scAEspy: a unifying tool based on autoencoders for the analysis of single-cell RNA sequencing data

Andrea Tangherloni[1,2,3,✉], **Federico Ricciuti**[4], **Daniela Besozzi**[4], **Pietro Liò**[5,†,✉], **and Ana Cvejic**[1,2,3,†,✉]

[1]Wellcome Trust–Medical Research Council Cambridge Stem Cell Institute, Cambridge, UK
[2]Department of Haematology, University of Cambridge, Cambridge, UK
[3]Wellcome Trust Sanger Institute, Wellcome Trust Genome Campus, Hinxton, UK
[4]Department of Informatics, Systems and Communication, University of Milano-Bicocca, Milan, Italy
[5]Department of Computer Science and Technology, University of Cambridge, Cambridge, UK
[†]These authors contributed equally to this work

Autoencoders (AEs) have been effectively used to capture the non-linearities among gene interactions of single-cell RNA sequencing (scRNA-Seq) data. However, their integration with the common scRNA-Seq bioinformatics pipelines still poses a challenge. Here, we introduce scAEspy, a unifying tool that embodies five of the most advanced AEs and different loss functions, including two novel AEs that we developed. scAEspy allows the integration of data generated using different scRNA-Seq platforms. We benchmarked scAEspy against principal component analysis (PCA) on five public datasets, showing that our new AEs outperform the existing solutions, achieving more than 20% increase of the Rand Index in the identification of cell clusters.

Autoencoders | scRNA-Seq | Dimensionality reduction | Clustering | Batch correction | Data Visualisation

Correspondence: *at860@cam.ac.uk pl219@cam.ac.uk as889@cam.ac.uk*

Background

Single-cell RNA sequencing (scRNA-Seq) is becoming increasingly prevalent in studies aimed at understanding the molecular processes that drive normal development and the onset of pathologies [1, 2]. One of the most important steps in scRNA-Seq analysis is the clustering of the cells into groups that correspond to both known and putatively novel cell types, by considering the expression of common sets of signature genes. However, this still remains a challenging task because applying clustering approaches in high-dimensional space can generate misleading results since the distance between most pairs of points is similar. As a consequence, finding an effective and efficient low-dimensional representation of the data is crucial. A common workflow of scRNA-Seq downstream analysis (Fig. 1) includes two dimensionality reduction steps: (1) the principal component analysis (PCA) [3] for an initial reduction of the dimensions based on the highly variable genes (HVGs), and (2) a non-linear dimensionality approach (e.g., t-distributed stochastic neighbour embedding (t-SNE) [4] or uniform manifold approximation and projection (UMAP) [5, 6]) on the PCA space for the visualization of the data (e.g., the labeled clusters) [7]. In addition, when multiple scRNA-Seq datasets are combined, the technical batch effects that might exist among the datasets must be taken into account [8–11].

Although commonly used approaches for dimensionality reduction achieved good performance when applied to scRNA-Seq data, novel and more robust dimensionality reduction strategies should be used to account for the intrinsic noise, unexpected dropout and burst effects [12], as well as the low amounts of RNA that are typically present in single cells. Autoencoders (AEs) showed outstanding performance in this regard [13–17] due to their ability to capture the strong non-linearities among the gene interactions that exist in the high-dimensional expression space. However, in the existing AEs the input data are usually codified in a specific format, making their integration into the existing scRNA-Seq analysis toolkits (e.g., SCANPY [18] and SEURAT [19]) difficult. The existing tools are implemented in Keras [20], TensorFlow [21] or PyTorch [22], and thus all the three libraries are required to run them; in addition, the currently available AEs cannot be directly exploited to obtain the latent space or to generate synthetic cells.

In order to overcome the described limitations, we developed scAEspy, which is a unifying, user-friendly and standalone tool that relies only on TensorFlow and allows easy access to different AEs by setting up only two user-defined parameters. Importantly, it implements different loss functions, which are fundamental to deal with the different sequencing platforms.

Results

scAEspy is comprised of five AEs, based on the variational AE (VAE) [23] and the maximum mean discrepancy (MMD) VAE [24]. Specifically, scAEspy contains the following most advanced AEs: VAE, Gaussian-mixture VAE (GMVAE), MMDVAE, and two novel Gaussian-mixture AEs that we developed, called GMMMD and GMMMDVAE. GMMMD is a modification of the MMDVAE where more than one Gaussian distribution is used to model different modes and only the MMD function is used as divergence function. On the contrary, GMMMDVAE is a combination of MMDVAE and GMVAE where both the MMD function [25] and the Kullback–Leibler divergence function [26] are used (for more details, see Methods section). scAEspy gives easy access to the latent space generated by the selected AE, which can be utilised to perform downstream analysis or to generate synthetic cells. Since scAEspy takes only a count matrix as input, it can be integrated into SCANPY and SEURAT pipelines.

We tested the performance of these five AEs when either multiple scRNA-Seq datasets or different scRNA-Seq platforms

DRAFT

are combined. Our analysis shows that GMMMD with the constrained negative binomial loss function is the best solution to combine datasets sequenced using droplet-based approaches. When plate-based methods are analysed, the Kullback–Leibler divergence function becomes fundamental in order to deal with the high values (up to millions) of gene expression. GMMMDVAE, coupled with the constrained Poisson loss function, shows outstanding performance in integrating multiple datasets generated using different scRNA-Seq platforms.

In all the tests, we used the MMDVAE by excluding the Kullback-Leibler divergence so that it can be directly compared to GMMMD. We modified the common workflow of scRNA-Seq downstream analysis by introducing scAEspy (Fig. 2). All the other steps can be performed by using both SCANPY and SEURAT toolkits.

AEs improve clustering of the cell types when multiple single-cell RNA-Seq datasets are combined. Various scRNA-Seq platforms are currently available (e.g., droplet-based and plate-based [27–36]) and their integration is often challenging due to the differences in biological sample batches as well as to the used experimental platforms. To test whether AEs can be effectively applied to combine multiple datasets generated using the same platform but under different experimental conditions, we used the publicly available datasets from peripheral blood mononuclear cells (PBMCs) (see Datasets section). The datasets include control (untreated) and treated (interferon-β stimulated) PBMCs.

We merged the control and treated datasets by using both the standard PCA and different AEs. After the construction of the neighbourhood graph, we performed the clustering by using the Leiden algorithm [37]. Since in the original paper 8 clusters were manually annotated for the PBMC datasets [38], we selected Leiden's resolutions that allowed us to obtain 8 distinct clusters and calculated the following metrics: Adjusted Rand Index (ARI), Adjusted Mutual Information Index (AMII), Fowlkes Mallows Index (FMI), Homogeneity Score (HS), Completeness Score (CS), and V-Measure (VM) (see Metrics section). In what follows, the calculated values of all metrics are given in percentages. For each metric, the higher the value the better the result.

We performed the same analysis by applying the batch balanced *k*-nearest neighbours (BBKNN) approach [9] to the PCA and latent spaces (Fig. 2). We selected BBKNN for the following reasons: (1) it has comparable or better performance in removing batch effect [9] with respect to the SEURAT implementation of the canonical correlation analysis (CCA) [8], Scanorama [11], and mnnCorrect [10]; (2) it is the fastest method considering both small and large datasets; (3) it is a lightweight graph alignment method that requires minimal changes to the classical workflow. BBKNN was used to modify the neighbourhood graph, built starting from the latent space.

Our analysis showed that the standard PCA has a mean RI equal to 67.90% (with standard deviation equal to 1.09%), while the PCA followed by BBKNN reached a mean RI of 67.82% ($\pm 1.56\%$). Among all the tested AEs, GMMMD (two Gaussian distributions and constrained negative binomial loss) outperformed the other AEs achieving a mean RI equal to 69.81% ($\pm 0.52\%$) without BBKNN, and equal to 85.27% ($\pm 1.43\%$) with BBKNN. Here, we used two Gaussian distributions because we merged two datasets. Regarding the AMII, the PCA achieved a mean value of 68.25% ($\pm 1.42\%$), while PCA followed by BBKNN achieved a mean value of 67.18% ($\pm 0.81\%$). GMMMD had better results, with mean values equal to 69.32% ($\pm 0.83\%$) without BBKNN, and 76.75% ($\pm 1.47\%$) when coupled with BBKNN. GMMMD obtained better results also in terms of FMS, HS, CS, and VS (see Table S2 and Fig. S1). The boxplots showing the distributions of the metrics obtained by applying PCA and all the AEs are depicted in Figs. S11-20.

In order to visually assess the quality of the sample alignment, we reduced the HVG space by either PCA or GMMMD and applied the BBKNN approach on the neighbourhood graphs. The 2D t-SNEs, as well as UMAPs, clearly show that the two samples were better aligned in the GMMMD space (Figs. 3A, 4A, S2A, and S3A). Next, we visually compared the distribution of the manually annotated clusters (Figs. 3B, 4B, S2B, and S3B) and the clusters obtained by applying the Leiden algorithm (Figs. 3C, 4C, S2C, and S3C). Our analysis showed that clustering the modified neighbourhood graph generated from GMMMD space allowed a better identification of the existing cell types when compared to PCA, thus confirming the RI results. We annotated the obtained clusters by calculating the marker genes per cluster using the Wilcoxon test [39] with Bonferroni correction [40]. The use of GMMMD resulted in more compact and homogeneous clusters as shown in the heatmaps (Figs. S4 (PCA) and S5 (GMMMD)). These results confirmed that the proposed GMMMD allowed a better identification of the cell types, thanks to its ability to capture the changes in the gene expression of cells in response to treatment.

AEs outperform linear approaches when different scRNA-Seq platforms are combined. Combining datasets from different studies and scRNA-Seq platforms can be a powerful approach to obtain complete information about the biological system under investigation. However, when datasets generated with different platforms are combined, the high variability in the gene expression matrices can obscure the existing biological relationships. For example, the gene expression values are much higher in data acquired with plate-based methods (i.e., up to millions) than in those acquired with droplet-based methods (i.e., a few thousands). Thus, combining gene expression data that spread across several orders of magnitude is a difficult task that cannot be tackled by using linear approaches. To examine how well AEs perform in resolving this task, we combined four different human pancreatic islet cell (PIC) datasets acquired with CEL-Seq [29], CEL-Seq2 [30], Fluidigm C1 [36], and Smart-Seq2 protocols [35] (see Datasets section).

In the first step, we integrated the datasets by using both the standard PCA and the different AEs. Since in the original paper 13 clusters were manually annotated for the PIC datasets [41], we clustered the neighbourhood graph using the Leiden

DRAFT

algorithm considering only the resolutions that allowed us to obtain 13 distinct clusters. We then calculated ARI, AMII, FMS, HS, CS, and VS metrics. We performed the same analysis by applying the BBKNN approach to the PCA and AE latent spaces. The calculated values of all metrics are given in percentages; for each metric, the higher the value the better the result.

PCA had a mean RI equal to 63.67% ($\pm 1.01\%$) without BBKNN, while the version with BBKNN achieved a mean RI of 60.67% ($\pm 0.61\%$). The best performance was achieved by GMMMDVAE (four Gaussian distributions and constrained Poisson loss), which reached a mean RI equal to 77.27% ($\pm 0.56\%$). Four Gaussian distributions were selected since we merged four datasets. When BBKNN was used the mean RI increased to 87.64% ($\pm 0.55\%$). Similar results were achieved for the AMII metric, where PCA had a mean RI equal to 72.01% ($\pm 0.67\%$) and PCA followed by BBKNN equal to 70.57% ($\pm 0.44\%$), whilse the mean RI obtained by GMMMDVAE is 71.86% ($\pm 0.52\%$) and by GMMMDVAE followed by BBKNN is 78.73% ($\pm 0.94\%$). GMMMDVAE outperformed PCA also in terms of FMS, CS, and VS (see Table S3 and Fig. S6). The distributions of the metrics obtained by applying PCA and all the AEs are shown in Figs. S21-30.

Next, we evaluated the sample alignment performed by BBKNN on the neighbourhood graphs obtained from both PCA and GMMMDVAE space. We visualised the cells (coloured by platform) using t-SNE and UMAP space (Figs. 5A, 6A, S7A, and S8A). In the t-SNE space, it is evident that GMMMDVAE performed a first alignment of the samples by distributing similar cells coming from different platforms in the four provided Gaussian distributions. The UMAPs showed that the BBKNN better removed the batch effects in the GMMMDVAE space (Figs. S7A, and S8A). We generated the clusters using the Leiden algorithm and the marker genes were inferred by the Wilcoxon test with Bonferroni correction. The visualisation of manually (Figs. 5B, 6B, S7B, and S8B) and automatically identified clusters (Figs. 5C, 6C, S7C, and S8C), in t-SNE and UMAP space, confirmed that the cell types were better annotated using GMMMDVAE. In line with this result, the heatmaps showing the expression of the top marker genes look more homogeneous when GMMMDVAE is used compared to PCA (Figs. S8 (PCA) and S9 (GMMMDVAE)). Taken together, our analysis shows that GMMMDVAE can efficiently identify the "shared" cell types across different platforms due to its ability to deal with the high variability in the gene expression matrices.

Conclusions

Non-linear approaches for dimensionality reduction can be effectively used to capture the non-linearities among the gene interactions that may exist in the high-dimensional expression space of scRNA-Seq data. Among the different non-linear approaches, AEs showed outstanding performance. Several AE-based methods have been developed so far, but their integration with the common single-cell toolkits is difficult because they usually require input data codified in a specific format. In addition, three different machine learning libraries are required to use them.

Here, we proposed scAEspy, a unifying and user-friendly tool that allows the user to use the most recent and promising AEs (i.e., VAE, MMDVAE, and GMVAE). We also designed and developed GMMMD and GMMMDVAE, two novel AEs that combine MMDVAE and GMVAE. We introduced a learnable prior distribution in the latent space to model the dimensionality of the subpopulations of cells composing the data or to combine multiple samples. The user can select the desired AE by setting up two user-defined parameters. Once the selected AE has been trained, it can be used to generate synthetic cells to increase the number of data for further downstream analyses (e.g. training classifiers). In scAEspy, the latent space is easily accessible and thus allows the user to perform different analyses, such as inference of differentiation trajectories. In this case, the latent space can be utilised to generate the "pseudotime" that measures transcriptional changes that a cell undergoes during the dynamic process. Thanks to its modularity, scAEspy can be extended to accommodate new AEs so that the user is able to utilise the latest and cutting-edge AEs [42], which can improve the downstream analysis of scRNA-seq data. Considering the achieved results on the identification of the clusters, scAEspy can be used at the basis of methods that aim to automatically identify the cell types composing the scRNA-seq datasets under analysis [43].

In addition, we integrated the AEs with BBKNN to remove the existing batch effects among different datasets. Our results showed that exploiting the latent space to remove the existing batch effects allows for a better identification of cells subpopulations. When different droplet-based data have to be combined, GMMMD coupled with BBKKNN using the constrained NB loss is the most effective method, with an $\sim 18\%$ increase of the ARI with respect to the standard PCA followed by BBKNN. In order to combine and analyse multiple datasets generated by using different scRNA-Seq platforms, our GMMMDVAE followed by BBKKNN and coupled with the constrained Poisson loss is the best solution. Indeed, the Kullback–Leibler divergence function is fundamental to handle data spreading various orders of magnitude, especially the high values introduced by plate-based methods. In such a case, our method increased the ARI of more than 25% compared to classic PCA followed by BBKNN.

Since AEs showed outsanding performance in the integration of multi-omics of cancer data [42], we plan to extend scAEspy to analyse other single cell omics. In that regard, scAEspy can be the starting point to build a more comprehensive toolkit designed to integrate multi single cell omics. For instance, AEs can be applied to analyse scATAC-seq, where the identification of the cell types is still more difficult due to technical challenges [44].

Datasets

Peripheral blood mononuclear cells. PBMCs from eight patients with systemic lupus erythematosus were collected

DRAFT

and processed using the $10\times$ Chromium Genomics platform [38]. The dataset is composed of a control group (6573 cells) and an interferon-β stimulated group (7466 cells).
The count matrices were downloaded from SEURAT's tutorial *"Integrating stimulated vs. control PBMC datasets to learn cell-type specific responses"*[1].

Pancreatic islet cells. PIC datasets were generated independently using four different platforms: CEL-Seq [45] (1004 cells), CEL-Seq2 [46] (2285 cells), Fluidigm C1 [47] (638 cells), and Smart-Seq2 [48] (2394 cells). The clusters across the datasets were identified in [41] by applying the PCA on the scaled integrated data matrix.
The count matrices were downloaded from SEURAT's tutorial *"Integration and Label Transfer"*[2].

Metrics

Adjusted Rand Index. The Rand Index (RI) is a similarity measure between the results obtained from the application of two different clustering methods. The first clustering method is used as ground truth (i.e., true clusters), while the second method has to be evaluated (i.e., predicted clusters). RI is calculated by considering all pairs of samples appearing in the clusters, namely, it counts the pairs that are assigned either to the same or different clusters in both the predicted and the true clusters. The Adjusted RI (ARI) [49] is the "adjusted for chance" version of RI. Its values vary in the range $[-1, 1]$: a value close to 0 means a random assignment, independently of the number of clusters, while 1 indicates that the clusters obtained with both clustering approaches are identical. Negative values are obtained if the index is less than the expected index.

Adjusted Mutual Information Index. The Mutual Information Index (MII) [50] represents the mutual information of two random variables, which is a similarity measure of the mutual dependence between the two variables. Specifically, it is used to quantify the amount of information that can be gained by one random variable observing the other variable. MII is strictly correlated with the entropy of a random variable, which quantifies the expected "amount of information" that is contained in a random variable. This index is used to measure the similarity between two labels of the same data. Similarly to ARI, the Adjusted MII (AMII) is "adjusted for chance" and its values vary in the range $[0, 1]$.

Fowlkes Mallows Index. The FMI [51] measures the similarity between the clusters obtained by using two different clustering approaches. It is defined as the geometric mean between precision and recall. Assuming that the first clustering approach is the ground truth, the precision is the percentage of the results that are relevant, while the recall refers to the percentage of total relevant results correctly assigned by the second clustering approach. The index ranges from 0 to 1.

Homogeneity Score. The result of the tested clustering approach satisfies the HS [52] if all of its clusters contain only cells which are members of a single cell type. Its values range from 0 to 1, where 1 indicates perfectly homogeneous labelling.
Notice that by switching true cluster labels with the predicted cluster labels, the Completeness Score is obtained.

Completeness Score. The result of the tested clustering approach satisfies the CS [52] if all the cells that are members of a given cell type are elements of the same cluster. Its values range from 0 to 1, where 1 indicates perfectly complete labelling.
Notice that by switching true cluster labels with the predicted cluster labels, the HS is obtained.

V-Measure. The V-Measure (VM) [53] is the harmonic mean between HS and CS; it is equivalent to MII when the arithmetic mean is used as aggregation function.

Methods

scAEspy can be easily integrated into both SCANPY and SEURAT pipeline as it directly works on the gene expression matrix (see Fig. 2). We integrated into a single tool the latest and most powerful existing AEs designed to resolve the problems underlying scRNA-seq data (e.g., sparsity, intrinsic noise, and dropout events). Specifically, scAEspy allows the user to use five different AEs (i.e., VAE, GMVAE, MMD-VAE, GMMMD, and GMMMDVAE) by setting up two user-defined parameters, α and λ, which are used to balance the Kullback–Leibler and the MMD divergence functions (see The proposed GMMMD and GMMMDVAE section). We designed and developed GMMMD and GMMMDVAE starting from scVAE [15] and InfoVAE [24]. In addition, a learnable mixture distribution was used for the prior distribution in the latent space, and also the marginal conditional distribution was defined to be a learnable mixture distribution with the same number of components as the prior distribution. Finally, the user can also select the following loss functions: NB, constrained NB, Poisson, constrained Poisson, zero-inflated Poisson, constrained zero-inflated Poisson, zero-inflated NB, constrained zero-inflated NB, and Mean Square Error.

The modified pipeline. We modified the workflow shown in Fig. 1 by replacing the PCA with AEs (Fig. 2).
We merged the gene expression matrices of E different samples ($E = 2$ and $E = 4$ for PBMC and PIC datasets, respectively). We applied both PCA and AEs on the space obtained by the top k HVGs calculated by using the latest implementation of SCANPY, where the top HVGs are separately selected within each batch and merged to avoid the selection of batch-specific genes. Here, we set $k = 1000$.
For what concerns the PCA, we firstly normalised and log-transformed the counts, then we applied a classic standardisation, that is, the distribution of the expression of each gene

[1] https://satijalab.org/seurat/v3.0/immune_alignment.html
[2] https://satijalab.org/seurat/v3.0/pancreas_integration_label_transfer.html

DRAFT

was scaled to zero mean and unit variance. We calculated the first 50 components; after that, we used the so-called "elbow method" to select the number of components for the downstream analysis. We used the first 12 and 14 components for PBMC and PIC datasets, respectively.

Regarding the AEs, we used the original counts since AEs showed to achieve better results when applied on raw data [15]. Indeed, using the counts allows for exploiting discrete probability distributions, such as Poisson and NB distributions, which obtained the best results in our tests. We performed a grid search to find the best set of their functional settings. Based on the obtained ARI, we selected: 100 epochs, a single hidden layer composed of 64 neurons and 16 neurons for the latent space, sigmoid activation functions, and the batch size was set 100 samples. In all tests we used the Adam optimizer [54].

We calculated the neighbourhood graph in both PCA and AE space by using the default parameter settings proposed in SCANPY. We clustered the obtained neighbourhood graphs with the Leiden algorithms by selecting the values of the resolution parameter such that the number of clusters was equal to the manually annotated clusters. Then, we calculated all the metrics for each found resolution. We performed the same analysis by replacing the neighbourhood graphs with those generated using BBKNN, which was applied to remove the batch effects.

The proposed generalised formulation. In this work, we used the notation proposed in [24] to extend MMDVAE with multiple Gaussian distributions as well as to introduce a learnable prior distribution in the latent space. The idea behind the introduction of learnable coefficients is that they might be suitable to model the diversity among the subpopulations of cells composing the data or to combine multiple samples.

We consider $p^*(\mathbf{x})$ as the unknown probability in the input space over which the optimisation problem is formulated, $\boldsymbol{z}$ is the latent representation of $\boldsymbol{x}$ with $|\boldsymbol{z}| \leq |\boldsymbol{x}|$. The encoder is identified by a function $e_{\boldsymbol{\phi}} : \boldsymbol{x} \mapsto \boldsymbol{z}$, while the decoder by a function $d_{\boldsymbol{\theta}} : \boldsymbol{z} \mapsto \boldsymbol{x}$.

We remind that in variational AEs, the input $\boldsymbol{x}$ is not mapped into a single point in the latent space, but it is represented by a probability distribution over the latent space. $q(\boldsymbol{z})$ is any possible distribution in the latent space and $y \in \{1, \ldots, K\}$ is a categorical random variable, where K corresponds to the number of desired Gaussian distributions. As general strict divergence function, we considered the maximum mean discrepancy $\mathrm{MMD}(\cdot)$ [25] divergence function.

The ELBO term proposed in this work, which is the measure maximised during the training of the AEs, is:

$$\begin{aligned}\mathrm{ELBO} = \mathbb{E}[\log(d(\boldsymbol{x}|\boldsymbol{z}, y))] \\ - (\alpha + \lambda - 1)\mathrm{MMD}(p_e(\boldsymbol{z})||q(\boldsymbol{z})) \\ - (1-\alpha)\mathbb{E}[\mathrm{KL}(p_e(\boldsymbol{z}, y|\boldsymbol{x})||q(\boldsymbol{z}, y))],\end{aligned} \tag{1}$$

where $\mathrm{KL}(\cdot)$ is the Kullback–Leibler divergence [26] between two distributions. All the mathematical details required to derive the generalised formula shown in Eq. 1 can be found in the Additional file 2.

Eq. 1 allows the user to easily exploit VAE, MMDVAE, GMVAE, GMMMD, and GMMMMDVAE. Indeed, by setting a single Gaussian distribution and $\alpha = 0, \lambda = 1$ the VAE model is obtained, while $\alpha = 1, \lambda = 1$ allows for using MMDVAE. When multiple Gaussian distributions are taken into account, GMVAE can be obtained by setting $\alpha = 0, \lambda = 1$. GMMMD and GMMMDVAE are obtained by setting $\alpha = 1, \lambda = 1$ and $\alpha = 0, \lambda = 2$, respectively.

Availability and requirements

scAEspy is written in Python programming language (version 3.6.5) and it relies on TensorFlow (version 1.12.0), an open-source machine learning library [21]. scAEspy also requires the following Python libraries: NumPy, SciKit-Learn, Matplotlib, and Seaborn. scAEspy's open-source code is available on GitLab: `https://gitlab.com/cvejic-group/scaespy` under the GPL-3 license. We provide a detailed description of scAEspy's parameters so that it be used by both novice and expert researchers for downstream analyses.

The repository contains all the scripts used to obtain the results shown in the paper. In addition, we provide Jupyter Notebooks showing the integration between scAEspy and SCANPY. In these Jupyter Notebooks, we also show how the data can be visualised and explored by using both scAEspy and SCANPY's functions.

References

1. Monika M Gladka, Bas Molenaar, Hesther De Ruiter, Stefan Van Der Elst, Hoyee Tsui, Danielle Versteeg, Grègory PA Lacraz, Manon MH Huibers, Alexander Van Oudenaarden, and Eva Van Rooij. Single-cell sequencing of the healthy and diseased heart reveals cytoskeleton-associated protein 4 as a new modulator of fibroblasts activation. *Circulation*, 138(2):166–180, 2018. doi: 10.1161/CIRCULATIONAHA.117.030742.
2. Hadas Keren-Shaul, Amit Spinrad, Assaf Weiner, Orit Matcovitch-Natan, Raz Dvir-Szternfeld, Tyler K Ulland, Eyal David, Kuti Baruch, David Lara-Astaiso, Beata Toth, et al. A unique microglia type associated with restricting development of alzheimer's disease. *Cell*, 169(7):1276–1290, 2017. doi: 10.1016/j.cell.2017.05.018.
3. Svante Wold, Kim Esbensen, and Paul Geladi. Principal component analysis. *Chemom. Intell. Lab. Syst.*, 2(1-3):37–52, 1987. doi: 10.1016/0169-7439(87)80084-9.
4. Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *J. Mach. Learn. Res.*, 9(Nov):2579–2605, 2008.
5. Leland McInnes, John Healy, and James Melville. UMAP: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, 2018.
6. Etienne Becht, Leland McInnes, John Healy, Charles-Antoine Dutertre, Immanuel WH Kwok, Lai Guan Ng, Florent Ginhoux, and Evan W Newell. Dimensionality reduction for visualizing single-cell data using umap. *Nature Biotech.*, 37(1):38, 2019. doi: 10.1038/nbt.4314.
7. Byungjin Hwang, Ji Hyun Lee, and Duhee Bang. Single-cell rna sequencing technologies and bioinformatics pipelines. *Exp. Mol. Med.*, 50(8):96, 2018. doi: 10.1038/s12276-018-0071-8.
8. Andrew Butler, Paul Hoffman, Peter Smibert, Efthymia Papalexi, and Rahul Satija. Integrating single-cell transcriptomic data across different conditions, technologies, and species. *Nature Biotech.*, 36(5):411, 2018. doi: 10.1038/nbt.4096.
9. Jong-Eun Park, Krzysztof Polański, Kerstin Meyer, and Sarah A Teichmann. Fast batch alignment of single cell transcriptomes unifies multiple mouse cell atlases into an integrated landscape. *bioRxiv*, page 397042, 2018.
10. Laleh Haghverdi, Aaron TL Lun, Michael D Morgan, and John C Marioni. Batch effects in single-cell rna-sequencing data are corrected by matching mutual nearest neighbors. *Nature Biotech.*, 36(5):421, 2018. doi: doi.org/10.1038/nbt.4091.
11. Brian L Hie, Bryan Bryson, and Bonnie Berger. Panoramic stitching of heterogeneous single-cell transcriptomic data. *BioRxiv*, page 371179, 2018.
12. Rhonda Bacher and Christina Kendziorski. Design and computational analysis of single-cell rna-sequencing experiments. *Genome Biol.*, 17(1):63, 2016. doi: 10.1186/s13059-016-0927-y.
13. Gökcen Eraslan, Lukas M Simon, Maria Mircea, Nikola S Mueller, and Fabian J Theis. Single-cell rna-seq denoising using a deep count autoencoder. *Nat. Commun.*, 10(1):390, 2019. doi: 10.1038/s41467-018-07931-2.

DRAFT

14. Romain Lopez, Jeffrey Regier, Michael B Cole, Michael I Jordan, and Nir Yosef. Deep generative modeling for single-cell transcriptomics. *Nat. Methods*, 15(12):1053, 2018. doi: 10.1038/s41592-018-0229-2.
15. Christopher Heje Grønbech, Maximillian Fornitz Vording, Pascal N Timshel, Casper Kaae Sønderby, Tune Hannes Pers, and Ole Winther. scvae: Variational auto-encoders for single-cell gene expression data. *bioRxiv*, page 318295, 2018. doi: 10.1101/318295.
16. Divyanshu Talwar, Aanchal Mongia, Debarka Sengupta, and Angshul Majumdar. Autoimpute: Autoencoder based imputation of single-cell rna-seq data. *Sci. Rep.*, 8(1):16329, 2018. doi: 10.1038/s41598-018-34688-x.
17. Dongfang Wang and Jin Gu. Vasc: dimension reduction and visualization of single-cell rna-seq data by deep variational autoencoder. *Genomics, proteomics & bioinformatics*, 16(5): 320–331, 2018. doi: 10.1016/j.gpb.2018.08.003.
18. F Alexander Wolf, Philipp Angerer, and Fabian J Theis. Scanpy: large-scale single-cell gene expression data analysis. *Genome Biol.*, 19(1):15, 2018. doi: 10.1186/s13059-017-1382-0.
19. Rahul Satija, Jeffrey A Farrell, David Gennert, Alexander F Schier, and Aviv Regev. Spatial reconstruction of single-cell gene expression data. *Nature Biotech.*, 33(5):495, 2015. doi: 10.1038/nbt.3192.
20. François Chollet et al. Keras. https://github.com/fchollet/keras, 2015. Online; accessed 17 July 2019.
21. Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. Tensorflow: A system for large-scale machine learning. In *Proceedings of the Symposium on Operating Systems Design and Implementation)*, pages 265–283, 2016.
22. Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017.
23. Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.
24. Shengjia Zhao, Jiaming Song, and Stefano Ermon. Infovae: Information maximizing variational autoencoders. *arXiv preprint arXiv:1706.02262*, 2017.
25. Arthur Gretton, Karsten Borgwardt, Malte Rasch, Bernhard Schölkopf, and Alex J Smola. A kernel method for the two-sample-problem. In *Proceedings of the Conference on Advances in Neural Information Processing Systems*, pages 513–520, 2007.
26. Solomon Kullback and Richard A Leibler. On information and sufficiency. *Ann. Math. Statist.*, 22(1):79–86, 1951. doi: 10.1214/aoms/1177729694.
27. Evan Z Macosko, Anindita Basu, Rahul Satija, James Nemesh, Karthik Shekhar, Melissa Goldman, Itay Tirosh, Allison R Bialas, Nolan Kamitaki, Emily M Martersteck, et al. Highly parallel genome-wide expression profiling of individual cells using nanoliter droplets. *Cell*, 161(5):1202–1214, 2015. doi: 10.1016/j.cell.2015.05.002.
28. Allon M Klein, Linas Mazutis, Ilke Akartuna, Naren Tallapragada, Adrian Veres, Victor Li, Leonid Peshkin, David A Weitz, and Marc W Kirschner. Droplet barcoding for single-cell transcriptomics applied to embryonic stem cells. *Cell*, 161(5):1187–1201, 2015. doi: 10.1016/j.cell.2015.04.044.
29. Tamar Hashimshony, Florian Wagner, Noa Sher, and Itai Yanai. Cel-seq: single-cell rna-seq by multiplexed linear amplification. *Cell Rep.*, 2(3):666–673, 2012. doi: doi.org/10.1016/j.celrep.2012.08.003.
30. Tamar Hashimshony, Naftalie Senderovich, Gal Avital, Agnes Klochendler, Yaron de Leeuw, Leon Anavy, Dave Gennert, Shuqiang Li, Kenneth J Livak, Orit Rozenblatt-Rosen, et al. Cel-seq2: sensitive highly-multiplexed single-cell rna-seq. *Genome Biol.*, 17(1):77, 2016. doi: 10.1186/s13059-016-0938-8.
31. Grace XY Zheng, Jessica M Terry, Phillip Belgrader, Paul Ryvkin, Zachary W Bent, Ryan Wilson, Solongo B Ziraldo, Tobias D Wheeler, Geoff P McDermott, Junjie Zhu, et al. Massively parallel digital transcriptional profiling of single cells. *Nat. Commun.*, 8:14049, 2017. doi: 10.1038/ncomms14049.
32. Todd M Gierahn, Marc H Wadsworth II, Travis K Hughes, Bryan D Bryson, Andrew Butler, Rahul Satija, Sarah Fortune, J Christopher Love, and Alex K Shalek. Seq-well: portable, low-cost rna sequencing of single cells at high throughput. *Nat. Methods*, 14(4):395, 2017. doi: 10.1038/nmeth.4179.
33. Saiful Islam, Una Kjällquist, Annalena Moliner, Pawel Zajac, Jian-Bing Fan, Peter Lönnerberg, and Sten Linnarsson. Characterization of the single-cell transcriptional landscape by highly multiplex rna-seq. *Genome Res.*, 21(7):1160–1167, 2011. doi: 10.1101/gr.110882.110.
34. Daniel Ramsköld, Shujun Luo, Yu-Chieh Wang, Robin Li, Qiaolin Deng, Omid R Faridani, Gregory A Daniels, Irina Khrebtukova, Jeanne F Loring, Louise C Laurent, et al. Full-length mrna-seq from single-cell levels of rna and individual circulating tumor cells. *Nature Biotech.*, 30(8):777, 2012. doi: 10.1038/nbt.2282.
35. Simone Picelli, Omid R Faridani, Åsa K Björklund, Gösta Winberg, Sven Sagasser, and Rickard Sandberg. Full-length rna-seq from single cells using smart-seq2. *Nat. Protoc.*, 9(1):171, 2014. doi: 10.1038/nprot.2014.006.
36. Diego Adhemar Jaitin, Ephraim Kenigsberg, Hadas Keren-Shaul, Naama Elefant, Franziska Paul, Irina Zaretsky, Alexander Mildner, Nadav Cohen, Steffen Jung, Amos Tanay, et al. Massively parallel single-cell rna-seq for marker-free decomposition of tissues into cell types. *Science*, 343(6172):776–779, 2014. doi: 10.1126/science.1247651.
37. Vincent A Traag, Ludo Waltman, and Nees Jan van Eck. From louvain to leiden: guaranteeing well-connected communities. *Sci. Rep.*, 9, 2019. doi: 10.1038/s41598-019-41695-z.
38. Hyun Min Kang, Meena Subramaniam, Sasha Targ, Michelle Nguyen, Lenka Maliskova, Elizabeth McCarthy, Eunice Wan, Simon Wong, Lauren Byrnes, Cristina M Lanata, et al. Multiplexed droplet single-cell rna-sequencing using natural genetic variation. *Nature biotechnology*, 36(1):89, 2018. doi: 10.1038/nbt.4042.
39. Frank Wilcoxon. Individual comparisons by ranking methods. In *Breakthroughs in statistics*, pages 196–202. Springer, 1992. doi: 10.1007/978-1-4612-4380-9$_$16.
40. Olive Jean Dunn. Estimation of the means of dependent variables. *Ann. Math. Stat.*, 29(4): 1095–1111, 1958.
41. Tim Stuart, Andrew Butler, Paul Hoffman, Christoph Hafemeister, Efthymia Papalexi, William M Mauck III, Yuhan Hao, Marlon Stoeckius, Peter Smibert, and Rahul Satija. Comprehensive integration of single-cell data. *Cell*, 2019. doi: 10.1016/j.cell.2019.05.031.
42. Nikola Simidjievski, Cristian Bodnar, Ifrah Tariq, Paul Scherer, Helena Andres-Terre, Zohreh Shams, Mateja Jamnik, and Pietro Liò. Variational autoencoders for cancer data integration: Design principles and computational practice. *bioRxiv*, page 719542, 2019.
43. Feiyang Ma and Matteo Pellegrini. Actinn: Automated identification of cell types in single cell rna sequencing. *Bioinformatics*, 2019. doi: 10.1093/bioinformatics/btz592.
44. Xi Chen, Ricardo J Miragaia, Kedar Nath Natarajan, and Sarah A Teichmann. A rapid and robust method for single cell chromatin accessibility profiling. *Nat. Commun.*, 9(1):5345, 2018. doi: 10.1038/s41467-018-07771-0.
45. Dominic Grün, Mauro J Muraro, Jean-Charles Boisset, Kay Wiebrands, Anna Lyubimova, Gitanjali Dharmadhikari, Maaike van den Born, Johan van Es, Erik Jansen, Hans Clevers, et al. De novo prediction of stem cell identity using single-cell transcriptome data. *Cell stem cell*, 19(2):266–277, 2016. doi: 10.1016/j.stem.2016.05.010.
46. Mauro J Muraro, Gitanjali Dharmadhikari, Dominic Grün, Nathalie Groen, Tim Dielen, Erik Jansen, Leon van Gurp, Marten A Engelse, Francoise Carlotti, Eelco JP de Koning, et al. A single-cell transcriptome atlas of the human pancreas. *Cell Syst.*, 3(4):385–394, 2016. doi: 10.1016/j.cels.2016.09.002.
47. Nathan Lawlor, Joshy George, Mohan Bolisetty, Romy Kursawe, Lili Sun, V Sivakamasundari, Ina Kycia, Paul Robson, and Michael L Stitzel. Single-cell transcriptomes identify human islet cell signatures and reveal cell-type–specific expression changes in type 2 diabetes. *Genome Res.*, 27(2):208–222, 2017. doi: 10.1101/gr.212720.116.
48. Åsa Segerstolpe, Athanasia Palasantza, Pernilla Eliasson, Eva-Marie Andersson, Anne-Christine Andréasson, Xiaoyan Sun, Simone Picelli, Alan Sabirsh, Maryam Clausen, Magnus K Bjursell, et al. Single-cell transcriptome profiling of human pancreatic islets in health and type 2 diabetes. *Cell Metab.*, 24(4):593–607, 2016. doi: 10.1016/j.cmet.2016.08.020.
49. Lawrence Hubert and Phipps Arabie. Comparing partitions. *Journal of classification*, 2(1): 193–218, 1985. doi: 10.1007/BF01908075.
50. Alexander Strehl and Joydeep Ghosh. Cluster ensembles—a knowledge reuse framework for combining multiple partitions. *J. Mach. Learn. Res.*, 3(Dec):583–617, 2002.
51. Edward B Fowlkes and Colin L Mallows. A method for comparing two hierarchical clusterings. *J. Am. Stat. Assoc.*, 78(383):553–569, 1983.
52. Nguyen Xuan Vinh, Julien Epps, and James Bailey. Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance. *J. Mach. Learn. Res.*, 11(Oct):2837–2854, 2010.
53. Andrew Rosenberg and Julia Hirschberg. V-measure: A conditional entropy-based external cluster evaluation measure. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 410–420, 2007.
54. Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Competing interests. The authors declare that they have no competing interests.

Author's contributions. AT conceived the project. AT and FR developed the software. AC, DB, and PL supervised the project and helped to interpret and present the results. AT and AC wrote the manuscript. DB and PL edited the manuscript. All authors read and approved the final manuscript.

Acknowledgements. This research was supported by Cancer Research UK grant number C45041/A14953 (AC and AT), European Research Council project 677501 – ZF_Blood (AC) and a core support grant from the Wellcome Trust and MRC to the Wellcome Trust – Medical Research Council Cambridge Stem Cell Institute.
We thank Dr. Leonardo Rundo (Department of Radiology, University of Cambridge) and Dr. Vladimir Kiselev (Cellular Genetics Informatics, Wellcome Trust Sanger Institute) for their critical comments.

DRAFT

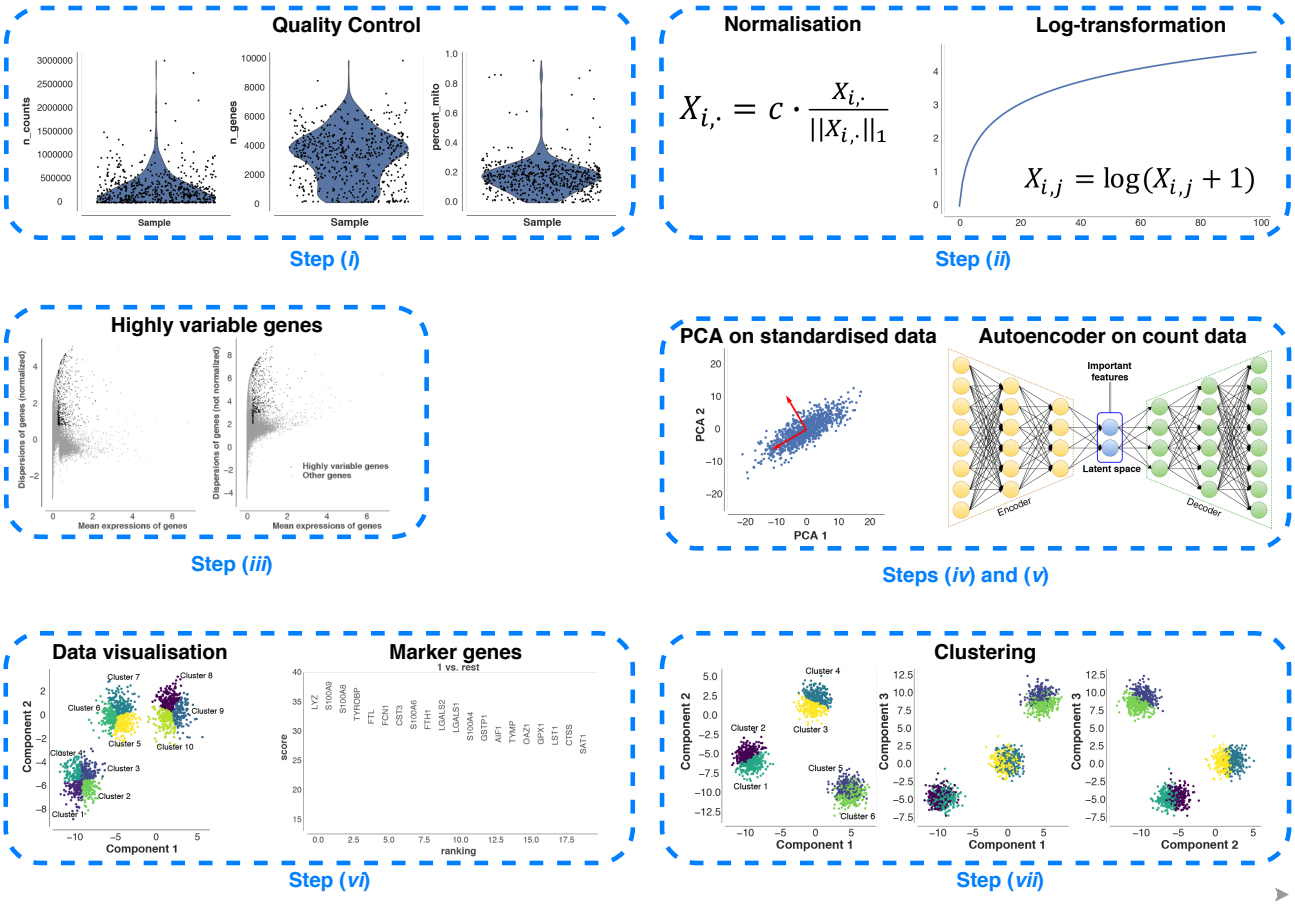

Fig. 1. The common workflow of a downstream analysis of scRNA-seq data. The workflow includes the following seven steps: (*i*) quality control to remove low-quality cells that may add technical noise, which could obscure the real biological signals; (*ii*) normalisation and log-transformation; (*iii*) identification of the HVGs to reduce the dimensionality of the dataset by including only the most informative genes; (*iv*) standardisation of each gene to zero mean and unit variance; (*v*) dimensionality reduction generally obtained by applying a linear approach (such as PCA); (*vi*) clustering of the cells starting from the low-dimensional representation of the data that are used to annotate the obtained clusters (i.e., identification of known and putatively novel cell types); (*vii*) data visualisation on the low-dimensional space generated by applying a non-linear approach on the reduced space calculated in step (*v*).

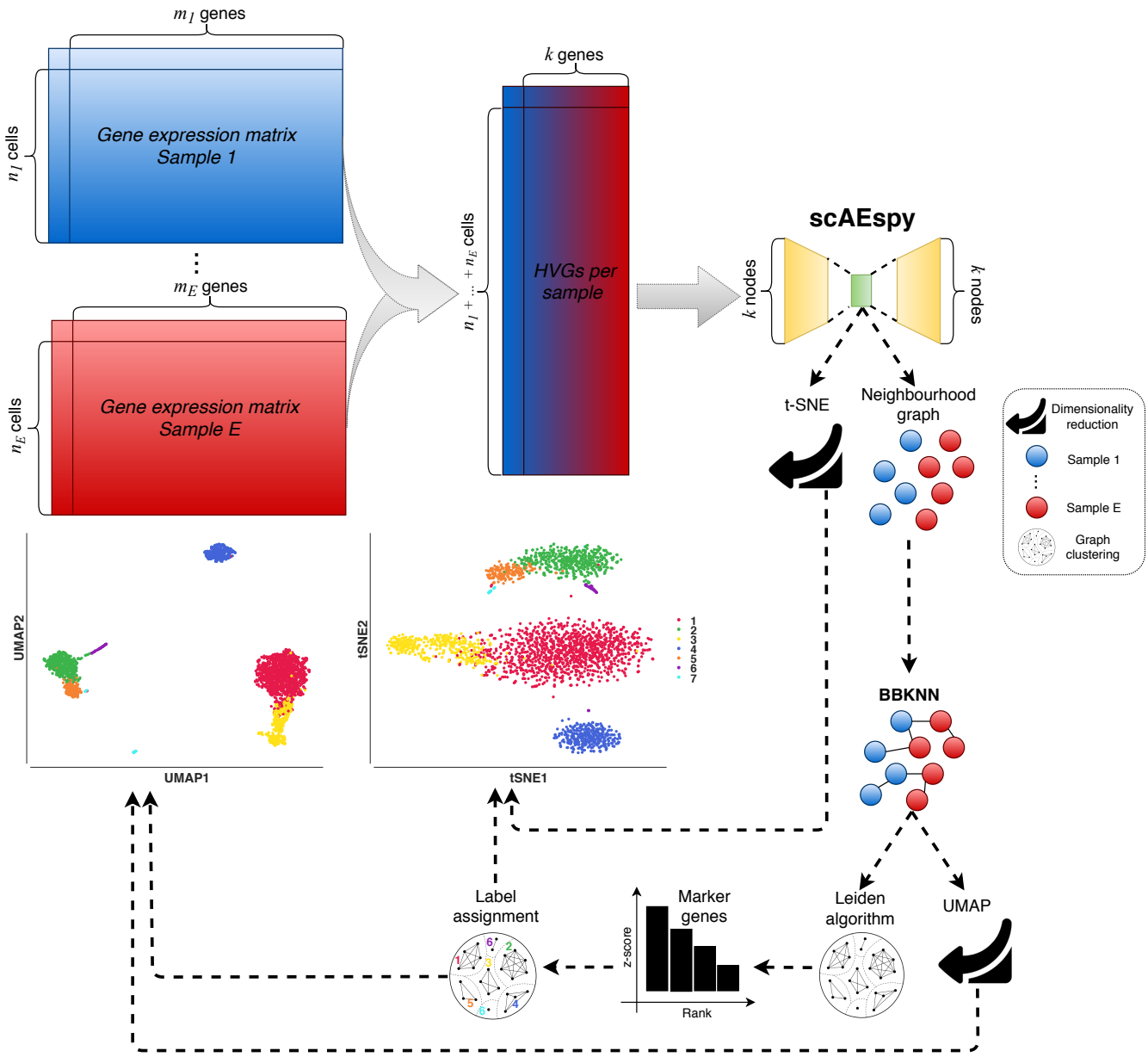

Fig. 2. The modified workflow when multiple samples are integrated. Given E different samples, their gene expression matrices are merged. Then, the top k HVGs are selected by considering the different samples. Specifically, they are selected within each sample separately and then merged to avoid the selection of batch-specific genes. scAEspy is used to reduce the HVG space (k dimensions), and the obtained latent space is used to calculate both the t-SNE space and the neighbourhood graph. BBKNN is applied to modify the neighbourhood graph by taking into account the batch effects that may exist among the samples. The modified neighbourhood graph is clustered by using the Leiden algorithm and used to calculate the UMAP space. In order to assign the correct label to the obtained clusters, the marker genes are calculated by using the Wilcoxon test. Finally, the annotated clusters are visualised in both t-SNE and UMAP space.

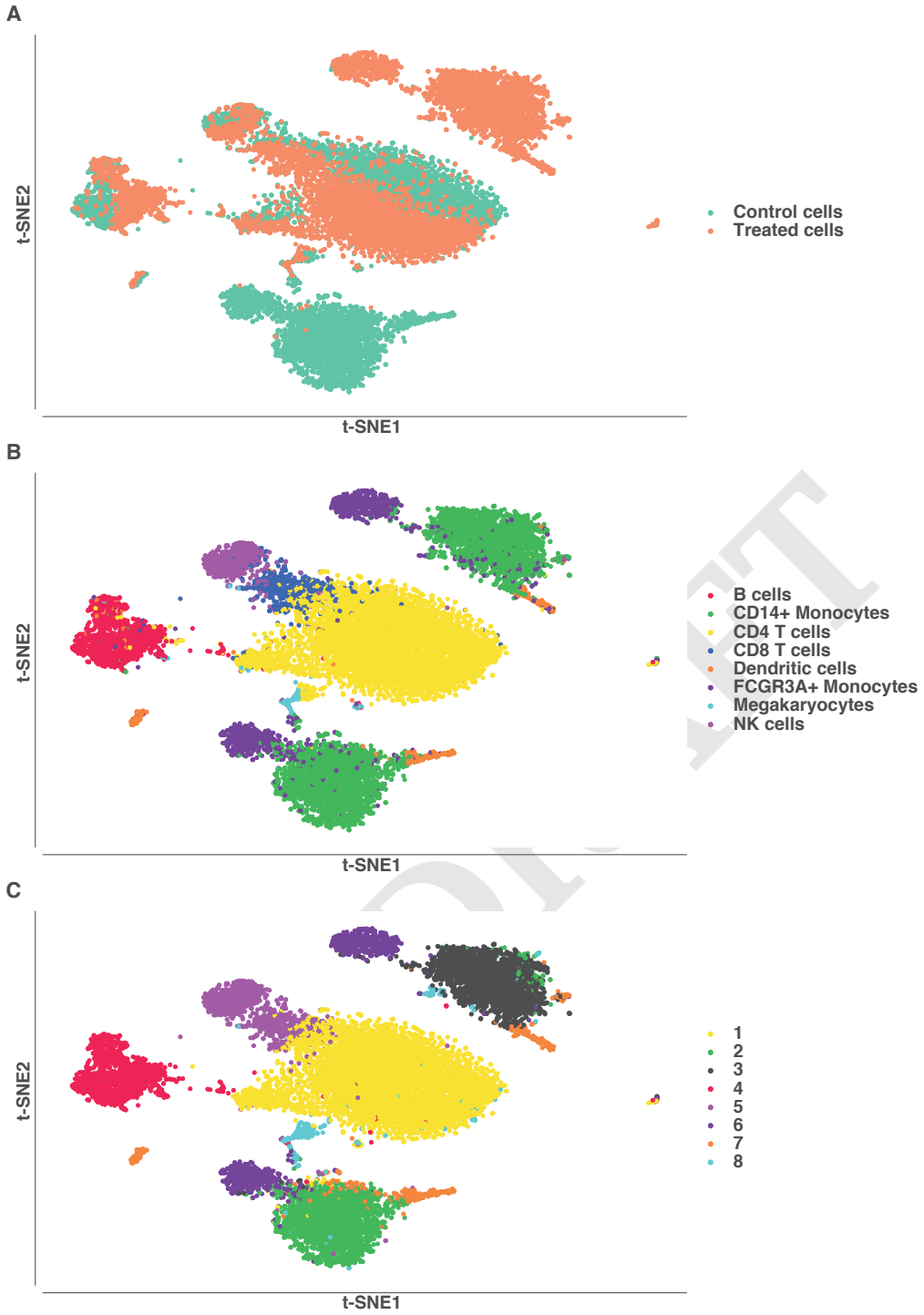

Fig. 3. The 2D t-SNE, calculated in the PCA followed by BBKNN space. A. The 2D t-SNE of control (light green) and treated (orange) cells. The t-SNE was calculated from the PCA followed by BBKNN space. **B**. The manually annotated clusters are visualised in the 2D t-SNE. **C**. Clusters obtained by applying the Leiden algorithm with the best resolution (highest RI value). The obtained clusters were labeled by the same colour as the manually annotated clusters based on similarity of marker genes between the two approaches. We used a grayscale palette for the clusters that did not correspond to any manually annotated cluster.

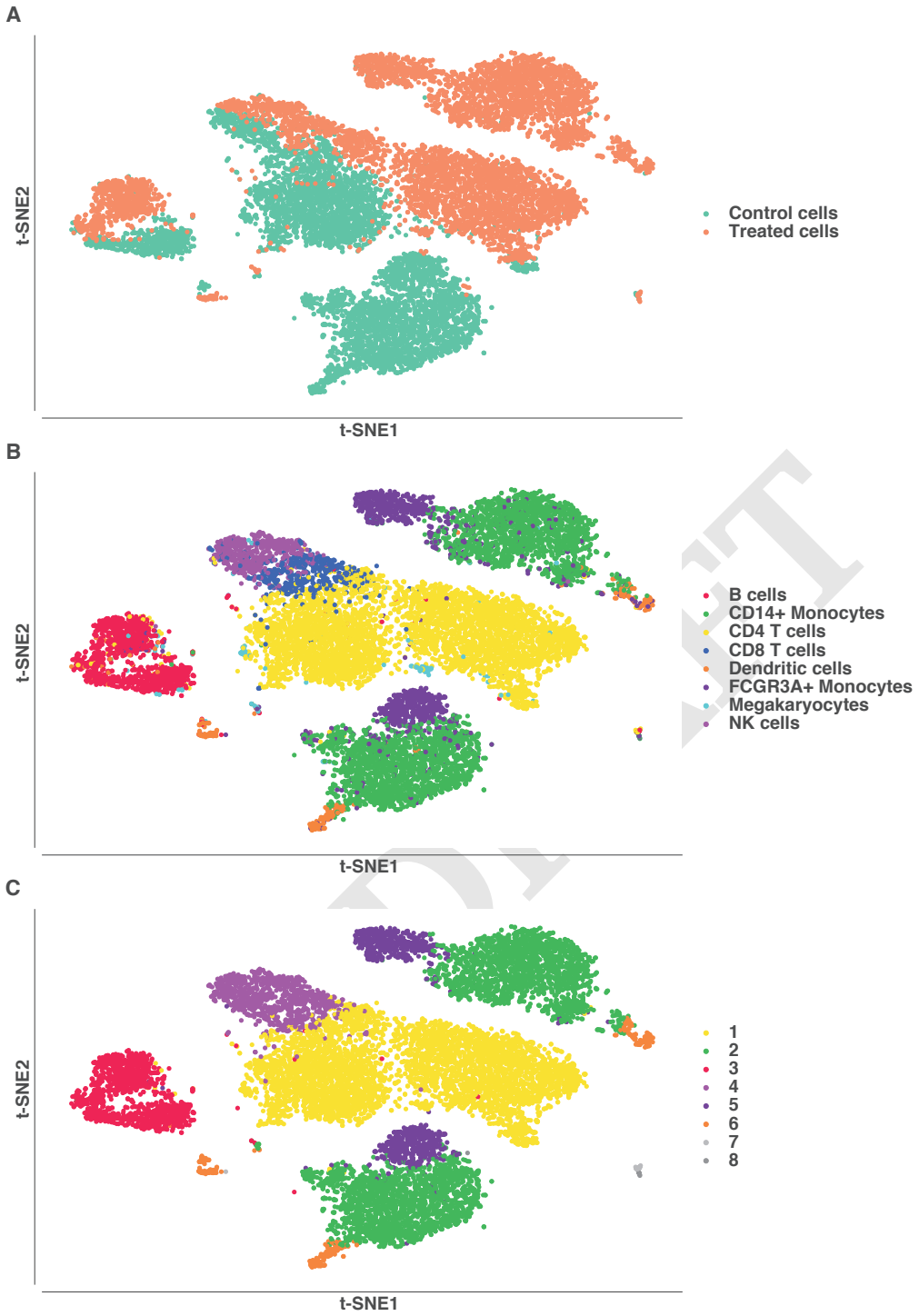

Fig. 4. The 2D t-SNE space, calculated in the GMMMD followed by BBKNN space. A. t-SNE showing the control (light green) and treated (orange) cells. **B**. The manually annotated clusters are visualised in the 2D t-SNE. **C**. Clusters obtained by applying the Leiden algorithm with the best resolution.

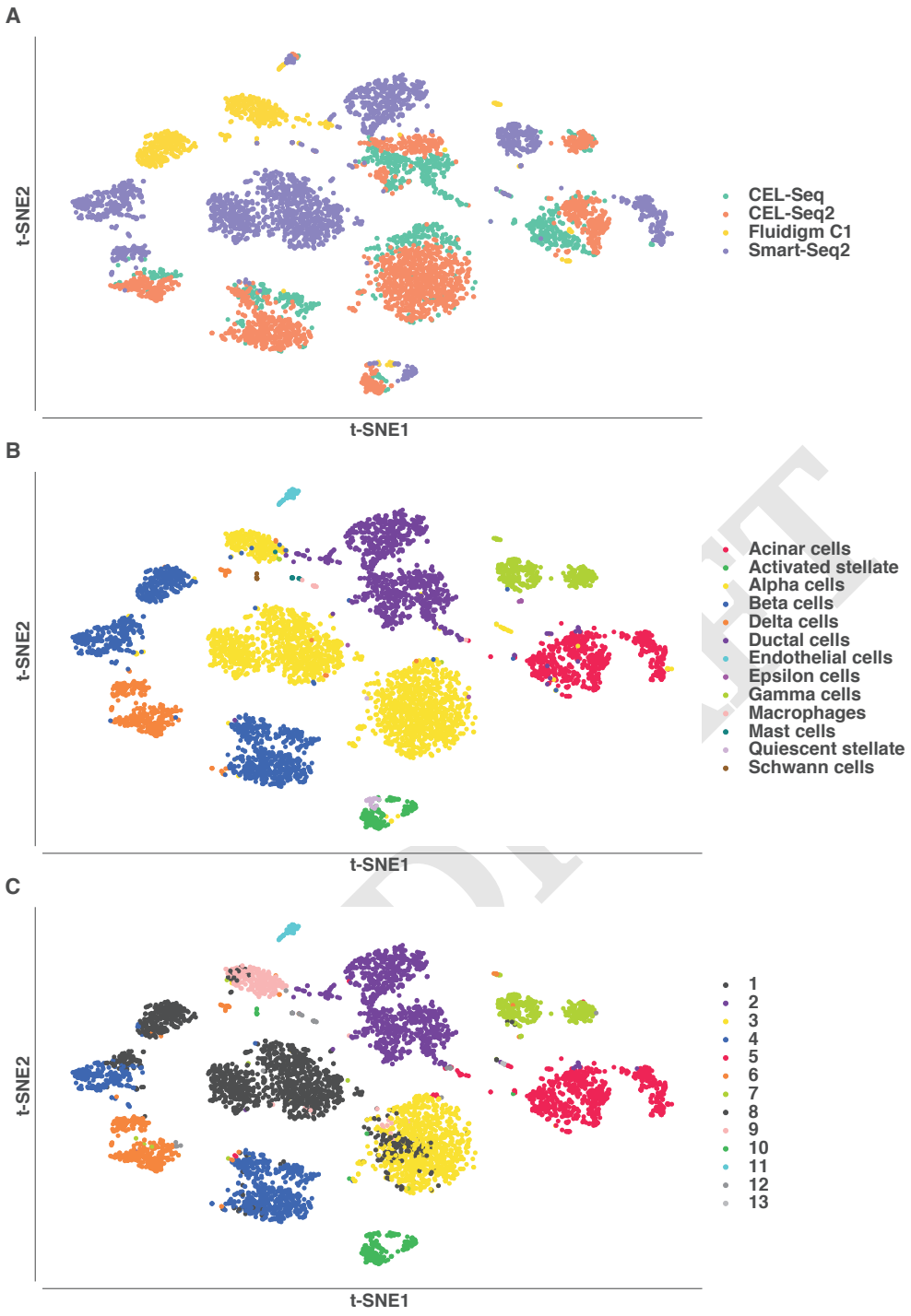

Fig. 5. The 2D t-SNE space, calculated in the PCA followed by BBKNN space. **A.** Cells in t-SNE plot are coloured by platform. **B.** The manually annotated clusters are visualised in the 2D t-SNE. **C.** Clusters obtained by applying the Leiden algorithm with the best resolution.

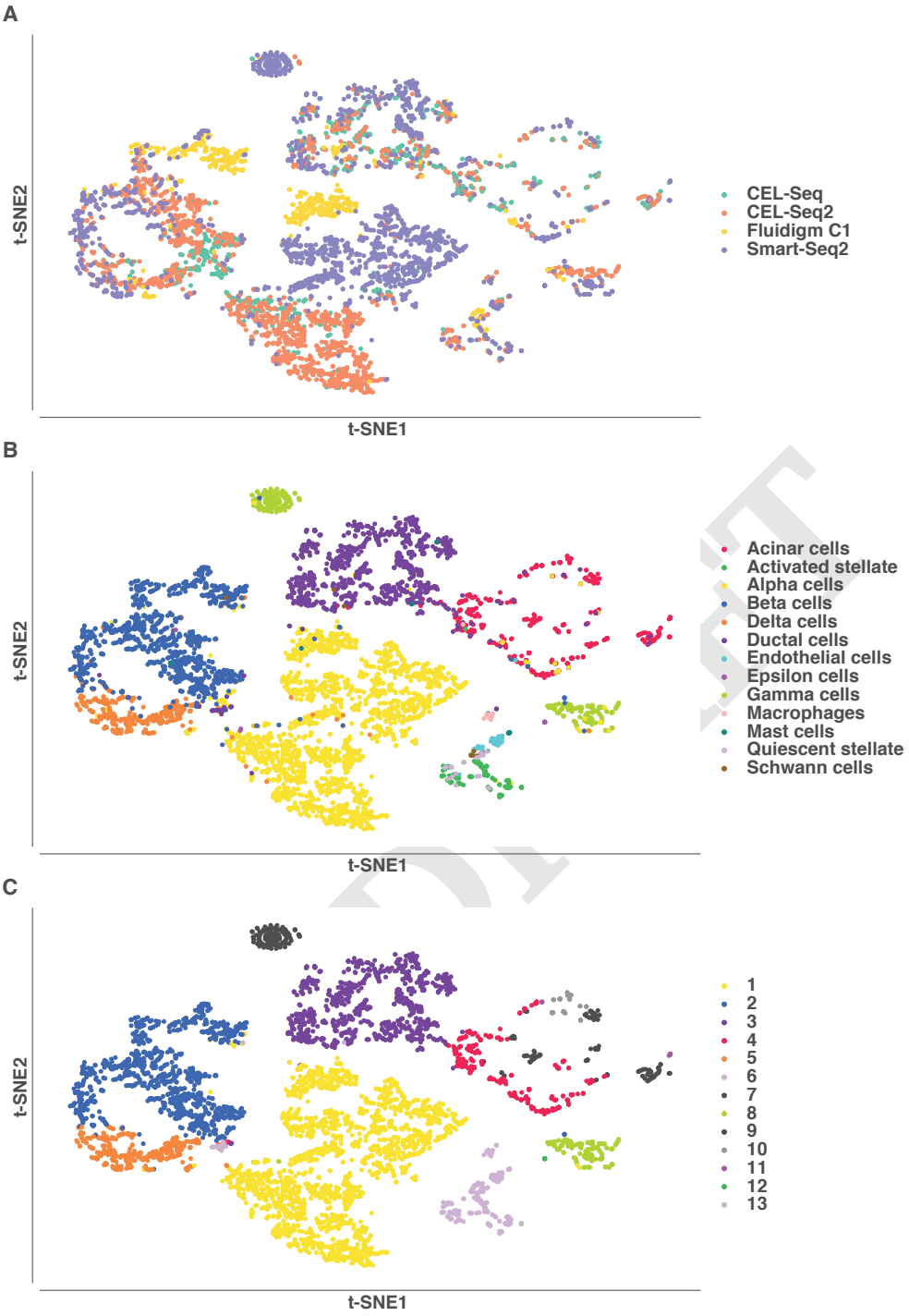

Fig. 6. The 2D t-SNE space, calculated in the GMMMDVAE followed by BBKNN space. A. The cells in t-SNE plot are coloured by platform. **B.** The manually annotated clusters are visualised in the 2D t-SNE. **C.** Clusters obtained by applying the Leiden algorithm with the best resolution.