Quantitative Biology 2023, 11(3): 246–259
https://doi.org/10.15302/J-QB-022-0318

RESEARCH ARTICLE

LIBRA: an adaptative integrative tool for paired single-cell multi-omics data

Xabier Martinez-de-Morentin[1,*], Sumeer A. Khan[2], Robert Lehmann[2], Sisi Qu[2], Alberto Maillo[2], Narsis A. Kiani[3], Felipe Prosper[4,5], Jesper Tegner[2,6,*], David Gomez-Cabrero[1,2,*]

[1] Navarrabiomed, Complejo Hospitalario de Navarra (CHN), Universidad Pública de Navarra (UPNA), IdiSNA, Pamplona 31001, Spain
[2] Biological and Environmental Sciences and Engineering Division, King Abdullah University of Science and Technology, Thuwal 23955, Saudi Arabia
[3] Algorithmic Dynamic Lab, Department of Oncology and Pathology, Center for Molecular Medicine, Karolinska Institute, Stockholm 17177, Sweden
[4] Division of Hemato-Oncology, Center for Applied Medical Research CIMA, Cancer Center University of Navarra (CCUN), Navarra Institute for Health Research (IDISNA), CIBERONC, Pamplona 31008, Spain
[5] Department of Hematology, Clinica Universidad de Navarra, CIBERONC Pamplona 31008, Spain
[6] Computer, Electrical and Mathematical Sciences and Engineering Division, King Abdullah University of Science and Technology, Thuwal 23955, Saudi Arabia
* Correspondence: xavier.martinez.demorentin@navarra.es; jesper.tegner@kaust.edu.sa; david.gomezcabrero@kaust.edu.sa

Received November 29, 2022; Revised December 29, 2022; Accepted January 3, 2023

***Background*: Single-cell multi-omics technologies allow a profound system-level biology understanding of cells and tissues. However, an integrative and possibly systems-based analysis capturing the different modalities is challenging. In response, bioinformatics and machine learning methodologies are being developed for multi-omics single-cell analysis. It is unclear whether current tools can address the dual aspect of modality integration and prediction across modalities without requiring extensive parameter fine-tuning.**

***Methods*: We designed LIBRA, a neural network based framework, to learn translation between paired multi-omics profiles so that a shared latent space is constructed. Additionally, we implemented a variation, aLIBRA, that allows automatic fine-tuning by identifying parameter combinations that optimize both the integrative and predictive tasks. All model parameters and evaluation metrics are made available to users with minimal user iteration. Furthermore, aLIBRA allows experienced users to implement custom configurations. The LIBRA toolbox is freely available as R and Python libraries at GitHub (TranslationalBioinformaticsUnit/LIBRA).**

***Results*: LIBRA was evaluated in eight multi-omic single-cell data-sets, including three combinations of omics. We observed that LIBRA is a state-of-the-art tool when evaluating the ability to increase cell-type (clustering) resolution in the integrated latent space. Furthermore, when assessing the predictive power across data modalities, such as predictive chromatin accessibility from gene expression, LIBRA outperforms existing tools. As expected, adaptive parameter optimization (aLIBRA) significantly boosted the performance of learning predictive models from paired data-sets.**

***Conclusion*: LIBRA is a versatile tool that performs competitively in both "integration" and "prediction" tasks based on single-cell multi-omics data. LIBRA is a data-driven robust platform that includes an adaptive learning scheme.**

Keywords: single-cell; multi-omic; Autoencoder; auto-finetuning

Author summary: There is a need for tools that integrate single-cell multi-omic data while addressing several integrative challenges simultaneously. To this end, we designed a deep-learning based tool LIBRA that performs competitively in both "integration" and "prediction" tasks based on single-cell multi-omics data. Furthermore, when assessing the predictive power across data modalities, LIBRA outperforms existing tools. LIBRA and its adaptive scheme aLIBRA, allow automatic fine-tuning for users with limited effort. Additionally, aLIBRA allows experienced users to implement custom configurations. The LIBRA toolbox is freely available as R and Python libraries.

INTRODUCTION

Single-cell genomics technologies set the stage for unraveling the intrinsic complex organization at single-cell resolution by simultaneously profiling several layers of transcriptional regulation [1−3]. Recent multi-omics single-cell technologies enable profiling of joint "chromatin accessibility & mRNA profiles" (*e.g.*, SNARE-seq [1], sci-CAR [2], SHARE-seq [3], Paired-seq [4], 10X Genomics [5]), "mRNA profiles & protein antibody-derived tags" (CITE-seq [6]), and even more than two omics such as "chromatin accessibility, DNA methylation, and transcriptome profiling" in the scNMT-seq [7] protocol. As a result of such novel technologies, it has become necessary to develop methods to integrate multi-omic profiles at single-cell level [8] (Fig.1A). The rationale is that current state-of-the-art bulk methodologies [9–12] and frameworks [13] cannot analyze single-cell data optimally [14,15,16]. Initially, methodologies such as Seurat3 [5] allowed single-cell multiome integrative analysis; however, Seurat3 [5] does not use the information derived from the *paired* nature of the data (that is, several omic profiles obtained from the same cell). More recently, methodologies using the paired information have been developed [8]. For example, machine learning tools such as MOFA+ [17] and Seurat4 [18] allow the identification of an integrated space that can be used for improved cell clustering. However, such tools have two potentially limiting factors: scalability and robustness. Furthermore, they do not generate models that allow the estimation of an omic profile from a second omic profile. Deep learning-based methodologies have been developed to overcome such limitations. The first one was BABEL [19] which was aimed at generating predictive models that "translate" between data types. Other methodologies

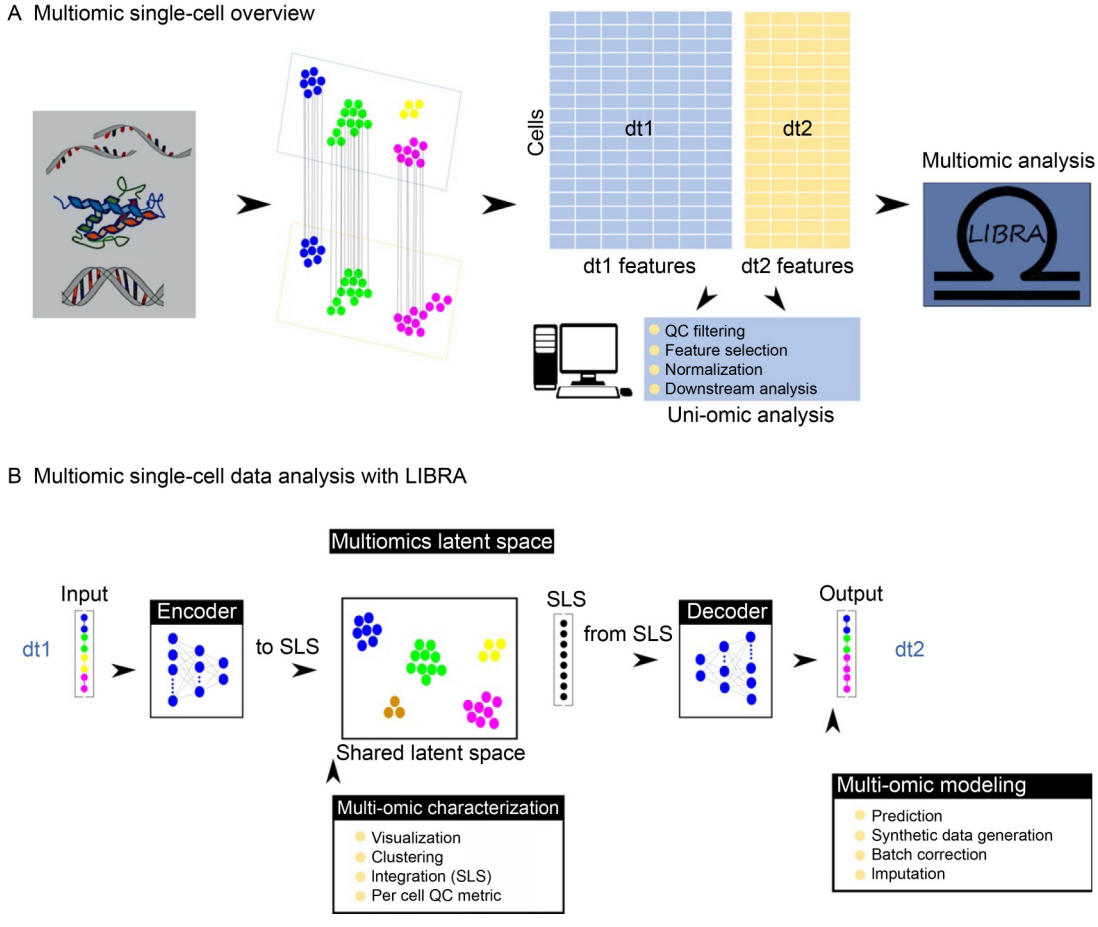


Figure 1. LIBRA overview. (A) The upper panel provides a visual representation of the task to be solved. From left to right: the initial samples and their omic profiles are represented. Then the multi-omic single-cell information is captured. Next, each omic is separately preprocessed. Finally, we propose LIBRA for the integrative analysis. (B) The conceptual idea used to develop LIBRA is represented in the lower panel. Through an encoding-decoding process, a latent space that contains the information from two omics is generated. Several measurements are carried out for quality and performance.

followed this idea, such as KPNNs [20], GAT [21], or scNym [22]. Increasingly complex models have been employed to solve specific tasks based on single-cell data. These include models based on "machine language translation" like long short-term memory (LSTM) for prediction [23] and the incoming transformers like scBERT [24] for annotation.

Recently, a Multi-modal Single-cell Data Integration Competition [25] organized by Neural Information Processing Systems (NeurIPS) was conducted. The NeurIPS challenge addressed several tasks, such as (i) predicting one modality from another (prediction), (ii) matching cells between modalities, and (iii) jointly learning representations of cellular identity (representation). In general, neural networks were the most popular and provided—in most cases—the best results. However, while the best methodologies used architectures of limited complexity, it became apparent during the competition that the methodologies required extensive fine-tuning of hyperparameters for each task and dataset regardless of the specific architecture. Furthermore, by observing that no method could win in more than one task, it was concluded that "no free lunch" [26] (no method works best for all) applies to the multi-omic analysis.

Therefore, we propose LIBRA (Fig.1B), an encoder-decoder architecture using AutoEncoders (AE) that can competitively perform two of the three tasks addressed in the NeurIPS challenge (prediction and representation). LIBRA is inspired by the ideas from neural machine translation [27]; however, the implementation selected is an AE-based framework. Similar to BABEL [19], LIBRA integrates single-cell multi-omics data by leveraging paired single-cell omics data. In the first part of the LIBRA development, we fine-tuned the LIBRA using a step-wise optimization strategy considering AE-associated quality measures and a new metric, the Preserved Pairwise Jaccard Index (PPJI). PPJI characterizes and quantifies whether the integrated space allows for finer granularity in detecting cellular subtypes. Interestingly, we observed that PPJI is a valuable metric for quantifying the added value of a multi-omic joint representation. Next, we compared LIBRA with the current state-of-the-art tools for several datasets, data modality combinations, and different tasks. LIBRA compared competitively in all cases. It is relevant to notice that LIBRA was among the top 10 in two NeurIPS sub-challenges (jointly learning and predicting modality) according to the NeurIPS challenge's metrics. Finally, to address the no-free-lunch observation, we combined LIBRA with an automatic fine-tuning paradigm [28] that allows LIBRA parameter optimization "on the run"; we denoted this version aLIBRA. It not only improves the results significantly and outperforms available methodologies but, most importantly, it provides instantiations of LIBRA that are competitive at both integration and prediction with reduced computational times. LIBRA is freely available in both R and Python (including tutorials) for multi-modal single-cell analysis.

RESULTS AND DISCUSSION

LIBRA framework

LIBRA "translates" [12] between omics. Implemented using Autoencoders, LIBRA encodes one omics and decodes the other omic to and from a reduced space. Here the decoder minimizes the distance to a second and paired data type (joined translation and projection). Briefly, LIBRA consists of two neural networks (NN) (Fig. 2A); the first NN (NN1) is designed similarly to an Autoencoder, but the difference is that input (dt1) and output (dt2) data correspond to two modalities of a paired multi-modal dataset (Fig. S1A).

Considering only one hidden layer, the encoder part of NN1 will aim to encode the input omic expression matrix denoted as $\mathrm{x} \in \mathbb{R}^d$ to the latent variables h following this formula:

$$h = \sigma(W\mathrm{x} + b) \tag{1}$$

where σ is the element-wise activation function, W is the weight matrix, and b is the bias vector. In the LIBRA implementation activation function used is leaky-relu [29]; this decision was taken due to a high rate of dead node generation using standard relu that produced lower performance and uncertainty on outcomes delivery. The activation function is then, instead of being 0 when $z < 0$, a small non-zero, constant gradient α where the function is as follows $R(z) = \begin{Bmatrix} z & z > 0 \\ \alpha z & z \leqslant 0 \end{Bmatrix}$, and its derivative is $R(z) = \begin{Bmatrix} 1 & z > 0 \\ \alpha & z < 0 \end{Bmatrix}$. This selection excludes LIBRA training to introduce death nodes due to the sparsity nature of the single-cell data. Weights and biases are initialized using distributed criteria Xavier uniform initializer and zeros. In the decoding part of the autoencoder, the output omic expression matrix is used to force the minimization of the loss based on the profiles of the second omic molecules instead of the original profiles. This process can be repeated during training using backpropagation for weight updating. The loss function employed is the mean squared error (MSE). An early stopping rule was added to save time when the evaluation function cannot retrieve better scores for MSE with a fixed patience value. In addition, a learning rate plateau callback was added to benefit from reducing the learning rate when

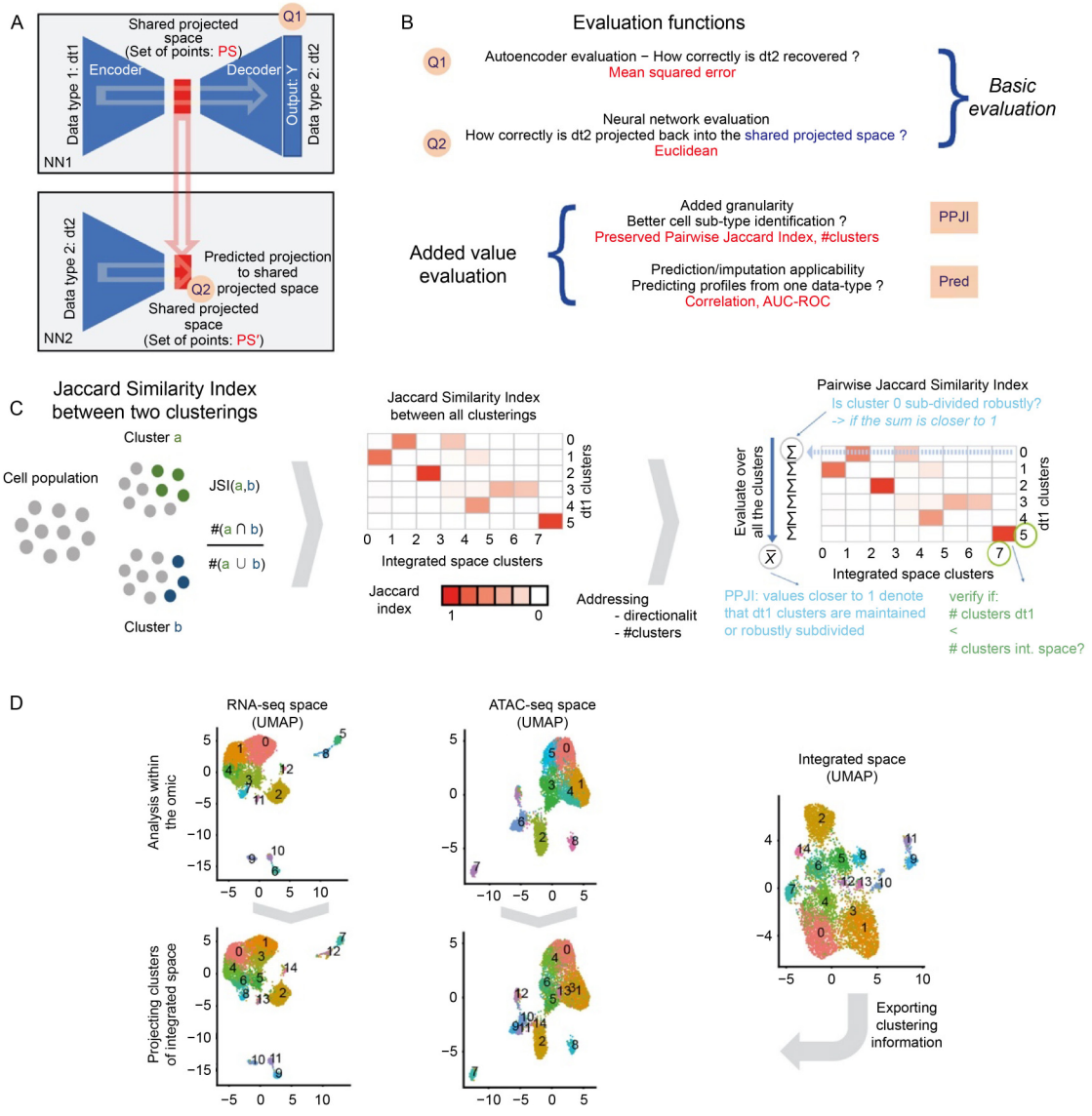


Figure 2. LIBRA design and challenge resolution. (A) Visual description of the LIBRA framework. LIBRA consists of two neural networks (NN); the first NN (NN1) is designed similarly to an Autoencoder, but with the difference that input (dt1) and output (dt2) data correspond to two modalities of a paired single-cell multi-modal dataset. The idea is to learn a shared latent space that contains the information of the two data-types, as shown in panel A. The second NN maps dt2 to the shared latent space to ensure that the projected space correctly embeds the dt2 information. See Fig. S2A, B for post-fine-tuning additional implementation details. (B) Summary overview of the evaluation functions used in the analysis and optimization of LIBRA. See Material and Methods for complete details. (C) Visual description of the PPJI. Left panel: visual description of the Jaccard Similarity Index. Middle panel: visual description of the Pairwise Jaccard Similarity Index. Right panel: visual description of the Preserved Pairwise Jaccard Index (PPJI); as shown in the figure PPJI investigates if the clusters derived from a single-omic data-analysis (dt1) are properly and robustly separated into the same or larger number of clusters. To this end, for each cluster *i* derived from dt1, the sum of JSI(*i*,x) for all clusters x in the integrated space is computed. And then, the average for all clusters from dt1 is computed as the final summary. A value of 1 denotes that dt1 clusters are perfectly identified in the integrated space; however, an added value (fine granularity at cell-type identification) requires large values of PPJI but also a larger number of clusters identified in the integrated space. An extended description of PPJI computation is provided in the methods section. (D) Example of the integrative challenge using dataset DS1. The integrated space was identified using LIBRA. Example of clustering resolution in the integrated space. Two left upper panels denote the UMAP projection and clustering for RNA and ATAC, respectively. The right panel shows the UMAP projection and clustering of cells in the integrated space (*e.g.*, the LIBRA optimized model). Finally, the two left bottom panels project the clustering information derived from the integrated space in the UMAP projections for RNA and ATAC, respectively.

no improvements are obtained on loss function during application of a fixed patience value. See Supplementary Materials and Methods for values and hyperparameters employed.

Thanks to this processing, a shared latent space (SLS) for two data types can be learned effectively (Fig. 2A). While NN1 identifies the SLS, we considered it necessary to implement a second NN(NN2) that maps dt2 to the generated SLS to ensure and quantify that the projected space correctly embeds the dt2 cells' information with a high quality (Fig. 2D). This NN2 (Fig. S1B) will use the same encoding strategy but with dt2 as input and will contain generated SLS as output. See Supplementary Materials and Methods for the complete formula.

Evaluation functions

To assess the performance of the LIBRA integration, we designed several quality metrics (Fig. 2B, C). The first set of metrics, Q1 and Q2 (Fig. 2B, upper part), is associated with the training of the neural network; the mean square error (MSE) and Euclidean distance are used to evaluate the training of NN1 and NN2, respectively. The second set of metrics (Fig. 2B, lower part) is implemented to evaluate the following LIBRA applications: (i) added value of integration, and (ii) predictive power between omic profiles. The following sub-sections describe each of the evaluation functions; for further technical details, see Supplementary Information.

Q1 has been computed as the MSE formula for NN1:

$$MSE_{NN} = \sum_{i=1}^{n} (dt_{2,i} - \widehat{dt}_{2,i})^2 \quad (2)$$

where $\widehat{dt}$ denotes the estimated value, and dt denotes the original value. n is the total number of cells for the given pair of single-cell data modalities.

Q2 has been computed as the Euclidean distances between generated SLS:

Since NN2 is trained using MSE as a loss function, the Euclidean distance Q2 between the SLSs generated in NN1, and the predicted output values of NN2 could easily be calculated as the square root of the MSE obtained in NN2 when calculated for all cells as:

$$Q2 = \sqrt[2]{MSE_{NN2}} \quad (3)$$

In the case of NN2 and based on the observed bimodal distribution (Fig. S1C), we evaluated separately the cells used in the training or validation in NN1.

Preserved Pairwise Jaccard Index

The Preserved Pairwise Jaccard Index (PPJI) is designed to quantify the added value of the paired integrative analysis to identify cell subtypes (better granularity) by providing a summary value over the Pairwise Jaccard Index (PJI) matrix (Fig. 2C). Briefly, PPJI provides a number between 0 and 1 that quantifies: "does the integrated space provide a finer cell-type definition than the cell-type definitions generated from a single-omics (*e.g.,* dt1)?". For a given cluster in dt1, the sum over the associated PJI matrix will be "1" if the cluster holds or separates perfectly into sub-clusters (Fig. 1C). Thus, PPJI calculates the average of the sums as a summary. PPJI computation will be as follows (See Supplementary Materials and Methods for a more detailed explanation):

$$\text{PPIJ} = \frac{1}{I} \sum \sum_{j \in B} \frac{(a_i \cap b_j)}{(a_i \cup b_j)} \quad (4)$$

where for each pair of clusters, $i \in$ A and $j \in$ B, a_i and b_j denote the set of cells in cluster i and j, respectively, when investigating how cluster a projects onto the reference cluster b (see Fig. 2D). Therefore, values closer to 1 denote that the clusters in dt1 are conserved or split into sub-clusters. It is important to note that the evaluation function PPJI must be combined with the "comparison between the number of clusters in dt1 and in the integrated space".

Synergy model performance ranking

To obtain a summary score of integration performance, a weighted average of the three metrics was calculated (See Supplementary Materials and Methods), where each metric is scaled and weighted equally. To do this, each time a set of combinations is compared, the results of training the AE 10 times for each combination are pooled. Then Q1, Q2, and PPJI are scaled and averaged equally to generate a final score that numerically represents overall performance.

Prediction-specific evaluation metrics

To evaluate the predictive power of LIBRA, the *pred* metric is used. The Pearson correlation and AUC-ROC curves were used for scRNA-seq and scATAC-seq, respectively. For ROC calculation, scATAC-seq predicted matrix was first binarized using a value of 0.25 as the cut-off point (based on the data distribution): values greater than 0.25 are considered 1, and values below or equal to 0.25 are considered 0. For more details on the implementation of this metric, see Supplementary Materials and Methods.

CITE-seq specific integration measurement

The last metric implemented is a CITE-seq [6] specific

for measuring integration performance. For this metric, a set of 25 reference expression proteins is used to measure how different the Spearman and Pearson correlation scores for the k-nearest neighboring cells (k = 20), in the entire feature space of the reference protein dataset, are for each of these proteins with respect to the expression of the k-nearest neighboring cells obtained in the SLS for the different methods used (LIBRA, Seurat4 [18], MOFA+ [17], totalVI [30]' and BABEL [19]) for each the 25 reference proteins.

LIBRA step-wise optimization

To identify default-tuned LIBRA's hyperparameters, we combined three quality measurements (Q1, Q2, PPJI) when conducting the analysis of the SNARE-seq [1] adult brain mouse dataset (DS1). Iteratively, we considered the optimization of the following parameters: (i) Autoencoder-type configuration = AE-based framework, (ii) number of dimensions of the projected space = 10, (iii) peak derived information for ATAC-seq, (iv) the ordering (dt1 = ATAC-seq and dt2 = RNA-seq), (v) using the most variable features only, and (vi) the number of hidden layers = 2. Table S1 includes the values for each evaluation metric. In all cases, a weighted score combining Q1, Q2, and PPJI was computed to determine the overall performance. Table S2 shows the final weighted *score* computed for each iteration in each combination. The best configuration was chosen based on overrepresentation, relative to other configurations, within the 10 highest values on each parameter selection step. (See additional details in Supplementary Materials and Methods).

Comparing LIBRA with existing tools

Next, we compared step-wise fine-tuned LIBRA using DS1 against existing tools Seurat3 [5], Seurat4 [18], MOFA+ [17], totalVI [30] and BABEL [19]. For that comparison, we used the PPJI measure (Fig. 2C, D), which quantifies the added value of multi-omic integration when identifying cell sub-types (Fig. 3A). We observed that only Seurat4 [18] outperforms default-tuned LIBRA, and it does so minimally. However, inspecting the clusters reveals an overwhelming similarity, as shown in Fig. S1D, E. Interestingly, LIBRA outperforms the other deep learning frameworks, including a concatenation of both data-type matrices in an Autoencoder to identify the shared latent space (unpaired AE). We investigated the cluster-specific markers from Seurat4 [18] and LIBRA to interrogate biological relevance. First, when taking Seurat4 [18] as a reference, the top markers identified at Seurat4 [18] are also recognized by LIBRA (Fig. S2A, B). LIBRA also identifies other markers (Fig. S2A, B). We conclude that both methodologies can recover a similar level of resolution for clusters, cell subtypes, and their associated biomarkers.

Sensitivity analysis

Next, we evaluated the robustness of both Seurat4 [18] and LIBRA by reducing the number of cells. To do so, we randomly selected and removed a certain percentage of cells while calculating the PPJI in each case. As expected, reducing the number of cells decreased the performance of Seurat4 [18] and LIBRA (Fig. 3B). Interestingly, when reducing the number of cells, LIBRA performs significantly better than Seurat4 [18]. Further robustness analysis shows that LIBRA can maintain high accuracy against randomization of matching information, dropout, and overtraining (see Table S3).

Generalization of the results

To assess the generalizability of LIBRA, we compared LIBRA with other methodologies using a wide range of datasets. We considered the following datasets: CITE-seq (Human Bone Marrow, DS2 [6]), PAIRED-seq (Mouse Adult Cerebral Cortex, DS3 [4]) and SHARE-seq (Mouse Skin, DS4 [3]), 10X (PBMC, DS5 [5]), 10XMultiome (Human Bone Marrow, DS6 [25]), CITE-seq (Human Bone Marrow, DS7 [25]) and scNMT-seq (Mouse Embryonic Stem Cells, DS8 [7]). Further details are provided in Fig. 3C and Table S4. PPJI based-comparison was not feasible in DS8 because of the limited number of cells and, as a result, the very limited number of clusters that were identified. A general observation (Fig. 3C) is that fine-tuned Seurat4 [18] surpasses all other methodologies in most cases. However, the differences between Seurat4 [18], MOFA+ [17], and LIBRA are limited, and depend on the dataset. BABEL [19] provides the worst results except for DS3 when compared against ATAC-seq as DS1. Interestingly, we found that DS3 ATAC-seq provides limited information on clusters, which is observed both in clusters based on Seurat4 integration [18] and LIBRA. However, BABEL [19] appears to prioritize the information from ATAC-seq in the integration as shown in Fig. S3. It is relevant to note that BABEL [19] development was aimed toward the prediction challenge, not cell-type identification. We also observed that the normalization procedure (*e.g.*, using or not SCT) has a limited effect on the PPJI analysis (see Table S5). In the case of DS5, being the set

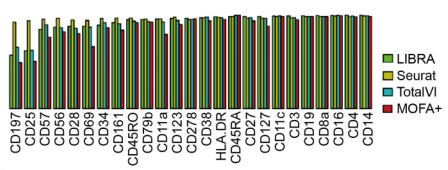
A

PPJI (to integrated)	RNA	ATAC
Ref to Seurat3	0.67	0.74
Ref to Seurat4	0.87*	0.80*
Ref to MOFA+	0.55*	0.73*
Ref to BABEL	0.48	0.50*
Ref to LIBRA	0.85*	0.78*
Ref to unpaired AE	0.48*	0.67*

B

PPJI		100%	80%	60%	40%	20%
Ref RNA vs	Seurat4	**0.87**	0.70	0.54	0.37	0.19
	LIBRA	0.85	**0.76**	**0.64**	**0.58**	**0.53**
Ref ATAC vs	Seurat4	**0.80**	0.66	0.52	0.35	0.19
	LIBRA	0.78	**0.76**	**0.77**	**0.77**	**0.74**

D



C

DataSet	Label	Details	Reference	Seurat4 (*) PPJI	Seurat4 (*) #cluster	MOFA+ PPJI	MOFA+ #cluster	totalVI PPJI	totalVI #cluster	BABEL PPJI	BABEL #cluster	LIBRA PPJI	LIBRA #cluster
DS1 GSE126074	SNARE-seq	Mouse Adult Brain Cortex	RNA	0,87	13/15	0.55	13/15	0.33	13/7	0.48	13/13	0.86	13/15
			ATAC	0.80	9/15	0.73	9/15	0.44	9/7	0.50	9/13	0.78	9/15
DS2 GSE128639	CITE-seq	Human Bone Marrow	RNA	0.72	39/40	0.76	39/37	0.72	39/38	(*)	--	0.72	39/39
			ADT	0.70	36/40	0.55	36/37	0.66	36/38	--	--	0,62	36/39
DS3 GSE130399	PAIRED-seq	Mouse Adult cerebral cortex	RNA	0.74	19/35	0.32	19/13	0.27	19/8	0.31	19/12	0.42	19/22
			ATAC	0.63	11/35	0.63	11/13	0.43	11/8	0.95	11/12	0.57	11/22
DS4 GSE140203	SHARE-seq	Mouse Skin Anagen	RNA	0.77	17/25	0.72	17/22	0.60	17/19	0.62	17/19	0.74	17/24
			ATAC	0.68	17/25	0.69	17/22	0.66	17/19	0.55	17/19	0.63	17/24
DS5 10X Genom	10X	Human PBMC	RNA	0.82	18/24	0.59	18/21	0.53	18/18	(**)	--	0.79	18/22
			ATAC	0.75	18/24	0.59	18/21	0.53	18/18	--	--	0.72	18/22
DS6 GSE194122	10X	Human PBMC	RNA	0.84	24/30	(****)	--	0.69	24/22	(*)	--	0.79(0.82)	24/28(53)
			ATAC	0.77	20/30	--	--	0.70	20/22	--	--	0.67(0.81)	20/28(53)
DS7 GSE194122	10X	Human PBMC	RNA	0.82	28/43	0.75	28/36	0.75	28/37	(**)	--	0.72	28/33
			ADT	0.67	43/43	0.55	43/36	0.58	43/37	--	--	0.58	43/33

Figure 3. LIBRA step-wise tuning and comparison with existing methodologies. (A) PPJI evaluation in DS1 for each of the methodologies considered in the analysis. * Indicates that the number of clusters in the integrated space is larger than is the case for the uni-omic analysis. (B) PPJI values derived from LIBRA and Seurat4 analysis, both using paired information when the total number of cells used to create the model is a percentage from the original total number (6735 cells). (C) PPJI-based comparison between the different methodologies in several datasets. (*) Not conducted in BABEL. (**) DS5 was analyzed using all genes instead of mvg. BABEL framework exceeded the limiting time of 1 week for running on a GPU infrastructure. (***) DS6 was analyzed using up to 2TB ram, but greater resources were required. (D) Analysis outcome in the CITE-seq dataset. Spearman protein expression correlation scores obtained on k-nearest neighboring (k = 20) integrated latent spaces for all integration methods and original reference CITE-seq dataset k-nearest neighboring (k = 20).

with the largest number of cells (at the time this analysis was conducted), we observed that using “all features" instead of “most-variables features” provided slightly better integration results (< 0.02 PPJI difference); consequently, we analyzed all methods using the “all features” option. It was not possible to run BABEL [19] within a reasonable time frame with the entire set of features on DS5.

As an extension to the current work, we compared LIBRA against the winning framework in the NeurIPS challenge (concatenated AE) on dataset DS6. Concatenated AE obtained a resolution of 23 clusters with a PPJI score of 0.72 and 0.64 for scRNA and scATAC, respectively. Considering the performance of LIBRA with default hyperparameters, we obtained resolution scores for 28 clusters and PPJI scores of 0.79 and 0.67, respectively. We conclude that LIBRA outperforms the concatenated AE in the resolution and preservation of biological information in SLS.

To evaluate LIBRA in other combinations of data modalities, we investigated the prediction in CITE-seq. To that end, we estimated the expression of 25 protein values in CITE-seq DS2 dataset [5] using the profiles from the neighboring cells as conducted in Seurat4 [18] analysis [17] and using previously explained metric computed for each of the SLS components. Seurat4 [18], LIBRA, totalVI [30], and MOFA+ [17] returned the best results for 14, 11, 1, and 1 of the 25 antibodies, respectively as shown in Fig. 3D. Overall, Seurat4 [18] provides the most stable results, followed by LIBRA.

Predictive power of LIBRA

While LIBRA is comparable with Seurat4 [18] using PPJI, the added value of the LIBRA framework is its use as a predictive model. The generation of a LIBRA model for a paired dataset allows the prediction of unknown biomolecule profiles from single-omic single-cell data of the same biological system. Given the dt1 = ATAC and dt2 = RNA, we quantified the predictive power for RNA profiles, predRNA, as the Pearson correlation between known and predicted profiles, as used in BABEL [19]. We acknowledge that predRNA can also be considered as an evaluation measure for NN1. We compared predRNA value between BABEL [19] and LIBRA on all datasets for which that was possible (Table S6); we observed that LIBRA outperforms BABEL [19] in all cases. We also observed that the prediction values estimation is valid for all clusters, and the correlation per cluster is not associated with the number of cells in the cluster (Tables S7, 8). Again, we observed that LIBRA outperforms BABEL [19] when using RNA to predict ATAC-seq profiles (0.87 vs. 0.85 predATAC, see Supplementary Materials and Methods).

Comparing running times

When comparing the computational cost (Table 1), Seurat4 [18] is the fastest. However, given that LIBRA training must be performed once or, at most, a few times in any single-cell multi-omics analysis, we consider the observed CPU time cost to be functional (Table 1). In particular, although BABEL [19] and LIBRA are both AE-inspired methodologies, the more complex architecture of BABEL [19] makes it much more time-consuming.

Adaptative LIBRA: automatic dataset specific auto-finetuning for LIBRA

From an observation derived from the NeuRIPS challenge, fine-tuning seems to be a necessary step to find optimal performance for all neural network architectures. To further investigate this hypothesis, we calculated LIBRA evaluation scores for different combinations of parameters, fixing the other parameters at a given value, using the largest dataset DS6. As shown in Fig. 4A, B, the different evaluation metrics used may show different associations to neural network parameters. Therefore, it is necessary to characterize the fitness landscape shown in Fig. 4C. In the example, aimed at optimizing the "integrative scores" we have compared 423 models (each model associated with a different parameter setting for LIBRA) using a grid of vectors for the different hyperparameters (Table S9; Materials and Methods).

A first observation is that, as shown in see Fig. 4D, fine-tuned LIBRA (aLIBRA) increased the PPJI scores for RNA (from 0.79 to 0.82) and ATAC (from 0.67 to 0.81). In the context of ATAC outperforming Seurat4 [18]. See also the extended clustering definition in Fig. 4E, F.

A second observation is that we can use the fine-tuning to investigate further the association between the parameters space and the tasks of prediction and integration. In the example shown in Fig. 4A, we observed that frameworks with lower nodes yield better predictions; see the section on Methods for the details of the two different parameter sets. We investigated whether there were combinations–of the overlapping parameter space–that returned good evaluations on both criteria (see Fig. 4G). We observed that there are Pareto optimals: competitive frameworks in both tasks.

Based on all those observations, we conclude that a predefined or even a step-wise model concept can be further improved if complemented by a fine-tuning strategy that fits the data (as is the case for aLIBRA).

LIBRA as a resource

LIBRA has been implemented as a Python package called sc-libra. It provides state-of-the-art performance while maintaining a competitive runtime (when compared with other Deep Learning based methods that require more CPU time to perform similar training processes). In addition, LIBRA, in the aLIBRA variant, includes the possibility to train hundreds of models in parallel for data-driven parameter fine-tuning.

LIBRA is a modular toolbox and, in our experience, is easy to use. All function outputs and the directory tree are generated "behind the scenes," and the required user interaction is very limited Fig. 5A.

Limitations, future developments and considerations about the evaluation functions

In our evaluation of LIBRA, two limitations are of relevance: (i) the current version of LIBRA is limited to two data types, and (ii) the evaluation functions used,

Table 1 Computational costs of each of the methodologies

DataSet	ID	Training time			Per epoch			MOFA+ (CPU) (h)	TotalVI (CPU) (h)	Seurat4 (CPU) (min)
		LIBRA (CPU)(min)	BABEL (GPU) (min)	BABEL GPU 2 CPU (h)	LIBRA (CPU) (s)	BABEL (GPU) (s)	BABEL GPU 2 CPU (s)			
DS1	GSE126074	12	10	6	3	12	360	6.5	15	<3
DS2	GSE128639	10	(*)		1	(*)		3.7	12	<3
DS3	GSE130399	10	5	3	5	8	240	0.9	4.5	<3
DS4	GSE140203	33	23	11.5	10	17	128	2.4	11.5	<3
DS5	10X Genom	25(*)	(**)		3	(**)		3.5	13	<3
DS6	GSE194122	42	(*)		10	17	128	Out of mem (2 TB ram)	>24	<3
DS7	GSE194122	10	(*)		3	(*)		5.5	19	<3
DS8	GSE109262	1	(***)		1	(***)		–	–	<3

Computational times required to generate the integrated spaces for each of the tested models. Estimations for GPUs (Tesla V100) or CPU based systems were computed based on Nvidia supplier specifications.

such as PPJI, are not a complete evaluation criterion of the integrative outcome. We observed that LIBRA is competitive when compared to existing methodologies of more than two omics (multigrate [31] and multiVI [32]), such an observation encourages the development of "> 2 omic data types" versions of LIBRA (see Fig. S4A). However, while performing the analysis, we also observed that measurements derived from challenges (*e.g.*, NeuRIPS' single-cell integration benchmark, (scib) [33]) provide different results from those derived from PPJI (which aims to quantify the added granularity derived from multiomics). When using PPJI we expect optimal multi-omic models to provide a higher granularity (more clusters), as shown in Fig. S4B, C. However, in the case of optimal scib-derived models, we observed an association with fewer clusters (see Fig. S4 D, E)—in some cases fewer than in the case of clusters derived from uni-omic analysis. Furthermore, the biological utility of the clusters identified by each one of the methods (and evaluation functions used) differ (see Fig. S5). While the design and nature of PPJI and scib are different, the observed results reflect the need for further investigation of the evaluation criteria and the biological value added.

CONCLUSIONS

To analyze single-cell multi-omics profiles [34], the research community needs powerful multi-omics data analysis software tools [8] capable of handling different combinations of omics. Moreover, these tools need to be adapted to different data modalities, to several challenges (multi-objective optimization) and to the specific characteristics of each dataset. To respond to this demand, we present LIBRA.

LIBRA is a tool that leverages paired-single-cell information using an AE framework to address two fundamental challenges in analyzing multi-modal single-cell data. Namely, to identify the joined space, thus facilitating cell-type resolution, and allowing prediction between different omics modalities. We observed that LIBRA competes with state-of-the-art tools in both tasks and is robust when the number of cells is reduced. Moreover, LIBRA's architecture and learning scheme are generalizable to any pair of omics. This allows LIBRA to be used in any biological context regardless of the nature of the biomolecule profiles used.

The limited CPU time demand of the model allows LIBRA to be easily fine-tuned to the characteristics of the different datasets (with its considerable effect on performance improvement), something that would be impossible with tools requiring higher CPU times such as totalVI [30], MOFA+ [17], or BABEL [19]. Furthermore, the simplicity of the LIBRA model, its limited CPU time requirements and its scalability allow LIBRA to be combined with a fine-tuning strategy. aLIBRA significantly refines and improves the model output and, consequently, outperforms other methodologies. Furthermore, aLIBRA allows the identification of frameworks during fine-tuning that are competitive in both prediction and integration, as shown in Fig. 4G.

LIBRA's limited computational time requirements make it a candidate for the analysis of large datasets such as the Human Cell Atlas [35], where the integration of this type of data involves additional technical complications (batch effects depending on technologies, laboratories, etc.).

In summary, LIBRA and aLIBRA are state-of-the-art tools for single-cell multi-modal prediction and projection analysis, whose implementations are available as OpenSource in R and Python (Fig. 5), with tutorials available. LIBRA is implemented as a Python package (under PyPI repository) called sc-libra, allowing users to efficiently perform all proposed analyses and metrics on any pair of paired single-cell omics. Online documentation for sc-libra is provided as a user's guide through this package.

MATERIALS AND METHODS

Preprocessing of scRNA-seq data

Following Seurat guidelines, several cell and feature quality filtering were applied. Cells with higher than 90% or lower than 10% for the "number of features" or "counts per cell" were filtered out. Cells with counts in less than 201 genes were filtered out. Genes with counts in less than 5 cells were filtered out. Cells with more than 5% reads mapping to mitochondrial genes were filtered out. When most variable genes (mvg) were used, the 2000 most variables genes were selected. Within each cell, the number of reads per gene was divided by the total number of reads in the cell and multiplied by a scale factor (10,000); then, a log-transformation was applied. Feature subspaces were obtained from most variable genes using principal component analysis (PCA) based on the top 15 principal components. Clustering was computed using the Louvain algorithm over principal components subspace. Bootstrap subsampling snakemake workflow was used to identify the optimal number of nearest neighbors and the resulting resolution. The ranges of the values for these parameters were 8−16 and 0.6−1.4, respectively. We used a subsampling rate of 0.8 for 20 subsamples, which generated a total of 500 samples for analysis. The clustering was repeated 1000 times with the final settings to discard spurious clusters. More details are available in the Supplementary Materials and Methods section.

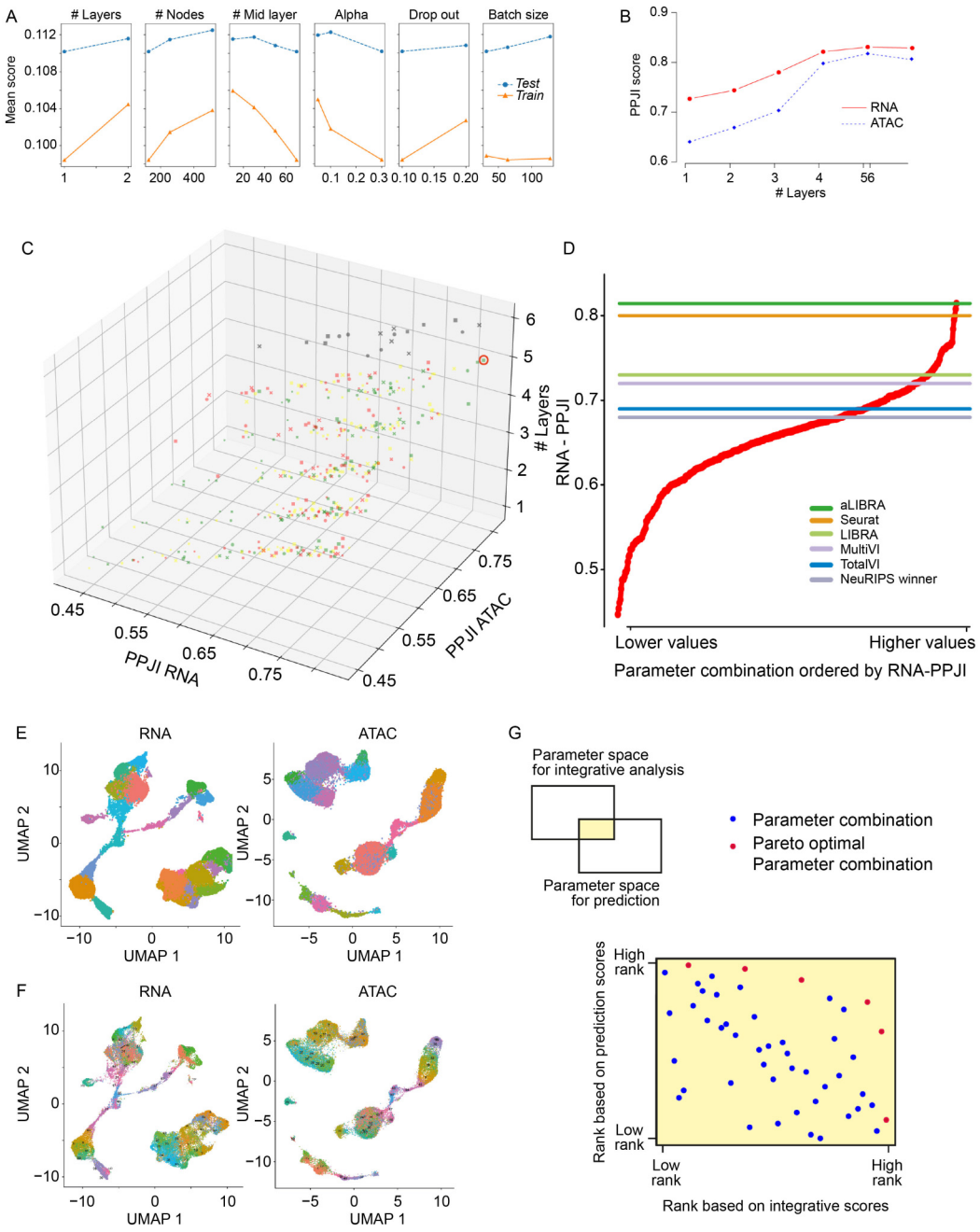


Figure 4. Additional evaluation. (A) Fine-tuned LIBRA using Q1. Orange represents the performance of model over training set (80% of cells in DS6) and blue the performance obtained over the test set (20% of cells in DS6). Score denotes the values associated to the training of NN1 (Q1). (B) Fine-tuned of LIBRA using PPJI. RNA preserved information score as red and ATAC preserved information score as blue. (C) aLIBRA results over DS6 (10XMultiome). The 3D representation provides the differences in performance because of hyperparameter differences. A version adapted to surpass fixed configurations limitations provides substantial performance improvements compared to that of fixed configuration. The model that has shown a higher performance over the other combinations of hyperparameters is highlighted by a red circle. Each dimension is denoted as; *x*-axis (RNA-seq PPJI), *y*-axis (ATAC-seq PPJI), *z*-axis (# of layers), size (%dropout, 0.1-smaller and 0.2-bigger), colors (#of nodes of first layer, 256-red, 512-yellow, 1024-green and 2048-black) and shape (#of nodes in middle layer, 10-comma, 50-cross and 70-circle). (D) Ranking of the parameter combinations from lower to higher values for the combinations investigated in the integrative analysis. Lines denote the values obtained by different methodologies. (E) Original RNA and ATAC clustering information is shown within UMAP representation. (F) As (E) but information provided corresponds to LIBRA fine-tuned model clustering outcomes. (G) Identification of Pareto optimals.

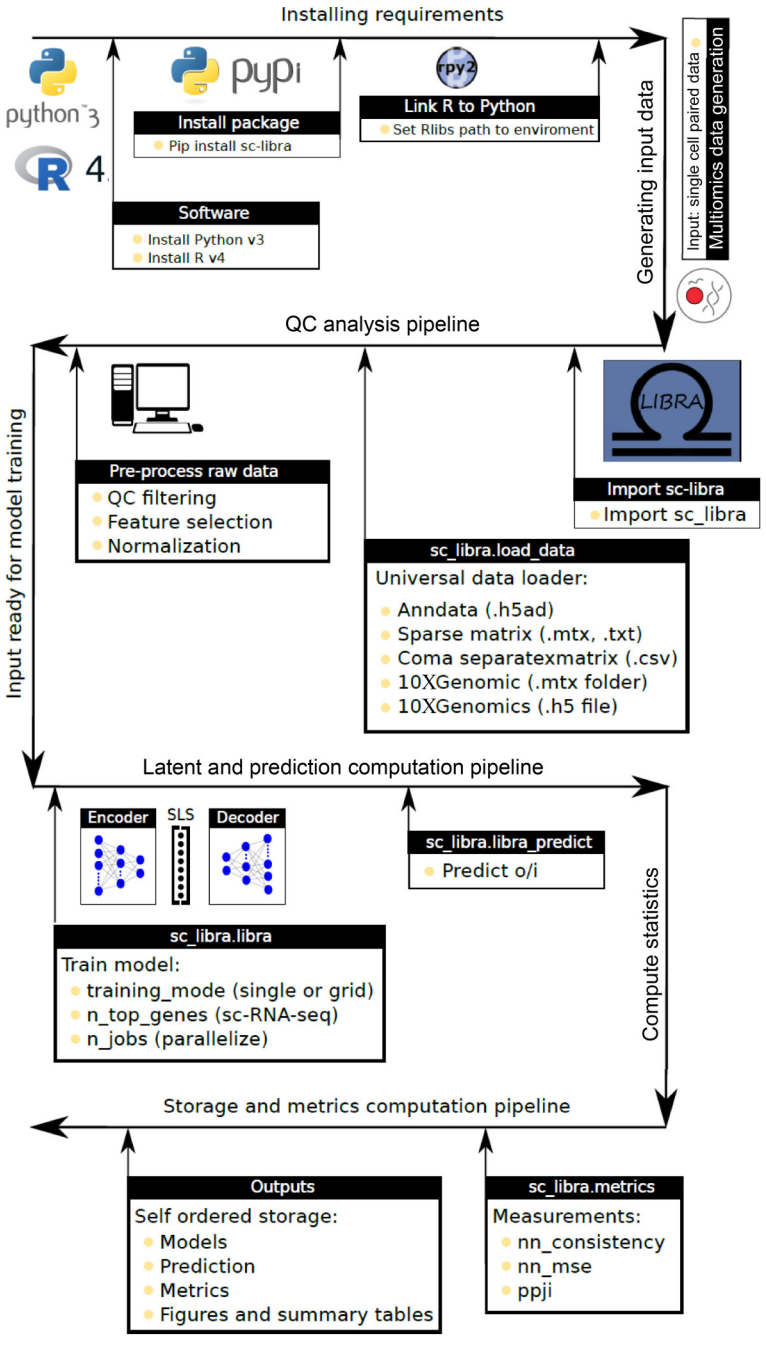


Figure 5. LIBRA and aLIBRA: sc-libra Python package graphical pipeline. Working from top to bottom, this provides a conceptual guide to using the LIBRA pipeline. Expected outputs and metrics are shown.

As a result, a robust latent space and clustering results were obtained for scRNA-seq, which can be used when comparing integrated-based approaches. The normalized and log-transformed scRNA-seq matrix will be the input to the LIBRA model.

Preprocessing of scATAC-seq data

scATAC-seq data was preprocessed similarly to scRNA-seq except when as described below. In scATAC-seq, a combined Seurat and Signac guideline was used.

Because of the greater sparsity of scATAC-seq, peaks were filtered out if they were profiled in less than 4 cells. Data matrix was normalized using the frequency-inverse document frequency (TF-IDF) method. In scATAC-seq, we used the entire feature space. Reduced feature subspaces were computed over all peaks feature space using singular value decomposition (SVD),

providing latent semantic indexing (LSI) as latent space with 50 components. The values and functions employed are detailed in the Supplementary Materials and Methods section.

The Signac *activity* estimation approach was used to conduct Seurat integrative analysis. An upstream from the Transcription Starting Site of 2.000 base pairs was used for "peak to gene relation" estimation. GRCH38 and mm10 reference genomes were used for human and mouse, respectively. See Supplementary Materials and Methods for Seurat integration parameter.

scATAC-seq reduced latent space and clustering results are used when evaluating the integrative analysis. A normalized scATAC-seq matrix will serve as input to the LIBRA model.

Preprocessing of CITE-seq data

The initial pipeline for the analysis of CITE-seq raw data was similar to previous data modalities. Differences are detailed below.

The entire protein space was used instead of selecting for most variable proteins sub-space. In addition, the protein expression measurements for each cell were normalized using centered log-ratio transformation (CLR). Values and functions employed are available in the Supplementary Materials and Methods section. The reduced latent space and clustering results obtained are used as antibody-derived tags (sc-ADT) reference for later performance metrics computation. A normalized scADT matrix will serve as input to the LIBRA framework.

Adaptative fine-tuning, aLIBRA

The first version of the LIBRA framework was instantiated using a step-wise optimization procedure over a single dataset; as a result, a set of parameters were selected, and such a framework was applied to all datasets. This step-wise optimization is detailed in the Results section. However, such parameter combinations may perform sub-optimally for different datasets or combinations of data modalities. Based on that assumption, we combine LIBRA with an automatic grid-based fine-tuning strategy to identify the optimal set of parameters for any given dataset; we denote the implementation as adaptive LIBRA (aLIBRA).

In aLIBRA, the optimal combinations of the number of layers, number of nodes, alpha, dropout, and mid-layers size for each of NN1 and NN2 are identified. A non-linear shrinkage was used for the following hidden layer size rule $Layersize_{N=}inputlayersize/2*N$ for both encoding and inverse boosting in the decoding part of the autoencoder; N denotes the position a layer has in a neural network. This consideration prevents LIBRA from generating layers smaller than the "middle layer" size for very large NN (which may be necessary for very large datasets).

Fine-tuning is executed twice for each of the tasks: integration and prediction. For integration, aLIBRA considered the following values: number of layers (1,2,3,4,5,6), number of nodes (256,512,1024,2048), alpha (0.1,0.3,0.5), dropout (0.1,0.2,0.3,0.4) and mid layers size (10,50,70). For prediction, aLIBRA considered the following values: number of layers (1,2), number of nodes (128,256,512), alpha (0.05,0.1,0.3), dropout (0.1,0.2), batch size (32,64,128) and mid layers size (10,30,50,70). These options are customizable in the Python implementation.

The fine-tuning of aLIBRA has been implemented with a parallelization strategy to decrease the computation time requirements. For further details, see Supplementary Materials and Methods.

AVAILABILITY AND REQUIREMENTS

Project name: LIBRA.
Project home page: "GitHub website (TranslationalBioinformaticsUnit/LIBRA)".
Operating system(s): platform independent. Tested on LINUX.
Programming language(s): sc-libra (LIBRA package implementation at PyPI), Python, Jupyter notebook, R and RMarkDown.
License: GPL-3.0 license.
Any restrictions to use by non-academics: none.

AVAILABILITY OF DATA AND MATERIALS

The datasets re-analyzed during the current study are available in the NCBI GEO repository via accession numbers GSE126074, GSE128639, GSE130399, GSE140203, GSE194122, GSE109262 and 10X Genomics website repository. The developed package and its online documentation and the code used for the re-analysis, are available at: sc-libra package: Pypi (sc-libra); sc-libra online docs: Read the docs (sc-libra); GitHub repository: GitHub website (TranslationalBioinformaticsUnit/LIBRA); Cone of GitHub repository plus data repository: Figshare (LIBRA).

ABBREVIATIONS

NN	Neural networks
GEO	Gene Expression Omnibus
SLS	Shared latent space
PJI	Pairwise Jaccard Index
DS	Data set
predRNA	Predicted RNA
predATAC	Predicted ATAC
MSE	Mean squared error

SNARE-seq	Droplet based technology to profile chromatin accessibility and gene expression from the same cells
CITE-seq	Qualitative information over gene expression and surface proteins with available antibodies on a single cell level
Paired-seq	Combinatorial indexing strategy to simultaneously tag both the open chromatin fragments generated by the Tn5 transposases and the cDNA molecules generated from reverse transcription
SHARE-seq	Strategy that uses three rounds of barcodes by ligating barcoded adaptors to both RNA (gene expression) and tagmented DNA (chromatin accessibility) to achieve the multi-omic profiling from the same single cells
10X	10X Genomics single-cell multiomics solutions
CITE-seq	Method for performing RNA sequencing along with gaining quantitative and qualitative information on surface proteins with available antibodies on a single cell level
scNMT-seq	Method to look at methylation (CpG) and chromatin accessibility (GpC)

SUPPLEMENTARY MATERIALS

The supplementary materials can be found online with this article at https://doi.org/10.15302/J-QB-022-0318.

AUTHORS CONTRIBUTIONS

XMM, DGC designed LIBRA and the computational experiments shown. XMM performed most of the experiments and analyzed the results. SQ conducted the initial experiments associated with BABEL. XMM, JT and DGC wrote the first draft and the final version. SQ, SK, RL, AM, NK, FP, and JT provided additional insights into the experiments and the text. All authors reviewed the manuscript before submission.

ACKNOWLEDGEMENTS

This work was supported by grants from the European Union under the Horizon 2020 programme (MultipleMS grant agreement 733161) to NK; and from the Spanish Government, through project PID2019-111192GA-I00 (MICINN) to DGC.

COMPLIANCE WITH ETHICS GUIDELINES

The authors Xabier Martinez-de-Morentin, Sumeer A. Khan, Robert Lehmann, Sisi Qu, Alberto Maillo, Narsis A. Kiani, Felipe Prosper, Jesper Tegner and David Gomez-Cabrero declare that they have no conflict of interest or financial conflicts to disclose.

This article does not contain any studies with human or animal materials performed by any of the authors.

OPEN ACCESS

REFERENCES

1. Chen, S., Lake, B. B. and Zhang, K. (2019) High-throughput sequencing of the transcriptome and chromatin accessibility in the same cell. Nat. Biotechnol., 37, 1452−1457
2. Cao, J., Cusanovich, D.A., Ramani, V., Aghamirzaie, D., Pliner, H.A., Hill, A.J., Daza, R.M., McFaline-Figueroa, J.L., Packer, J.S., Christiansen, L., *et al*. (2018) Joint profiling of chromatin accessibility and gene expression in thousands of single cells. Science, 361, 1380−1385
3. Ma, S., Zhang, B., LaFave, L. M., Earl, A. S., Chiang, Z., Hu, Y., Ding, J., Brack, A., Kartha, V. K., Tay, T., *et al*. (2020) Chromatin potential identified by shared single-cell profiling of RNA and chromatin. Cell, 183, 1103−1116.e20
4. Zhu, C., Yu, M., Huang, H., Juric, I., Abnousi, A., Hu, R., Lucero, J., Behrens, M. M., Hu, M. and Ren, B. (2019) An ultra high-throughput method for single-cell joint analysis of open chromatin and transcriptome. Nat. Struct. Mol. Biol., 26, 1063−1070
5. Stuart, T., Butler, A., Hoffman, P., Hafemeister, C., Papalexi, E., Mauck, W. M. 3rd, Hao, Y., Stoeckius, M., Smibert, P. and Satija, R. (2019) Comprehensive integration of single-cell data. Cell, 177, 1888−1902.e21
6. Stoeckius, M., Hafemeister, C., Stephenson, W., Houck-Loomis, B., Chattopadhyay, P. K., Swerdlow, H., Satija, R. and Smibert, P. (2017) Simultaneous epitope and transcriptome measurement in single cells. Nat. Methods, 14, 865−868
7. Clark, S. J., Argelaguet, R., Kapourani, C. A., Stubbs, T. M., Lee, H. J., Alda-Catalinas, C., Krueger, F., Sanguinetti, G., Kelsey, G., Marioni, J. C., *et al*. (2018) scNMT-seq enables joint profiling of chromatin accessibility DNA methylation and transcription in single cells. Nat. Commun., 9, 781
8. Argelaguet, R., Cuomo, A. S. E., Stegle, O. and Marioni, J. C. (2021) Computational principles and challenges in single-cell data integration. Nat. Biotechnol., 39, 1202−1215
9. Rohart, F., Gautier, B., Singh, A. and Lê Cao, K.-A. (2017) mixOmics: An R package for 'omics feature selection and multiple data integration. PLOS Comput. Biol., 13, e1005752
10. Argelaguet, R., Velten, B., Arnol, D., Dietrich, S., Zenz, T., Marioni, J. C., Buettner, F., Huber, W. and Stegle, O. (2018) Multi-omics factor analysis−a framework for unsupervised integration of multi-omics data sets. Mol. Syst. Biol., 14, e8124

11. Lock, E. F., Hoadley, K. A., Marron, J. S. and Nobel, A. B. (2013) Joint and individual variation explained (Jive) for integrated analysis of multiple data types. Ann. Appl. Stat., 7, 523−542
12. Teschendorff, A. E., Jing, H., Paul, D. S., Virta, J. and Nordhausen, K. (2018) Tensorial blind source separation for improved analysis of multi-omic data. Genome Biol., 19, 76
13. Gomez-Cabrero, D., Tarazona, S., Ferreirós-Vidal, I., Ramirez, R. N., Company, C., Schmidt, A., Reijmers, T., Paul, V. V. S., Marabita, F., Rodríguez-Ubreva, J., *et al*. (2019) STATegra, a comprehensive multi-omics dataset of B-cell differentiation in mouse. Sci. Data, 6, 256
14. Stegle, O., Teichmann, S. A. and Marioni, J. C. (2015) Computational and analytical challenges in single-cell transcriptomics. Nat. Rev. Genet., 16, 133−145
15. Perkel, J. M. (2021) Single-cell analysis enters the multiomics age. Nature, 595, 614−616
16. Marx, V. (2022) How single-cell multi-omics builds relationships. Nat. Methods, 19, 142−146
17. Argelaguet, R., Arnol, D., Bredikhin, D., Deloro, Y., Velten, B., Marioni, J. C. and Stegle, O. (2020) MOFA+: a statistical framework for comprehensive integration of multi-modal single-cell data. Genome Biol., 21, 111
18. Hao, Y., Hao, S., Andersen-Nissen, E., Mauck, W. M. 3rd, Zheng, S., Butler, A., Lee, M. J., Wilk, A. J., Darby, C., Zager, M., *et al*. (2021) Integrated analysis of multimodal single-cell data. Cell, 184, 3573−3587.e29
19. Wu, K. E., Yost, K. E., Chang, H. Y. and Zou, J. (2021) BABEL enables cross-modality translation between multiomic profiles at single-cell resolution. Proc. Natl. Acad. Sci. USA, 118, e2023070118
20. Fortelny, N. and Bock, C. (2020) Knowledge-primed neural networks enable biologically interpretable deep learning on single-cell sequencing data. Genome Biol., 21, 190
21. Ravindra, N., Sehanobish, A., Pappalardo, J. L., Hafler, D. A. and Van Dijk, D. "Disease state prediction from single-cell data using graph attention networks," *ACM CHIL 2020 - Proc. 2020 ACM Conf. Heal. Inference, Learn.*, pp. 121–130, 2020
22. Kimmel, J. C. and Kelley, D. R. (2021) Semisupervised adversarial neural networks for single-cell classification. Genome Res., 31, 1781−1793
23. Sargent, B., Jafari, M., Marquez, G., Mehta, A. S., Sun, Y. H., Yang, H. Y., Zhu, K., Isseroff, R. R., Zhao, M. and Gomez, M. (2022) A machine learning based model accurately predicts cellular response to electric fields in multiple cell types. Sci. Rep., 12, 9912
24. Yang, F., Wang, W., Wang, F., Fang, Y., Tang, D., Huang, J., Lu, H. and Yao, J. (2022) scBERT as a large-scale pretrained deep language model for cell type annotation of single-cell RNA-seq data. Nat. Mach. Intell., 4, 852−866
25. Lücken, M. D., Burkhardt, D., Cannoodt, R., Lance, C., Agrawal, A., Aliee, H., Chen, A., Deconinck, L., Detweiler, A., Granados, A. *et al*. (2021) A sandbox for prediction and integration of DNA, RNA, and protein data in single cells. In: NeurIPS 2021 Track Datasets Benchmarks, pp. 1−13
26. Lockett, A. J. (2020) No free lunch theorems. Nat. Comput. Ser., 1, 287−322
27. Cho, K., van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H. and Bengio, Y. (2014) Learning phrase representations using RNN encoder–decoder for statistical machine translation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 1724−1734
28. Pedregosa, F., Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R. (2011) Scikit-learn: machine learning in Python. J. Mach. Learn. Res., 12, 2825−2830
29. Xu, B., Wang, N., Chen, T. and Li, M. (2015) Empirical evaluation of rectified activations in convolutional network. arXiv, 1505.00853v2
30. Gayoso, A., Steier, Z., Lopez, R., Regier, J., Nazor, K. L., Streets, A. and Yosef, N. (2021) Joint probabilistic modeling of single-cell multi-omic data with totalVI. Nat. Methods, 18, 272−282
31. Lotfollahi, M., Litinetskaya, A. and Theis, F. J. (2022) Multigrate: single-cell multi-omic data integration. bioRxiv, 2022.03.16.484643
32. Ashuach, T., Gabitto, M. I., Jordan, M. I. and Yosef, N. (2021) MultiVI: deep generative model for the integration of multi-modal data. bioRxiv, 2021.08.20.457057
33. Luecken, M. D., Büttner, M., Chaichoompu, K., Danese, A., Interlandi, M., Mueller, M. F., Strobl, D. C., Zappia, L., Dugas, M., Colomé-Tatché, M., *et al*. (2022) Benchmarking atlas-level data integration in single-cell genomics. Nat. Methods, 19, 41−50
34. Mimitou, E. P., Lareau, C. A., Chen, K. Y., Zorzetto-Fernandes, A. L., Hao, Y., Takeshima, Y., Luo, W., Huang, T. S., Yeung, B. Z., Papalexi, E., *et al*. (2021) Scalable, multimodal profiling of chromatin accessibility, gene expression and protein levels in single cells. Nat. Biotechnol., 39, 1246−1258
35. Rood, J. E., Maartens, A., Hupalowska, A., Teichmann, S. A. and Regev, A. (2022) Impact of the Human Cell Atlas on medicine. Nat. Med., 28, 2486−2496

