A Comparison of the Effect of Dimensionality Reduction Techniques on the Clustering Tendency of Single-Cell RNA-Seq Data

Forrest Koch

1 Introduction

Dimensionality reduction is an integral part of many single-cell RNA sequencing (scRNA-seq) pipelines.[1–3] It is an effective approach to overcome the so-called “curse of dimensionality”,[4] with many successful applications to real-world data in medical imaging, signal processing, language processing, and gene expression.[1,5,6] Researchers have at their disposal a wide range of both general purpose and domain specific techniques that can be used for this task; however, the lack of systematic comparisons utilizing relevant datasets presents a significant hurdle to those wishing to chose the appropriate technique for the task at hand.[5]

Recent advancements in high-throughput sequencing technology mean that researchers now have the ability to measure the gene expression of thousands to millions of cells in a single experiment.[7–9] This high-resolution data is ideal for detecting previously unknown, rare cell populations that were previously masked by more prevalent cell types in bulk-sequencing technologies.[10] However, little consensus has been reached regarding which methods should be used in the analysis of scRNA-seq or how various methods should be benchmarked against one another.[11]

This study attempts to address these issues by exploring the effect of various dimensionality reduction techniques on the clustering tendency of several scRNA-seq datasets. We will also present a straight-forward, yet novel approach to assess the performance of dimensionality reduction methods on annotated scRNA-seq data.

2 Methods

2.1 Data

The seven datasets used in this comparison were curated by a previous study. They are available through the *R* package *DuoClustering* and are summarized in Table 1.[9] The curators used the *scater* package[12] to perform quality control as well as normalization based on the normalization factors of deconvolution by the *scran* package.[13] Genes with no expression in any samples were removed but were otherwise unfiltered. Refer to the reference papers for further details.

Data set	**Cells**	**Features**	**Groups**	**Description**
Koh[14]	531	48,981	9	FACS purified H7 human embryonic stem cells in different differention stages
Kumar[15]	246	45,159	3	Mouse embryonic stem cells, cultured with different inhibition factors
SimKumar4easy[16]	500	43,606	4	Simulation using different proportions of differentially expressed genes
SimKumar4hard[16]	499	43,638	4	Simulation using different proportions of differentially expressed genes
SimKumar8hard[16]	499	43,601	8	Simulation using different proportions of differentially expressed genes
Zhengmix4eq[7]	3,994	15,568	4	Mixtures of FACS purified peripheral blood mononuclear cells
Zhengmix8eq[7]	3,994	15,716	8	Mixtures of FACS purified peripheral blood mononuclear cells

Table 1: Summary of datasets used in the study. This table has been reproduced from the information in Table 1, Duò et. al. 2018[9]

2.2 Dimensionality Reduction Techniques

The following is a brief overview of the dimensionality reduction techniques employed by this study. For a more in depth discussion, Van Der Maaten (2009)[5] gives an excellent summary of several of these techniques. For the UMAP technique, in addition to the reference paper, the authors have included a more casual explanation at (https://umap-learn.readthedocs.io/en/latest/how_umap_works.html). Each reduction technique was used to produce an embeddings of varying dimensions ranging from 2 up to 10,000, or the maximum allowed by the technique.

2.2.1 Factor Analysis (FA)

Factor Analysis is a linear method that, similar to PCA, is used to find a low dimensional representation of the data to explain the observed variance in the model.[17] FA, however, models the low dimensional factors as Gaussian latent variables which are found through the expectation-maximizing (EM) algorithm.[18]

2.2.2 Fast Independent Component Analysis (FastICA)

This linear method models the underlying dimensions as non-Gaussian latent variables and was originally used in signal processing. These variables are found by minimizing certain contrast functions and are selected such that the components are statistically independent from one another.[19]

2.2.3 Isometric Mapping (Isomap)

Building on multidimensional scaling, Isomap is a non-linear technique that attempts to preserve pairwise geodesic distances between points.[20] To estimate geodesic distance, a manifold is constructed by connecting each point to it's *k* nearest neighbours, and the shortest path between two points is taken as the distance metric. Despite a few weaknesses relating to the constructed graph, Isomap has been shown to perform well on a variety of tasks.[5]

2.2.4 Locally Linear Embedding (LLE)

Similar to Isomap, this non-linear technique also builds a graph of *k* nearest neighbors, however, each point is then expressed as a linear combination of these neighbors. The algorithm attempts to find a low dimensional reconstruction that preserves these linear combinations, thus preserving the local space around each point.[21]

2.2.5 Latent Dirichlet Allocation (LDA)

Latent Dirichlet Allocation is a generative Bayesian model originally developed for document classification by the analysis of word counts and was inspired by Latent Semantic Indexing (LSI).[22] Each sample is modeled as a random mixture of latent variables characterized by a distribution over the input dimensions. This method is implemented using a variational-Bayes algorithm optimized through an online stochastic gradient descent approach.[23]

2.2.6 Multidimensional Scaling (MDS) using SMACOF

Multidimensional Scaling is an approach that, when applied using classical methods, is roughly equivalent to PCA. Here we employ the SMACOF algorithm, a modern approach which attempts to solve the MDS problem by majorization of an objective function.[24]

2.2.7 Principal Component Analysis (PCA)

PCA is a linear dimensionality reduction method that uses single value decomposition to find a low dimensional, orthogonal basis. Each basis vector can be expressed as a linear combination of the high dimensional basis. The low dimensional basis is constructed in a way that maximizes the observed variance within the data.

2.2.8 Laplacian Eigenmaps

Laplacian Eigenmaps is a sparse spectral decomposition technique which aims to minimize the distance between each point and their *k* nearest neighbors in the low dimensional embedding. This minimization is done in a weighted manor such that the distances to the i^{th} nearest neighbor is given a higher weighting than the distance to the $(i+1)^{th}$ neighbor. This is achieved by minimizing the Laplacian of the weight *k* nearest neighbor adjacency matrix.[5,25]

2.2.9 Stacked Denoising Autoencoder (SDAE)

Stacked denoising autoencoders are neural networks trained to reproduce the input after being passed through a bottleneck layer containing fewer nodes than the input layer. Unlike traditional autoencoders, smaller layers are gradually added between the network to gradually reduce the dimensionality of the data. Furthermore, the input is corrupted to promote the de-noising properties of the network.[26] One drawback of this method is that the size of each hidden layer must be defined by the user resulting in an essentially infinite parameter space.

2.2.10 t-Distributed Stochastic Neighbor Embedding

This non-linear technique has gained the favor of many researchers for it's ability to produce well defined clusters in it's low dimensional representation of the data; however, many implementations struggle to handle a large number of input/output dimensions. It works by minimizing the KL-divergence of the distributions of the data with it's low dimensional representation.[28]

2.2.11 Uniform Manifold Projection and Approximation (UMAP)

The UMAP algorithm first constructs a fuzzy topological representation of the high-dimensional data. A low-dimensional embedding is then optimized to have a similar topological representation.[29] This algorithm is a recent development in the field, but is widely considered to rival t-SNE in ability.

2.2.12 Non-negative Matrix Factorization (NMF)

Given a non-negative *n x p* matrix *X*, NMF finds two smaller, non-negative matrices *W* (*n x k*)and *H* (*k x p*) such that their product approximates *X*. *W* can then be used as a *k* dimensional embedding.

Technique	Linear?	Class	Optimization Catagory	References
Factor Analysis	Yes	Convex (full)	Euclidean distance	[17,18]
FastICA	Yes	Convex (full)	Euclidean distance	[19]
Isomap	No	Convex (full)	Geodesic distance	[5,20]
LLE	No	Convex (sparse)	Reconstruction weights	[21]
LDA	No	?	?	[22,23]
MDS	Yes	Convex (full)	Euclidean distance	[24]
PCA	Yes	Convex (full)	Euclidean distance	
Laplacian Eigenmaps	No	Convex (sparse)	Neighborhood graph Laplacian	[5,25]
SDAE	No	Non-Convex	Neural Network	[26]
T-SNE	No	Non-Convex	KL-divergence	[28]
NMF	?	?	?	
UMAP	No	Convex?	Fuzzy set cross entropy	[29]

Table 2: Summary of dimensionality reduction methods

2.3 Validation Metrics

The following validation metrics were calculated for each of the embeddings produced by the techniques outlined in *2.2*.

2.3.1 Variance Ratio Criterion (VRC)

The Varaince Ratio Criterion, also known as the Calinski-Harabaz score, is the ratio of between cluster dispersion to within cluster dispersion, and is commonly used for estimating the ‘correct’ number of clusters.[30] It is defined as:

$$VRC=\frac{(n-k)\,BGSS}{(k-1)\,WGSS}$$

Where *BGSS* is the between group sum of squares, and *WGSS* is the within group sum of squares. Larger values of the *VRC* are said to indicate a better clustering.

2.3.2 Davies-Bouldin Score (DBS)

The Davies-Bouldin score is a measure of how well separated clusters are. It is the ratio of within cluster distances to between cluster distances and is defined as:

$$\frac{1}{c}\sum_{i=1}^{c}\max_{(1<j<c,\,i\neq j)}\left\{\frac{\Delta(X_i)+\Delta(X_j)}{\delta(X_i,X_j)}\right\}$$

Where $\Delta(X_i)$ is the intra-cluster distance and $\delta(X_i,X_j)$ is inter-cluster distance.[31,32]

2.3.3 Dunn Index (DI)

The Dunn index is given by the ratio of the minimum intra-cluster distance to the maximum inter-cluster distance.[33]

$$\frac{\min\limits_{(1<i<c,\,1<j<c,\,j\neq i)}\delta(X_i,X_j)}{\max\limits_{(1<k<c)}\Delta(X_k)}$$

It is helpful for establishing the worst-case performance of a clustering partition, with low values indicating that all clusters are well separated.

2.3.1 Silhouette Score (SS)

The silhouette score for a single sample is calculated as:

$$s(i)=\frac{1}{c}\sum_{i=1}^{c}\frac{b(i)-a(i)}{\max\{a(i),b(i)\}}$$

Where $a(i)$ is the distance of point *i* to it's furthest within-cluster neighbor, and $b(i)$ is the distance of sample *i* to it's nearest neighbor from a separate cluster.[34] It can be used as a measure of confidence of the *i*^*th*^ point's membership to it's cluster with values ranging from -1 (low confidence) to 1 (high confidence).[32] A global score for a particular embedding can be generated by averaging over all points.[35]

2.4 Methods and Software Packages

2.4.1 Dimensionality Reduction

The Python package *sklearn* was used for it's implementations of FA, FastICA, Isomap, LLE, MDS, PCA, Laplacian Eigenmaps, and NMF.[36] The reference version of UMAP was used and is available on github (https://github.com/lmcinnes/umap),[29] the t-SNE implementation used is also available from github (https://github.com/DmitryUlyanov/Multicore-TSNE/),[28,37] and the SDAE implementation is available on github (https://github.com/vlukiyanov/pt-sdae).[38]

Default settings were used for each implementation except for the SDAE. Inputs to the SDAE were log transformed and each dimension was scaled to have a between sample range of 0 to 1. For the sake of consistency between trials, we decreased the dimensionality by 80% at each layer until the target dimension was reached. The network was trained using a similar approach as Xie et al (2015)[27]. Each training step was over 500 epochs with a learning rate decay of 0.985.

2.4.2 Validation Metrics

The Python package *sklearn* provides implementations of VRC, DBS, and SS.[36] An implementation of the Dunn Index was written with the aid of *sklearn*.

2.4.3 Analysis

All code written and used for this study and analysis are available on github (https://github.com/ForrestCKoch/SVR2019-DL-Models/).

3 Results

We compared the performance of 1333 embeddings generated from 12 reduction methods applied to 7 datasets. LDA, SDAE, t-SNE, and UMAP where the best performing methods accounting for all but one of the top measures (summarized in Table 3). These results, however, were highly dataset dependent with one method generally winning out most of the measures for a given dataset. For example, UMAP performed best on both of the Zhengmix datasets in all measures except for DBS in Zhengmix4eq. Similarly, t-SNE obtained the best performance on the Simk4 datasets, again, in all measures except for DBS. It is worth noting that although UMAP did not always perform the best, it was generally close to the top performing method.

With the exception of t-SNE, lower dimensions (< 40) yielded the best results among these top performing methods, and performance in general decreased with increasing number of dimensions. Refer to the supplementary material for plots of performance against dimensions for each method and metric.

dataset	VRC	DBS	DI	SS
Koh	lda-10	lda-10	lda-10	lda-10
Kumar	sdae-10	tsne-2500	sdae-10	sdae-10
Simk4easy	tsne-5000	umap-250	tsne-5000	tsne-5000
Simk4hard	tsne-5000	fa-10	tsne-5000	tsne-5000
Simk8hard	lda-12	lda-18	lda-12	lda-12
Zhengmix8eq	umap-40	umap-10	umap-40	umap-40
Zhengmix4eq	umap-30	tsne-5000	umap-30	umap-30

Table 3: Summary of best performing methods according to dataset and internal-validation metric. The entries indicate the method as well as the number of dimensions.

A few methods (UMAP, PCA, Isomap, Laplacian Eigenmaps, and NMF) possesed an upper limit to the number of dimensions which was imposed by the number of samples. Furthermore, LDA became computationally unfeasible somewhere between 500 and 2500 dimensions for most datasets.

As previously mentioned, most methods performed best at a relatively low number of dimensions, and performance generally degraded with increasing dimensionality. UMAP and, to a lesser extent, LDA were relatively immune to this effect with performance remaining more or less constant at higher dimensions (see Figure 1). T-SNE, however, was fairly unstable having non-monotonic performance with respect to dimensions in many datasets (see Figure 1). Furthermore, both PCA and MDS displayed a tendency to converge on the ground-truth values which is not unexpected given the nature of the techniques. This is nonetheless a noteworthy property as many methods (e.g FA, FastICA, NMF, etc …) exhibit significant under-performance, particularly in higher dimensions.

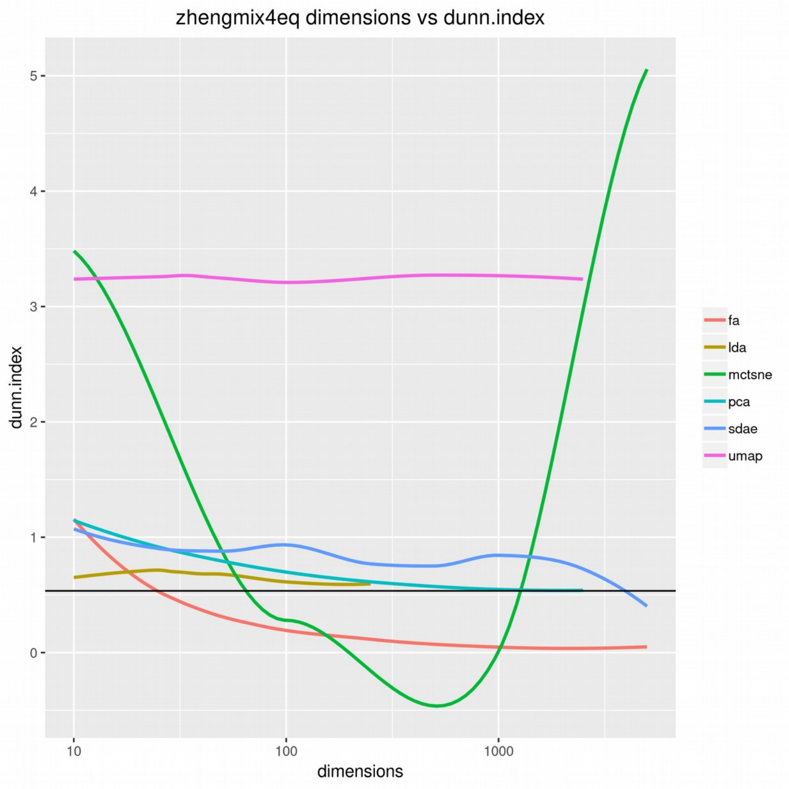

Figure 1: Performance of FA (red), LDA (yellow), tSNE (green), PCA (light blue), SDAE (blue), and UMAP (pink) on Dunn Index for Zhengmix4eq dataset. Higher is better.

4 Discussion

4.1 Discussion of results and some suggestions

The present study investigated the use of 12 unsupervised dimensionality reduction techniques and assessed their ability to produce embeddings with well-formed, biologically relevant clusters. This performance was judged through the use of the internal validation metrics: VRC, DBS, DI, and SS. These metrics have traditionally been used as an unsupervised method for selecting the best clustering of data[39]; however, we have repurposed these metrics to assess the performance of a given embedding of data Because these validation metrics reflect how well defined, separated, and compact a clustering partition is, they can also be utilized to determine the clustering tendency of an embedding given the ground truth labels. To the best of the author’s knowledge, this is the first application of these measures in a supervised approach to compare dimensionality reduction methods.

UMAP, t-SNE, LDA, and SDAE were the top performing methods; however, results were highly dataset dependent. As no single method consistently out performed the rest, we warn that being too reliant on any single reduction technique may result in sub-optimal embeddings thus having implications for downstream clustering and analysis. Furthermore, the ideal number of dimensions also varies between datasets and may also take some experimentation to work out. Unfortunately, many existing scRNA pipelines fail to provide flexibility in these regards. Although this allows the tool to be more user friendly, it may ultimately be a hindrance in the pursuit of achieving an optimal clustering.[1,2,40]

The stability of the method to be used should also be considered during application. A few methods (e.g t-SNE), were non-monotonic with respect to dimensionality with drastic changes in performance on either side of their optima. This is in contrast to other methods (e.g UMAP, LDA), which demonstrated relatively level performance regardless of dimensionality.

4.2 Some notes on the top performing methods.

4.2.1 t-SNE

This method is perhaps one of the most extensively used approaches in scRNA analysis. It is most frequently used to collapse data down to 2 dimensions for visualization purposes, and although we only explored down to 10 dimensions, our results suggest that there may be more optimal reductions at higher dimensions.

4.2.2 UMAP

Recently developed, UMAP has attracted plenty of interest in recent scRNA studies for it’s ability to rival t-SNE in producing well-formed clusters for 2D visualization.[41–43] We have demonstrated the stability of this algorithm across a range dimensions and datasets. If the researcher does not wish to explore multiple methods and parameter choices, UMAP may be one of the safer options.

4.2.3 LDA

Despite being a long-standing, popular technique in natural language processing, relatively few studies have explored LDA in scRNA. A few scRNA pipelines, namely *CellTree* and *Cis-Topic*, have found success using this approach.[44,45] There is a large body of research surrounding LDA and many extensions have been developed to achieve better results.[46,47] Considering the maturity of LDA research, and the success we found using the "vanilla" approach, this could be a fruitful avenue of exploration.

4.2.4 SDAE

At the time of writing, this is the only study to use stacked denoising autoencoders for dimensionality reduction in scRNA; however, other studies have been successful in employing other variants of the autoencoder method.[35,48,49] When successfully trained, this approach can yield very good results; however, architecture definition and parameter selection pose significant hurdles to those wishing to apply this technique.

4.3 Limitations of the study

This study possesses several limitations that should be taken into consideration. Most of the methods studied here have additional parameters beyond dimensionality that need to be set. Where possible, we used the implementation's default parameters; however, the SDAE required some additional choices that are noted in the methods. We chose not to optimize over the parameter space to reduce the complexity of this study as well as to avoid implicit bias in model selection. In practice, multiple parameter choices should be explored to obtain an optimal solution.

Furthermore, high-dimensional space has some counter-intuitive properties the result in the breakdown of traditional distance metrics.[6,50] This study employed the Euclidean distance metric; however, other metrics such as cosine, or Pearson correlation have been shown to be more robust at higher dimensions and may give better results in some circumstances.[6] Additionally, the maximum number of dimensions able to be achieved by a few techniques is dependent on the sample size and was a limitation in several of the datasets used.

We did not reduce beyond 10 dimensions. It is not uncommon for studies to perform clustering in as few as 2 dimensions, and so this study cannot comment on how these approaches perform in this range.

It is common practice to use multiple reduction techniques in succession. For example Bhaduri et al (2018) first used PCA to reduce down to 50 dimensions, and then employed t-SNE to bring this down to 2.[51] We did not explore this combination approach which may complicate comparisons to other studies.

4.4 Areas for further study

Although our validation approach attempted to identify embeddings which resulted in well defined clusters, we have not verified that unsupervised clustering algorithms actually perform better on these embeddings. Further work might compare unsupervised clustering approaches on these “optimal” embeddings using the internal validation metrics of VRC, DBS, DI, and SS as indicators of fitness. Accuracy of the top performing methods can then be assessed according to ground truth labels.

Successful training of an SDAE is an important preliminary step in implementing Deep Embedded Clustering (DEC), an unsupervised deep learning clustering technique that aims to minimize the KL-divergence between the distribution of the centroids and a target auxiliary function.[27] This approach has been shown to perform well on the MNIST, STL, and REUTERS datasets; however, it’s performance has yet to be demonstrated on scRNA datasets. This study lays the groundwork necessary to successfully perform DEC on scRNA data.

4.5 Conclusions

UMAP, LDA, t-SNE, and SDAE were found to be the top performing dimensionality reduction methods on a benchmark of 7 scRNA datasets. A relatively low number of dimensions (<40) seems to produce the best results. Researchers should take caution when relying on a single reduction method, as results tended to be highly dataset dependent. That said, certain algorithms (e.g UMAP and LDA) showed more stable performance than others (e.g t-SNE) indicating some methods warrant more caution than others. Furthermore, this study provides some valuable insights regarding training SDAE which is a necessary step for the implementation of DEC.

1. Ding, J., Condon, A. & Shah, S. P. Interpretable dimensionality reduction of single cell transcriptome data with deep generative models. *Nat. Commun.* **9**, 2002 (2018).
2. Pierson, E. & Yau, C. ZIFA: Dimensionality reduction for zero-inflated single-cell gene expression analysis. *Genome Biol.* **16**, 241 (2015).
3. Risso, D., Perraudeau, F., Gribkova, S., Dudoit, S. & Vert, J.-P. A general and flexible method for signal extraction from single-cell RNA-seq data. *Nat. Commun.* **9**, 284 (2018).
4. Chen, L. Curse of Dimensionality. in *Encyclopedia of Database Systems* (eds. LIU, L. & ÖZSU, M. T.) 545–546 (Springer US, 2009).
5. Van Der Maaten, L., Postma, E. & Van den Herik, J. Dimensionality reduction: a comparative review. *J Mach Learn Res* **10**, 66–71 (2009).
6. Paukkeri, M.-S., Kivimäki, I., Tirunagari, S., Oja, E. & Honkela, T. Effect of Dimensionality Reduction on Different Distance Measures in Document Clustering. in *Neural Information Processing* (eds. Lu, B.-L., Zhang, L. & Kwok, J.) 167–176 (Springer Berlin Heidelberg, 2011).
7. Zheng, G. X. Y. *et al.* Massively parallel digital transcriptional profiling of single cells. *Nat. Commun.* **8**, 14049 (2017).
8. Han, X. *et al.* Mapping the Mouse Cell Atlas by Microwell-Seq. *Cell* **172**, 1091–1107.e17 (2018).
9. Duò, A., Robinson, M. D. & Soneson, C. A systematic performance evaluation of clustering methods for single-cell RNA-seq data. *F1000Research* **7**, (2018).
10. Rostom, R., Svensson, V., Teichmann, S. A. & Kar, G. Computational approaches for interpreting scRNA-seq data. *FEBS Lett.* 2213–2225 (2017). doi:10.1002/1873-3468.12684@10.1002/(ISSN)1873-3468.SINGLECEL
11. Tian, L. *et al.* scRNA-seq mixology: towards better benchmarking of single cell RNA-seq protocols and analysis methods. *bioRxiv* 433102 (2018). doi:10.1101/433102
12. McCarthy, D. J., Campbell, K. R., Lun, A. T. L. & Wills, Q. F. Scater: pre-processing, quality control, normalization and visualization of single-cell RNA-seq data in R. *Bioinformatics* **33**, 1179–1186 (2017).
13. L. Lun, A. T., Bach, K. & Marioni, J. C. Pooling across cells to normalize single-cell RNA sequencing data with many zero counts. *Genome Biol.* **17**, 75 (2016).

14. Koh, P. W. *et al.* An atlas of transcriptional, chromatin accessibility, and surface marker changes in human mesoderm development. *Sci. Data* **3**, 160109 (2016).

15. Kumar, R. M. *et al.* Deconstructing transcriptional heterogeneity in pluripotent stem cells. *Nature* **516**, 56–61 (2014).

16. Zappia, L., Phipson, B. & Oshlack, A. Splatter: simulation of single-cell RNA sequencing data. *Genome Biol.* **18**, 174 (2017).

17. Bartholomew, D. J., Steele, F., Galbraith, J. & Moustaki, I. *Analysis of multivariate social science data*. (Chapman and Hall/CRC, 2008).

18. Dempster, A. P., Laird, N. M. & Rubin, D. B. Maximum Likelihood from Incomplete Data via the EM Algorithm. *J. R. Stat. Soc. Ser. B Methodol.* **39**, 1–38 (1977).

19. Hyvärinen, A. & Oja, E. Independent component analysis: algorithms and applications. *Neural Netw.* **13**, 411–430 (2000).

20. A Global Geometric Framework for Nonlinear Dimensionality Reduction | Science. Available at: http://science.sciencemag.org/content/290/5500/2319. (Accessed: 7th February 2019)

21. Nonlinear Dimensionality Reduction by Locally Linear Embedding | Science. Available at: http://science.sciencemag.org/content/290/5500/2323. (Accessed: 7th February 2019)

22. Blei, D. M., Ng, A. Y. & Jordan, M. I. Latent Dirichlet Allocation. *J. Mach. Learn. Res.* **3**, 993–1022 (2003).

23. Hoffman, M., Bach, F. R. & Blei, D. M. Online Learning for Latent Dirichlet Allocation. in *Advances in Neural Information Processing Systems 23* (eds. Lafferty, J. D., Williams, C. K. I., Shawe-Taylor, J., Zemel, R. S. & Culotta, A.) 856–864 (Curran Associates, Inc., 2010).

24. Leeuw, J. de & Mair, P. Multidimensional Scaling Using Majorization: SMACOF in R. (2011).

25. Belkin, M. & Niyogi, P. Laplacian Eigenmaps for Dimensionality Reduction and Data Representation. *Neural Comput.* **15**, 1373–1396 (2003).

26. Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y. & Manzagol, P.-A. Stacked Denoising Autoencoders: Learning Useful Representations in a Deep Network with a Local Denoising Criterion. *J. Mach. Learn. Res.* **11**, 3371–3408 (2010).

27. Xie, J., Girshick, R. & Farhadi, A. Unsupervised Deep Embedding for Clustering Analysis. *ArXiv151106335 Cs* (2015).

28. Maaten, L. van der & Hinton, G. Visualizing Data using t-SNE. *J. Mach. Learn. Res.* **9**, 2579–2605 (2008).

29. McInnes, L., Healy, J. & Melville, J. UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction. *ArXiv180203426 Cs Stat* (2018).

30. Caliński, T. & Harabasz, J. A dendrite method for cluster analysis. *Commun. Stat.* **3**, 1–27 (1974).

31. Davies, D. L. & Bouldin, D. W. A Cluster Separation Measure. *IEEE Trans. Pattern Anal. Mach. Intell.* **PAMI-1**, 224–227 (1979).

32. Bolshakova, N. & Azuaje, F. Cluster validation techniques for genome expression data. *Signal Process.* **83**, 825–833 (2003).

33. Dunn, J. C. A Fuzzy Relative of the ISODATA Process and Its Use in Detecting Compact Well-Separated Clusters. *J. Cybern.* **3**, 32–57 (1973).

34. Rousseeuw, P. J. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *J. Comput. Appl. Math.* **20**, 53–65 (1987).

35. Hu, Q. & Greene, C. S. Parameter tuning is a key part of dimensionality reduction via deep variational autoencoders for single cell RNA transcriptomics. in *Biocomputing 2019* 362–373 (WORLD SCIENTIFIC, 2018).

36. Pedregosa, F. *et al.* Scikit-learn: Machine Learning in Python. *J. Mach. Learn. Res.* **12**, 2825–2830 (2011).

37. Ulyanov, D. *Parallel t-SNE implementation with Python and Torch wrappers.: DmitryUlyanov/Multicore-TSNE*. (2019).

38. Lukiyanov, V. *PyTorch implementation of SDAE (Stacked Denoising AutoEncoder): vlukiyanov/pt-sdae*. (2019).

39. Yeung, K. Y., Haynor, D. R. & Ruzzo, W. L. Validating clustering for gene expression data. *Bioinformatics* **17**, 309–318 (2001).

40. Tian, L. *et al.* scPipe: A flexible R/Bioconductor preprocessing pipeline for single-cell RNA-sequencing data. *PLOS Comput. Biol.* **14**, e1006361 (2018).

41. Wu, D. *et al.* Comparison Between UMAP and t-SNE for Multiplex-Immunofluorescence Derived Single-Cell Data from Tissue Sections. *bioRxiv* 549659 (2019). doi:10.1101/549659
42. Becht, E. *et al.* Evaluation of UMAP as an alternative to t-SNE for single-cell data. *bioRxiv* 298430 (2018). doi:10.1101/298430
43. Oetjen, K. A. *et al.* Human bone marrow assessment by single-cell RNA sequencing, mass cytometry, and flow cytometry. *JCI Insight* **3**,
44. duVerle, D. A., Yotsukura, S., Nomura, S., Aburatani, H. & Tsuda, K. CellTree: an R/bioconductor package to infer the hierarchical structure of cell populations from single-cell RNA-seq data. *BMC Bioinformatics* **17**, 363 (2016).
45. González-Blas, C. B. *et al.* Cis-topic modelling of single cell epigenomes. *bioRxiv* 370346 (2018). doi:10.1101/370346
46. Ramage, D., Hall, D., Nallapati, R. & Manning, C. D. Labeled LDA: A Supervised Topic Model for Credit Attribution in Multi-labeled Corpora. in *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1 - Volume 1* 248–256 (Association for Computational Linguistics, 2009).
47. Teh, Y. W., Jordan, M. I., Beal, M. J. & Blei, D. M. Sharing clusters among related groups: Hierarchical Dirichlet processes. in *Advances in neural information processing systems* 1385–1392 (2005).
48. Eraslan, G., Simon, L. M., Mircea, M., Mueller, N. S. & Theis, F. J. Single-cell RNA-seq denoising using a deep count autoencoder. *Nat. Commun.* **10**, 390 (2019).
49. Wall, M., Dreisbach, C., Zaidi, A., Flower, A. & Overall, C. Using autoencoders and text mining to characterize single cell populations in the hippocampus and cortex. in *2018 Systems and Information Engineering Design Symposium (SIEDS)* 106–111 (2018). doi:10.1109/SIEDS.2018.8374718
50. Zaki, M. J., Meira Jr, W. & Meira, W. *Data mining and analysis: fundamental concepts and algorithms*. (Cambridge University Press, 2014).
51. Bhaduri, A., Nowakowski, T. J., Pollen, A. A. & Kriegstein, A. R. Identification of cell types in a mouse brain single-cell atlas using low sampling coverage. *BMC Biol.* **16**, 113 (2018).