**Supp. Information**

# Machine Translation between paired Single-Cell Multi-Omics Data


SUPPLEMENTARY METHODS 2

**Datasets.** **2**

**Pre-analysis and data processing:** **4**

**Visualization:** **7**

**Clustering for integrated and independent omic modalities:** **7**

**Overview of the Quality Measures.** **7**

**Model performance ranking** **8**

**Supplementary NN2 (mapper).** **9**

**Computing the Preserved Pairwise Jaccard Index (PPJI)** **9**

**Q1 and Q2 evaluation.** **10**

**PredRNA** **10**

**PredATAC** **11**

**Default model development.** **11**

**LIBRA structure and additional hyperparameters.** **12**

**LIBRA fine-tune (deep adaptive structure).** **13**

GITHUB REPOSITORY 14

SUPPLEMENTARY FIGURE LEGENDS. 15

SUPPLEMENTARY TABLE LEGENDS. 19

REFERENCES 21

## Supplementary Methods

### *Datasets.*

Eight paired multi-omic single-cell data sets were used to develop and test the performance and quality of LIBRA. Additional information is provided in Fig.3C.

DataSet1, SNARE-seq (Chen et al., 2019):

- GSE126074.
- Data modalities: single-cell RNA-seq and single-cell ATAC-seq.
- Single-cell nucleus.
- Cells selected: Mouse Adult Cerebral Cortex
- The total number of cells before filtering: 10,309 nuclei.
- The total number of cells after filtering: 6,735 nuclei.

DataSet2, CITE-seq (Hao et al., 2020; Stoeckius et al., 2017):

- GSE128639.
- Data modalities: single-cell RNA-seq and ADT panel for 25 antibodies.
- Cells selected: Human Bone Marrow cells
- The total number of cells before filtering: 33,454 cells.
- The total number of cells after filtering: 30,672 cells.

DataSet3, Paired-seq (Zhu et al., 2019):

- GSE130399.
- Data modalities: single-cell RNA-seq and single-cell ATAC-seq.
- Cells selected: Mouse Adult Cerebral Cortex
- The total number of cells before filtering: 11,544 cells.
- The total number of cells after filtering: 7,743 cells.

DataSet4, SHARE-seq (Ma et al., 2020):

- GSE140203.
- Data modalities: single-cell RNA-seq and single-cell ATAC-seq.
- Single-cell nucleus.
- Cells selected: Mouse Skin
- The total number of cells before filtering: 42,948 cells.

- The total number of cells after filtering: 34,774 cells.

DataSet5, 10X Multiome:

- 10X Genomics website repository.
- Data modalities: 10X Genomics single-cell RNA-seq and single-cell ATAC-seq.
- Single-cell nucleus.
- Cells selected: Human PBMC
- The total number of cells before filtering: 11,709 cells.
- The total number of cells after filtering: 10,412 cells.

DataSet6, 10X Multiome:

- GSE194122
- Data modalities: 10X Genomics single-cell RNA-seq and single-cell ATAC-seq.
- Cells selected: Bone Marrow
- The total number of cells after filtering: 42,492 cells.

DataSet7, CITE-seq:

- GSE194122
- Data modalities: single-cell RNA-seq and ADT panel for 134 antibodies.
- Cells selected: Bone Marrow.
- The total number of cells after filtering: 66,175 cells.

DataSet8, scNMT-seq (Clark et al., 2018):

- GSE109262.
- Data modalities: single-cell RNA-seq, single-cell ATAC-seq and single-cell DNA Methylation.
- Single-cell nucleus.
- Cells selected: Mouse Embryonic Stem Cells.
- The total number of cells before filtering: 100 cells.
- The total number of cells after filtering: 69 cells.

Importantly, during the pre-processing and after the quality filter, the total number of cells used in the analysis was reduced. See Table S4 for the final total number of cells used for all the analysis.

### *Pre-analysis and data processing:*

DataSet1 (Detailed example, similar pipeline for paired RNA and ATAC data-sets):

*Quality filter – low-quality features*: removing low-quality features and cells for both modalities. All cells with an overall abundance level of "*number of features per cell*" and "*number of counts per cell*" lower than quantile 0.1 or higher than quantile 0.9 were removed before the downstream analysis. Minimum abundance filtering was applied for mRNA (ATAC): genes (peaks) profiled in less than 4 cells (3 cells), and cells will lower than 201 genes quantified were filtered. No minimum of peaks per cell was required. After quality and abundance filtering, we selected a total of 8,086 cells for scRNA and 8,214 cells for scATAC adult samples.

*ATAC-derived gene activity*: when ATAC-derived gene activity was computed, the Seurat3(Stuart et al., 2019) 'CreateGeneActivityMatrix' function was used with parameter "upstream=2000" bases. GRC 38 reference genome was used.

*Quality filter – mitochondrial:* cells with more than 5% mitochondrial reads were filtered in scRNA-seq analysis. In scATAC gene activity derived from peaks was used for filtering.

*Variable feature selection*: For variable features selection in scATAC the function 'FindTopFeatures' from Signac(Stuart et al., 2020) library was applied with cut-offs of q0 and q5 (later selected q0 based on higher performance results). The top 2,000 most variable genes (*mvg*) were selected for scRNA. It is relevant to mention that we repeated the analysis with mvg=3,000 with no improvement in performance – data not shown.

*Component parameters*: A total of 15 principal components (PCA) were selected for scRNA dimension reduction, and 50 latent semantic indexing components (LSI) were selected for scATAC.

*The final number of cells*: from the resulting pipeline, a total of 6,735 paired cell profiles were considered for the downstream analysis.

*Integration*: To generate a reference integrated version of this dataset Seurat 3 (unpaired) and Seurat 4 (paired) has been used **over the independent modalities** for further processing and later integration. Using standard normalization and integration guides for both Seurat3 and Seurat4 Weighted Nearest Neighbour Analysis vignette (Hao et al., 2021). In Seurat3, *FindTransferAnchors* function using RNA as a reference and ATAC as query modalities employing CCA as reduction method was used for the anchorset generation, followed by *TransferData* function where anchorset generated was used for transfer into ATAC modality the RNA derived information using LSI dimensional reduction for the weighting anchors. Finally, data was merged for combined dataset generation. *FindMultimodalNeighbors* function from Seurat4 was used for identifying multimodal neighbours.

Specifics for DataSet2:

- We followed the same pre-processing for the scRNA-seq data analysis as in DataSet1.
- ADT dataset was analysed following Seurat4 Weighted Nearest Neighbour Analysis vignette (Hao et al., 2021). The exact same functions and parameters where used.

Specifics for DataSet6 and DataSet7:

- Data has been used as provided in Open Problems in Single-Cell Analysis NeuRIPScompetition (2021). Data is already filtered based on quality controls and normalized using Seurat standard pipeline.

Specifics for DataSet8:

- We followed the same pre-processing for the scRNA-seq and scATAC-seq data analysis as in DataSet1.
- For DNA methylation analysis first, a summarized matrix containing unique CpG positions identified in at least a single-cell as rows and all the cells as columns was generated and filled with obtained percentages of methylation for each corresponding cell and position. Next, DNA methylation values were scaled from 0 to 1, and a binarization was conducted by replacing values greater or equal to 0.5 by 1 and the rest by 0.

*SCT uni-omics integration pre-processing.*

The impact of the normalization during data pre-processing has been tested using *sctransform* proposed by the Seurat team (SCT) in DataSet5 (PBMC cells). *SCTransform* function was used with default parameters in the RNA dataset, generating the resulting Pearson normalized residuals later exported for the LIBRA model training.

### *Visualization:*

UMAP dimensionality reduction method was applied for the generation of visualisations over reduced components generated for either independent or integrated modalities. UMAP 2-dimensional reduced space was computed from corresponding low dimensional spaces: RNA(pca), ADT(pca), ATAC(lsi) and DNA methylation(lsi). The implementation used was RunUMAP function from Seurat package.

### *Clustering for integrated and independent omic modalities:*

All the clustering analysis has been conducted using Seurat Louvain clustering implementation(Waltman & van Eck, 2013). Depending on the analysis, different inputs are considered:

- Single-cell RNA-seq data: PCA components.
- Single-cell ATAC-seq data: LSI components.
- ADT: PCA components.
- Single-cell DNA methylation: LSI components.
- Integrated space: cells were clustered based on shared components generated by the methods considered (LIBRA, Seurat4, MOFA+(Argelaguet et al., 2020), totalVI(Gayoso et al., 2021) and BABEL(Wu et al., 2021)). The components used over these shared spaces were 10, 15, 15, 10 and 16, respectively.

Louvain resolution has been fixed to default 0.8 value for integrated subspaces; the starting seed has been computed over 1,000 times to exclude spurious clusters that can appear because of an arbitrary starting seed selection for Louvain clustering. The number of nearest neighbours was K=30 (default) but optimized using a *snakemake workflow* [1] which is based on subsampling and repeated clustering using *scclusteval*(Tang et al., 2020) as a reference. The bootstrap parameters used: subsample rate of 0.8 for a total of 20 subsamples using values of k from 8 to 16 (in steps of +2) and subsample resolutions from 0.6 to 1.4 (in steps of +0.2) generating a total of 500 analysis.

### *Overview of the Quality Measures.*

To evaluate the quality of LIBRA, we designed several quality metrics. The first metric measures the quality of the first NN (**referred to as Q1, and computed as MSE**), while the second, **Q2**, measures the accuracy of the dt2 projection (Fig.2B) using an Euclidean distance (which is the square root of the MSE used during training). When evaluating Q2 per-cell, we identified a bimodal distribution of the Euclidean estimated distances per

[1] https://github.com/crazyhottommy/pyflow_seurat_parameter

cell; such distribution was associated with whether the cells were part of the training or not (Fig.S1C); thus, both distributions of the bimodal were evaluated separately.

To evaluate the additional value within the shared latent space (SLS), we designed the **Preserved Pairwise Jaccard Index** (PPJI), a non-symmetric distance metric between two clusterings. PPJI co pares clusters derived from single-omics and multi-omics profiling (Fig.2B). For instance, when comparing RNA derived clustering (A), and CPS derived clusterings (B), the PPJI computes the Jaccard Index between every pair of clusters from A and B and summarizes the value (e.g., sum) over the clusters of B (Fig.2C). Unless the two clusterings have the same number of clusters, the outcome will not be symmetric. PPJI is aimed to measure how well the CPS projection provides a finer granularity than a single-omics.

Using these three quality measurements (Q1, Q2, PPJI) and the SNARE-seq (Chen et al., 2019) adult brain mouse dataset, we selected the hyperparameters for LIBRA (see details in the "Default Model Development Section" and Table S1):

(i) Autoencoder-type configuration,
(ii) The number of dimensions of the projected space,
(iii) Selection between peak and gene-activity derived information for ATAC-seq,
(iv) The assignation of dt1 and dt2,
(v) The features to be included (all vs most variable features (MVF)) and
(vi) The number of intermediate layers.

In all cases, we favoured first maximal PPJI values, then minimal Q2, and finally, minimal Q1 values. The LIBRA's selected parameters were: *AE-based framework, ten dimensions for the middle layer, dt1=ATAC > dt2=RNA, MVP features included, and two hidden layers* (see details in the "Default Model Development Section").

### *Model performance ranking*

To summarize the overall integration performance of generated models in a final score, a weighted average of the three metrics was calculated (see Table S2). Each time a set of combinations is compared, the results of training the AE 10 times for each combination are pooled. Then for Q1 (Q2), the maximum (*maxQ1*) and minimum (*minQ1*) are computed. For each value of *i,* a *snormQ1* (*snormQ2*) is computed as follows:

$$snormQ1(i) = \frac{Q1(i) - maxQ1}{minQ1 - maxQ1}$$

Higher values are associated with a lower error. For PPJI, considering the aim is to maximize the value, the values are computed as follows:

$$snormPPJI(i) = \frac{PPJI(i) - minPPJI}{maxPPJI - minPPJI}$$

The weighted combination (*score*) for network training *i* is computed as follows:

$$Score(i) = snormQ1(i) * 0{,}33 + snormQ2(i) * 0{,}33 + sminPPJI(i) * 0{,}33$$

Larger *score* values denote a better performance, generally associated with lower Q1 and Q2, and higher PPJI values.

### *Supplementary NN2 (mapper).*

This Neural Network maps *dt2* to the generated SLS on the NN1

$$h'' = \sigma'(\mathrm{W}'\mathrm{y} + \mathrm{b}')$$

Where $h''$ is the encoded SLS generated by NN1 and $\mathrm{y} \in \mathbb{R}^d$ denotes the output omic expression matrix.

### *Computing the Preserved Pairwise Jaccard Index (PPJI)*

Given a set of cells S and two clustering's A and B of the cells, Jaccard Similarity Index (JSI) is computed as follows for every pair of clusters i ∈ A and $j$ ∈ B ($a_i$ and b*j* denote respectively the set of cells in cluster *i* and *j*):

$$JSI(a_i, b_j) = \frac{(a_i \cap b_j)}{(a_i \cup b_j)}$$

The function *PairWiseJaccardSetsHeatmap* from scclusteval(Tang et al., 2020) package was used for Jaccard Similarity Index estimation. The Pairwise Jaccard Index is a matrix that computes JSI for all pairs of clusters *i* ∈ A and $j$ ∈ B.

When we aim to investigate how A clustering projects on B clustering, by using the matrix, for each cluster *i* ∈ A, the following statistic is computed:

$$S_i = \sum_{j \in B} JSI(a_i, b_j)$$

Finally, PPJI is computed as follows, where *I* is the total number of clusters in A:

$$\mathbf{PPIJ} = \frac{\mathbf{1}}{\boldsymbol{I}} \sum \boldsymbol{S}_{\mathbf{i}}$$

It is important to note that PPJI is not a symmetric measure, especially in the case where the number of clusters is different.

### *Q1 and Q2 evaluation.*

Mean square error (MSE) is used as a loss function in the training of networks NN1 and NN2. For instance, in the case of NN1:

$$Q1 = MSE_{NN1} = \sum_{i=1}^{n} (dt_{2,i} - \widehat{dt}_{2,i})^2$$

Where $\widehat{dt}$ denotes the estimated value, and *dt* denotes the original value. *n* is the total number of cells. Q1 is reported as two values:

- Q1 training, the MSE over the training set of cells, when computed for all cells.
- Q1 test, the MSE value when applied to all test cells. Test cells are a 20% of the original number of cells.

And, while NN2 is trained using MSE as a loss function, the Q2 returned is the root square of MSE when computed for all cells, which effectively is the Euclidian distance between predicted value and expected value (for the mapping of dt2 in the shared space).

$$Q2 = \sqrt[2]{MSE_{NN2}}$$

Q2 is computed separately for two different set of cells, due to the observation that Q2 follows a bimodal distribution (Fig.1S(c)) statistics are computed for each of the two distributions identified in the bimodal.

### *PredRNA*

RNA prediction was obtained by first loading hdf5 format stored trained LIBRA model from ATAC(dt1) to RNA(dt2), then *predict* Keras function was used for getting output from output layer (RNA dt2 like) from ATAC(dt1) input data as parameters.

Evaluation for quality check was performed by computing Pearson correlation between each pair of cells from predicted RNA and original RNA training input data. This computation was performed using *cor* function from *stats* package.

### *PredATAC*

ATAC prediction was obtained by first loading hdf5 format stored trained LIBRA model from RNA(dt1) to ATAC(dt2), then *predict* Keras function was used for getting output from output layer (ATAC dt2 like) from RNA(dt1) input data as parameters.

Evaluation for quality check was performed by computing AUC-ROC between original ATAC training input data (dt2) and predicted ATAC. Before ROC computation, matrices where transform into binarized numeric vectors. The threshold employed for binarization is 0.25. This computation was performed using *roc and auc* functions from *pROC* package.

### *Default model development.*

Computing scores: Neural networks have a certain degree of variability while learning a model; therefore, the results shown have been generated using the mean of 10 trained networks for the case of LIBRA, Autoencoder, Variational Autoencoder or Beta Variational Autoencoder. The highest performance scores in each set of 10 trained models have been highlighted in appropriate cases.

Hyperparameter selection: In order to generate the best configuration for LIBRA according to the nature of the single-cell data, a series of analyses have been conducted using DataSet1. Among them, hyperparameters, layers, number of nodes, activation function used, or optimizer applied has been tested over a basic structure presented in Table S1; as a result, the final framework is shown in Fig.S1A,B.

(i) *Autoencoder-type configuration* ("Selection of Autoencoder Model" table): several neural network models have been investigated. Results show that models based on distribution fits such as VAE or BVAE provide a lower model quality (higher values of Q1 and Q2) but also return worse granularity on cell sub-type identification (lower values of PPJI). As a result, we selected the AE architecture for the NN1 model.

(ii) *The number of dimensions of the projected space* ("Selection of # of middle layer neurons" table): we observed that for a paired sample of about 5,000 to 10,000 cells the highest performance in biological preservation (PPJI) was obtained when considered 5 to 10 dimensions. Such selection has as an additional trade-off: a worse model quality (high Q1) but better dt2 encoding (lower Q2). We selected ten as the number of dimensions in the middle layer.

(iii) *Selection between peak and gene-activity derived information for ATAC-seq*: we observed that the use of the original peaks matrix information from scATAC outperforms the use of the transformed activity expression when considering PPJI (cell sub-type identification) and model quality (Q1). The trade-off is a worse dt2 encoding (Q2).

(iv) *The assignation of dt1 and dt2* ("Selection of the order" table): we observed that setting dt1=ATAC and dt2=RNA returns better PPJI (cell sub-type identification)

and model quality (Q1); as a trade-off dt2 encoding (Q2) is worse. We consider that an explanation for this is the different levels of granularity derived from scRNA and scATAC.

(v) *The features to be included* ("Selection of features to include: all vs most variable" table): the use of only the MVF shows a better model quality (Q1) and dt2 encoding (Q2). When comparing cell sub-type identification (PPJI), we observed that on average "all features" has better results, but the "MVP" returns the best model from 10 runs. Signac library has been used for selecting among different percentages of most variable peaks obtaining a worse performance (data not shown).

(vi) *The number of intermediate layers*: we compared using one or two hidden layers. When using 2 hidden layers, we observed improvement for all quality measures Q1, Q2 and PPJI.

### *LIBRA structure and additional hyperparameters.*

LIBRA neural networks were implemented by using Keras R and Python packages. The final structure used for LIBRA consists of two neural networks; the primary/first one (NN1) implemented to identify the shared latent space (SLS). The second one (NN2) is used for generating a mapping (encoding) of dt2 into the SLS modalities. NN1 contains two parts the encoder and decoder ones while NN2 only includes an encoder. The characteristics of the networks and encoding are:

- The number of layers and nodes used for each of these networks are shown in Fig.S1(a,b).
- Because of the sparse nature of single-cell data, Leaky ReLU has been used as an activation function to avoid zero components in the middle layer, making middle layer size an attribute that can be modified based on the number of paired cells used on the data.
- Adam has been used as an optimizer, and the loss function used is the mean squared error (MSE). Adam optimizer has been configured with; 0.001 learning rate, 0.9 for beta_1 and 0.999 for beta_2 values. The values for epsilon was set to the default, i.e., 0.001 and decay to 0.
- A dropout rate of 0.2 has been used for all activation functions except for the last to the output layer where no dropout is used.
- The maximum number of epochs was set at 1.500, but an early stopping rule with a patience value of 20 over loss score was used, causing the learning process to stop if no improvement was obtained after 20 epochs. We used a large batch size of 7000 to speed up computations on CPU-based systems (limited parallelization capabilities), while smaller datasets will use the maximum available samples; otherwise, BABEL has been fixed to 512 batch size, because it was trained on a GPU (NVIDIA Tesla V100 with an excellent parallelization capabilities) reducing epoch training time and the number of epochs required to achieve the convergence.

- Learning rate plateau callback was added with a factor value of 0.1, starting from 0.001 and updating it with a patience of 15 until reaching a learning rate of 0.00001.

When using LIBRA, the following applies:

- The components generated in the middle layer of the main LIBRAS's neural network (NN1) were used for the Louvain clustering application and, therefore, cluster identification. The resolution used is 0.8 and k=30, in the same way as before the clustering was repeated over 1,000 different combinations of starting points to have a robust clustering outcome.
- UMAP dimensionality reduction method was applied for visualization generation over the middle layer obtained components.

(vii) Q1, Q2 and PPJI.

### *LIBRA fine-tune (deep adaptive structure).*

We change the static default model to a dynamic one based on the data. To do this, we perform an exhaustive search using a grid of hyperparameters ranging as: 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 integration task and 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].

To increase the computation speed, it has been implemented in parallel to train multiple models at the same time.

A convex decreasing curve was used for number of nodes assignation to layers to avoid reduce it linearly based on number of layers used. This was done dividing the maximum number of nodes used by a factor of “2 * layer position” for a given layer; Example: for a 2048 number of nodes hyperparameter having 6 layers nodes of encoder will by assigned as, 2048 (1º intermediate layer), 1024 (2º intermediate layer), 512 (3º intermediate layer), 341 (4º intermediate layer), 256 (5º intermediate layer) and 204 (6º intermediate layer). By doing this we allow a high increase of the layers without the need to assign an extremely high value in the first layers so as not to suffer from a negligible value of nodes in the last ones, which would happen if we used a linear decrement Example: dividing by 2 based on previous layer size.

In our tests python implementation of LIBRA has shown to be able to train the model significantly faster than in R outcomes obtained, python version requires less resources (RAM). For this reason, it has been implemented in both to extend access to users and take advantage of python properties.

### *Parameters for the integrative tools:*

BABEL(Wu et al., 2021): We used the implementation from https://github.com/wukevin/babel with default parameters for all datasets. Same normalized data input used for the training of the rest of the models was used as input, generating a BABEL shared space of 16 components. Adam optimizer was used, with an initial learning rate of 0.01 and a batch size of 512 were set by default. Shared space generated by BABEL was used for clustering computation and PPJI estimation. NVIDIA Tesla V100 GPU with 32 GB RAM was used for the training and inference of BABEL on all datasets.

totalVI(Gayoso et al., 2021): The publicly available implementation from totalVI at https://docs.scvi-tools.org/en/stable/user_guide/notebooks/totalVI.html has been used. Default model values have been selected generating a shared space of a total of 10 components. Following the tool guidelines, count data from already filtered datasets (same input for all models) has been used as an input for totalVI by using also the 2.000 mvg for transcriptomic data. In order to compare the integration quality with the rest of the models, shared space generated has been used for clustering and PPJI computation.

MOFA+(Argelaguet et al., 2020): As for the rest of the models tested, corresponding guidelines present at https://github.com/bioFAM/MOFA have been used. A maximum of 15 factors has been generated in the shared space using MOFA+ for the different datasets employed. MOFA+ reduced factors' space has later been used for integration performance measurement by using PPJI metric after clustering computation over this space.

### *CITE-seq additional quality measurement*

To estimate the level of integration quality for DataSet2, CITE-seq-GSE128639, an additional quality metric has been investigated.

First the nearest cells to each cell have been obtained by applying the k-nearest neighbours algorithm (k-NN) with k=20 over the integrated shared space components from the corresponding model applied (LIBRA, Seurat4, MOFA+, totalVI and BABEL). Then, the mean of the expression values of the 20 cells closest to each cell for each of the 25 proteins was calculated and next the Spearman correlation and Pearson correlation were computed between these generated values for each of the 25 proteins and the original protein expression values.

## GitHub Repository

The code to reproduce the generation of LIBRA models and the entire analysis described in the manuscript is available at:

https://github.com/TranslationalBioinformaticsUnit/LIBRA.

It includes a Jupyter notebook and a RMarkDown file.

## Supplementary Figure Legends.

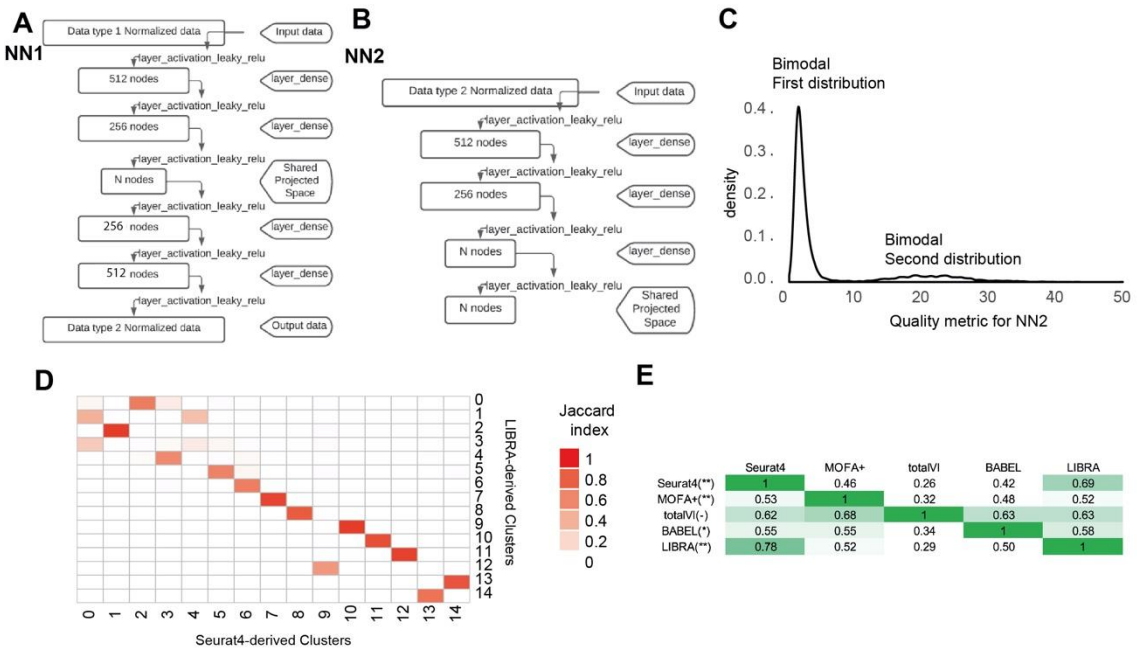


**Figure S1. LIBRA optimization using DS1. A.** Fina c nfiguration of NN1 in LIBRA. **B.** Final configuration of NN2 in LIBRA. **C**. Exam le of bimodal distribution identified when computing Q2 values per cells. Q2 is associated with NN2 quality. **D**. Pairwise Jaccard Index between the clusterings derived from Seurat4 and LIBRA. **E**. Heatmap. Color and values denote the PPJI distance between the clusters generated using the integrated space of each methodology.

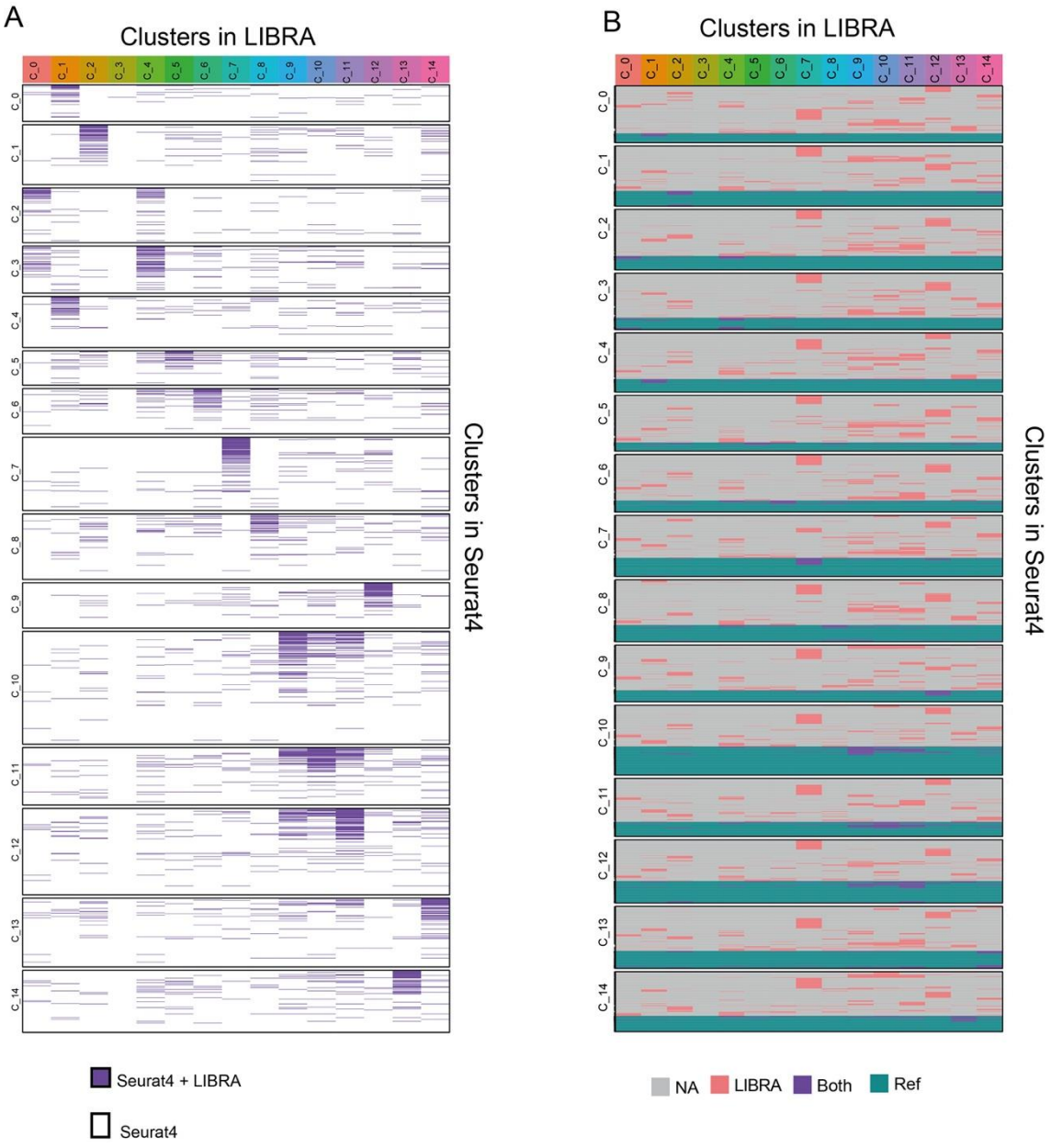


**Figure S2. LIBRA a d the identification of additional markers.** A. For each cluster identified in Seurat4 (rows), each line denotes a gene and purple denotes if LIBRA also identifies the gene. B. For each cluster identified in Seurat4 (rows), each line denotes a gene and, if it is identified in the Seurat4 cluster, by LIBRA or by both.

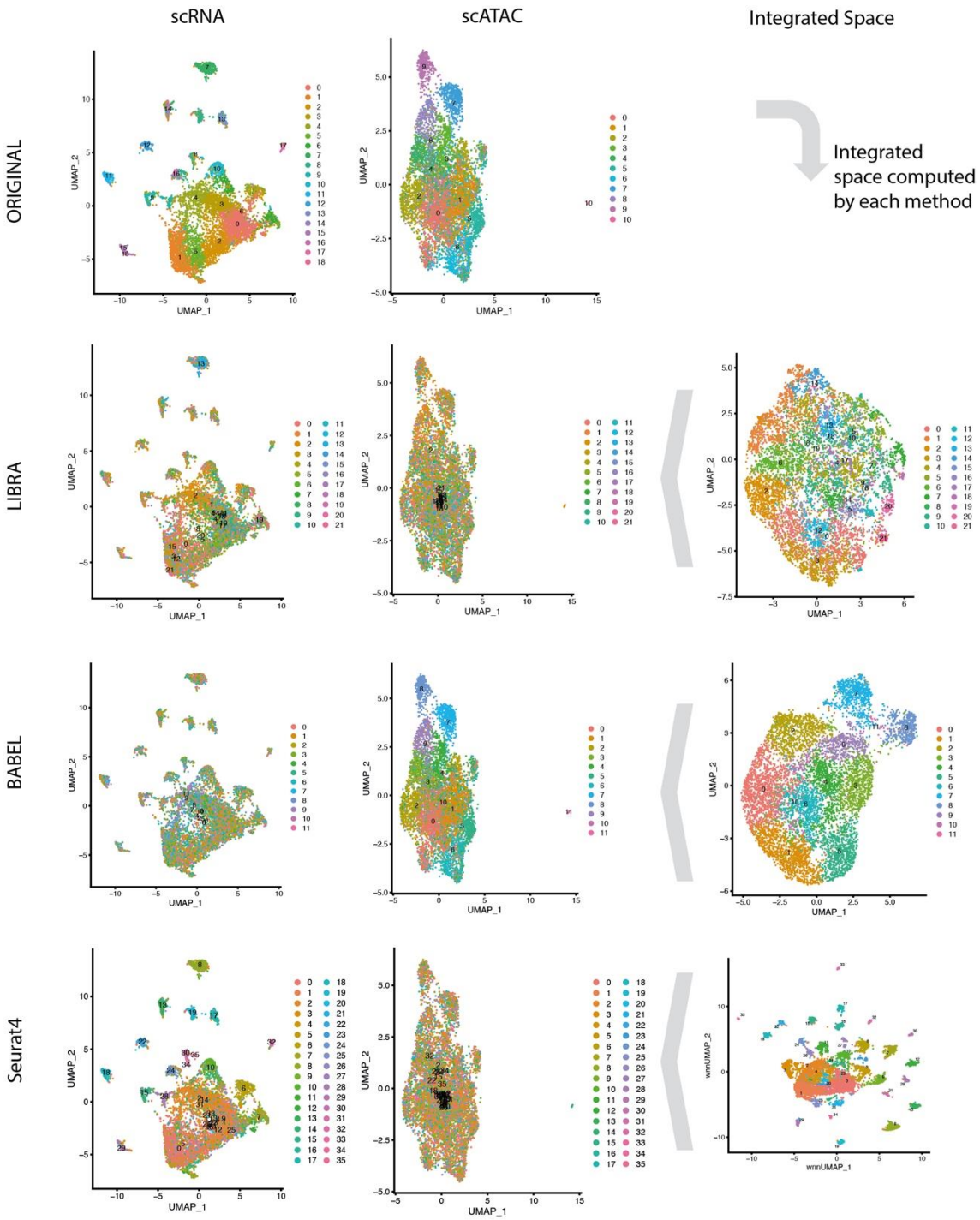


**Figure S3. Integra ive analysis in DS4 for Seurar4, BABEL and LIBRA.** Two upper panels to denote the UMAP projection and their clustering for RNA and ATAC, respectively. Then, the following rows are associated to the integrative analysis for the different methods: LIBRA, BABEL and Seurat4. In each row, the right panel shows the integration-based clustering, and the two left panels show the integrated based clustering “projected” into the RNA and ATAC UMAPS, respectively.

A

| | multigrate | totalVI | multiVI | LIBRA | aLIBRA | LIBRA-scib-best |
|---|---|---|---|---|---|---|
| PPJI_RNA | 0.61 | 0.69 | 0.76 | 0.79 | 0.82 | 0.54 |
| PPJI_ATAC | 0.66 | 0.70 | 0.68 | 0.67 | 0.81 | 0.53 |
| Average | 0.63 | 0.70 | 0.72 | 0.73 | 0.82 | 0.54 |
| Average(scib) | 0.62 | 0.73 | 0.71 | 0.69 | 0.59 | 0.73 |

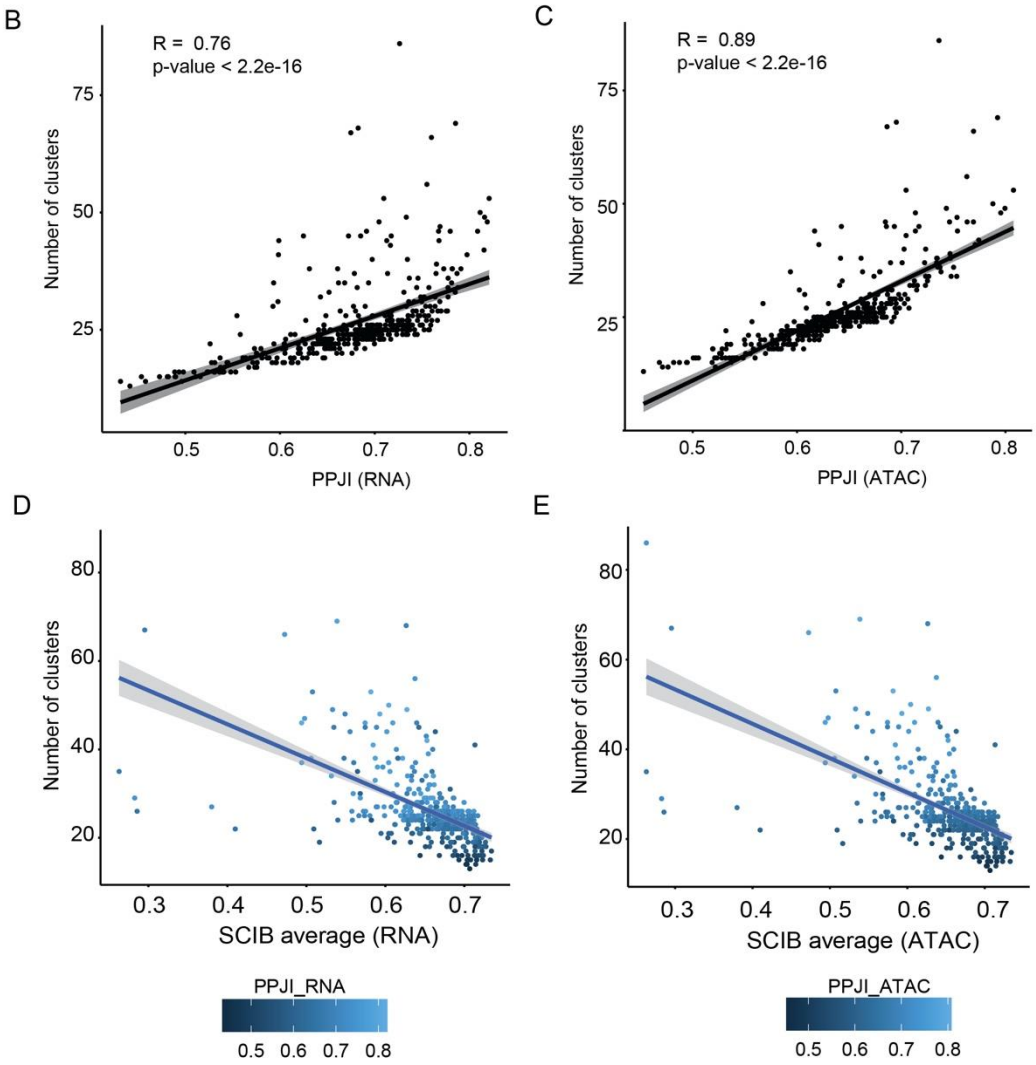


**Figure S4. Future venues of research for aLIBRA.** A. Evaluation metrics according to latent spaces obtained with multigrate, totalVI, multiVI, LIBRA (default), and aLIBRA based on the PPJI metric. Last column shows the aLIBRA PPJI values for the best models derived from scib average based optimization. B. PPJI RNA vs number of clusters. Each dot corresponds to a model trained over DS6 during aLIBRA fine tuning. R denot s Spearman correlation. C. Same as B for PPJI_ATAC. D. SCIB average RNA vs number of clusters. Each dot corresponds to a model trained over DS6 during aLIBRA fine tuning using the SCIB average as the optimization. R denotes Spearman correlation. The gra ient shows the PPJI_RNA value of the model. E. Same as D for SCIB_ATAC.

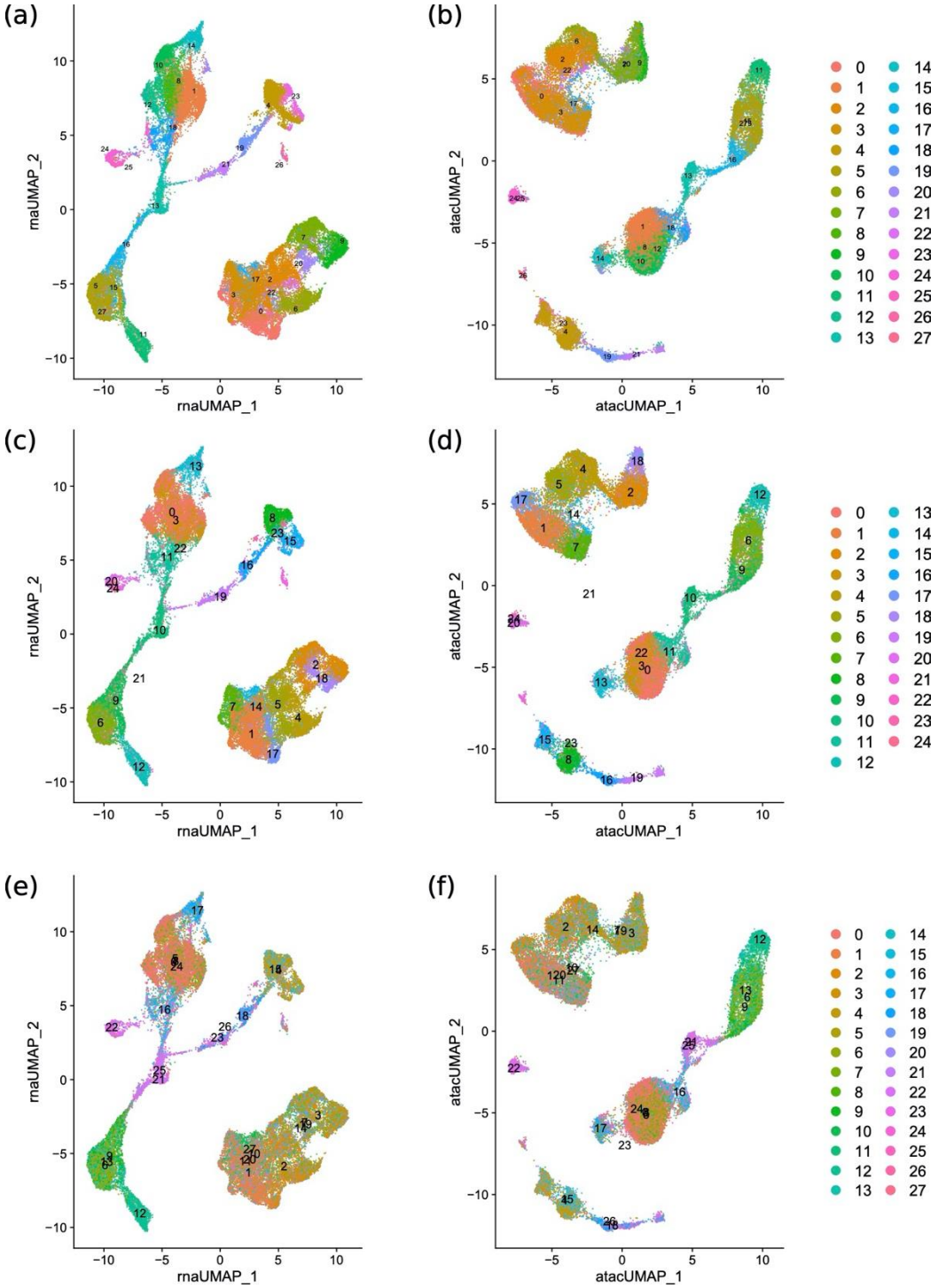


**Figure S5. Extended benchmarking over DS6.** RNA (le t) and ATAC (right) derived UMAP projections of DS6 dataset. A, B co respond to LIBRA(default) integration-based derived clusters. C, D co respond to MultiVI integration-based derived clusters. E,F correspond to multitegra integration-based derived clusters. The cells have been coloured using the cluster assignment obtained in the integrated space generated by each of the methods with default values.

## Supplementary Table Legends.

**Table S1. Fine-tuning of LIBRA using DS1: all measures.** Several sub-tables showing Q1, Q2 and PPJI used for the selection of the hyperparameters of LIBRA. In gree , the selected version in each case is highlighted. Selecti ns made in upper tables are applied to lower tables. In a la ge majority of the cases we identified an increased number of clusters in the integrated space.

**Table S2. Fine-tuning of LIBRA using DS1: score.** Scores for each training for each of the fine-steps. Same ordering as table S1, but introducing the weighted *score* for each training. In green the top values are provided.

**Table S3. Robustness and Sensitivity analysis in DS1.** Tables showing Q1, Q2 and PPJI used for the sensitivity and robustness analysis of LIBRA. (a) Qua ity analysis when pairing information is not correct for different percentages of cells. (b) Quality analysis when dropout is added. (c) Quality analysis for different number of epochs in the training of the model. In a large majority of the cases we identified an increased number of clusters in the integrated space.

**Table S4**. **Number of cells per dataset.**

**Table S5. PPJI values between Seurat 4 and LIBRA comparing the two strategies for normalization.** SCT stands for the sctTransform normalization procedure implemented in Seurat package.

**Table S6**. **Predict on-based comparison between BABEL and LIBRA.** (*) 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 are required for BABEL.

**Table S7. predRNA values for all cells and per cluster in DataSet1**. Rho and p-value denotes the Spearman correlation and significance between the number of cells and *predRNA* computed.

**Table S8**. **predRNA values for all cells and per cluster in DataSet5**. Rho and p-value denote the Spearman correlation and significance between the number of cells and *predRNA* computed.

## REFERENCES


Argelaguet, R., Arnol, D., Bredikhin, D., Deloro, Y., Velten, B., Marioni, J. C., & Stegle, O. (2020). MOFA+: A statistical framework for comprehensive integration of multi-modal single-cell data. *Genome iology*, *21*(1), 1–17.

Chen, S., Lake, B. B., & Zhang, K. (2019). High-throughput sequencing of the transcriptome and chromatin accessibility in the same cell. *Nature iotechnology*, *37*(12), 1452–1457.

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., Stegle, O., & Reik, W. (2018). ScNMT-s q enables joint profiling of chromatin accessibility DNA methylation and transcription in single cells e. *Nature ommunications*, *9*(1), 1–9.

Gayoso, A., Steier, Z., Lopez, R., Regier, J., Nazor, K. L., Streets, A., & Yosef, N. (2021). Joint probabilistic modeling of single-cell multi-omic data with totalVI. *Nature Methods*, *18*(3), 272–282.

Hao, Y., Hao, S., Andersen-Nissen, E., Mauck, W. M., Zheng, S., Butler, A., Lee, M. J., Wilk, A. J., Darby, C., Zagar, M., Hoffman, P., Stoeckius, M., Papalexi, E., Mimitou, E. P., Jain, J., Srivastava, A., Stuart, T., Fleming, L. B., Yeung, B., … Satija, R. (2020). Integra ed analysis of multimodal single-cell data. *BioRxiv* 2020.10.12.335331.

Hao, Y., Hao, S., Andersen-Nissen, E., Mauck, W. M., Zheng, S., Butler, A., Lee, M. J., Wilk, A. J., Darby, C., Zager, M., Hoffman, P., Stoeckius, M., Papalexi, E., Mimitou, E. P., Jain, J., Srivastava, A., Stuart, T., Fleming, L. M., Yeung, B., … Satija, R. (2021). Integra ed analysis of multimodal single-cell data. *Cell*, *1 4*(13), 3573-3587.e29.

Ma, S., Zhang, B., LaFave, L. M., Earl, A. S., Chiang, Z., Hu, Y., Ding, J., Brack, A., Kartha, V. K., Tay, T., Law, T., Lareau, C., Hsu, Y. C., Regev, A., & Buenrostro, J. D. (2020). Chromat n Potential Identified by Shared Single-Cell Profiling of RNA and Chromatin. *Cell*, *1 3*(4), 1103-1116.e20.

Stoeckius, M., Hafemeister, C., Stephenson, W., Houck-Loomis, B., Chattopadhyay, P. K., Swerdlow, H., Satija, R., & Smibert, P. (2017). Simultaneous epitope and transcriptome measurement in single cells. *Nature ethods*, *14*(9), 865–868.

Stuart, T., Butler, A., Hoffman, P., Hafemeister, C., Papalexi, E., Mauck, W. M., Hao, Y., Stoeckius, M., Smibert, P., & Satija, R. (2019). Comprehensive Integration of Single-Cell Data. *Cell*, *1 7*(7), 1888-1902.e21.

Stuart, T., Srivastava, A., Lareau, C., & Satija, R. (2020). Multimodal single-cell chromatin analysis with Signac. *BioRxiv* 2020.11.09.373613.

Tang, M., Kaymaz, Y., Logeman, B. L., Eichhorn, S., Liang, Z. S., Dulac, C., & Sackton, T. B. (2020). Evaluating single-cell cluster stability using the Jaccard similarity index. *Bioinfo matics*.

Waltman, L., & van Eck, N. J. (2013). A smart local moving algorithm for large-scale modularity-based community detection. *The Eur pean Physical Journal B*, *86*(11), 471.

Wu, K. E., Yost, K. E., Chang, H. Y., & Zou, J. (2021). BABEL e ables cross-modality translation between multiomic profiles at single-cell resolution. *Proceed ngs of the National Academy of Sciences*, *118*(15), e2023070118.

Zhu, C., Yu, M., Huang, H., Juric, I., Abnousi, A., Hu, R., Lucero, J., Behrens, M. M., Hu, M., & Ren, B. (2019). An ultra high-throughput method for single-cell joint analysis of open chromatin and transcriptome. *Nature  tructural and Molecular Biology*, *26*(11), 1063–1070.