Contrastive self-supervised clustering of scRNA-seq data

Journal:	*NAR Genomics and Bioinformatics*
Manuscript ID	NARGAB-2020-158
Manuscript Type:	Methods Article
Date Submitted by the Author:	01-Dec-2020
Complete List of Authors:	Ciortan, Madalina; Université Libre de Bruxelles, Interuniversity Institute of Bioinformatics in Brussels Defrance, Matthieu; Université Libre de Bruxelles, Interuniversity Institute of Bioinformatics in Brussels
Keywords:	Clustering, Contrastive learning, Single cell RNA-seq, Deep learning, Self training

Contrastive self-supervised clustering of scRNA-seq data

Madalina Ciortan, Matthieu Defrance

Interuniversity Institute of Bioinformatics in Brussels

Université Libre de Bruxelles, Belgium

Corresponding author: matthieu.dc.defrance@ulb.ac.be

ABSTRACT

Clustering single cell RNA-seq (scRNA-seq) data allows the exploration of cell identities and their associated marker genes. The high dimensionality of data, its significant sparsity due the presence of frequent dropout events makes this clustering computationally challenging.
In this work, we propose *contrastive-sc*, an unsupervised deep learning method that allows the clustering of scRNA-seq data using contrastive representation learning in a two-phased process. A self-supervised representation learning is first used on strongly augmented cell data to learn an embedding which is clustered in a second phase. Several strategies for clustering have been explored, ranging from direct clustering of the internal representation to combined representation (ensemble) clustering. All explored clustering strategies provided encouraging results, showing that this approach opens new research paths for the unsupervised analysis of scRNA-seq data. The contrastive self-supervised representation learning succeeds in learning a meaningful data embedding and shows encouraging results compared to state-of-the-art methods when the dropout rate or the number of clusters are increasing.

INTRODUCTION

Single cell RNA sequencing (scRNA-seq) has emerged has a new efficient technique to study the transcriptome of individual cells. Indeed, this technique allows the study of transcription dynamics, the composition of tissues or the relationships within gene-networks [1]. Clustering analysis is usually performed in order to discover new cellular subtypes or to identify known identities as well as the underlying marker genes. Single-cell clustering analysis comes with a number of computational challenges, consisting primarily in the high dimensionality of data (the number of transcripts is usually greater than 20,000) leading to “the curse of dimensionality” and its high sparsity, due to low mRNA expression level and dropout events. As a result, numerous computational methods have been made available for the analysis of scRNA-seq data, as shown in various review papers [2]–[5]. CIDR [6] addresses the dropout with a data imputation method and clusters the PCA-reduced result with hierarchical clustering. RaceID [7] replaces K-means with K-medoids in order to identify rare cell types. Seurat [8] performs a cell-community detection on top of the shared nearest neighbor graph. The Scanpy [9] python package makes available several computation methods for scRNA-seq data, including Seurat.

More recently, deep learning techniques have been adapted to analyzing scRNA-seq data. A deep count autoencoder, DCA [10] has been proposed to denoise and impute the original data by learning three components: the count distribution, the sparsity and the overdispersion in the data using a zero-inflated negative binomial (ZINB) model. The ZINB components are estimated by corresponding fully connected layers, representing a triple autoencoder output. This architecture became a baseline for several other state-of-the-art clustering analysis methods. ScDeepCluster [11] enriches the DCA model with a clustering layer attached to the autoencoder’s learned embedding space (i.e. the bottleneck layer) similar to the DEC [12] method, originally proposed for image and text unsupervised analysis. ScDeepCluster employs as loss a linear combination of the ZINB loss and the Kullback-Leibler (KL) divergence between the distribution of soft labels of the embedding layer (measured by a Student’s t distribution) and the derivation of the target distribution. This combined loss helps to preserve the local structure of the data generating distribution while refining the clusters. The DESC model [13] is similar to scDeepCluster, but separates the data construction from the clustering phase. Sczi [14] proposes a weighted soft K-means clustering (instead of hard clustering) and attempts to enhance the association between similar cells under the same cluster. The proposed loss is a linear combination of (1) the ZINB loss, (2) a weighted soft K-means loss and (3) a KL divergence between the Student’s t distribution of the embedding space and that of a target distribution proposed in DEC [12]. The soft K-means employs a Gaussian kernel function as weight measure which gives to each sample embedding a weight proportional to its proximity to the cluster center. The KL divergence loss, being based on the pairwise similarity of data points in the latent space, encourages similar points to be clustered to the same cluster.

Despite the abundance of clustering methods, there is no consensus regarding the best approach under every circumstance. Freytag et. al. showed in the Cluster Headache publication [4] that 11 state-of-the-art methods for scRNA-seq clustering produced different results, also having little in common with a supervised labeling approach. This analysis highlights both the limitations and the challenges of the field.

MATERIALS AND METHODS

In this work, we propose an unsupervised deep learning method to cluster scRNA-seq data using contrastive representation learning. Our method, *contrastive-sc*, analyzes a count matrix $D = \{x_{ij}\} \in R^{n*d}$ having ***n*** samples (e.g. cells) and ***d*** features (e.g. transcripts) in a two-phased process in order to identify a partition of well-separated sample clusters. First, a representation learning phase creates an embedding of the original data, which is then clustered in the second phase. Our contribution consists in applying self-supervised contrastive learning to scRNA-seq data to create a sample embedding, followed by the assessment of several strategies to cluster the learned representation.

Similar to recent state of the art contributions ([11], [14]), our method starts with the standard preprocessing phase proposed by the Scanpy package [9]. Only the most variable genes are kept, and the count matrix D is normalized using the following rule: each cell is divided by its global counts sum and the values are multiplied by the median of the total expression across the dataset. A natural logarithm is then applied and subsequently the data is standardized such that each gene has a mean equal to 0 and a standard deviation of 1. Removal of less variable and low-expression genes also brings computational gain.

Self-supervised contrastive learning has been recently explored in the context of image classification in semi-supervised [15] and unsupervised [16] [17] settings. It consists in applying strong data augmentation to the input x_i (e.g. the cells) in order to generate two distinct views of the same data: x'_i and x''_i. Our data augmentation technique consists in using a limited random selection of the most variable genes (up to 20% of the genes). An artificial neural network composed of several fully connected layers (in our experiments 3 linear layers have been used) processes the input data and outputs an embedding (a vector of size 32). The contrastive loss proposed in [15] is applied on this final layer in order to guide the model to map similar views to neighboring representations and dissimilar ones to non-neighboring ones (Figure 1). Given a minibatch with N samples, the two sample views are concatenated creating a structure indexed by $1 \leqslant i \leqslant 2N$. The loss takes the form:

$$L^{contrastive} = \sum_{i=1}^{2N} L_i^{contrastive}, \text{ with}$$

$$L_i^{contrastive} = -log \frac{exp\left(\frac{z_i \cdot z_{pair(i)}}{\tau}\right)}{\sum_{i=1}^{2N} 1_{i \neq j} exp\left(\frac{z_i \cdot z_{pair(i)}}{\tau}\right)},$$

where $pair(i)$ is the batch index of the pair for sample i, z_i is the embedding of x_i, $z_i \cdot z_{pair(i)}$ represents the dot product between the normalized embedding vectors, τ is a temperature parameter (set in our experiments to the recommended value of 0.07) and $1_B \in \{0,1\}$ is the indicator function returning 1 iff B evaluates to true.

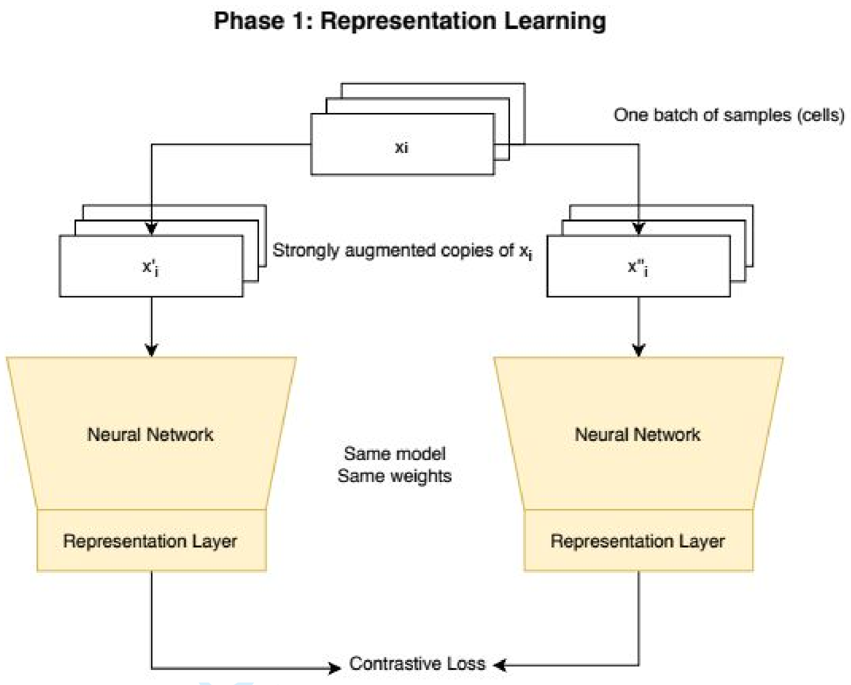

Figure 1. Representation learning phase overview: two strongly augmented copies of the input data (x'_i and x''_i) are passed through the network trained with the unsupervised contrastive loss. This guides the model to map similar views to neighboring representations and dissimilar ones to non-neighboring ones. The data augmentation consists of adding random ratios of neural-network dropout on each data view, on up to 90% of the features (e.g. genes).

Unlike images, scRNA-seq data is affected by dropout events, representing missing gene measurements producing incorrect zero count observations. Existing scRNA-seq clustering methods have addressed the dropout problem using models such as NB or ZINB autoencoder [18] [14] [19], which models the expected negative binomial or zero inflated negative binomial distribution of count data. Our method constitutes an alternative to this analytical framework relying on strong data augmentation to acquire robustness to dropout.

After the first representation learning phase, the second clustering phase can be performed. The following approaches have been explored:

(a) Baseline approach: clustering directly the embedding with a general clustering algorithm (e.g. K-means);

(b) Clustering Network: adding a soft clustering layer to the representation network and training as proposed in [14];

(c) Combined representation clustering: consisting in training multiple networks until the representation phase, combining the embedding and clustering the result with an arbitrary clustering algorithm (e.g. Leiden, K-means).

Since our experiments have shown that the last approach performed best on both simulated and biological data (see results section for a detailed view), the method description will focus, here, only on this approach. The other two approaches are described in Supplementary Materials.

Due to the differences in the views generated using strong data augmentation, training several identical models leads to complementary representations. As illustrated in Figure 2, three models have been trained on the representation learning task. The resulting embeddings are combined and clustered with a general algorithm (e.g. Leiden, K-means). This approach benefits from the improved generalization provided by model ensembling and allows the possibility to choose a suitable clustering algorithm. For instance, employing a community algorithm such as Leiden [20] can mitigate the need for the prior knowledge of the number of clusters and addresses known pitfalls of K-means (e.g. loss of performance in the presence of a strong class imbalance).

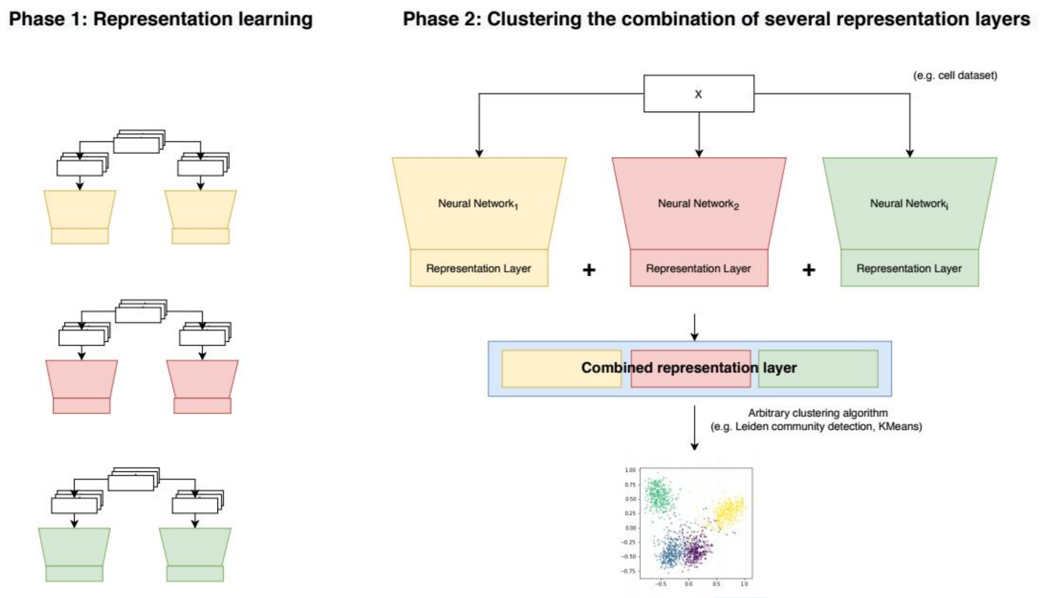
Phase 1: Representation learning
Phase 2: Clustering the combination of several representation layers
X
(e.g. cell dataset)
Neural Network$_1$
Neural Network$_2$
Neural Network$_i$
Representation Layer
Representation Layer
Representation Layer
+
+
Combined representation layer
Arbitrary clustering algorithm
(e.g. Leiden community detection, KMeans)

Figure 2. Overview of the combined representation clustering. Several models having the same architecture are trained in parallel (Phase 1, detailed in Figure 1). The combined representation is then clustered using a general clustering algorithm (Phase 2).

RESULTS

Several experiments have been performed employing both simulated and biological data.

Analysis of simulated data

The data simulation strategy consisted in generating both balanced and imbalanced datasets (i.e. uniform and non-uniform distribution of cluster sizes). The R package splatter [21] has been used to generate datasets with an arbitrary number of clusters, samples, genes and dropout rates. The balanced datasets consist of 2500 genes, 4, 8, 16 and 32 clusters and dropout rates ranging from 5% to 30% (shape = -1, type = “experiment”, facScale = 0.2 and mid in [-1, 0, 1, 1.5]). A constant cluster size of 250 samples has been employed and the total sample size is thus proportional to the number of embedded clusters (i.e. 4 * 250 to 32 * 250). The imbalanced datasets consist of 2500 genes, 3000 cells, 4, 8, 16 and 32 clusters with size ratio ranging from 0.6 to 0.01 and dropout rates ranging from 5% to 30%.

We started by analyzing the performance of the presented methods (a, b, c) on the generated datasets. For the ensemble representation (c) both K-Means and Leiden community detection have been explored. On balanced datasets (Figure 3 a, b) all methods achieve similar performances. However, on imbalanced datasets (Figure 3 c, d) the ensemble method using Leiden community detection provides better results than K-means, which can be explained by the tendency of the latter to identify equal sized clusters. As information about class ratio is not available when analyzing biological datasets, we consider the last method (ensemble using Leiden) as best performing and we refer to it as *contrastive-sc*.

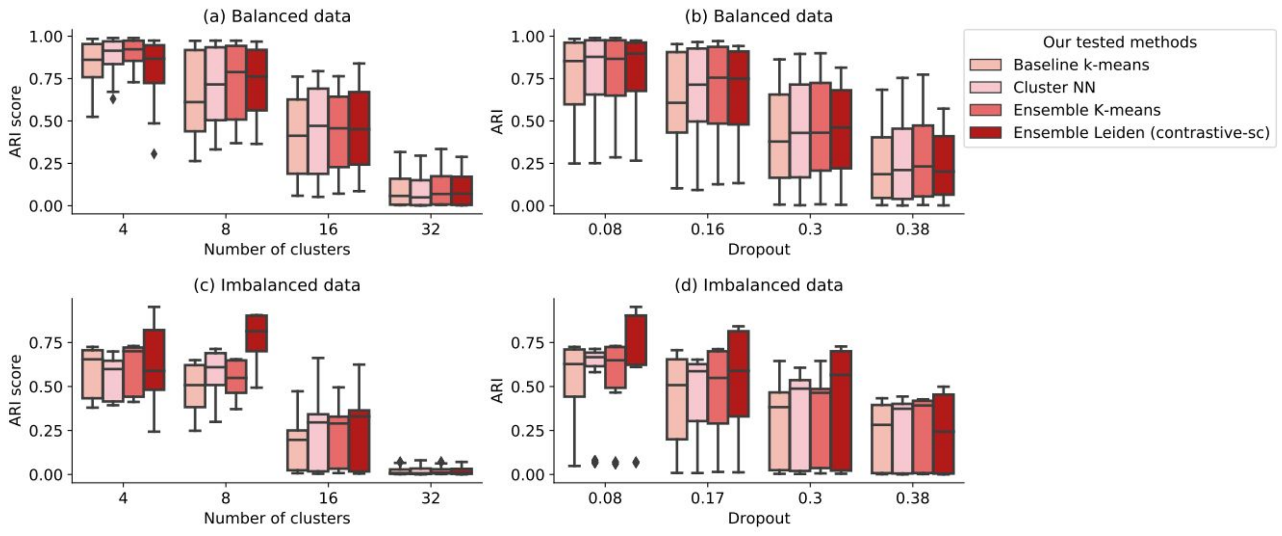

Figure 3. Comparison of our tested methods on balanced (a, b) and imbalanced data (c, d). The ensemble method using Leiden community detection outperforms the other methods on imbalanced data, when the dropout rate or the number of clusters are increasing. The plot depicts the results of 3 runs for each dataset using different initialization seeds.

Next, the proposed method has been compared with a baseline method consisting of a PCA followed by a K-means clustering as well as the following state of the art methods: scDeepClustering[11], soft K-means (sczi) [14] and Seurat[8] / Scanpy [9]. Similar to sczi, our method employs a filtering step that selects the 500 most variable genes and the training was carried out for 500 epochs. To assess the validity of this selection step, we performed an analysis of the impact of the selected number of genes on the final model performance.

In the case of balanced data, our method compares favorably to the state-of-the-art candidates when the number of clusters or the dropout rate is increasing (Figure 4). When the number of clusters grows beyond 30 and the dropout rate above 0.2, none of the methods manages to correctly find the input clusters. Our method appears to be more stable than the analyzed alternatives (e.g. scDeepCluster, sczi). In the case of imbalanced stimulated data, *contrastive-sc* outperforms the other methods (Figure 5). A similar loss of performance across methods is observed as the number of clusters or the dropout rate increases, but contrastive-sc maintains good performance for up to 16 clusters.

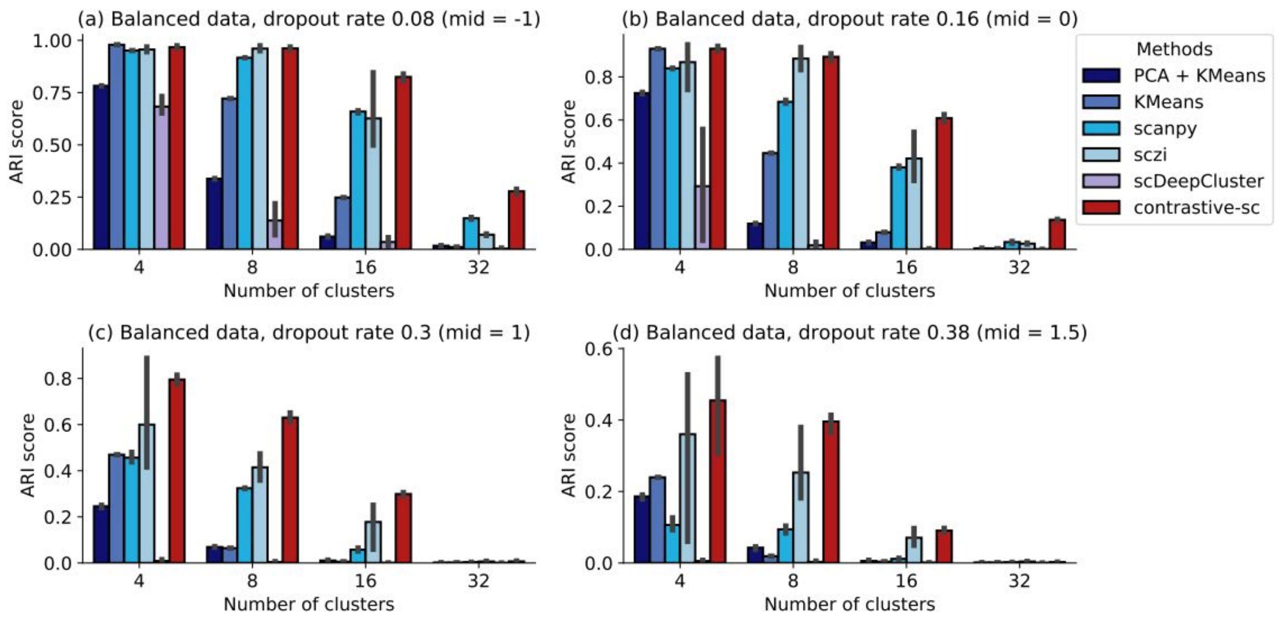

Figure 4. Method comparison on balanced simulated data for 4 levels of dropout rate: 0.08 (a), 0.16 (b), 0.3 (c) and 0.38 (d). For each experiment, the Adjusted Rand Index score has been computed. All methods display a loss of performance as the number of clusters or the dropout rate are increasing, but *contrastive-sc* appears to be more stable. The plot depicts the results of 3 runs for each dataset using different initialization seeds.

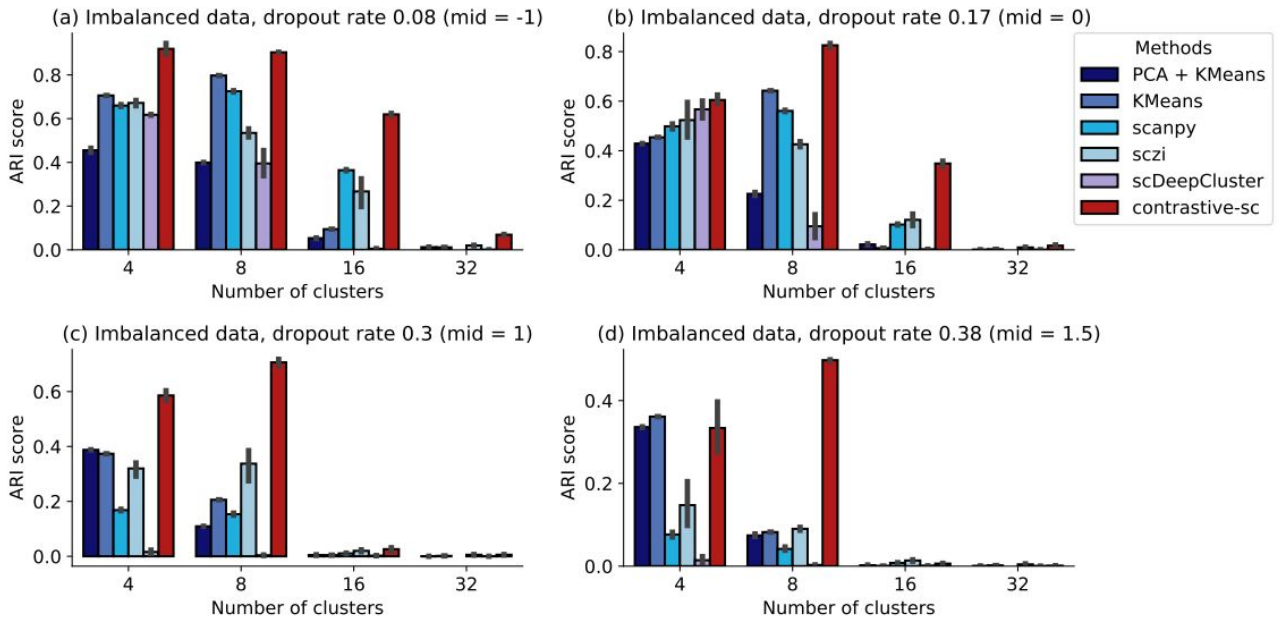

Figure 5. Method comparison on imbalanced simulated data for 4 levels of dropout rate: 0.08 (a), 0.16 (b), 0.3 (c) and 0.38 (d). For each experiment, the Adjusted Rand Index score has been computed. All methods display a loss of performance as the number of clusters or the dropout rate are increasing. Similar to the balanced data, the performance of all methods degrades when increasing the number of clusters or the dropout rate. The plot depicts the results of 3 runs for each dataset using different initialization seeds.

Analysis of scRNA-seq datasets

The collection scRNA-seq datasets published along with [11] have been selected for benchmarking the model performance (see Table 1 for details). The data had been collected using four different sequencing platforms: 10X genomics platform for the PBMC cells [22], droplet barcoding for mouse embryonic stem cells [23], Microwell-seq for mouse bladder cells [24] and sci-RNA-seq for worm neuron cells[25]. All datasets are imbalanced and have 4 to 16 annotated clusters. The preprocessed input dataset contained the top 1500 most variable genes and the model trained for 100 epochs. *contrastive-sc* performs best on the 10X PBMC 8 clusters datasets and favorably on the Worm neuron cell and Mouse bladder cell datasets (Figure 6). The results are highly dependent of the input data and should be independently extended to draw a final conclusion. We also replicated the experiment conducted in the associated paper [11], which compared for each dataset, the results using all cells with those of a random selection of 2100 cells from each dataset. Our experiments also revealed that the random selection of cells does not affect the performance of *contrastive-sc* (Figure S4).

Dataset name	Nb cells	Nb genes	Nb clusters	Cluster sizes
10 PBMC	4271	16653	8	From 135 to 1292
Mouse ES cells	2717	24175	4	From 303 to 933
Worm neuron cell	4186	13488	10	From 70 to 1015
Mouse bladder cell	2746	20670	16	From 13 to 717

Table 1. Description of scRNA-seq datasets used for benchmarking.

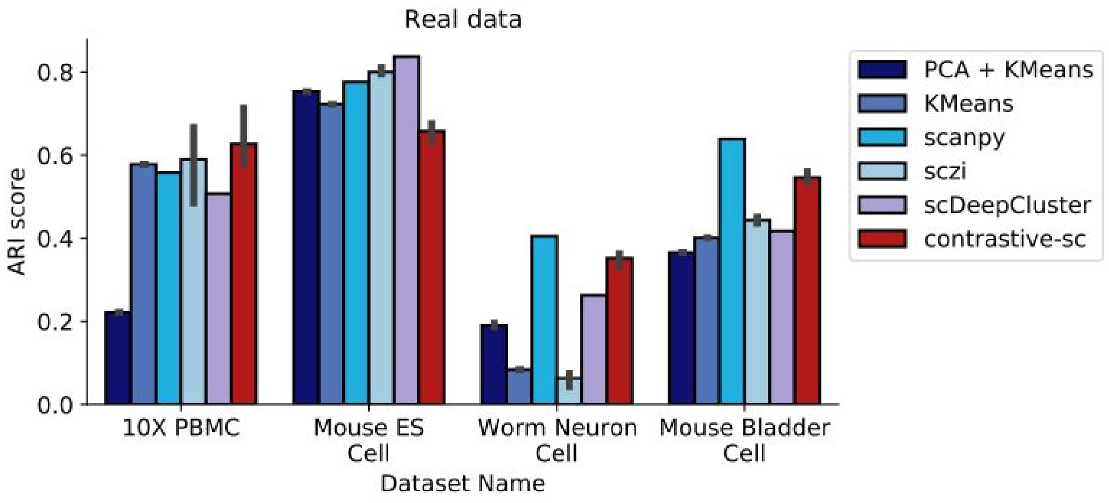

Figure 6. Method results on biological datasets. *contrastive-sc* performs best on the 10X PBMC 8 clusters datasets. The random selection of cells doesn't have a significant impact on any of the tested methods. The plot depicts the results of 3 runs using different initialization seeds.

To summarize the results across all sets of experiments (Figure 7): our method provides the best results on both simulated balanced and imbalanced datasets. Our experiments show that there is no consensus regarding the best state of the art method, the result being heavily dependent on the input dataset. From all proposed contrastive-based methods, the best results are achieved on average with the combined ensemble of representations followed by Leiden community detection (*contrastive-sc*, Figure S4).

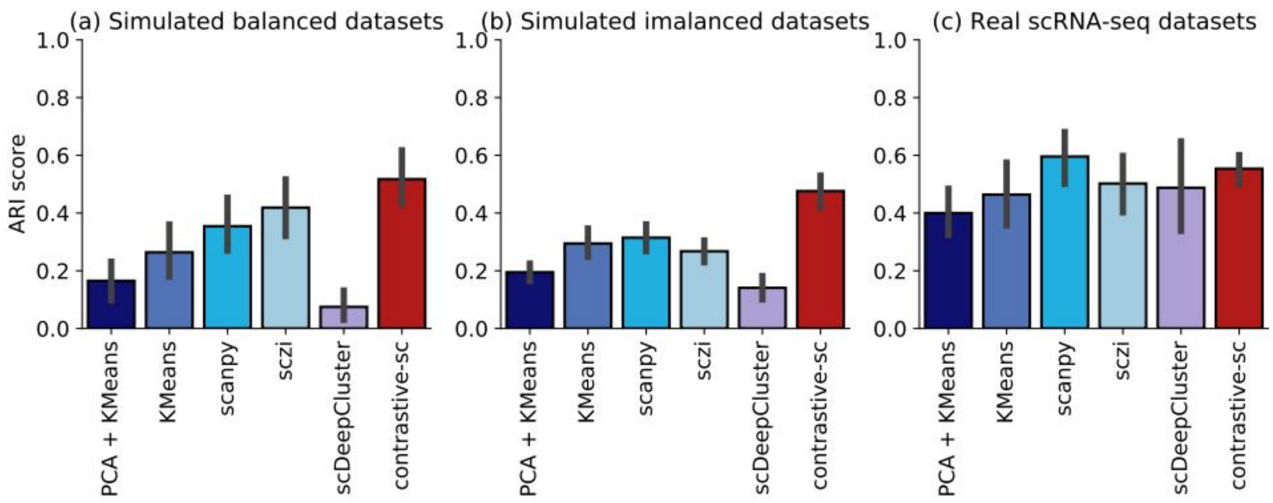

Figure 7. Average results of state-of-the-art methods over all explored simulated balanced (a), simulated imbalanced (b) and real scRNA-seq datasets. *contrastive-sc* brings an important performance gain on imbalanced datasets.

DISCUSSION

The encouraging experimental results presented in the previous section show that self-supervised contrastive learning constitutes a good alternative to the analytical way of modeling the dropout in scRNA-seq data, using NB or ZINB autoencoders ([10] [11] [14]). Moreover, the performance gain of employing our method is more significant as the dropout rate increases (Figures 4 c, d and 5 c, d).

Analysis of data augmentation techniques

We analyzed the importance of the data augmentation techniques by running the method on the simulated data while employing various ratios of kept data. This technique is equivalent to a neural network dropout applied on the input layer. Similar to the results reported on image classification ([15] [26]), contrastive learning performs best when employing strong data augmentation, here by using only up to 20% of the input features size (Figure 8 a).

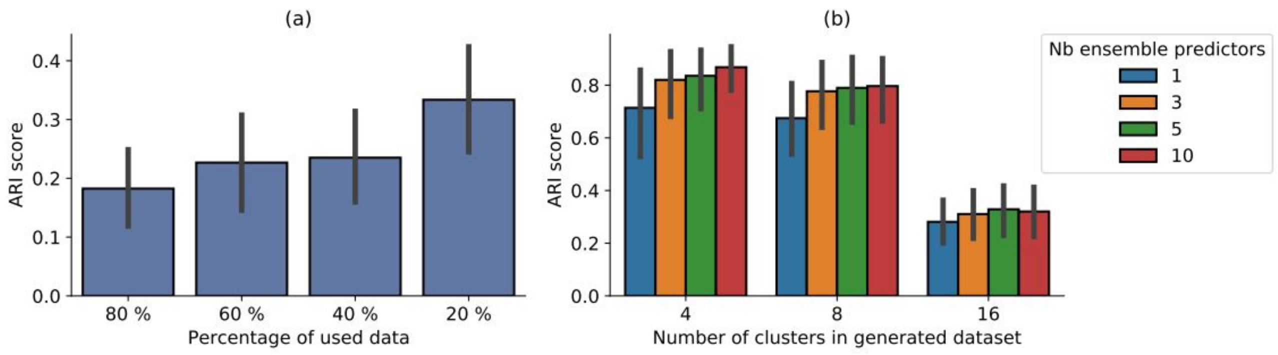

Figure 8. Impact of data augmentation (a) and number of predictors (b). Impact of the amount of data used as data augmentation (a). The best performance is recorded when only up to 20% of selected genes are being used. Impact of the number of predictors on the results of the ensemble method (b). There is no significant improvement as the number of predictors increases. These experiments have been performed on the simulated data described previously.

For visual representation, Chen et al. [15] recommend using larger batch sizes (e.g. 4096 samples). However, the studied scRNA-seq datasets have on average less than 4000 samples. In our experiments we employed a fixed batch size of 300 samples. We foresee exploring in future works more recent and improved approaches to representation learning, leveraging techniques providing state of the art results with smaller batch sizes, such as the Moco frameworks [27] [28].

Impact of the number of estimators in the representation ensemble

The performance of the baseline model (a) can be improved using the representation ensembling technique of creating a combined embedding. We performed experiments using up to 10 predictors. The results depicted in Figure 8 b show that there is only a small improvement when choosing more than 3 predictors. The performance gain of using ensembles instead of a single predictor is decreasing when the number of input clusters grows.

Employing multiple representation learners improves the score with respect to the baseline but introduces an additional computational cost.

Impact of the selected clustering algorithm

Employing Leiden clustering on the learned embedding outperforms K-means and brings a significant accuracy gain especially on imbalanced datasets. This behavior can be explained by the fact that community detection is not biased towards identifying equal sized clusters. Moreover, from a usage perspective, Leiden clustering does not require prior knowledge about the expected number of clusters, which makes it more suitable to being employed in a knowledge discovery setting. Our experiments employed the Leiden implementation integrated in the Scanpy package [9]. The Leiden [20] graph-clustering performs community detection based on optimizing modularity and processes the neighborhood graph of cells. In the case of Scanpy, the graph is computed using a selected PCA representation of the data matrix while our method uses directly the learned embedding. As depicted in Figures 4 and 5 the differences between our method and Scanpy highlight the importance of the representation learning phase.

Impact of the number of selected genes

Next, the impact of the number of selected genes in the input dataset has been analyzed on both simulated and real data. We performed experiments on balanced data where the full set or only top 100, 200, 500, 1000, 1500 most variable genes have been selected from the original dataset. The best performance is achieved when using top 500 most variable genes (Figure 9), result which is also in line with the default settings employed in [11] [14]. Using the entire dataset brings a loss of performance which can be explained by the inclusion of low-expressed genes, more affected by dropout events.

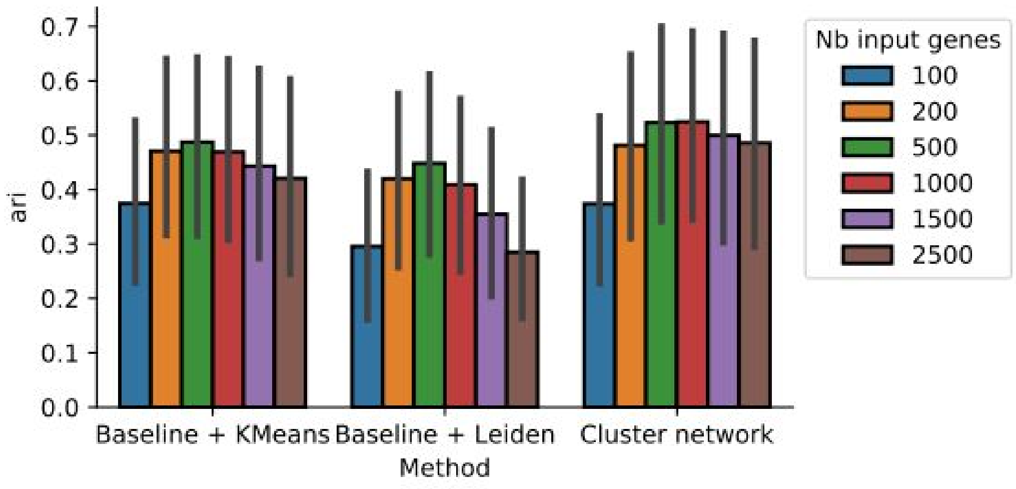

Figure 9. Impact of the number of selected genes on simulated data. A selection of the top variable genes (100 to 2500) was used. The model performs best when the top 500 most variable genes are selected as input to the representation phase. The presented balanced simulated data has been employed for this experiment, having a total of 2500 genes.

We conducted the same experiment with scRNA-seq data. Because real datasets have more genes (from 13488 to 24175) than the simulated data, we extended the selection range to 8000. The optimal number of genes is dataset specific but on average the best results are achieved when using above 1000 genes (Figure 10). The performance decreases when too many genes with low expression level are included, as shown in the transition from 5000 to 8000 genes. A value of 1500 genes has been employed in our benchmark results.

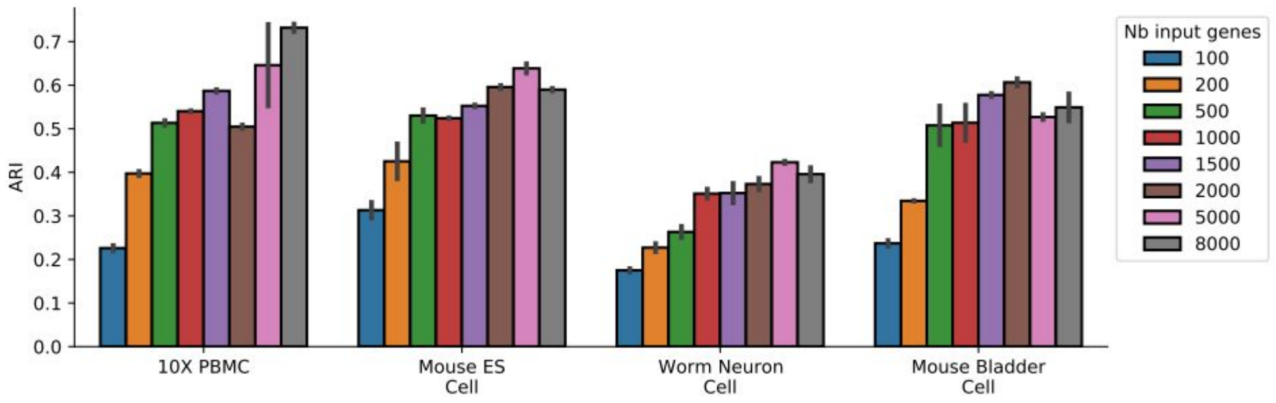

Figure 10. Impact of the number of selected genes on real data. A selection of the top variable genes (100 to 8000) was used. The best scores are achieved when using above 1000 genes, however, on most datasets the performance declines when using more than 5000 genes, which corresponds to including many genes with low level of expression. The depicted results correspond to two runs of our method for each input size.

Computational analysis

The training time grows linearly with the number of samples. Both the representation training and the community detection methods are insensitive to the number of input clusters. The training time to learn meaningful representations has been analysed in Figure 11 in terms of the model performance recorded after an arbitrary number of epochs (ranging from 5 to 300), averaged on 2 different runs. Our results show that 50 epochs are enough to learn meaningful representations and that continuing the training changes the performance only marginally. As a comparison, other methods (e.g. sczi, scDeepCluster) have been configured with a default of 500 epochs for the data denoising phase.

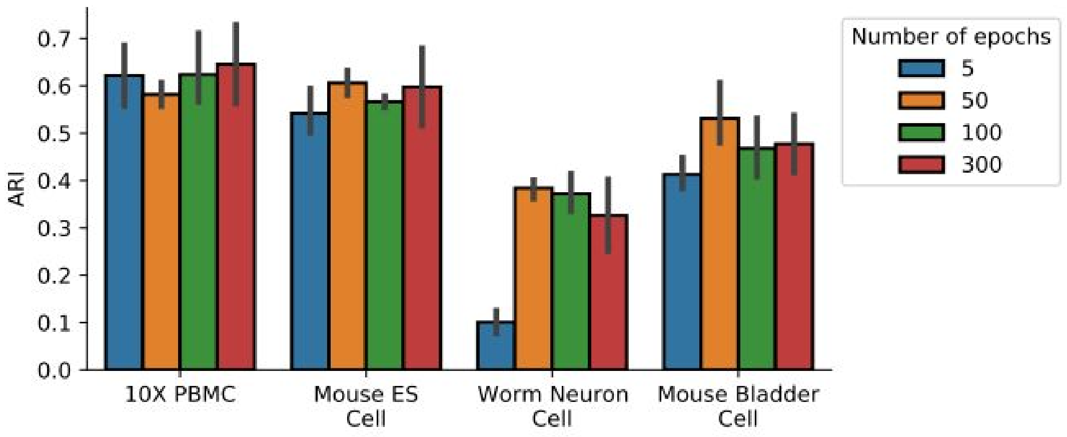

Figure 11. Analysis of training convergence times (as number of epochs) on biological datasets. For most datasets 50 epochs are enough for the model to learn meaningful representations, which brings a computational speed gain compared to other deep-learning competitor methods. This plot depicts the results of two runs of the proposed method for each of the selected number of epochs.

Future works

Following the first set of experiments presented in this publication, we foresee a number of possible optimizations. One promising idea is to leverage the cluster assignment predictions of the representation embedding as pseudo-labels and formulate the clustering problem as a classification task under label noise [29]. The soft K-means clustering network can be optimized by adding to the loss a representation learning term in order to ensure that the clustering does not alter the local structure of data. Another topic which has been studied only in a few publications [30] is the early stopping condition for representation learning. Most publications [15] [28] [26] propose a dataset-related number of epochs and do not elaborate on the evolution of performance or overfitting. We foresee an investigation in terms of the stability of learned representations which could be used as an early stopping condition. Last but not least, the current method could be extended and applied to other types of omics data (e.g. bulk RNA-seq, DNA methylation) or in a different context.

Conclusion

In this paper, we introduced, *contrastive-sc*, a new method leveraging contrastive self-supervised learning for clustering scRNA-seq data. We analyzed several strategies to cluster the learned embedding and promising results were achieved when employing Leiden clustering on the combined embedding of multiple representation learning estimators. We conducted experiments to illustrate the importance of the data augmentation strategy, the impact of the input size, of imbalanced cluster sizes, of the number of estimators in the combined representation. The main contributions can be summarized as follows:

- Adaptation of contrastive self-supervised training to scRNA-seq data and analysis of data augmentation techniques to integrate in the representation learning process. All explored methods showed encouraging results across all analyzed datasets and showed that contrastive self-supervised learning is a promising direction to explore in future bioinformatics research;

- Analysis of several clustering strategies by directly applying an arbitrary clustering algorithm on the learned embedding or on the combined representation of multiple estimators, as well as by leveraging a soft K-means clustering network. Our experiments showed that the best results are achieved when using a combined embedding and Leiden clustering.

In future works, we foresee an improvement of the baseline model by exploring the most recent techniques for representation learning [28] as well as an optimization of the clustering network to avoid the need for ensemble learning. This approach would provide a computational gain when analyzing large sample datasets and bypass the need for traditional clustering.

DATA AVAILABILITY

All data needed to reproduce the presented results has been made available on GitHub (https://github.com/ciortanmadalina/contrastive-sc).

Abbreviations

D – input dataset

n – number of observations

d – number of dimensions (features)

ARI – Adjusted Rand Index

Kullback-Leibler – KL

Keywords

Contrastive learning, Clustering, Single cell RNA-seq, Self-training, Deep learning, Optimization.

Declarations

Ethics approval and consent to participate

Not applicable.

Consent for publication

Not applicable.

FUNDING

Not applicable.

Authors' contributions

MC developed the method, analyzed and interpreted the data. MC and MD contributed in writing the manuscript. All authors read and approved the final manuscript.

Acknowledgments

Not applicable.

REFERENCES

[1] A. A. Kolodziejczyk, J. K. Kim, V. Svensson, J. C. Marioni, and S. A. Teichmann, "The Technology and Biology of Single-Cell RNA Sequencing," *Molecular Cell*, vol. 58, no. 4. Cell Press, pp. 610–620, May 21, 2015, doi: 10.1016/j.molcel.2015.04.005.

[2] V. Menon, "Clustering single cells: a review of approaches on high-and low-depth single-cell RNA-seq data. - PubMed - NCBI." https://www.ncbi.nlm.nih.gov/pubmed/29236955 (accessed May 08, 2020).

[3] V. Y. Kiselev, T. S. Andrews, and M. Hemberg, "Challenges in unsupervised clustering of single-cell RNA-seq data," *Nat. Rev. Genet.*, vol. 20, no. 5, pp. 273–282, 2019, doi: 10.1038/s41576-018-0088-9.

[4] S. Freytag, I. Lonnstedt, M. Ng, and M. Bahlo, "Cluster Headache: Comparing Clustering Tools for 10X Single Cell Sequencing Data," doi: 10.1101/203752.

[5] R. Qi, A. Ma, Q. Ma, and Q. Zou, "Clustering and classification methods for single-cell RNA-sequencing data," *Brief. Bioinform.*, no. May, 2019, doi: 10.1093/bib/bbz062.

[6] P. Lin, M. Troup, and J. W. K. Ho, "CIDR: Ultrafast and accurate clustering through imputation for single-cell RNA-Seq data," *bioRxiv*, p. 068775, Jan. 2017, doi: 10.1101/068775.

[7] D. Grün *et al.*, "Single-cell messenger RNA sequencing reveals rare intestinal cell types," *Nature*, vol. 525, no. 7568, pp. 251–255, Sep. 2015, doi: 10.1038/nature14966.

[8] R. Satija, J. A. Farrell, D. Gennert, A. F. Schier, and A. Regev, "Spatial reconstruction of single-cell gene expression data," *Nat. Biotechnol.*, vol. 33, no. 5, pp. 495–502, May 2015, doi: 10.1038/nbt.3192.

[9] F. A. Wolf, P. Angerer, and F. J. Theis, "SCANPY: Large-scale single-cell gene expression data analysis," *Genome Biol.*, vol. 19, no. 1, p. 15, Feb. 2018, doi: 10.1186/s13059-017-1382-0.

[10] G. Eraslan, L. M. Simon, M. Mircea, N. S. Mueller, and F. J. Theis, “Single-cell RNA-seq denoising using a deep count autoencoder,” *Nat. Commun.*, vol. 10, no. 1, pp. 1–14, Dec. 2019, doi: 10.1038/s41467-018-07931-2.

[11] T. Tian, J. Wan, Q. Song, and Z. Wei, “Clustering single-cell RNA-seq data with a model-based deep learning approach,” *Nat. Mach. Intell.*, vol. 1, no. 4, pp. 191–198, 2019, doi: 10.1038/s42256-019-0037-0.

[12] J. Xie, R. Girshick, and A. Farhadi, “Unsupervised Deep Embedding for Clustering Analysis,” 2016. Accessed: Jul. 31, 2020. [Online]. Available: https://github.com/piiswrong/dec.

[13] X. Li *et al.*, “Deep learning enables accurate clustering with batch effect removal in single-cell RNA-seq analysis,” *Nat. Commun.*, vol. 11, no. 1, pp. 1–14, Dec. 2020, doi: 10.1038/s41467-020-15851-3.

[14] L. Chen, W. Wang, Y. Zhai, and M. Deng, “Deep soft K-means clustering with self-training for single-cell RNA sequence data,” *NAR Genomics Bioinforma.*, vol. 2, no. 2, Jun. 2020, doi: 10.1093/nargab/lqaa039.

[15] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, “A Simple Framework for Contrastive Learning of Visual Representations,” 2020. Accessed: Sep. 14, 2020. [Online]. Available: https://github.com/google-research/simclr.

[16] W. Van Gansbeke, S. Vandenhende, S. Georgoulis, M. Proesmans, and L. Van Gool, “SCAN: Learning to Classify Images without Labels,” 2020, Accessed: Sep. 04, 2020. [Online]. Available: http://arxiv.org/abs/2005.12320.

[17] Y. Tian, D. Krishnan, G. Research, and P. Isola, “Contrastive Multiview Coding.” Accessed: Oct. 24, 2020. [Online]. Available: http://github.com/HobbitLong/CMC/.

[18] R. M. Suresh, K. Dinakaran, and P. Valarmathie, “Model based modified k-means clustering for microarray data,” in *Proceedings - 2009 International Conference on Information Management and Engineering, ICIME 2009*, 2009, pp. 271–273, doi: 10.1109/ICIME.2009.53.

[19] G. Eraslan, L. M. Simon, M. Mircea, N. S. Mueller, and F. J. Theis, “Single-cell RNA-seq denoising using a deep count autoencoder,” *Nat. Commun.*, vol. 10, no. 1, pp. 1–14, Dec. 2019, doi: 10.1038/s41467-018-07931-2.

[20] V. A. Traag, L. Waltman, and N. J. van Eck, “From Louvain to Leiden: guaranteeing well-connected communities,” *Sci. Rep.*, vol. 9, no. 1, pp. 1–12, Dec. 2019, doi: 10.1038/s41598-019-41695-z.

[21] L. Zappia, B. Phipson, and A. Oshlack, “Splatter: Simulation of single-cell RNA sequencing data,” *Genome Biol.*, vol. 18, no. 1, p. 174, Sep. 2017, doi: 10.1186/s13059-017-1305-0.

[22] G. X. Y. Zheng *et al.*, “Massively parallel digital transcriptional profiling of single cells,” *Nat. Commun.*, vol. 8, no. 1, pp. 1–12, Jan. 2017, doi: 10.1038/ncomms14049.

[23] A. M. Klein *et al.*, “Droplet barcoding for single-cell transcriptomics applied to embryonic stem cells,” *Cell*, vol. 161, no. 5, pp. 1187–1201, May 2015, doi: 10.1016/j.cell.2015.04.044.

[24] X. Han *et al.*, “Mapping the Mouse Cell Atlas by Microwell-Seq,” *Cell*, vol. 172, no. 5, pp. 1091-1107.e17, Feb. 2018, doi: 10.1016/j.cell.2018.02.001.

[25] J. Cao *et al.*, “Comprehensive single-cell transcriptional profiling of a multicellular organism,” *Science (80-.).*, vol. 357, no. 6352, pp. 661–667, Aug. 2017, doi: 10.1126/science.aam8940.

[26] P. Khosla *et al.*, “Supervised Contrastive Learning,” 2020, Accessed: Sep. 10, 2020. [Online]. Available: http://arxiv.org/abs/2004.11362.

[27] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, “Momentum Contrast for Unsupervised Visual Representation Learning.” Accessed: Oct. 27, 2020. [Online]. Available: https://github.com/facebookresearch/moco.

[28] X. Chen, H. Fan, R. Girshick, and K. He, “Improved Baselines with Momentum Contrastive Learning,” 2020, Accessed: Oct. 27, 2020. [Online]. Available: http://arxiv.org/abs/2003.04297.

[29] H. Song, M. Kim, D. Park, and J.-G. Lee, “Learning from Noisy Labels with Deep Neural Networks: A Survey,” 2020, Accessed: Aug. 22, 2020. [Online]. Available: http://arxiv.org/abs/2007.08199.

[30] M. Li, M. Soltanolkotabi, and S. Oymak, “Gradient Descent with Early Stopping is Provably Robust to Label Noise for Overparameterized Neural Networks,” 2019.

Contrastive self-supervised clustering of scRNA-seq data

Supplementary materials

Madalina Ciortan, Matthieu Defrance

Interuniversity Institute of Bioinformatics in Brussels

Université Libre de Bruxelles, Belgium

Corresponding author: matthieu.dc.defrance@ulb.ac.be

Alternative clustering methods

This section details alternative methods presented but not detailed in the main manuscript. The baseline method (a) directly clusters the embedding with an arbitrary clustering algorithm. In our work, both K-Means and Leiden community detection have been employed, as illustrated in Figure S1.

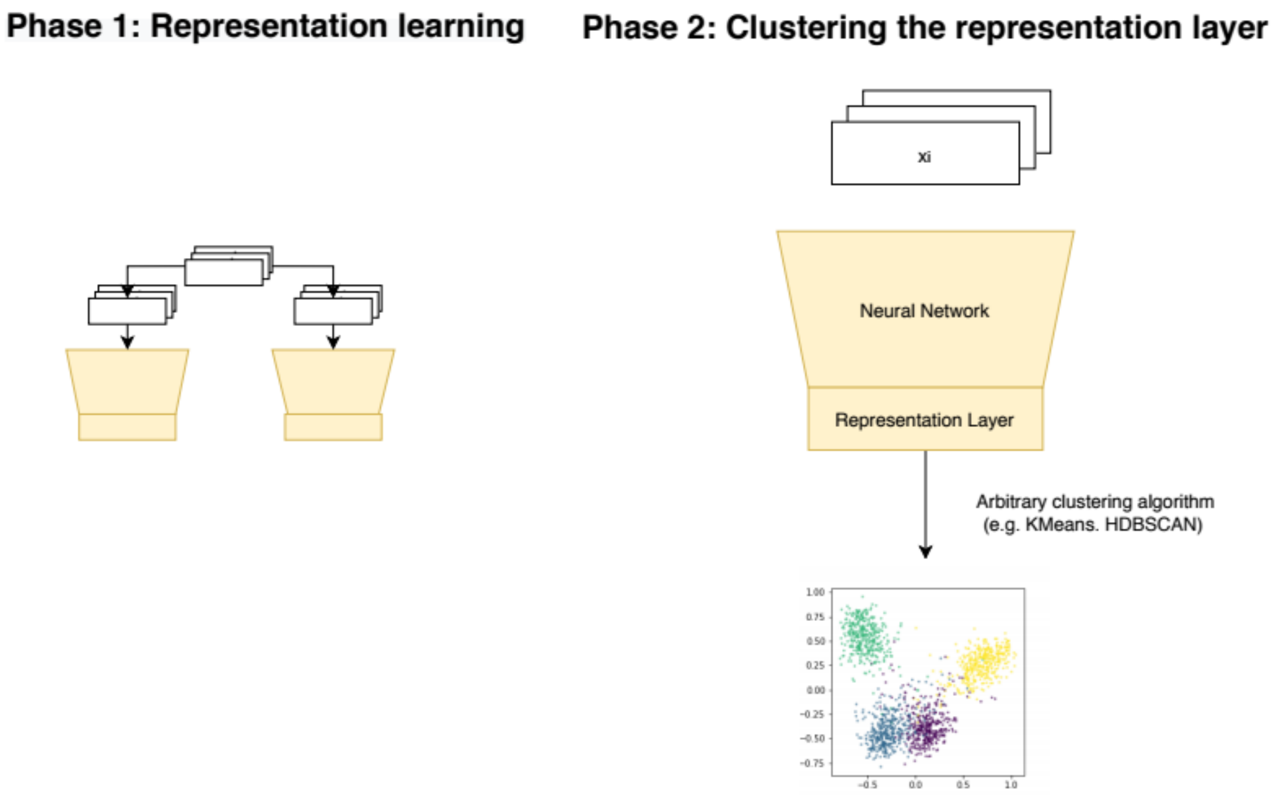
Phase 1: Representation learning
Phase 2: Clustering the representation layer
xi
Neural Network
Representation Layer
Arbitrary clustering algorithm
(e.g. KMeans. HDBSCAN)

Figure S1. Baseline method model consists in clustering directly the representation layer using an arbitrary clustering algorithm. In our experiments, both K-Means and HDBSCAN have been employed.

The second approach (Clustering network (b), Figure S2) consists in leveraging a soft clustering loss on the representation space. Given that Z_i is the learned embedding of x_i, the model will cluster the data into K clusters with centers $v_r (1 \leqslant r \leqslant K)$, where K is a user input, and the initial values of the cluster centers are computed by clustering Z_i. A combined loss has been employed consisting of a linear combination between a weighted K-Means loss and KL divergence loss:

$$L_{total} = L_{KMeans} + L_{KL}$$

$L_{KMeans} = \sum_{i=1}^{n} \sum_{r=1}^{K} w_{ir} \, \|Z_i - v_r\|^2$where

$$v_r = \frac{\sum_{i=1}^{n} w_{ir} Z_i}{\sum_{i=1}^{n} w_{ir}}, w'_{ir} = \frac{exp(-\|Z_i - v_r\|^2)}{\sum_{k=1}^{K} exp\,(-\|Z_i - v_k\|^2)}, w_{ir} = \frac{w'^{\alpha}_{ir}}{\sum_{j=1}^{K} w'^{\alpha}_{ij}}$$

and $L_{KL} = KL(q \parallel p) = \sum_i \sum_j q_{j|i} \, log \frac{q_{j|i}}{p_{j|i}}$

We have explored adding to the above defined loss the contrastive loss used in the representation learning $L^{contrastive}$, but our experiments revealed that this combination destabilizes the convergence.

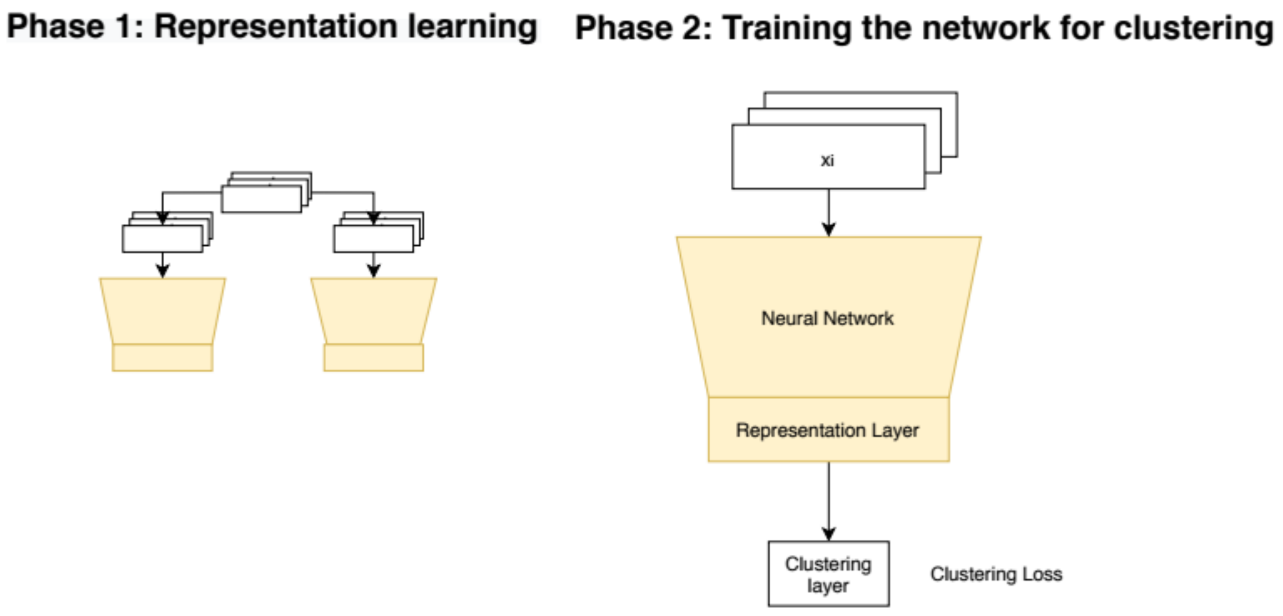
Phase 1: Representation learning
Phase 2: Training the network for clustering
xi
Neural Network
Representation Layer
Clustering layer
Clustering Loss

Figure S2. Clustering network implementing soft KMeans clustering on the representation layer. The employed loss is a linear combination between the weighted soft KMeans loss and the KL divergence

Another approach has been investigated but left aside due to its high computational complexity with the number of cells. As depicted in Figure S3, the baseline approach can be applied on several models trained in parallel and return for each one a clustering prediction. A unified similarity matrix is computed by summing the co-association matrix of each clustering prediction. The similarity matrix allows to embed the samples on a graph and then to apply community detection algorithms, such as Louvain, in order to obtain the final sample cluster assignment prediction.

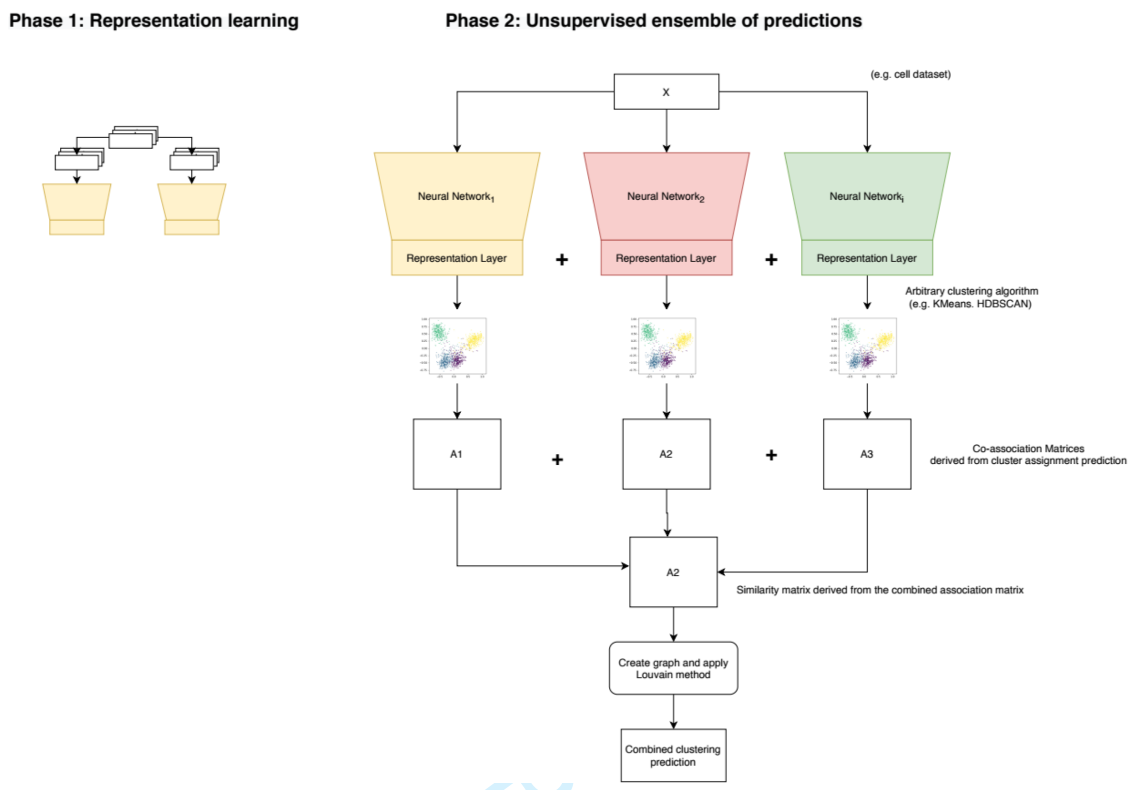
Phase 1: Representation learning
Phase 2: Unsupervised ensemble of predictions
X
(e.g. cell dataset)
Neural Network1
Neural Network2
Neural Networki
Representation Layer
Representation Layer
Representation Layer
+
+
Arbitrary clustering algorithm
(e.g. KMeans. HDBSCAN)
A1
A2
A3
Co-association Matrices
derived from cluster assignment prediction
A2
Similarity matrix derived from the combined association matrix
Create graph and apply Louvain method
Combined clustering prediction

Figure S3. Unsupervised prediction ensemble: Several models having the same architecture are trained in parallel. The combined representation is then clustered using an arbitrary clustering algorithm.

Supplementary Figures

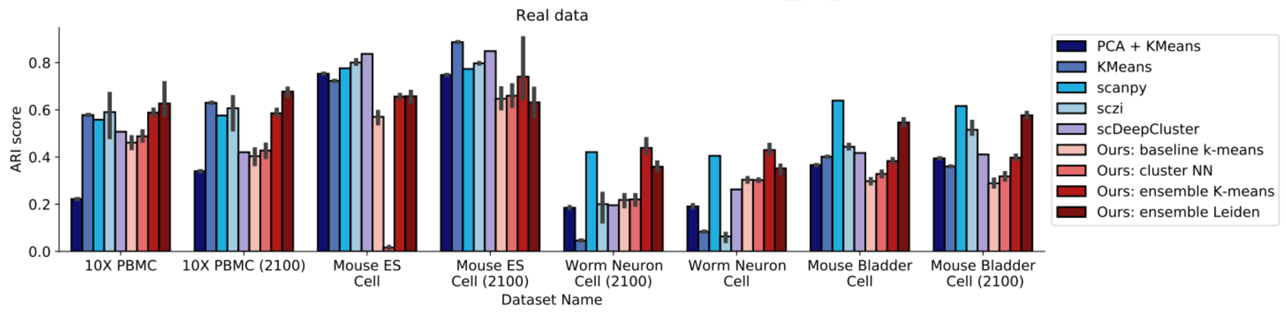

Figure S4. Method results on biological datasets. Leiden community detection outperforms K-means clustering. Our method performs best on the 10X PBMC 8 clusters datasets. The random selection of cells (2100 cells) doesn't have a significant impact on any of the tested methods. *contrastive-sc* is referred here as "Ours: ensemble Leiden".