OXFORD

Briefings in Bioinformatics, 2025, **26(3)**, bbaf243
https://doi.org/10.1093/bib/bbaf243
Problem Solving Protocol

scaLR: a low-resource deep neural network-based platform for single cell analysis and biomarker discovery

Saiyam Jogani[1,‡], Anand Santosh Pol[1,‡], Mayur Prajapati[2,‡], Amit Samal[2], Kriti Bhatia[1], Jayendra Parmar[2], Urvik Patel[2], Falak Shah[2], Nisarg Vyas[2], Saurabh Gupta[2,*]

[1]Department of Generative AI & Bioinformatics, Infocusp Innovations, Laxman Nagar Baner, Pune 411045, Maharashtra, India
[2]Department of Generative AI & Bioinformatics, Infocusp Innovations, Gala-hub, Bopal, Ahmedabad 380058, Gujarat, India
*Corresponding author. Department of Generative AI & Bioinformatics, Infocusp Innovations, Gala-hub, Bopal, Ahmedabad 380058, Gujarat, India.
E-mail: saurabh@infocusp.com
‡Saiyam Jogani, Anand Santosh Pol, and Mayur Prajapati contributed equally to this work.

Abstract

Single-cell ribonucleic acid (RNA) sequencing (scRNA-seq) produces vast amounts of individual cell profiling data. Its analysis presents a significant challenge in accurately annotating cell types and their associated biomarkers. Different pipelines based on deep neural network (DNN) methods have been employed to tackle these issues. These pipelines have arisen as a promising resource and can extract meaningful and concise features from noisy, diverse, and high-dimensional data to enhance annotations and subsequent analysis. Existing tools require high computational resources to execute large sample datasets. We have developed a cutting-edge platform known as scaLR (Single-cell analysis using low resource) that efficiently processes data into feature subsets, samples in batches to reduce the required memory for processing large datasets, and running DNN models in multiple central processing units. scaLR is equipped with data processing, feature extraction, training, evaluation, and downstream analysis. Its novel feature extraction algorithm first trains the model on a feature subset and stores the importance of the features for all the features in that subset. At the end of the training of all subsets, the top-K features are selected based on their importance. The final model is trained on top-K features; its performance evaluation and associated downstream analysis provide significant biomarkers for different cell types and diseases/traits. Our findings indicate that scaLR offers comparable prediction accuracy and requires less model training time and computational resources than existing Python-based pipelines. We present scaLR, a Python-based platform, engineered to utilize minimal computational resources while maintaining comparable execution times and analysis costs to existing frameworks.

Keywords: cell type; annotation; classification; ML/DNN; feature selection

Introduction

Single-cell ribonucleic acid (RNA) sequencing (scRNA-seq) is increasingly prevalent because it provides more comprehensive data than bulk RNA-seq. Advances in scRNA-seq technology, driven by continuous development and optimization from various companies, have improved accessibility, increased data throughput, and reduced costs, making the technique more widely available [1]. scRNA-seq data analysis facilitates the identification of cell types based on their gene expression profiles across different biological samples [2]. Accurate annotation of cells into types, subtypes, and states is essential for robust single-cell analysis, especially in the context of disease characterization, therapeutic evaluation, biomarker discovery, and subsequent analyses [3].

Traditionally, cell type annotation has been based on marker genes, correlation-based methods, and supervised and unsupervised learning techniques [4]. Recently, deep neural network (DNN) based models have been introduced to enhance the accuracy and scalability of cell type annotation [5–7]. The selection of marker genes relies on the prior knowledge of researchers, which can introduce biases and inaccuracies. Furthermore, marker genes may not be readily available for all cell types of interest, especially for novel cell types that lack an established set of marker genes. Moreover, many cell types are defined by a combination of genes rather than a single marker gene [8]. Without a robust method to incorporate expression data from multiple marker genes, ensuring a consistent and precise annotation of cell types to each cluster becomes challenging, time-consuming, laborious, and less scalable for large datasets [9–12].

Clustering methods also remain widely used, where cell types are assigned to clusters formed through unsupervised learning, often guided by manually curated marker genes derived from published literature [12]. Recent advancements have introduced innovative techniques for cell type and rare cell type annotation, including clustering integrated with graph learning [13], consistency learning methods [14], network-based [15], and a deep generative model designed for improved cell identification [16]. These developments continue to expand the capabilities of scRNA-seq data analysis, enabling more precise insights into cellular diversity and function. Tools such as cellMeSH [17], CellAssign [18],

Received: December 16, 2024. **Revised:** April 14, 2025. **Accepted:** May 2, 2025

scCATCH [19], SCINA [20], SCSA [21], scSorter [22], and scType [23] perform clustering first and then assign a cell type identity to each cluster.

Correlation-based methods such as DUBStepR [24], Scmap [25], SingleR [26], scMatch [27], CHETAH [28], and Seurat [29] measure the correlation of gene expression profiles between query samples and the reference dataset [12], and mainly affected by the batch effect of platforms and experiments [30]. Numerous batch-effect correction methods are available, but distinguishing genuine biological diversity from technical disparities remains challenging, thereby preserving crucial biological variations poses a challenge [31]. The annotation of cell types using supervised and semi-supervised classification methods, including DNN-based approaches, aligns with the conventional framework of machine learning. These methods are designed to recognize intricate patterns in gene expression profiles and efficiently transfer labels from well-annotated datasets to unlabeled data, thereby improving the accuracy and scalability of cell type identification [32]. Several prominent tools such as ACTINN [33], Celltypist [34], devCellPy [35], scBalance [36], scVI-tools [37], scDeepinsight [38], SingleCellNet also known as pySingleCellNet [39], and cellPLM [7] have emerged that leverage these methodologies, demonstrating enhanced performance and adaptability in cell type annotation applications. These machine learning and DNN-based tools have gained significant popularity due to their ability to handle data noise, manage biological variability, and operate independently of manually selected marker genes. Such flexibility enables more reliable annotation in complex and heterogeneous single-cell datasets. However, these advanced methods also pose computational challenges; they require substantial resources to efficiently process large-scale datasets, especially those containing over 1 million single-cell samples [40]. This computational demand underscores the need for optimized frameworks that balance performance with resource efficiency to ensure scalable single-cell analysis.

To overcome the computational challenges associated with large-scale scRNA-seq data analysis, we developed scaLR, a low-resource deep neural network-based platform designed for efficient and scalable cell type annotation. We rigorously tested scaLR on a resource-constrained virtual machine (VM) equipped with 128GB RAM, 12GB GPU, and 8 CPUs, as well as on a local system with 32GB RAM and 8 CPUs. scaLR effectively manages datasets containing millions of cells by strategically partitioning the data into manageable batches, ensuring smooth processing even on systems with limited computational resources. To further enhance performance, scaLR integrates multiple DNN architectures, allowing flexible model training tailored to dataset complexity and size. In addition to its robust cell type annotation capabilities, scaLR extends its functionality to downstream gene analysis. This feature enables the identification of key genes that influence cell identity, traits, and disease associations, providing valuable insights for biological research and clinical applications. By combining efficient resource utilization with comprehensive analytical capabilities, scaLR presents a versatile solution for large-scale single-cell data analysis.

Results

ScaLR

Various machine learning (ML) and DNN-based automated frameworks have been recently developed for cell type classification, which requires a significant amount of GPU memory and RAM to process raw data and generate a model that classifies cells. We developed scaLR, a Python-based platform that provides cell annotations and associated important biomarkers to address these constraints. scaLR comprises data processing, feature extraction, training, evaluation, and downstream analysis (Fig. 1). The data processing module is designed to handle large sample datasets and segment them into training, testing, and validation sets. After that, it undergoes preprocessing of the dataset by performing sample-wise and/or standard scale normalization. The feature extraction modules play a pivotal role in scaLR's uniqueness, as they identify class-specific top-K features from each subset through single or multilayer DNN models. In the feature selection model training phase, these feature subsets are trained using a separate DNN model. After training all feature subsets, feature importance is assessed across the entire feature set using either a Linear or SHAP(SHapley Additive exPlanations) based scoring method to identify the most informative features. A new model is then trained on the top K selected features and evaluated using metrics such as precision, recall, F1-score, and accuracy, along with performance reports including receiver-operating characteristic and area under the curve (ROC-AUC) [41] and gene recall curves based on cell type annotations. Additionally, key biomarkers associated with specific cell classes and differentially expressed genes (DEGs) distinguishing disease and normal cell types are identified. scaLR is implemented in Python, with detailed information available on the official website and the code accessible via the GitHub repository.

Performance evaluation of scaLR using all genes as features

scaLR introduces a distinctive feature extraction capability, providing users with the flexibility to either select the top-K features from the input dataset or utilize the entire feature set for classification. This adaptable feature selection strategy enhances scaLR's versatility, enabling efficient model training across datasets of varying complexity. To evaluate scaLR's performance, its prediction accuracy, memory usage, and execution time for cell type and cell state classification were benchmarked against existing Python frameworks that utilize ML and DNN models. These evaluations were conducted in the PBMCs-bacterial sepsis (PBMCs-BS) dataset (Table 1). The results demonstrate that scaLR achieves comparable accuracy to established frameworks such as scVI-tools (including scVI and scANVI) [37], SingleCellNet [39], CellTypist (Logistic, SGD, and SGD + FS-TRP) [34], ACTINN [33], and devCellPy [35] when using a single-layer neural network (Table 2). For cell state classification, scaLR maintains accuracy levels comparable to CellTypist-Logistic, while outperforming in comparison to scBalance [36] and SingleCellNet. However, in certain conditions, slightly higher accuracy was observed in frameworks such as ACTINN, CellTypist-SGD, CellTypist-SGD + FS-TRP, scVI-tools, and devCellPy. Despite this, scaLR consistently excels in terms of minimal memory consumption and faster execution time (with one exception), reinforcing its efficiency as a low-resource solution. Notably, a key limitation encountered with scVI-tools was its inability to process all 22,858 features simultaneously due to extensive memory demands requiring over 128GB of RAM and a high-performance GPU to construct the latent spaces. To mitigate this constraint, scVI-tool analyses were performed using the default latent space size of 30, with incremental increases up to 3,500 latent dimensions, while carefully managing GPU memory limitations. Similarly, cellPLM restricts its feature matrix to the top 3,000 features to ensure model feasibility and maintain stable execution. Overall, the comparison underscores that when utiliz-

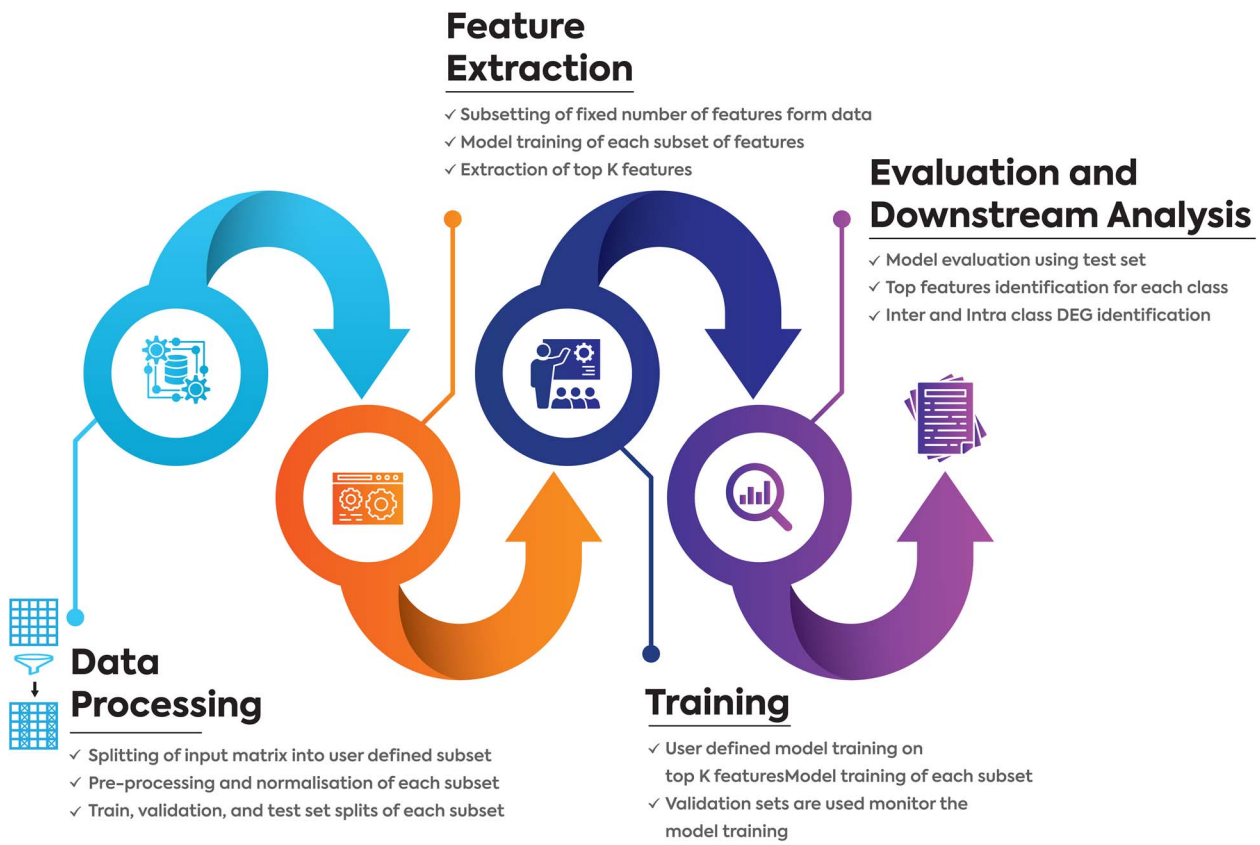

Figure 1. Schematic representation of different steps of scaLR.

Table 1. List of scRNA-seq datasets used to evaluate scaLR performance and compare it with other frameworks.

Datasets	Number of cells	Number of genes	Number of cell-type	Reference
PBMCs-BS[a]	1,26,351	22,858	6	[42]
PBMCs-Normal[b]	6,85,024	36,771	25	[43]
PBMCs-C19-F[c]	8,36,148	37,292	17	[44]
PBMCs-C19-H[d]	9,00,360	45,821	35	[45–47]
PBMCs-AIDA[e]	10,58,909	36,161	33	[48]
PBMCs-SLE[f]	12,63,676	30,867	11	[49]
HBSFG[g]	2,28,479	38,353	8	[50, 51]
HBPFC[h]	3,60,190	50,197	9	[52–54]
HBCA[i]	21,22,065	15,166	26	[55]
HED[j]	40,62,980	46,397	72	[56]
MED[k]	114,41,407	45,854	134	[57]

[a]PBMCs-BS: PBMCs-Bacterial Sepsis. [b]PBMCs-Normal: PBMCs from a healthy human. [c]PBMCs-C19-F: PBMCs-COVID-19-Flu (PBMCs of COVID-19, sepsis and flu patient). [d]PBMCs-C19-H: PBMCs-COVID-19-Harmonized(PBMCs from various COVID-19 studies were pooled together and harmonized). [e]PBMCs-AIDA: PBMCs-Asian immune diversity atlas. [f]PBMCs-SLE: PBMCs-Systemic lupus erythematosus. [g]HBSFG: Human brain superior frontal gyrus. [h]HBPFC: Human brain prefrontal cortex. [i]HBCA: Human breast cell atlas. [j]HED: Human embryonic development. [k]MED: Mouse embryonic development.

ing the full feature set for cell type annotation, scaLR achieves superior accuracy, memory efficiency, and execution speed. Its ability to efficiently operate in low-resource environments while delivering robust annotation results positions scaLR as a powerful and scalable platform for comprehensive single-cell RNA-seq data analysis.

Comparison of scaLR extracted top 3,500 features as input for different pipelines

In this comparison, we extracted the top 3500 features of the PBMCs-BS dataset using the scaLR feature extraction algorithm that uses an stochastic gradient descent (SGD) optimizer. We ran different pipelines using these features to evaluate prediction accuracy, memory consumption (maximum memory usage during the pipeline run), and wall clock time (end-to-end pipeline execution time) in a VM instance with a 12GB GPU, 128GB RAM, and 8 CPUs. This experiment's results indicate that wall clock time and maximum memory usage of scaLR for cell type and state are better than all other pipelines in Table 3. However, prediction accuracy for cell type and state is better than SingleCellNet [39], CellTypist(all variant runs), scBalance [36], ACTINN [33], and cellPLM [7], and similar to scVI-tools (scVI and scANVI) [37], and devCellPy [35]. However, while running the devcellpy, we have turned off the fine-tuning of parameters, probably leading to running the model faster using standard parameters defined inside the tool. Table 3 consists of pipeline accuracy, time, maximum memory, and parameters used to execute each tool.

Table 2. Accuracy, wall-clock time, and memory usage of different pipelines executed using all features and samples from the PBMCs-BS dataset.

	Cell type			Cell state		
Classifier	**Accuracy**	**Time**	**Memory**	**Accuracy**	**Time**	**Memory**
scaLR[a]	**0.942**	27:50	**9.778**	0.840	40:50	**8.679**
Svi-tools(scVI & scANVI)[b]	0.939	53:52	23.914	**0.870**	54:24	23.645
SingleCellNet[c]	0.929	27:42	57.821	0.811	1:12:02	98.973
scBalance	0.750	24:47	38.268	0.640	**22:56**	38.272
CellTypist-Logistic[d]	0.940	09:28:31	34.271	0.820	38:15:01	34.27
CellTypist-SGD[e]	0.930	17:54.32	34.273	0.850	37:23	34.269
CellTypist-SGD + FS-TRP[f]	0.930	**16:51**	34.272	**0.870**	36:05	34.271
devCellPy	0.940	30:01	37.7	0.780	41:28	37.702
ACTINN	0.939	41:34	55.738	0.843	40:34	55.741

[Time]Wall-clock time, reported in hours, minutes, and seconds (HH:MM: SS or MM: SS), refers to the end-to-end runtime of the program. [Memory]Maximum memory(gigabytes) utilized at any point of time. [a]Used single layer, epochs = 100. [b]Used number of latent = 30, number of layers = 2, scVI epochs = 100, and scANVI epochs = 25. [c]Random forest using nTrees = 500. [d]CellTypist using logistic classifier. [e]CellTypist using using SGD. [f]CellTypist using feature selection method followed by an SGD classifier.

Table 3. Accuracy, wall clock time, and memory use in different pipelines executed on top 3,500 features with samples extracted from the PBMCs-BS dataset using the scaLR features extraction module.

	Cell type (SGD)			Cell state (SGD)		
Classifier	**Accuracy**	**Time**	**Memory**	**Accuracy**	**Time**	**Memory**
scaLR[a]	**0.941**	**2:33**	**1.70**	0.840	**6:23**	**1.70**
scvi-tools (scVI & scANVI)[b]	0.940	21:25	3.76	**0.860**	22:47	3.77
SingleCellNet[c]	0.920	18:30	43.84	0.801	51:42	80.32
scBalance	0.790	14:12	6.35	0.690	14:16	6.35
CellTypist-Logistic[d]	0.930	51:45	4.38	0.830	2:30:31	4.38
CellTypist-SGD[e]	0.920	6:19	4.38	**0.860**	8:58	4.38
CellTypist-SGD + FS-TRP[f]	0.910	8:22	4.38	**0.860**	13:08	4.38
devCellPy	0.940	6:58	6.62	0.840	8:08	6.64
ACTINN	0.930	8:40	8.77	0.830	8:05	8.77
CellPLM[g]	0.890	9:05:19	3.86	0.840	7:02:41	3.74

[Note]SGD in the cell type and state columns indicates the input features. [Time]Wall-clock time, reported in hours, minutes, and seconds (HH:MM: SS or MM: SS), refers to the end-to-end runtime of the program. [Memory]Maximum memory(gigabytes) utilized at any point of time. [a]Used single layer, epochs = 100. [b]Used number of latent = 30, number of layers = 2, scVI epochs = 100 and scANVI epochs = 25. [c]Random forest using nTrees = 500. [d]CellTypist using logistic classifier. [e]CellTypist using using SGD. [f]CellTypist using feature selection method followed by an SGD classifier. [g]CellPLM-The time is not included as we have run it on CPU, and the model expects the GPU architecture for faster runs.

Evaluating the performance and robustness of scaLR on large sample datasets

To evaluate the performance and robustness of scaLR in large datasets, we selected a range of datasets with various characteristics (~22–46 k) and sample sizes from 0.12 to 11.4 million cells (Table 1). scaLR has no constraints on the number of data samples that can be used for training, as we perform feature subsetting for feature extraction (if opted) and batch-wise data training. This allows users to train a model based on their available resources. We aim to assess the capability of other tools like CellTypist, devCellPy, SingleCellNet, scBalance, ACTINN, and CellPLM. We ran these pipelines on datasets with varying sample sizes (Table 1) using the same VM used in scaLR. Initially, we tested these tools on the PBMCs-SLE dataset, which has over 1.2 million samples, using ~0.8 million samples for model training. Running each tool individually on this dataset caused memory crashes and failures to produce results. We then tested these frameworks on the PBMCs-normal dataset, which has over 0.6 million cells, using ~0.4 million cells for model training. We observed that only CellPLM can run in this dataset, while other tools require more than 128GB of memory and are unable to function with limited or moderate resources. This issue arises because these tools attempt to load the entire dataset into memory simultaneously, which is not feasible for large datasets with limited resources.

On the other hand, scaLR enables the analysis of cross-cohort datasets exceeding 11.5 million cells, effectively managing batch effects generated by various single-cell sequencing platforms. These datasets include PBMCs, human breast, and human and mouse embryonic development stages, each divided into the train, test, and validation sets as shown in Table 1. This experiment is crucial for ensuring the generalizability and reliability of scaLR in low-resource environments. Performance evaluation involves measuring the platform's accuracy, precision, recall, F1-score (Supplementary Table S1–S7), memory usage, and experiment runtime across diverse datasets, encompassing various biological conditions and technical variations (Table 4).

Robustness evaluation assesses the scaLR's ability to maintain consistent performance despite changes in input data characteristics, such as varying noise levels, batch effects, and data sparsity across all selected datasets. By systematically testing this platform on a comprehensive collection of datasets, including benchmark datasets, one can identify strengths and limitations, ensuring that the platform is not overfitted to specific conditions. The associated results of these datasets are listed in supplementary tables and figures. This evaluation process is essential for confirming that the platform can accurately annotate cell types and subtypes, perform downstream analysis, and produce reliable results across different experimental settings, ultimately validating its applicability in various scRNA-seq studies.

Table 4. Performance evaluation and robustness check of scaLR in large sample datasets.

	Training[a]			Evaluation (1CPU + 1GPU)[b]			Evaluation (Multiple CPUs)[c]		
Datasets	Samples size	Epoch	Layers	Accuracy	Memory	Time	Accuracy	Memory	Time
PBMCs-C19-F	5,67,761	23	5,000,200,17	0.932	8.053	02:44:57	0.935	5.87	01:19:09
PBMCs-Normal	4,56,682	40	5,000,200,25	0.905	8.335	03:46:29	0.906	5.413	01:45:30
PBMCs-C19-H	6,35,132	15	5,000,200,35	0.862	12.206	05:02:59	0.88	7.727	01:36:34
PBMCs-AIDA	7,09,215	28	5,000,200,33	0.908	8.424	04:51:17	0.89	7.401	01:32:40
PBMCs-SLE	8,44,488	30	5,000,200,11	0.979	9.448	04:29:14	0.979	8.385	02:22:03
HBCA	14,12,546	17	5,000,200,26	0.894	9.717	05:49:04	0.894	9.727	02:17:55
HED	32,13,236	26	5,000,200,72	0.944	11.39	16:48:12	0.919	14.33	06:50:30
MED	86,13,510	10	5,000,1,000,300,134	0.927	29.072	63:27:59	0.932	28.175	21:08:30

[Note1]Used GPU configuration—NVIDIA TITAN XP GPU—12GB memory. [Samples size]Number of samples in training. [Epoch]Number of epochs used to converge the model out of 200 input epochs. [Layers]Used DNN layers for the final model run. [Time]Wall clock time in hours, minutes, and seconds (HH:MM: SS or MM: SS) is referred to as end-to-end pipeline run time, excluding downstream analysis time. [Memory]Maximum memory(gigabytes) utilized at any point of time. [a]The training section covers factors like input sample size, epochs, and DNN layers, which ensure the effective execution of scaLR across various datasets. The evaluation section provides performance insights by presenting metrics such as accuracy, memory utilization, and total pipeline execution time. [b]The performance of scaLR, using 1 CPU and 1 GPU, includes all steps run. [c]The performance of scaLR with 3 and 5 CPUs used in data ingestion and feature selection, respectively.

Table 5. Performance and analysis cost comparison of scaLR with scVI-tools and cell-typist.

Dataset	Pipelines	Accuracy	Time	Memory	VM-Configuration	Analysis cost
PBMCs-SLE	**scaLR**	**0.979**	5:06:07	**9.200**	n1-standard-4	**$5.50**
PBMCs-SLE	scvi-tools (scVI & scANVI)	0.970	6:54:59	99.76	n1-highmem-16	$11.87
PBMCs-SLE	CellTypist-SGD	0.970	**3:35:14**	389.597	n1-highmem-64	$17.94
PBMCs-SLE	CellTypist-SGD + FS-TRP	0.965	3:47:04	389.599	n1-highmem-64	$17.94
PBMCs-HED	**scaLR**	0.930	18:46:03	**12.763**	n1-standard-4	**$13.06**
PBMCs-HED	scvi-tools (scVI & scANVI)	**0.94**	**15:27:38**	592.161	n1-highmem-64	$116.06

[Note1]The cost calculation is done based on max memory and time taken by the tool to perform the PBMCs-SLE analysis and requires VM configuration. [Note2]In this comparison, the time and memory of the sacLR run were performed using one CPU only. [Note3]All the above experiments were carried out in Google Cloud VMs with maximum memory and respective calculated estimated cost of each dataset run (details about cost calculation are listed in Supplementary Table S15)

Performance and analysis cost comparison

For this comparison, we used the PBMCs-SLE and PBMCs-HED datasets, which contain over 1.25 and 2 million cells, respectively. We performed cell type annotations using scaLR and compared the results and analysis cost with two gold standard tools: scVI-tools and celltypist. Table 5 summarizes the performance and cost of analysis for these tools. Although the classification accuracy was similar across all three tools, scVI-tools and celltypist required significantly more memory than scaLR. Although scaLR had a slightly longer runtime than celltypist (both variants) in the case of PBMCs-SLE, its lower memory consumption reduced the overall cost. Notably, scVI-tools took the longest time for end-to-end analysis but also benefited from moderate memory usage, which helped reduce costs in comparison to CellTypist (Supplementary Table S15). Overall, scaLR was the most cost-efficient, as its low memory requirements and the capability of running without a GPU allowed it to run on a standard machine, whereas the other tools required specialized computing resources.

Features of scaLR

The scaLR is equipped with a range of advanced features designed to streamline and enhance the analysis of scRNA-seq data. It begins with the splitting of input datasets into the train, validation, and test subsets. Then, user-defined normalization can correct the differences in sequencing depth and technical noise, ensuring consistent and comparable gene expression levels across cells. Feature extraction is the soul of this platform, enabling loading large sample datasets in different subsets and extracting the top-K features using an SGD optimizer. The platform uses these top-K features to accurately classify cells into distinct types and subtypes using DNN models. It also provides cell type and disease-associated top 100 biomarkers. An integrated differential gene expression (DGE) analysis allows users to get DEGs between different conditions or cell types based on the top-K features of the model or considering all features. To evaluate the performance of the model classification, ROC & AUC curves are used to illustrate the classification accuracy, plotting true positive rates against false positive rates. In contrast, gene recall curves assess the platform's ability to recover known gene sets, providing insights into its sensitivity and robustness. Figure 2 shows the various downstream analysis plots generated by scaLR. Together, these features ensure a comprehensive, accurate, and reliable analysis of scRNA-seq data, making the platform an invaluable tool for researchers and industry personnel. However, we have also performed specific analysis using scaLR as described in further subsections.

Cell type-specific biomarker discovery

Comparative analysis of the top 100 cell type-specific biomarkers identified by scaLR and CellTypist-Logistic (the model with the highest accuracy) revealed significant overlaps and unique contributions from each tool in terms of literature or validated cell-specific biomarkers (Supplementary Table S8). Out of 223 reported T-cell biomarkers, scaLR identified 21, while CellTypist predicted 19. For megakaryocyte cells, out of 39 biomarkers, scaLR predicted 10, while CellTypist predicted only 5. Similarly, for monocytes, Dendritic cells (DCs), and Natural killer (NK) cells, scaLR predicted more biomarkers than CellTypist (Supplementary Fig. S1). Only for B cells, CellTypist captures more biomarkers (33) compared to scaLR (32) within the top 100 features. DCs showed the highest overlap between these tools across different cell types, while T and megakaryocyte cells had less overlap. We also plotted the gene recall for cell-specific biomarkers (Supplementary Table S8) as a reference concerning

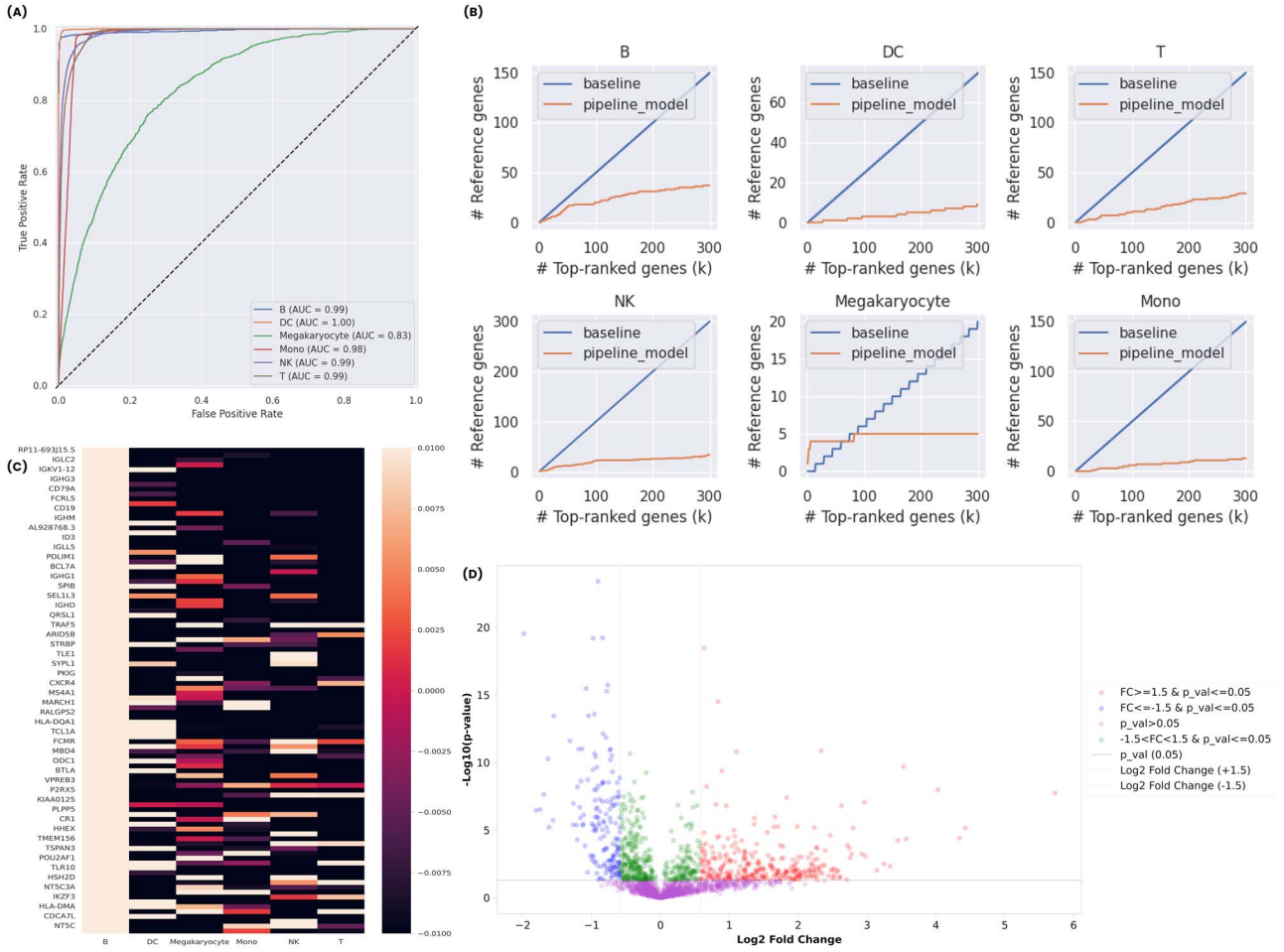

Figure 2. Different downstream analysis plots (an example) produced by scaLR (A) ROC & AUC curve showing the classification accuracy of predicted class, (B) gene recall curves of B lymphocyte (B) cells, dendritic cells (DC), thymocytes (T) cells, natural killer (NK) cells, megakaryocyte and monokaryocyte (mono) cells with cell-type specific biomarkers identified in top-ranked genes predicted by model with respect to literature/reference genes, (C) an example heatmap indicates the top-ranked genes of a cell-type with their association in other cell type and (D) DGE plot showing DEGs for particular disease and normal condition.

the identified biomarkers in the top 5,000 features of scaLR and CellTypist (Supplementary Fig. S2). Both comparisons indicate that scaLR identifies more biomarkers than CellTypist. scaLR also generates a heatmap of the top 20 biomarkers identified through SHAP analysis for each cell type with their expression in other cells (Supplementary Fig. S3).

Leveraging the power of SHAP, scaLR identifies the top 100 genes specific to each cell type, which are then compared to reference biomarkers for each cell type (Supplementary Table S8) using a gene recall curve for the PBMCs-SLE (Supplementary Fig. S4 and Supplementary Table S9) and PBMCs-C19-F (Supplementary Fig. S5 and Supplementary Table S10) datasets. These gene recall curves reveal that most biomarkers appear within the top 100 features for each cell type. In comparison to control patient samples, the scaLR DGE module pinpoints cell type-specific DEGs in PBMCs-SLE (Supplementary Table S11 and Supplementary Fig. S6) and PBMCs-C19-F (Supplementary Table S12 and Supplementary Fig. S7). This demonstrates scaLR's accuracy in detecting cell-specific biomarkers, while other pipelines struggled to process these datasets on low-resource instances.

Diseases specific biomarker discovery

scaLR identifies the top 100 biomarkers for the human brain superior frontal gyrus (HBSFG) and human brain prefrontal cortex (HBPFC) datasets (Table 1), specific to Alzheimer's disease and control groups, as listed in Supplementary Tables S13 and S14, respectively. Functional analysis of the top 10 genes reveals that scaLR effectively detects biomarkers related to Alzheimer's disease and neurological disorders in both datasets. For the HBSFG dataset, seven of the top 10 genes are linked to neurological disorders; in contrast, in the HBPFC region dataset, only two of the top 10 genes are associated with such conditions. The functions of these genes are outlined in Supplementary Tables S13 and S14. Although we annotated the top 10 genes, additional high-ranking genes may also represent potential biomarkers, pending further functional annotation.

Comparison of top-K differential gene expression features between full and test set samples

DGE analysis in scaLR can be conducted in two ways, mainly utilizing all samples and test sets with the top-K features identified using feature selection algorithms using pseudo-bulk and linear mixed effects models (LMEM) approaches. These approaches rely on the top-K features generated by feature selection and model training steps. Further, these methods operate on an AnnData object containing the gene expression matrix for the selected features along with their corresponding cell and gene metadata.

To ensure significant identification of DEGs, the correct sample sizes must be considered. To verify this, we have compared DGE analysis results on myocardial data (191 k cells and 28 k genes)

[58] using the top 5,000 features of test samples and all samples. The used dataset encompasses 10 cell types across two clinical conditions: "Myocardial infarction" and "Normal". For each cell type, we identified up-and down-regulated genes in "Myocardial infarction" relative to "Normal" samples. For the "Pseudo Bulk" analysis, we applied a cutoff of ±1.5-fold change in gene expression with a *P*-value threshold of 0.05. In the LMEM analysis, coefficients greater or less than zero, with a *P*-value threshold of 0.05, were considered significant. A consistent trend of increased DEGs count was observed when using all samples compared to test samples alone. For example, in "Smooth muscle myoblasts," 41 upregulated genes were identified with the Pseudo bulk approach in test samples, increasing to 254 when using all samples. In the LMEM analysis, the number of upregulated genes increased from 92 in the test samples to 130 in the full dataset. A similar trend was observed for down-regulated genes: in Pseudo bulk analysis, 22 genes were downregulated in the test samples, increasing to 164 with all samples. In the LMEM analysis, 256 downregulated genes were found in test samples, rising to 332 when all samples were included.

Discussion

Recent advancements in single-cell sequencing technologies have revolutionized transcriptomic research, enabling the generation of large-scale datasets containing millions of individual cells. These technological improvements have facilitated deeper insights into cellular heterogeneity, developmental processes, and disease mechanisms [1, 59]. A crucial step in the downstream analysis of scRNA-seq data is accurate cell-type annotation, which is vital for identifying distinct cell populations, understanding cellular states, and discovering biomarkers with potential diagnostic or therapeutic value [2, 3]. As the number of publicly available cell atlases continues to expand, the demand for automated annotation tools has surged. These tools leverage ML and DNN models to enhance cell type identification's accuracy, reproducibility, and scalability [32]. Despite these advancements, existing frameworks still encounter significant challenges when processing large-scale scRNA-seq datasets, particularly those exceeding 0.5 million cells. The complexity arises from the need to manage extensive feature matrices, which require substantial computational resources. Most conventional python-based tools struggle to efficiently load, process, and train models using such large datasets in resource-constrained environments. For example, systems with 12GB GPU memory, 128GB RAM, and 8 CPUs often face performance bottlenecks when attempting to utilize all available features for model training. These limitations highlight the urgent need for optimized frameworks that ensure scalability, computational efficiency, and compatibility with low-resource environments to support comprehensive single-cell data analysis.

In this work, we introduce scaLR, an innovative deep neural network-based platform designed to address the computational challenges associated with largescale scRNA-seq data analysis. scaLR incorporates a suite of optimized modules, including sample batching, normalization and batch processing, feature extraction, feature ranking using multiple scoring metrics, and a specialized gene recall module. These integrated components enable scaLR to manage large datasets while efficiently minimizing computational resource demands. By leveraging sample batching and streamlined data handling techniques, scaLR effectively partitions extensive datasets into manageable subsets, ensuring stable performance even in low-resource environments. The platform's feature extraction and ranking modules further enhance model efficiency by identifying and prioritizing key genes that contribute significantly to cell type prediction, improving both accuracy and interpretability. The inclusion of a dedicated gene recall module facilitates downstream analyses, supporting gene-specific insights into cell identity, trait associations, and disease mechanisms. Through comprehensive benchmarking across diverse scRNA-seq datasets varying in size, generation protocols, and data imbalance, we have demonstrated that scaLR consistently outperforms existing frameworks in key metrics. In particular, scaLR achieves superior annotation accuracy, faster execution speeds, and reduced computational costs, making it an optimal solution for scalable and resource-efficient single-cell analysis.

Notably, we have compared scaLR performance of most of the widely used Python-based cell-type annotation tools such as scVI-tools (scVI and scANVI) [37], SingleCellNet [39], CellTypist [34], scBalance [36], ACTINN [33], devCellPy and [35]. scaLR has shown excellence in cell type annotation and features identification ability using all features and the top 3,500 features (Tables 2 and 3). In addition, several key metrics were considered to assess the performance and robustness of scaLR in large sample datasets (Table 4), including precision, memory usage, and end-to-end execution time. These metrics were evaluated in varying dataset sizes to gauge the scalability and efficiency of scaLR in handling high-dimensional data. Robustness checks were performed by running the pipeline on multiple large datasets to ensure consistent results under different input conditions. Additionally, sensitivity analyses were performed by altering key input parameters, such as epochs and DNN layers, to evaluate the stability of the model's predictions. This evaluation demonstrates that scaLR maintains high accuracy, low resource utilization, and cost of the analysis, even with increased batch size, confirming its robustness for large-scale data analysis. The latest version of scaLR is enabled with multi-CPU execution, effectively reducing the time and cost of the analysis (Table 4). This enhancement significantly improves scalability, making it more efficient for large-scale scRNA-seq datasets.

scaLR can also identify disease-specific biomarkers using single-cell control and disease-patient datasets (Supplementary Tables S10–S14). Our analysis of the human brain's two regions: superior frontal gyrus (SFG) and prefrontal cortex (PFC) datasets provides the top 100 ranked genes, and most of them are possibly associated with Alzheimer's disease and different neurological disorders (explained in the Results section). We have performed a functional annotation of the top 10 ranked genes, which are mainly in the HBSFG dataset. Ferritin heavy chain 1 is involved in iron storage and the regulation of oxidative stress. Its abnormal expression is linked to neurodegenerative diseases like Alzheimer's and Parkinson's due to iron accumulation [60]. Gene semaphorin-3C is involved in the emerging development of axons and dendrites of the cortex [61, 62]. Diazepam-binding inhibitor is involved in the altered expression linked to anxiety, depression, and epilepsy, possibly due to its role in neurosteroid production [63]. Overexpression stanniocalcin-1 alleviates oxidative stress-induced injury, reduces neuroinflammation, and improves cognitive function [64]. Angiopoietin-like protein 4 plays a role in the process of white matter damage and cognitive impairment (CI) in patients with cerebral small vessel disease [65]. Histidine triad nucleotide-binding protein 1 is involved in transcription regulation and apoptosis. Mutations in this gene are linked to neuropathy, schizophrenia, and CIs [66].

In the case of the HBPFC dataset, most of the top 10 ranked genes are long noncoding RNAs with unknown functions, while only two genes, i.e. FERM, RhoGEF, and pleckstrin domain-containing protein1 (FARP-1) and Ribosomal protein S6 (RSP6),

are found as potential biomarkers. FARP-1 is involved in Autism spectrum disorder and CI, particularly in its role in neural development [67]. RSP6 actively participates in protein synthesis and cell growth. Its phosphorylation is commonly used as a marker for neuronal activity [68].

In summary, we believe that scaLR represents a significant advancement in cell type annotation and biomarker identification, particularly for low-resource environments. Its ability to efficiently process datasets containing millions of cells while delivering accurate cell type annotations makes it a valuable tool for large-scale scRNA-seq analysis. Beyond its core annotation capabilities, scaLR offers additional features that enhance downstream analysis. These include DGE analysis, gene recall curves, and heat maps that highlight commonly associated cell type-specific genes, providing deeper insights into cellular heterogeneity and gene regulation. While scaLR demonstrates strong performance in various scenarios, it has a few limitations. Notably, its memory usage is inversely proportional to execution time, meaning that reducing memory consumption may increase processing time. Additionally, scaLR's performance may vary based on the input data format, requiring users to ensure proper data preprocessing for optimal results. Future improvements will aim to address these limitations by enhancing memory management strategies and expanding compatibility with diverse data formats. Moreover, upcoming updates may introduce additional features to further strengthen scaLR's capabilities, ensuring it remains a cutting-edge platform for scalable and efficient scRNA-seq data analysis.

Methods

Single-cell RNA-Seq data selection and download

Peripheral Blood Mononuclear Cells (PBMCs) constitute a crucial component of the immune system and are pivotal in combating a spectrum of infections stemming from harmful pathogens. They serve as a fundamental tool for investigating the immune response, infectious ailments, cancer, and the development of vaccines [2, 69]. To design and compare the scaLR platform performance, the PBMCs-BS dataset was used. Furthermore, to showcase the efficacy of scaLR and compare it with other pipelines, we downloaded and curated various PBMC studies, COVID-19, human breast, and embryonic development scRNA-seq datasets. Along with this, human brain scRNA-seq data of different cortices are downloaded to identify the Alzheimer's disease-specific biomarkers (Table 1). These individual studies were primarily sourced from Cellxgene [70] and the single-cell portal of Broad Institute [31].

The scaLR platform

The scaLR platform consists of four primary modules: data processing, feature extraction, model training, and evaluation & downstream analysis (Fig. 1).

Data processing

Large samples of transcripts per million (TPM) matrix accompanied by metadata in the h5ad file (AnnData) are divided into training, validation, and test sets. Each of these sets was stored in sample-wise subsets. Subset-wise preprocessing (normalization) is adopted to optimize resource usage, and the data are read in backed mode, i.e. the entire data are not loaded in memory at once, which ensure low memory usage. Only the concerned data subset is loaded in memory to carry out preprocessing like normalization, and then the updated data subset is written to disk for further usage. Subsequently, batch correction (if the user opts) is handled by the scaLR dataloader module on each subset to remove the noise from gene expression and eliminate batch effects [71]. This approach effectively manages RAM load as per system specifications without impacting computational time. Notably, the subset sample size is a critical factor in memory consumption across the platform. This enables the reduction of the subset size significantly and reduces memory usage without introducing excessive computational overhead.

Feature extraction

The normalized input dataset comprises thousands of sparse features, emphasizing the need for faster, more efficient, and superior model training. Identifying important features from all the features is an important section of the platform. We have developed the unique Feature Extraction Algorithm 1 to address this. This algorithm partitions the input dataset into feature subsets, and a single-layered neural network model is fitted for a given training dataset. Subsequently, the validation set is employed to assess the model's effectiveness. The weights for each subset of features are then stored. This process iterates all feature subsets, ensuring no repetition of data. Each subset is used to fit exactly one model; each model is trained on a distinct subset of features. The weights matrix generated from each model is utilized to evaluate the contribution of features toward predicting specific classes. It is important to note that each model was trained in the same condition, using the same optimizer, and all their weights were initialized to zero to ensure symmetry and fairness. All subset weight matrices are combined to form a weight matrix for all features across all classes. Then, the mean of absolute weights of each feature across all classes is performed, representing the score of a feature contributing towards a prediction. The features are then ranked according to their scores, which are calculated by the scorer module. Then, top-K features are chosen to train the final model, further explained in Fig. 3.

Algorithm 1 Feature Extraction Algorithm used in scaLR.

```
# Required inputs from the user.
Dataset: D
Number of features to be extracted: K
Feature subset size: S
Parameters of model: params

# Number of subsets based on user inputs.
n_subsets = length(D.Features)/S

for feature_subsets_i in range 0 to n_subsets:
    # Define the model.
    model = LinearModel(params)

    # Train the model on the feature subsets.
    model.fit(feature_subsets_i)

    # The store features importance class-wise.
    feature_class_matrix = model.feature_importance()

# Concatenate feature_class_matrix for all the n_subsets.
feature_imp_all_classes = Concat(feature_class_matrix)

# Aggregate feature imp for every feature across all classes.
final_feature_imp = Aggregate(feature_imp_all_classes)

# Select the top K features based on the final weights of
# features using different scorers.
top_K_features = Select_top_k_features(final_feature_imp)

returns top_K_features
```

Training

During DNN model training, the top-K feature dataset is utilized to train the model. A validation dataset is employed to monitor

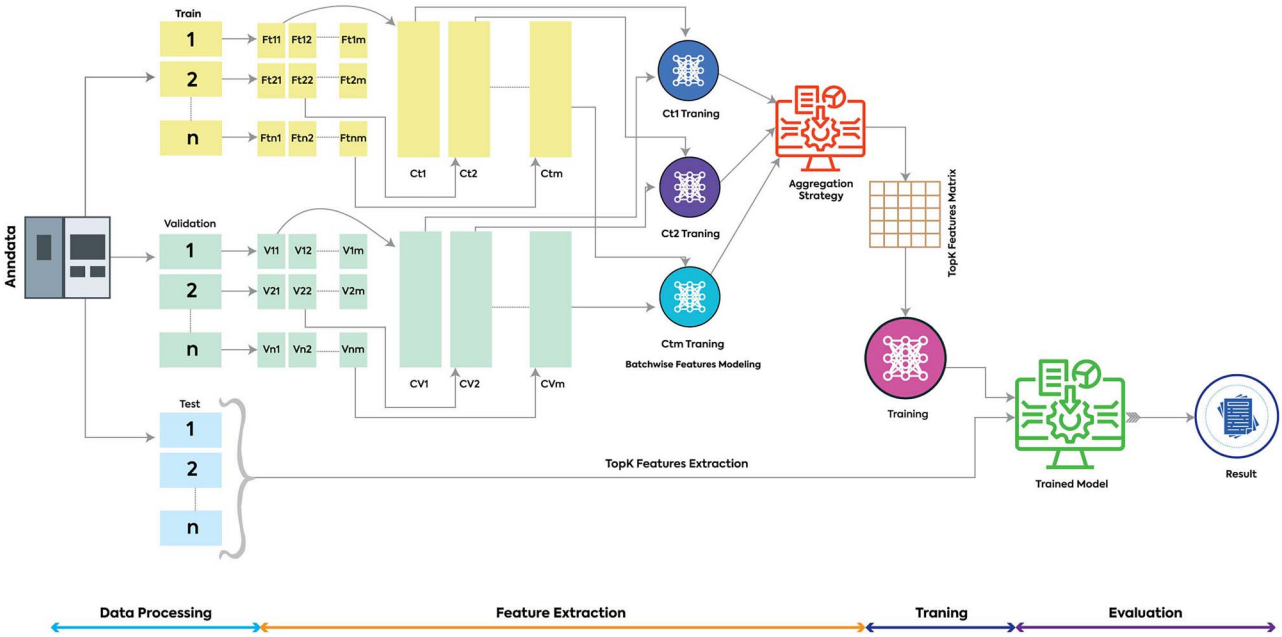

Figure 3. Overview of scaLR, a multilayered machine-learning platform for cell type classification and disease-specific features identification. It also performs a downstream analysis of genes for relevant biomarkers. User-defined top-K features can be extracted using a single or multilayer neural network for all feature subsets by performing batch-wise data processing and feature extraction. These top-K features are used in the final DNN model training, and these models are evaluated using a test set.

performance and identify the optimal model from the training model checkpoints. To adhere to the principle of minimal resource consumption, the training data is directly read from disk, and training is conducted in batches, thereby ensuring efficient processing of considerably large datasets containing tens of thousands of features. This approach allows scaLR to be executed on any machine.

Evaluation and downstream analysis

The trained model is then evaluated on the test set using metrics such as precision, recall, f1-score, and accuracy scores. A detailed classification report will show the model's performance for each class. In the case of the linear model, the weights are used to identify the rank of the top features that contributed to predicting a particular class. While for multiple layered models, the SHAP algorithm [72] is used to identify class-specific top features/biomarkers. The scaLR downstream analysis module is embedded with functions of gene recall curves, ROC & AUC, heatmaps, and DGE analysis. DGE analysis can be performed using Pseudo bulk [73] and LMEM [74] approaches, and the module will generate their respective volcano plots and associated gene lists.

scaLR performance comparison with different pipelines

To compare the performance of scaLR with other pipelines, we used all features and the top 3500 features extracted by the scaLR feature extraction module from the PBMCs-BS dataset (Table 1). This enabled us to evaluate the results obtained across different ML and DNN pipelines, which primarily include:

Automated cell type identification using neural networks (ACTINN) employs NN with three hidden layers and is trained on datasets with predetermined cell types. It uses the trained parameters to predict cell types in other datasets [33].

CellTypist is an automated cell type annotation tool utilizing logistic regression classifiers optimized by the stochastic gradient descent (SGD) algorithm. We have experimented with three variants of this tool, i.e. (i) using a logistic classifier, (ii) using an SGD classifier, and (iii) using a feature selection method followed by an SGD classifier with all default parameters mentioned in the classifier's scikit-learn packages [34].

CellPLM is the first pretrained large language transformer capable of encoding intercellular relationships, integrating spatially resolved transcriptomic data, and utilizing a justified prior distribution. For model training on the PBMCs-BS dataset, we used 200 epochs for all features and 100 epochs for the top 3,000 selected features [7]. devCellPy offers various parameters that can be tuned based on the input dataset.

We used a variant that trains all layers without cross-validation and metrics, skipping the timepoint prediction. Additionally, the initial parameter fine-tuning was turned off, allowing the tool to run with the standard default parameters set in the tool itself [35]. SingleCellNet or PySingleCellNet classifies cell types and states within heterogeneous cell populations by employing top-pair transformation followed by training a random forest classifier. It demonstrates the ability to classify across platforms and species with comparable sensitivity and specificity scores [39].

scVI-tools are designed for probabilistic modeling on scRNA-seq data, and they utilize the inference procedure for scVI and scANVI models relying on NN and stochastic optimization and employ variational autoencoders to infer latent representations of single-cell data in low-dimensional space [37].

scBalance employs a standard neural network architecture with 3–4 densely connected layers, incorporating batch normalization and dropout for better generalization, and uses the exponential linear unit (ELU) activation function. Before training, they opted to scale the data and applied a weighted sampling technique to emphasize rare cell types [36].

We have designed custom scripts to run all the above pipelines that are present in the scaLR comparison GitHub repository. In addition to comparing the accuracy of scaLR cell annotation with

these tools, we also compared the top 100 genes identified for each cell type predicted by scaLR and Cell Typist (all variant runs) with cell-specific biomarkers identified by various studies based on scRNA-seq and downloaded from CellMarker2.0 [75]. We cannot extract the features, or there is no explanation about how to get the top features for each cell type from the generated models, scBalance, scvi, devcellpy, and cellPLM.

To assess the performance and robustness of the scaLR platform, we have analyzed large sample datasets of different tissues of humans and mice. We also compared the DGE analysis across various cell types in PBMCs-SLE and PBMCs-C19-F datasets using the scaLR-enabled DEG module, which uses the top 5,000 features provided by the model as well as whole input data. scaLR is also capable of identifying disease-specific biomarkers using its novel feature extraction algorithm and associated model scorer. For this, we have used the HBSFG and prefrontal cortex (HBPFC) (Table 1) datasets of normal and Alzheimer patients. To validate the identified disease-specific top 10 genes, we have performed literature mining of each gene for both the cortex datasets.

Key Points

- Leveraged advanced AI and ML algorithms for precise and reliable cell type classification using NN models.
- Facilitates the identification of key biomarkers, driving forward personalized medicine and targeted therapies.
- Designed to run efficiently on large sample datasets, even on systems with limited resources.
- Supports a wide range of downstream analysis, enhancing the depth and breadth of your single-cell genomics research.
- User-friendly, intuitive design for seamless integration into your research workflow.

Supplementary data

Supplementary data are available at *Briefings in Bioinformatics* online.

Conflict of interest

The authors declare that the research was conducted without any commercial or financial relationships that could be construed as a potential conflict of interest.

Funding

This project was funded by Infocusp Innovation Research funds.

References

1. Brendel M, Su C, Bai Z. *et al.* Application of deep learning on single-cell rna sequencing data analysis: a review. *Genomics Proteomics Bioinformatics* 2022;**20**:814–35. https://doi.org/10.1016/j.gpb.2022.11.011.
2. Verhoeckx K, Cotter P, López-Expósito I. *et al. The Impact of Food Bioactives on Health: in Vitro and Ex Vivo Models*. Cham: Springer International Publishing, 2015.
3. Zhao X, Wu S, Fang N. *et al.* Evaluation of single-cell classifiers for single-cell rna sequencing data sets. *Brief Bioinform* 2020;**21**:1581–95. https://doi.org/10.1093/bib/bbz096.
4. Ma W, Su K, Wu H. Evaluation of some aspects in supervised cell type identification for single-cell rna-seq: classifier, feature selection, and reference construction. *Genome Biol* 2021;**22**:1–23. https://doi.org/10.1186/s13059-021-02480-2.
5. Yang F, Wang W, Wang F. *et al.* Scbert as a large-scale pretrained deep language model for cell type annotation of single-cell rna-seq data. *Nat Mach Intell* 2022;**4**:852–66. https://doi.org/10.1038/s42256-022-00534-z.
6. Cui H, Wang C, Maan H. *et al.* Scgpt: toward building a foundation model for single-cell multi-omics using generative ai. *Nat Methods* 2024;**21**:1470–80. https://doi.org/10.1038/s41592-024-02201-0.
7. Wen H, Tang W, Dai X. *et al.* Cellplm: pre- training of cell language model beyond single cells. bioRxiv. 2023.
8. Qiu Y, Wang J, Lei J. *et al.* Identification of cell-type-specific marker genes from co-expression patterns in tissue samples. *Bioinformatics* 2021;**37**:3228–34. https://doi.org/10.1093/bioinformatics/btab257.
9. Pliner HA, Shendure J, Trapnell C. Supervised classification enables rapid annotation of cell atlases. *Nat Methods* 2019;**16**:983–6. https://doi.org/10.1038/s41592-019-0535-3.
10. Abdelaal T, Michielsen L, Cats D. *et al.* A comparison of automatic cell identification methods for single-cell rna sequencing data. *Genome Biol* 2019;**20**:1–19. https://doi.org/10.1186/s13059-019-1795-z.
11. Huang Q, Liu Y, Du Y. *et al.* Evaluation of cell type annotation r packages on single-cell rna-seq data. *Genomics Proteomics Bioinformatics* 2021;**19**:267–81. https://doi.org/10.1016/j.gpb.2020.07.004.
12. Pasquini G, Arias JER, Sch¨afer, P., Busskamp, V. Automated methods for cell type annotation on scrna-seq data. Computational and structural. *Biotechnol J* 2021;**19**:961–9. https://doi.org/10.1016/j.csbj.2021.01.015.
13. Wu W, Zhang W, Hou W. *et al.* Multi-view clustering with graph learn- ing for scrna-seq data. *IEEE/ACM Trans Comput Biol Bioinform* 2023;**20**:3535–46. https://doi.org/10.1109/TCBB.2023.3298334.
14. Wang H, Liu Z, Ma X. Learning consistency and specificity of cells from single-cell multi-omic data. *IEEE J Biomed Health Inform* 2024;**28**:3134–45. https://doi.org/10.1109/JBHI.2024.3370868.
15. Wu W, Zhang W, Ma X. Network-based integrative analysis of single-cell transcriptomic and epigenomic data for cell types. *Brief Bioinform* 2022;**23**:546. https://doi.org/10.1093/bib/bbab546.
16. Wang H, Ma X. Learning discriminative and structural samples for rare cell types with deep generative model. *Brief Bioinform* 2022;**23**:317. https://doi.org/10.1093/bib/bbac317.
17. Mao S, Zhang Y, Seelig G. *et al.* Cellmesh: probabilistic cell-type identification using indexed literature. *Bioinformatics* 2022;**38**:1393–402. https://doi.org/10.1093/bioinformatics/btab834.
18. Zhang AW, O'Flanagan C, Chavez EA. *et al.* Probabilistic cell-type assignment of single-cell rna-seq for tumor microenvironment profiling. *Nat Methods* 2019;**16**:1007–15. https://doi.org/10.1038/s41592-019-0529-1.
19. Shao X, Liao J, Lu X. *et al.* Sccatch: automatic annotation on cell types of clusters from single-cell rna sequencing data. *Science* 2020;**23**:100882. https://doi.org/10.1016/j.isci.2020.100882.
20. Zhang Z, Luo D, Zhong X. *et al.* Scina: a semi-supervised subtyping algorithm of single cells and bulk samples. *Genes* 2019;**10**:531. https://doi.org/10.3390/genes10070531.
21. Cao Y, Wang X, Peng G. Scsa: a cell type annotation tool for single-cell rna-seq data. *Front Genet* 2020;**11**:524690. https://doi.org/10.3389/fgene.2020.00490.
22. Guo H, Li J. Scorter: assigning cells to known cell types according to marker genes. *Genome Biol* 2021;**22**:69. https://doi.org/10.1186/s13059-021-02281-7.

23. Ianevski A, Giri AK, Aittokallio T. Fully-automated and ultra-fast cell-type identification using specific marker combinations from single-cell transcriptomic data. *Nat Commun* 2022;**13**:1246. https://doi.org/10.1038/s41467-022-28803-w.
24. Ranjan B, Sun W, Park J. *et al.* Dubstepr is a scalable correlation-based feature selection method for accurately clustering single-cell data. *Nat Commun* 2021;**12**:5849. https://doi.org/10.1038/s41467-021-26085-2.
25. Kiselev VY, Yiu A, Hemberg M. Scamp: projection of single-cell rna-seq data across data sets. *Nat Methods* 2018;**15**:359–62. https://doi.org/10.1038/nmeth.4644.
26. Aran D, Looney AP, Liu L. *et al.* Reference-based analysis of lung single- cell sequencing reveals a transitional profibrotic macrophage. *Nat Immunol* 2019;**20**:163–72. https://doi.org/10.1038/s41590-018-0276-y.
27. Hou R, Denisenko E, Forrest AR. Scratch: a single-cell gene expression pro- file annotation tool using reference datasets. *Bioinformatics* 2019;**35**:4688–95. https://doi.org/10.1093/bioinformatics/btz292.
28. De Kanter JK, Lijnzaad P, Candelli T. *et al.* Chetah: a selective, hierarchical cell type identification method for single-cell rna sequencing. *Nucleic Acids Res* 2019;**47**:e95–5. https://doi.org/10.1093/nar/gkz543.
29. Hao Y, Stuart T, Kowalski MH. *et al.* Dictio- nary learning for integrative, multimodal and scalable single-cell analysis. *Nat Biotechnol* 2024;**42**:293–304. https://doi.org/10.1038/s41587-023-01767-y.
30. Haghverdi L, Lun AT, Lun AT. *et al.* Batch effects in single- cell rna-sequencing data are corrected by matching mutual nearest neighbors. *Nat Biotechnol* 2018;**36**:421–7. https://doi.org/10.1038/nbt.4091.
31. Tran HTN, Ang KS, Chevrier M. *et al.* A benchmark of batch-effect correction methods for single-cell rna sequencing data. *Genome Biol* 2020;**21**:1–32. https://doi.org/10.1186/s13059-019-1850-9.
32. Asada K, Takasawa K, Machino H. *et al.* Single-cell analysis using machine learning techniques and its application to medical research. *Biomedicine* 2021;**9**:1513. https://doi.org/10.3390/biomedicines9111513.
33. Ma F, Pellegrini M. Actinn: automated identification of cell types in single cell rna sequencing. *Bioinformatics* 2020;**36**:533–8. https://doi.org/10.1093/bioinformatics/btz592.
34. Domınguez Conde C, Xu C, Jarvis L. *et al.* Cross-tissue immune cell analysis reveals tissue-specific features in humans. *Science* 2022;**376**:5197. https://doi.org/10.1126/science.abl5197.
35. Galdos FX, Xu S, Goodyer WR. *et al.* Devcellpy is a machine learning-enabled pipeline for automated annotation of complex multilayered single-cell transcriptomic data. *Nat Commun* 2022;**13**:5271. https://doi.org/10.1038/s41467-022-33045-x.
36. Cheng Y, Fan X, Zhang J. *et al.* A scalable sparse neural network framework for rare cell type annotation of single-cell transcriptome data. *Commun Biol* 2023;**6**:545. https://doi.org/10.1038/s42003-023-04928-6.
37. Gayoso A, Lopez R, Xing G. *et al.* A python library for probabilistic analysis of single-cell omics data. *Nat Biotechnol* 2022;**40**:163–6. https://doi.org/10.1038/s41587-021-01206-w.
38. Jia S, Lysenko A, Boroevich KA. *et al.* Scdeepinsight: a supervised cell-type identification method for scene-seq data with deep learning. *Brief Bioinform* 2023;**24**:266. https://doi.org/10.1093/bib/bbad266.
39. Tan Y, Cahan P. Singlecellnet: a computational tool to classify single cell rna- seq data across platforms and across species. *Cell Syst* 2019;**9**:207–213.e2. https://doi.org/10.1016/j.cels.2019.06.004.
40. Erfanian N, Heydari AA, Feriz AM. *et al.* Deep learning applications in single-cell genomics and transcriptomics data analysis. *Biomed Pharmacother* 2023;**165**:115077. https://doi.org/10.1016/j.biopha.2023.115077.
41. Hajian-Tilaki K. Receiver operating characteristic (roc) curve analysis for med- ical diagnostic test evaluation. *Caspian J Intern Med* 2013;**4**:627–35.
42. Reyes M, Filbin MR, Bhattacharyya RP. *et al.* An immune- cell signature of bacterial sepsis. *Nat Med* 2020;**26**:333–40. https://doi.org/10.1038/s41591-020-0752-4.
43. Heimlich JB, Bhat P, Parker AC. *et al.* Multiomic profiling of human clonal hematopoiesis reveals genotype and cell-specific inflammatory pathway activation. *Blood Adv* 2024;**8**:3665–78. https://doi.org/10.1182/bloodadvances.2023011445.
44. Ahern DJ, Ai Z, Ainsworth M. *et al.* A blood atlas of covid-19 defines hallmarks of disease severity and specificity. *Cell* 2022;**185**:916–938.e58. https://doi.org/10.1016/j.cell.2022.01.012.
45. Jin K, Bardes EE, Mitelpunkt A. *et al.* An interactive single cell web portal identifies gene and cell networks in covid-19 host responses. *Science* 2021;**24**:103115. https://doi.org/10.1016/j.isci.2021.103115.
46. Szabo PA, Dogra P, Gray JI. *et al.* Longitudinal profiling of respiratory and systemic immune responses reveals myeloid cell-driven lung inflammation in severe covid-19. *Immunity* 2021;**54**:797–814.e6. https://doi.org/10.1016/j.immuni.2021.03.005.
47. Consortia, C.Z.I.S.-C.C, Ballestar E, Farber DL. *et al.* Single cell profiling of covid-19 patients: an international data resource from multiple tissues MedRxiv. 2020.
48. Kock KH, Tan LM, Han KY. *et al.* Single-cell analysis of human diversity in circulating immune cells. bioRxiv. 2024.
49. Perez RK, Gordon MG, Subramaniam M. *et al.* Single-cell transseq reveals cell type–specific molecular and genetic associations to lupus. *Science* 2022;**376**:1970. https://doi.org/10.1126/science.abf1970.
50. Leng K, Li E, Eser R. *et al.* Molecular characterization of selectively vulnerable neurons in Alzheimer's disease. *Nat Neurosci* 2021;**24**:276–87. https://doi.org/10.1038/s41593-020-00764-7.
51. Yang AC, Vest RT, Kern F. *et al.* A human brain vascular atlas reveals diverse mediators of Alzheimer's risk. *Nature* 2022;**603**:885–92. https://doi.org/10.1038/s41586-021-04369-3.
52. Lau S-F, Cao H, Fu AK. *et al.* Single-nucleus transcriptome analysis reveals dysregulation of angiogenic endothelial cells and neuroprotective glia in Alzheimer's disease. *Proc Natl Acad Sci* 2020;**117**:25800–9. https://doi.org/10.1073/pnas.2008762117.
53. Morabito S, Miyoshi E, Michael N. *et al.* Single-nucleus chromatin accessibility and transcriptomic characterization of Alzheimer's disease. *Nat Genet* 2021;**53**:1143–55. https://doi.org/10.1038/s41588-021-00894-z.
54. Otero-Garcia M, Mahajani SU, Wakhloo D. *et al.* Molecular signatures underlying neurofibrillary tangle susceptibility in Alzheimer's disease. *Neuron* 2022;**110**:2929–2948.e8. https://doi.org/10.1016/j.neuron.2022.06.021.
55. Reed AD, Pensa S, Steif A. *et al.* A single-cell atlas enables mapping of homeostatic cellular shifts in the adult human breast. *Nat Genet* 2024;**56**:652–62. https://doi.org/10.1038/s41588-024-01688-9.
56. Cao J, O'day DR, Pliner HA. *et al.* A human cell atlas of fetal gene expression. *Science* 2020;**370**:7721. https://doi.org/10.1126/science.aba7721.
57. Qiu C, Martin BK, Welsh IC. *et al.* A single-cell time-lapse of mouse prenatal development from gastrula to birth. *Nature* 2024;**626**:1084–93. https://doi.org/10.1038/s41586-024-07069-w.

58. Kuppe C, Ramirez Flores RO, Li Z. *et al.* Spatial multi- omic map of human myocardial infarction. *Nature* 2022;**608**:766–77. https://doi.org/10.1038/s41586-022-05060-x.
59. Molla Desta G, Birhanu AG. Advancements in single-cell rna sequencing and spatial transcriptomics: transforming biomedical research. *Acta Biochim Pol* 2025;**72**:13922. https://doi.org/10.3389/abp.2025.13922.
60. Shieh JT, Tintos-Hernandez JA, Murali CN. *et al.* Heterozygous nonsense variants in the ferritin heavy-chain gene fth1 cause a neuroferritinopathy. *Hum Genet Genom Adv* 2023;**4**:100236. https://doi:10.1016/j.xhgg.2023.100236.
61. Gonthier B, Nasarre C, Roth L. *et al.* Functional interaction between matrix metalloproteinase- 3 and semaphorin-3c during cortical axonal growth and guidance. *Cereb Cortex* 2007;**17**:1712–21. https://doi.org/10.1093/cercor/bhl082.
62. Du H, Xu Y, Zhu L. Role of semaphorins in ischemic stroke. *Front Mol Neurosci* 2022;**15**:848506. https://doi.org/10.3389/fnmol.2022.848506.
63. Everlien I, Yen T-Y, Liu Y-C. *et al.* Diazepam binding inhibitor governs neurogenesis of excitatory and inhibitory neurons during embryonic development via gaba signaling. *Neuron* 2022;**110**:3139–3153.e6. https://doi.org/10.1016/j.neuron.2022.07.022.
64. Wang P, Li X-L, Cao Z-H. Stc1 ameliorates cognitive impairment and neu- roinflammation of alzheimer's disease mice via inhibition of erk1/2 pathway. *Immunobiology* 2021;**226**:152092. https://doi.org/10.1016/j.imbio.2021.152092.
65. Zhao J, Zhang S, Wang X. *et al.* Correlation between serum angptl4 levels and white matter hyperintensity and cognitive impairment in patients with cerebral small vessel disease. *Brain Behav* 2024;**14**:3401. https://doi.org/10.1002/brb3.3401.
66. Morel V, Campana-Salort E, Boyer A. *et al.* Hint1 neuropathy: expanding the genotype and phenotype spectrum. *Clin Genet* 2022;**102**:379–90. https://doi.org/10.1111/cge.14198.
67. Cucinotta F, Ricciardello A, Turriziani L. *et al.* Farp-1 deletion is associated with lack of response to autism treatment by early start Denver model in a multiplex family. *Mol Genet Genomic Med* 2020;**8**:1373. https://doi.org/10.1002/mgg3.1373.
68. Biever A, Valjent E, Puighermanal E. Ribosomal protein s6 phosphorylation in the nervous system: from regulation to function. *Front Mol Neurosci* 2015;**8**:75. https://doi.org/10.3389/fnmol.2015.00075.
69. Netla VR, Shinde H, Kumar G. *et al.* A comparative analysis of single-cell transcriptomic technologies in plants and animals. *Curr Plant Biol* 2023;**35-36**:100289. https://doi.org/10.1016/j.cpb.2023.100289.
70. CZI Cell Science Program, Abdulla S, Aevermann B, Assis P, Badajoz S, Bell SM, Bezzi E, Cakir B, Chaffer J, Chambers S, Cherry JM. CZ CELLxGENE Discover: a single-cell data platform for scalable exploration, analysis and modeling of aggregated data. *Nucleic Acids Res* 2025;**53**:D886–900.
71. Čuklina J, Lee CH, Williams EG. *et al.* Diagnostics and correction of batch effects in large-scale proteomic studies: a tutorial. *Mol Syst Biol* 2021;**17**:10240. https://doi.org/10.15252/msb.202110240.
72. Lundberg SM, Lee S-I. A unified approach to interpreting model predictions. *Adv Neural Inf Proces Syst* 2017;**30**:4768–77.
73. Muzellec B, Telenczuk M, Cabeli V. *et al.* Pydeseq2: a python package for bulk rna-seq differential expression analysis. *Bioinformatics* 2023;**39**:btad547. https://doi.org/10.1093/bioinformatics/btad547.
74. Lindstrom MJ, Bates DM. Newton—Raphson and em algorithms for lin- ear mixed-effects models for repeated-measures data. *J Am Stat Assoc* 1988;**83**:1014–22. https://doi.org/10.1080/01621459.1988.10478693.
75. Hu C, Li T, Xu Y. *et al.* Cellmarker 2.0: an updated database of manually curated cell markers in human/mouse and web tools based on scrna-seq data. *Nucleic Acids Res* 2023;**51**:D870–6. https://doi.org/10.1093/nar/gkac947.

Briefings in Bioinformatics, 2025, **26(3)**, bbaf243
https://doi.org/10.1093/bib/bbaf243
Problem Solving Protocol