ground truth (x-axis) and the 1-metrics with batch labels as the ground truth (y-axis) on the pSCoPE_Huffman and plexDIA datasets. **c**, Heatmap showing the estimated uncertainties of each peptide signal across cells, colored by the estimated uncertainty calculated on the pSCoPE_Huffman dataset. The batch information and protein information are shown below the heatmap and on the right-hand side of the heatmap, respectively.

https://doi.org/10.1038/s41592-024-02214-9



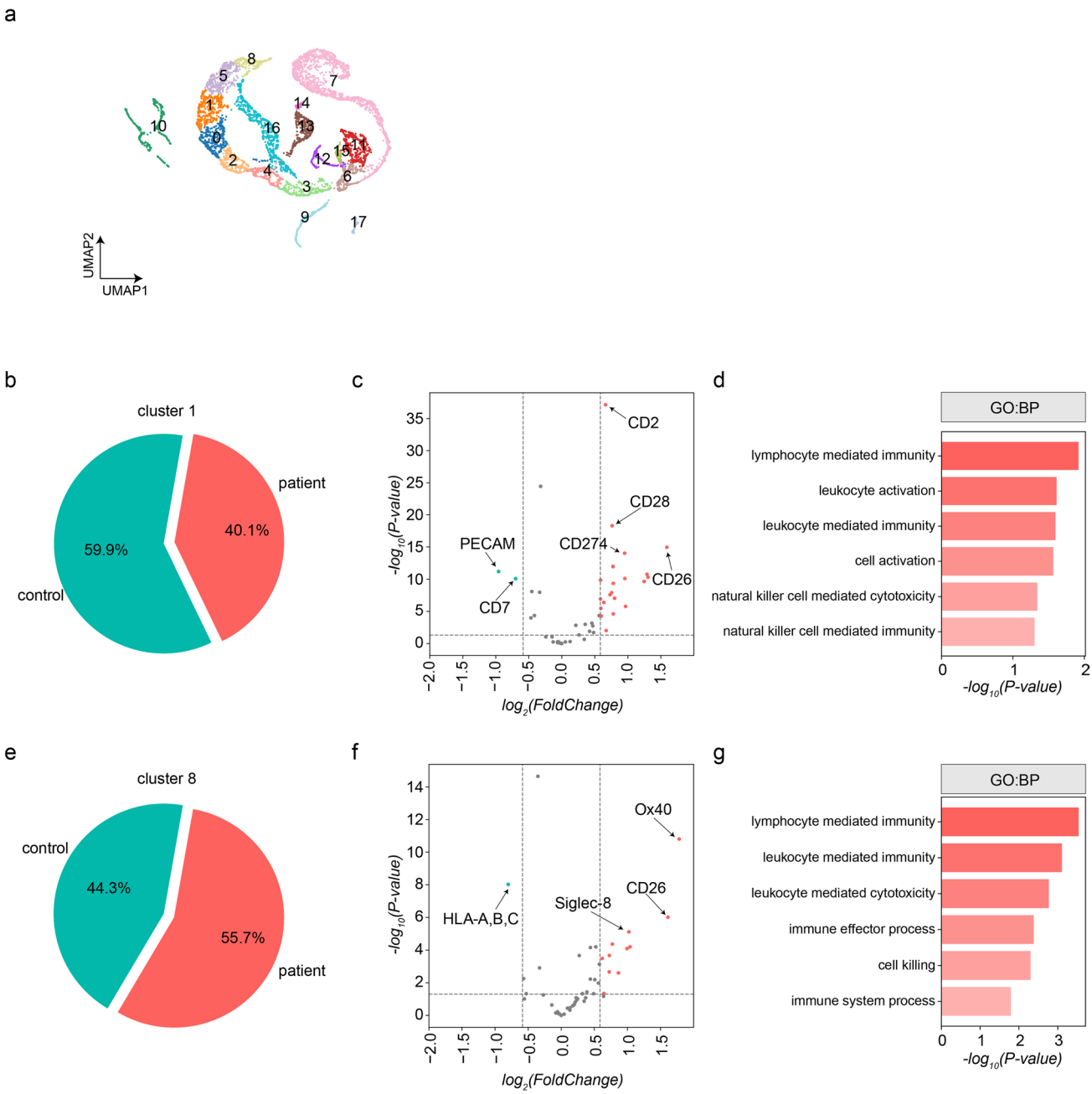
Extended Data Fig. 5 | Data integration results obtained on the pSCoPE_Leduc and plexDIA datasets. a, t-SNE plots showing the cells of the pSCoPE_Leduc and plexDIA datasets, colored by their data acquisitions and cell lines. Melanoma and U-937 are the shared cell types between in the two datasets. **b**, ARI_cell, ASW_cell, NMI_cell, and PS_cell results produced by scPROTEIN and the comparison methods with the cell type labels as the ground truth (x-axis) and the 1-metrics with batch labels as ground truth (y-axis) on the pSCoPE_Leduc and plexDIA datasets.

https://doi.org/10.1038/s41592-024-02214-9



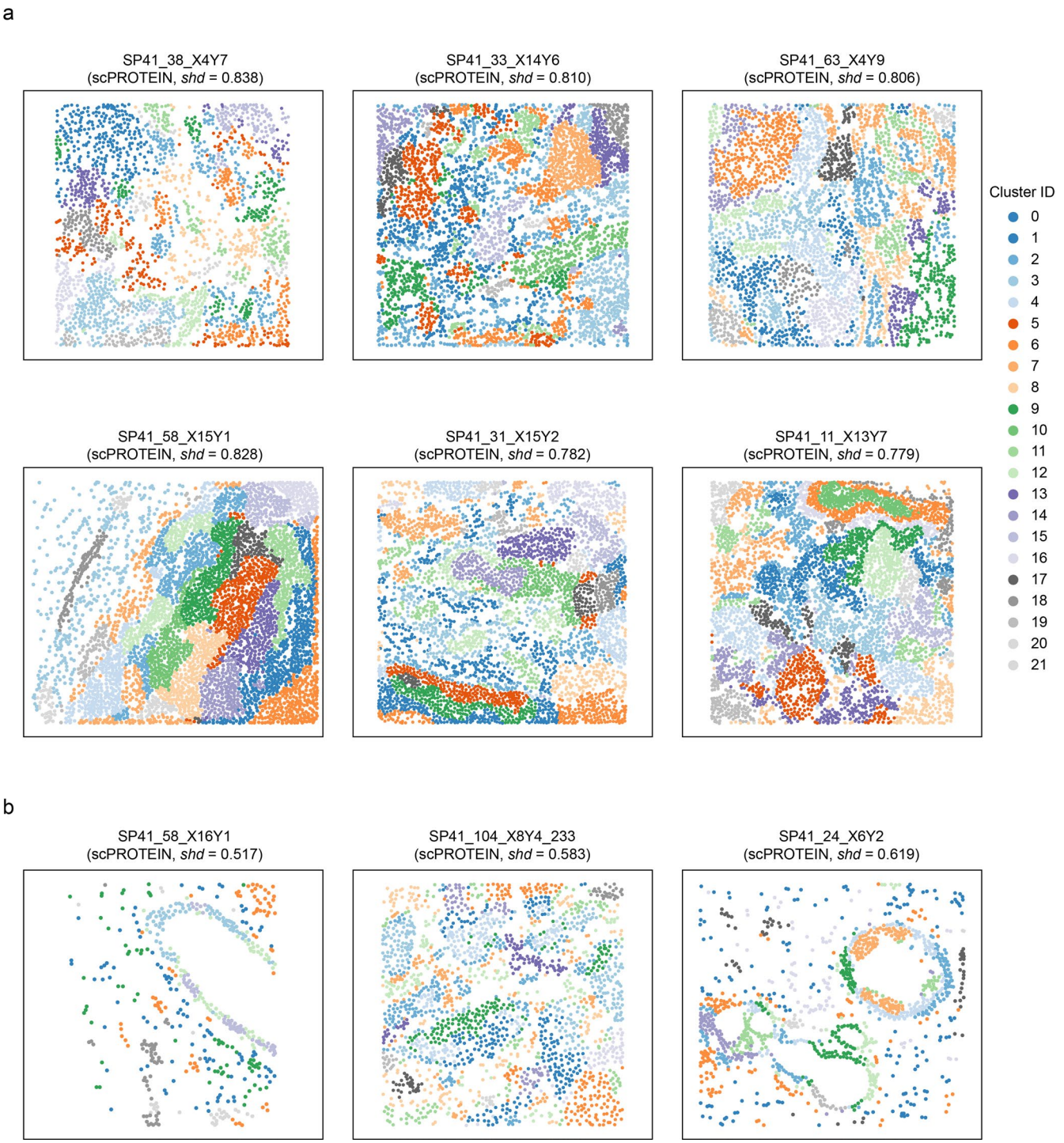
Extended Data Fig. 6 | Data integration results obtained on the pSCoPE_Leduc and SCoPE2_Leduc datasets. a, t-SNE plots showing the cells of the pSCoPE_Leduc and SCoPE2_Leduc datasets, colored by their data acquisitions and cell lines. U-937 is the shared cell type between the two datasets. **b**, ARI_cell, ASW_cell, NMI_cell, and PS_cell results obtained by scPROTEIN and the comparison methods with the cell type labels as the ground truth (x-axis) and the 1-metrics with batch labels as ground truth (y-axis) on the pSCoPE_Leduc and SCoPE2_Leduc datasets.

https://doi.org/10.1038/s41592-024-02214-9



Extended Data Fig. 7 | Application of scPROTEIN to clinical proteomic dataset. a, UMAP of the scPROTEIN embeddings, which shows the cells colored by their clustering results. **b**, Detailed ratio of the cluster 1 cells for the control donor and CTCL donor. **c**, Volcano plot showing the differentially expressed proteins found by contrasting the healthy cells and CTCL cells in cluster 1. **d**, Top GO terms in the BP for the identified upregulated proteins of the CTCL cells in cluster 1. The p-values are computed using Fisher's one-tailed test and adjusted by the multiple-hypotheses testing method (g:SCS) of gProfiler. **e**, Detailed ratio of the cluster 8 cells for the control donor and CTCL donor. **f**, Volcano plot showing the differentially expressed proteins found by contrasting the healthy cells and CTCL cells in cluster 8. **g**, Top GO terms in the BP for the identified upregulated proteins of the CTCL cells in cluster 8. The p-values are computed using Fisher's one-tailed test and adjusted by the multiple-hypotheses testing method (g:SCS) of gProfiler.

https://doi.org/10.1038/s41592-024-02214-9



Extended Data Fig. 8 | Application of scPROTEIN to spatial proteomic data. a, Visualizations of the learned spatial informative embeddings and the spatial heterogeneity degrees within tumor samples. **b**, Visualizations of the learned spatial informative embeddings and the spatial heterogeneity degrees within nontumor samples.

nature portfolio

Corresponding author(s): Jianhua Yao

Last updated by author(s): Jan 23, 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	scanpy==1.8.2
Data analysis	Custom software: https://github.com/TencentAILabHealthcare/scPROTEIN Public softwares: torch==1.10.0 pandas==1.3.5 numpy==1.22.3 torch_geometric==2.0.4 scikit-learn==1.1.1 scipy==1.8.1 scanpy==1.8.2 scrublet==0.2.3 scanorama==1.7.3 harmony-pytorch==0.1.7 magic-impute==3.0.0 rliger==1.0.0 AutoClass (https://github.com/datapplab/AutoClass)

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

All data used in this study are publicly available and the usages were fully illustrated in the Methods section. The SCoPE2_Specht dataset was downloaded from

https://scp.slavovlab.net/Specht_et_al_2019. The nanoPOTS dataset was downloaded at MassIVE data repository with ID: MSV000084110. The N2 dataset was downloaded from MassIVE data repository with ID: MSV000086809. The SCoPE2_Leduc dataset was downloaded from https://scp.slavovlab.net/Leduc_et_al_2021. The plexDIA dataset was downloaded from https://scp.slavovlab.net/Derks_et_al_2022. The pSCoPE_Huffman dataset was downloaded from https://scp.slavovlab.net/Huffman_et_al_2022_v1 (derived from their original "Benchmarking experiments: Figure 1b/e data"). The pSCoPE_Leduc dataset was downloaded from https://scp.slavovlab.net/Leduc_et_al_2022. The ECCITE-seq dataset was downloaded from Gene Expression Omnibus with accession number GSE126310. The BaselTMA dataset was downloaded at Zenodo (https://doi.org/10.5281/zenodo.3518284). The T-SCP dataset was downloaded from the PRIDE partner repository (accession no. PXD024043).

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	In this work, we investigated the model performance on a variety of independent and publicly available single-cell proteomics datasets : (1) SCoPE2_Specht contains 1490 cells. (2) nanoPOTS dataset contains 61 cells. (3) N2 dataset contains 108 cells. (4) SCoPE2_Leduc dataset contains 163 cells. (5) plexDIA dataset contains 164 cells. (6) pSCoPE_Huffman dataset contains 206 cells. (7) pSCoPE_Leduc dataset contains 1543 cells. (8) ECCITE-seq dataset contains 13000 cells. (9) The number of cells for BaselTMA dataset varies across different slices (9-6940 cells). (10) T-SCP dataset contains 225 cells. All the used datasets are publicly available and illustrated in the data availability statement. The datasets are used in different experiments with elaborated experiment design in the method section. A total of 10 independent benchmark datasets were assembled to encompass the current mainstream single-cell proteomics experiment protocols, including the most comprehensive cell types, the largest cell numbers of mass spectrometry single-cell proteomics to date, as well as covering the main species. Thus the datasets should be sufficient as in most single-cell studies focused on demonstration of methodological development.
Data exclusions	No data was excluded from the analysis.
Replication	We didn't have control over the experimental design because we utilized pre-existing public datasets. Thus, this is not applicable
Randomization	We didn't have control over the experimental design because we utilized pre-existing public datasets. Thus, this is not applicable
Blinding	We didn't have control over the experimental design because we utilized pre-existing public datasets. Thus, this is not applicable

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