EXPLORING VARIATIONAL AUTOENCODERS POTENTIAL TO CLASSIFY SINGLE-CELL RNA-SEQ DATA

Begoña Bolos[1], *Felix Pacheco* [1], *Paula Rodriguez*[1], *Laura Sans-Comerma*[1], *Ole Winther*[2,3]

[1] DTU Bioinformatics, Technical University of Denmark,[2] The Bioinformatics Centre, Department of Biology, University of Copenhagen,
[3] DTU Compute, Technical University of Denmark

ABSTRACT

Deep generative models, such as variational auto-encoders (VAEs) can be used to analyze single-cell RNA sequencing (scRNA-seq) data, generating latent representations (LRs) which capture most variability of gene expression levels. The library single-cell variational auto-encoder (scVAE) applies VAEs on raw scRNA-seq data, omitting all pre-processing steps and producing a latent representation useful for downstream analyses [1]. In this study, the potential of deep generative models such as scVAE has been evaluated on a cell type classification problem, using 10x-PBMC-68k data set [2]. For this purpose, a LR of the scRNA-seq data has been generated with scVAE and subsequently inputted into a feed-forward neural network (FFNN) model to classify the cell types. This model has been compared to other two FFNNs trained with (1) sparse raw data and (2) the components resulting from applying principal component analysis (PCA) on the sparse raw data. Contrary to our hypothesis, results do not show clear differences in classification accuracy among the three different training inputs. These findings should be further validated with larger sample sizes to truly assess the potential of VAEs to classify scRNA-seq data.

Index Terms— Deep generative models, variational auto-encoders, feed-forward networks, single-cell RNA-seq, cell type classification.

1. INTRODUCTION

The development of next-generation sequencing (NGS) technologies in the recent years has provided highly valuable insights into biological systems from cancer genomics to diverse microbial communities. Data resulting from RNA sequencing (RNA-seq), typically performed for the study of the transcriptome or analysis of gene expression levels, represents an average of gene expression patterns across thousands to millions of cells, which can hide potentially significant and biologically important differences between cells. scRNA-seq has taken transcriptome profiling to the next level, analyzing individual cells by uncovering their gene expression variability, providing a higher resolution of cellular differences [3].

Transcriptome profiling can be used to classify cell types, partitioning the data into clusters of single cells, where each cluster is defined by a unique gene expression signature relative to other clusters [4, 5]. However, the human body contains approximately 40 trillion cells and complex mammals contain $\sim$ 30,000 genes in their genomes. Hence, data from scRNA-seq experiments is often high dimensional, but also sparse, introducing challenges in computational analyses.

Dimensionality reduction is one of the approaches used to deal with the high-dimensional gene expression data obtained from scRNA-seq experiments, in which the data is projected into a lower dimensional space. PCA or t-distributed stochastic neighbor embedding (tSNE) are commonly used. Furthermore, machine learning algorithms have been used for identifying cell types as unsupervised clustering problem in different studies, including algorithms as K-means, hierarchical clustering, density-based clustering or graph-based clustering. Clustering algorithms are commonly combined with dimensionality reduction and/or feature selection to identify cell types [6, 7].

Recently, new methods that model the gene expression levels directly as counts have been developed, skipping the preprocessing for analyzing scRNA-seq data [8, 9]. Particularly, VAEs are deep generative models which use many different data distributions from the training data using unsupervised learning. VAEs learn compressed LRs of the data by denoising the input using an encoder-decoder structure. Thus VAEs capture most of the variance from the data by generating LRs with a lower dimensionality than the input. The reconstruction error is minimized by computing the evidence lower bound (ELBO) [10].

Some advantages of VAE approach are its probabilistic nature and the fact that different likelihood functions for modelling the data distributions can be applied. This method has been recently applied for modelling directly raw count data from RNA-seq [11] or scRNA-seq experiments for cell-type classification [1]. In the last case, the publicly available library scVAE offers the possibility of modelling scRNA-seq data to the user by choosing different the likelihood functions, and giving vast range of the hyperparameters values to tune [1].

In this study, the potential of VAE generative models will be explored on a cell type classification problem based on available framework for unsupervised modelling scVAE [1]. To do so, a LR of scRNA-seq data, generated with scVAE, will be used to train a FFNN to classify the cell types. These results will be compared with (1) a FFNN classifier trained with sparse raw data and (2) a FFNN classifier after applying dimensionality reduction through PCA.

2. METHODS

The main goal of this study is to validate the ability of VAEs to classify scRNA-seq data. To do so, a workflow was designed, as seen in figure 1. For the study, three data sets must be obtained: (1) raw data, (2) LRs of the raw data obtained by running scVAE and (3) resulting features from computing PCA on the raw data, obtaining the same number of components as LR dimensions used in data set (2). Subsequently, all three data sets will be fed to a simple FFNN to classify the cell type.

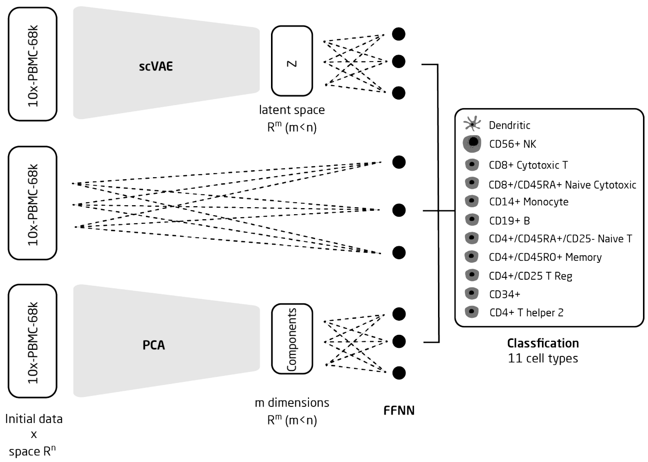

Fig. 1. Overview of the workflow

2.1. Data set

The data set used in this study is the peripheral blood mononuclear cells (PBMC) from 10X Genomics Fresh Donor A (10x-PBMC-68k), used in Zheng, et al. 2017 [2]. The data set consists of scRNA-seq data from $\sim 68,000$ cells, that checks counts for $\sim 32,000$ genes. It was sequenced using Illumina NextSeq 500 High throughput technologies with $\sim 20,000$ reads per cell, with a final number of reads of $\sim 1,400$M. The sequencing output was aligned using the *hg19* transcriptome. In supplementary figure S1, two tSNE projections of the data set can be observed.

Due to time and computational power restraints, it was decided that the study would be performed with a subset of the data set. The models and analyses were done with 2% of the total number of cells, being 1,300 cells. However, after seeing the results, new models were performed, this time with a subset of 25% of the total number of cells, being 15,000 cells. Nonetheless, it is important to note that only the LRs with the subset of 15,000 cells could be generated. The sparse, raw scRNA-seq subset of 15,000 cells could not be obtained despite considerable efforts and therefore PCA could not be performed on this subset either.

2.2. Variational Auto-Encoders

As mentioned above, scVAE was run on a subset of the 10x-PMBC-68k data set containing 1,300 cells using different hyperparameters. The different models were computed by combining two hyperparameters: the number of hidden units and the dimensions of the LRs. Combining the number of hidden units, ($H\epsilon[100, 250, 500, 1000]$); and the dimensions of the LRs, ($Z\epsilon[10, 25, 50, 100]$), 16 models were generated. Subsequently, 4 more scVAE models, using the outperforming number of hidden units for each of the four latent dimensions, were run on a larger subset of 10x-PMBC-68k data containing 15,000 cells.

Other hyperparameters were left fixed for all models, as shown in table 1. Since scRNA-seq data represents different cell types it is desirable to be able to classify it in different classes, or in this case, cell types. For this reason gaussian mixture variational auto-encoder (GMVAE) was used as the model to fit the output. The likelihood function chosen to model the count data was the negative binomial function, following the results shown in Grønbech, et al. 2020 [1].

Table 1. Fixed parameters for the all the trained models. (*) Split fraction refers to the fraction of the subset used for training. The remaining fraction is equally divided for validation and test.

Parameter	Fixed value
Model	GMVAE
Likelihood function	Negative binomial
Epochs	100
Split Fraction (*)	0.8
Clusters	11
Learning rate	10^{-4}
Hidden layers	1

2.3. Principal component analysis

In order to compare the generated LRs with simpler dimensionality reduction techniques, a subset of 10x-PMBC-68k data set containing 1,300 cells was used to perform PCA. For the reasons described in *Methods* section 2.1, PCA could only be performed on this subset, and not on the one with 15,000 cells. Firstly, data standardization was performed with the *StandardScaler* library, which standardizes features, or genes in this case, by removing the mean and scaling to unit variance. Secondly, *PCA* library was used in order to perform the dimensionality reduction of the subset [12]. In order to obtain a comparable output with scVAE latent representations, the number of components k, was set to $k\epsilon[10, 25, 50, 100]$. After applying PCA, the resulting components were stored to be used as input for the FFNN classifier.

2.4. Feed-forward neural network

Cell type classification was carried by a FFNN model, trained on the three data sets previously described. For each latent dimension, only the LR generated by the outperforming scVAE model was used, hence four scVAE input sets were used to train the network. Likewise, four PCA input sets were used to train the network, containing the same number of components as dimensions used in the LRs. FFNN architecture was constructed with PyTorch [13], using fixed parameter values shown in table 2 for all three input sets used. It was decided to maintain a simplistic FFNN architecture for all input sets, while letting most of the complexity to be included in scVAE model. Some parameters, such as Xavier Glorot initialization and Adam optimizer were selected based on literature [1, 9, 14, 15]. The data set was randomly split in 80% training, 10% validation and 10% test. Model training was performed with the fixed values contained in table 2.

Table 2. Fixed parameters for FFNN architecture and training. (*) ReLU function was used as activation function for the hidden layer, while Softmax function was used for the output layer. (**) The remaining fraction of the data set was equally divided for validation and test.

Architecture Parameters	Value
Initialization	Xavier Glorot
Hidden layers	1
Hidden units	516
Output classes	11
Activation function	ReLU/Softmax (*)

Training Parameters	Value
Batch size	10
Epochs	100
Training fraction (**)	0.8
Optimizer	Adam
Learning rate	10^{-4}
Loss criterion	Cross Entropy Loss

3. RESULTS

To study the benefits of using LRs to classify cell types with scRNA-seq data, a classification task was carried by a FFNN trained on the LRs generated with scVAE. The FFNN classifier was also trained with two other data inputs: (1) raw data and (2) the k principal components resulting from applying PCA. These different inputs were included in the present study to assess whether the use of LRs for classification outperforms the use of raw data or classical dimensionality reduction techniques such as PCA.

3.1. Generation of latent representations of the data

The LRs were evaluated based on the ELBO and Clustering Rand Index. These two parameters were used since they provide information on the training quality and how suitable is the LR for clustering cell types, respectively. The results of the different models can be seen in table 3. As it can be observed, the obtained Rand Index and ELBO values are considerably low.

Table 3. Initial scVAE 16 models test performance comparison from 1,300 cells subset. Z = latent dimensions; H = hidden units.

Model	Z	H	ELBO	Rand Index
1	10	100	-5971.4	0.033
2		250	-4420.9	0.118
3		500	-3139.4	0.058
4		1000	-2860.9	0.045
5	25	100	-5030.0	0.037
6		250	-3152.0	0.011
7		500	-2704.2	0.110
8		1000	-2605.3	0.090
9	50	100	-4519.8	0.098
10		250	-2868.5	0.111
11		500	-2567.1	0.073
12		1000	-2513.3	0.092
13	100	100	-4144.3	0.004
14		250	-2597.5	0.010
15		500	-2387.5	0.106
16		1000	-2437.8	0.014

Given that the models obtained to generate LRs showed low values of Rand Index and ELBO, the outperforming four scVAE models were re-computed with a larger subset of 10x-PBMC-68k data set, containing 15,000 cells. The results can be observed in table 4.

In figure 2, the learning curve and the tSNE visualization of

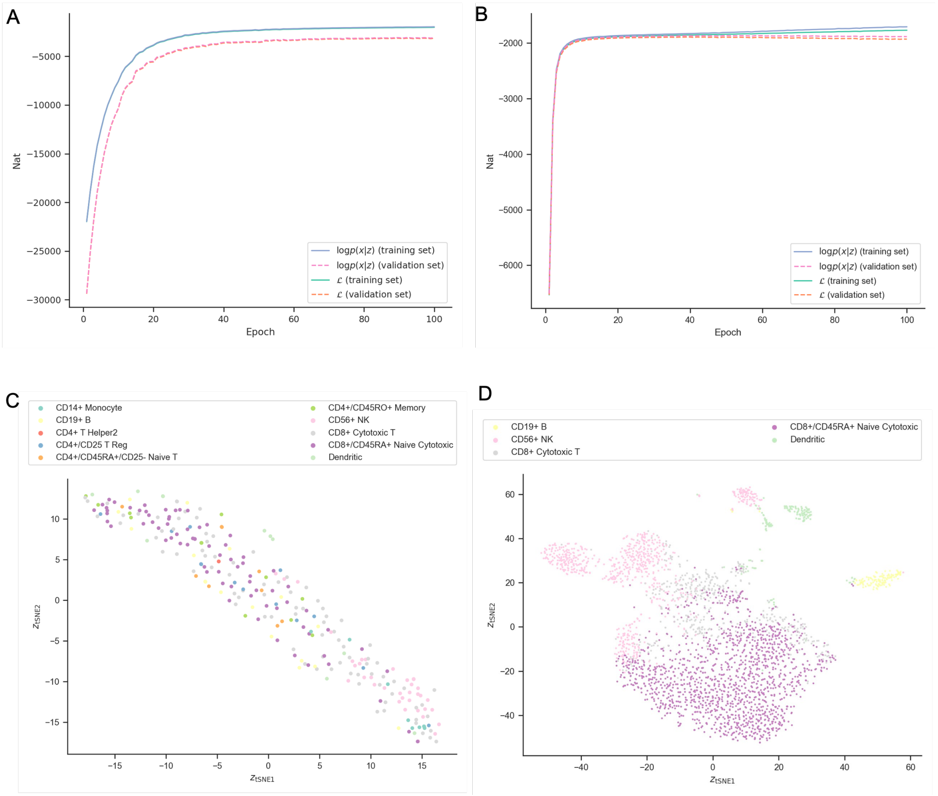

Fig. 2. scVAE model 15: learning curve (A) and tSNE visualization of the LR (C), scVAE model 20: learning curve (B) and tSNE visualization of the LR (D).

Table 4. Four scVAE models test performance from 15,000 cells subset. Z = latent dimensions; H = hidden units.

Model	Z	H	ELBO	Rand Index
17	10	250	-1871.9	0.169
18	25	500	-1890.5	0.188
19	50	250	-1876.4	0.236
20	100	500	-1880.7	0.212

the LR for models 15 and 20 are shown. It can be appreciated that the models trained with a larger subset of the data reach a higher value of ELBO. Concerning the clustering, model 20 also achieves a better Rand Index, which can be observed in the tSNE representation.

3.2. Cell type classification using FFNN

To assess the power of scVAE to classify scRNA-seq data, a FFNN was trained on (1) the raw count data, (2) the features resulting from applying PCA on the raw counts and (3) the LRs generated with scVAE. Four scVAE input sets were used, corresponding to the outperforming models in terms of ELBO, for each latent dimension tried ($Z\epsilon[10, 25, 50, 100]$). Likewise, for an adequate comparison, four input PCA sets were used for the FFNN model, containing same the number of components, k, as latent dimensions modelled ($k\epsilon[10, 25, 50, 100]$). All these input sets were generated using a subset of 1,300 cells of the original data set. A fourth type of input set was included afterwards, corresponding to the LRs generated with scVAE using a subset of 15,000 cells (see *Methods* section 2.1).

As it can be observed in table 5, the classification accuracy does not clearly differ among the input data used to train the FFNN. It was initially thought this outcome would be due to the small sample size used, only slight improvements were observed using as input the scVAE LRs generated with the subset of 15,000 cells (25% of original data set). Moreover, the number of dimensions in the PCA or the LR inputs does not cause any clear effect in the classification accuracy.

Figure 3 shows the learning curves and test classification results of the FFNN models trained scVAE outperforming model (Z=100, H=500), on PCA features with an equivalent number of components (k=100) and the raw data. The graphs corresponding to the rest of the FFNN models run can be found on supplementary figures S3-S6. The FFNN learning

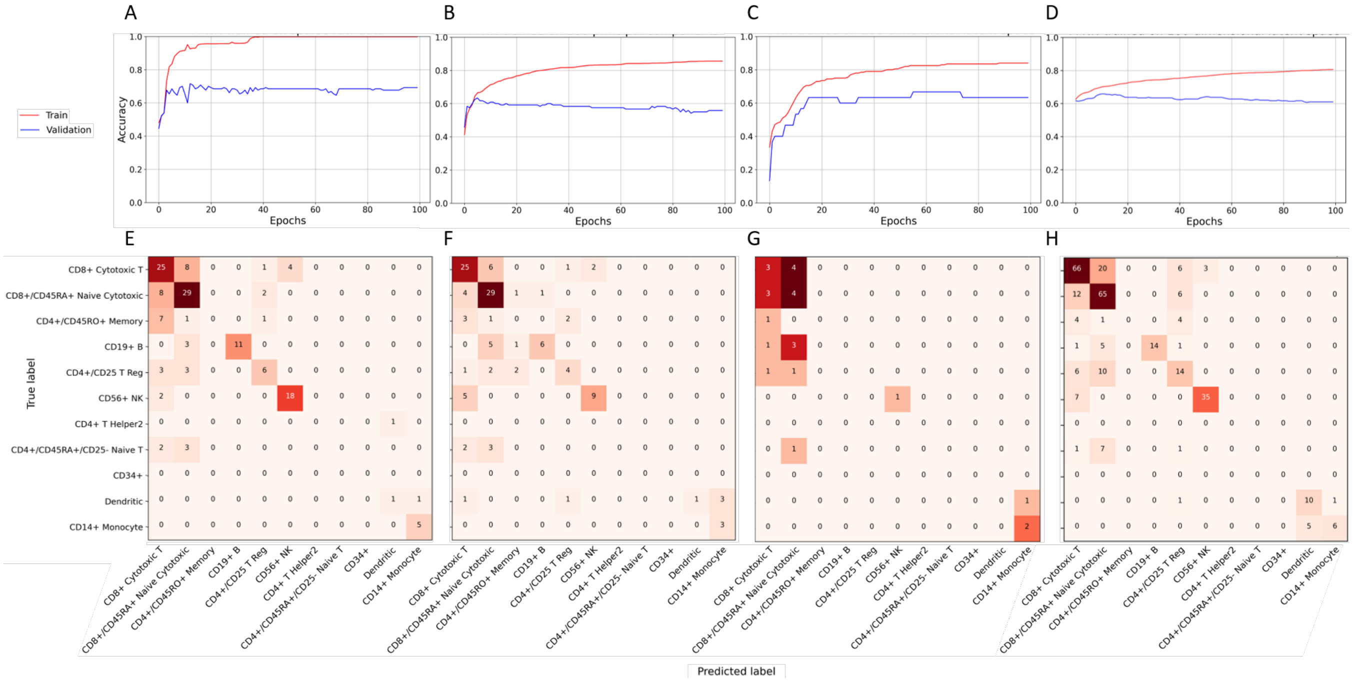

Fig. 3. FFNN classification results. A-D: learning curves of model trained on raw data (A), model trained on PCA features ($k = 100$) (B), model trained on outperforming scVAE LR ($Z = 100$, $H = 500$) (C), model trained on outperforming scVAE LR ($Z = 100$, $H = 500$) using the subset of 15000 cells (D). Red curve: training accuracy; blue curve: validation accuracy. E-F: confusion matrices of test set from the model trained on raw data (E), model trained on PCA features ($k = 100$) (F), model trained on outperforming scVAE LR ($Z = 100$, $H = 500$) (G), model trained on outperforming scVAE LR ($Z = 100$, $H = 500$) using the subset of 15,000 cells (H).

Table 5. Cell-type classification accuracy of test set from all FFNN models built. Model = scVAE model numbering used in tables 3 and 4; Subset = number of cells; Dim = number of components (k) or latent dimensions (Z); H = number of hidden units used in scVAE to generate the LRs.

Input	Subset	Model	Dim	H	Accuracy
raw	1,300	-	-	-	0.655
PCA	1,300	-	10	-	0.580
		-	25	-	0.556
		-	50	-	0.569
		-	100	-	0.621
scVAE	1,300	2	10	250	0.536
		7	25	500	0.622
		10	50	250	0.765
		15	100	500	0.585
scVAE	15,000	17	10	250	0.687
		18	25	500	0.656
		19	50	250	0.687
		20	100	500	0.673

curves shown in figure 3 reveal a considerably higher accuracy in the training process than in the validation. This event, slightly stronger using raw and PCA input, is a sign of overfitting, that does not seem to be solved by increasing the sample size, as observed in figure 3.D.

Lastly, confusion matrices shown in figure 3.E-H reveal a strong class imbalance: the majority of cell types in the data set corresponds to CD8+ cytotoxic T cells. This event might be another reason for the low performance observed.

4. DISCUSSION

The scVAE model trained on 1,300 cells does not achieve pure clusters and does not reach good values of ELBO. For all the range of parameters tested, there is not any apparent clustering when observing the tSNE plots from the small subsets. The rest of the parameters used were the optimal parameters reported by Grønbech et al. 2020 [1]. These results suggested that the selected subset of the data might need to be scaled-up, which led to the evaluation of a larger subset of 15,000 cells.

Modelling on the larger subset showed more differentiated clusters on the tSNE plots and the training reached ELBO values closer to zero. This outcome confirms that 1,300 cells are not enough to obtain meaningful LRs of the data. Regardless of the subset size, the LR dimension or the number of hidden units do not seem to have an effect on the quality of the computed models included in this study. To further test the importance of the sample size, it would be interesting to see which sizes represent a good trade-off between small subsets, and thus low computational intensity, and meaningful LRs.

Another aspect worth mentioning is that for all 20 models, regardless of the subset size, the learning curve reached a plateau before 50 epochs. This might also be an indicator that the sample size should be increased.

In spite of the low performance observed in scVAE models, the LRs of the outperforming models were used to train a FFNN to classify the cell types and compare the potential of scVAE LRs to other inputs, such as PCA components and raw data. Contrary to our hypothesis, the resulting classification accuracies do not show a clear difference among the three input sets used, and neither among the number of input dimensions used.

On one side, performing PCA prior to training the FFNN does not improve the classification accuracy compared to direclty using raw data. This could be due to the fact that the first 100 principal components can only explain approximately 20% of the variance, as shown in supplementary figure S2. This suggests the use of PCA might not be helpful in this case. On the other side, using the LR obtained from scVAE does not seem to improve the accuracy either, regardless of the subset size, by direct comparison to the PCA components. Moreover, a strong sign of overfitting has been observed in all FFNN models.

The subset size might be one of the main causes for this outcome, as the model trained on the LR generated with the larger subset shows a slight improvement in accuracy and overfitting compared to the model trained on the LR generated with the smaller subset. Unfortunately, due to unavailability of a raw subset of 15,000 cells, as described in *Methods* section 2.1, direct comparisons between the performance of a FFNN classifier using raw data, PCA components or scVAE LRs on the larger subset cannot be precisely done. Nevertheless, and in line with the insights revealed by scVAE models, these results still suggest that training the FFNN on a larger subset will lead to better performances.

The classification outcome could have also been affected by a strong class imbalance, probably resulting from sampling a small fraction of the original data set (2%). An assessment of class imbalance was thereby conducted and, as it can be observed in supplementary figure S7, an inherent class imbalance was observed in the complete data set. This is thus reflected in the subset, but it is not an effect resulting from sub-sampling. Still, this imbalance might be hampering the classification results.

Overall, we can not conclude in this study that the use of scVAE LRs improves the performance of cell-type classification. However, the findings obtained shed light onto the fact that increasing the sample size can lead to higher FFNN classification accuracies. Larger subsets, as mentioned in *Methods* section 2.1, could not be used in this study due to limited computational resources. Therefore, further studies following the same methodology but on larger data sets could be interesting and beneficial to appropriately assess the true potential of VAEs, and particularly scVAE, to classify high-dimensional, sparse data such as scRNA-seq data.

5. ACKNOWLEDGMENTS

The authors of the project would like to acknowledge the help and guidance on scVAE received from C. H. Grønbech, as well as the supervision from O. Winther.

6. AUTHOR CONTRIBUTIONS

All the authors of the project contributed equally to all parts of the project: (1) study design, (2) model creation and coding, (3) results interpretation and (4) poster design and article writing.

Begoña Bolós Sierra - s193036
Felix Pacheco Pastor - s192496
Paula Rodríguez García - s192448
Laura Sans Comerma - s192437

7. SUPPLEMENTARY MATERIALS

All the analyses and results conducted and generated in this project can be found online at 02456_scVAE repository. Supplementary figures can be found on the supplementary PDF document attached to this report. A poster containing the main insights about this work can be found in Github.

8. REFERENCES

[1] Christopher Heje Grønbech, Maximillian Fornitz Vording, Pascal N Timshel, Casper Kaae Sønderby, Tune H Pers, and Ole Winther, "scvae: variational auto-encoders for single-cell gene expression data," *Bioinformatics*, vol. 36, no. 16, pp. 4415–4422, 05 2020.

[2] Grace X.Y. Zheng, Jessica M. Terry, Phillip Belgrader, Paul Ryvkin, Zachary W. Bent, Ryan Wilson, Solongo B. Ziraldo, Tobias D. Wheeler, Geoff P. McDermott, Junjie Zhu, Mark T. Gregory, Joe Shuga, Luz Montesclaros, Jason G. Underwood, Donald A. Masquelier, Stefanie Y. Nishimura, Michael Schnall-Levin, Paul W. Wyatt, Christopher M. Hindson, Rajiv Bharadwaj, Alexander Wong, Kevin D. Ness, Lan W. Beppu, H. Joachim Deeg, Christopher McFarland, Keith R. Loeb, William J. Valente, Nolan G. Ericson, Emily A. Stevens, Jerald P. Radich, Tarjei S. Mikkelsen, Benjamin J. Hindson, and Jason H. Bielas, "Massively parallel digital transcriptional profiling of single cells," *Nature Communications*, vol. 8, no. 1, pp. 14049, 2017.

[3] Thale Kristin Olsen and Ninib Baryawno, "Introduction to single-cell RNA sequencing," *Current Protocols in Molecular Biology*, vol. 122, no. 1, Apr. 2018.

[4] Karthik Shekhar and Vilas Menon, "Identification of cell types from single-cell transcriptomic data," in *Methods in Molecular Biology*, pp. 45–77. Springer New York, 2019.

[5] Byungjin Hwang, Ji Hyun Lee, and Duhee Bang, "Single-cell rna sequencing technologies and bioinformatics pipelines," *Experimental & Molecular Medicine*, vol. 50, no. 8, pp. 96, Aug 2018.

[6] Tallulah S. Andrews and Martin Hemberg, "Identifying cell populations with scrnaseq," *Molecular Aspects of Medicine*, vol. 59, pp. 114 – 122, 2018, The emerging field of single-cell analysis.

[7] Azam Peyvandipour, Adib Shafi, Nafiseh Saberian, and Sorin Draghici, "Identification of cell types from single cell data using stable clustering," *Scientific Reports*, vol. 10, no. 1, pp. 12349, Jul 2020.

[8] Romain Lopez, Jeffrey Regier, Michael B. Cole, Michael I. Jordan, and Nir Yosef, "Deep generative modeling for single-cell transcriptomics," *Nature Methods*, vol. 15, no. 12, pp. 1053–1058, Nov. 2018.

[9] Gökcen Eraslan, Lukas M. Simon, Maria Mircea, Nikola S. Mueller, and Fabian J. Theis, "Single-cell RNA-seq denoising using a deep count autoencoder," *Nature Communications*, vol. 10, no. 1, Jan. 2019.

[10] Diederik P. Kingma and Max Welling, "Auto-encoding variational bayes," *2nd International Conference on Learning Representations, Iclr 2014 - Conference Track Proceedings*, 2014.

[11] Gregory P. Way and Casey S. Greene, "Extracting a biologically relevant latent space from cancer transcriptomes with variational autoencoders," in *Biocomputing 2018*. Nov. 2017, WORLD SCIENTIFIC.

[12] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[13] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems 32*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds., pp. 8024–8035. Curran Associates, Inc., 2019.

[14] Feiyang Ma and Matteo Pellegrini, "Automated identification of cell types in single cell RNA sequencing," Jan. 2019.

[15] Nelson Johansen and Gerald Quon, "scAlign: a tool for alignment, integration, and rare cell identification from scRNA-seq data," *Genome Biology*, vol. 20, no. 1, Aug. 2019.