**Nonlinear Dimensionality Reduction of Single Cell Data using Autoencoders**

Parth Vora, Robert Waxman

Introduction

Single cell technologies, such as single cell RNA sequencing (scRNA-seq) and mass cytometry, have allowed for extremely high resolution data to be collected for tissue samples. These technologies can help elucidate genetic and metabolic pathways at the cellular level, therefore having the potential to vastly increase our knowledge of cellular differentiation, disease progression, and other topics. However, due to its high resolution, single cell datasets often have a large number of dimensions and tens of thousands of samples, making the data difficult to work with. Simply applying PCA is not an excellent solution as PCA struggles to account for the underlying structure of the data. Therefore, many have turned to nonlinear dimensionality reduction techniques to interpret and work with single cell datasets.

Two commonly used nonlinear dimensionality reduction techniques are t-distributed stochastic neighbour embeddings (tSNE) and uniform manifold approximation and projection (UMAP). tSNE is essentially an unsupervised, nonlinear dimensionality reduction method that works by calculating similarity measures between points in a high dimensional space and low dimensionality space and then trying to optimize the two similarity measures with a KL-divergence cost function. The benefits of tSNE over PCA is that tSNE is capable of learning nonlinearly structured data such as the famous Swiss Roll dataset because of tSNE’s emphasis on preserving pairwise distances and local similarities. The main drawbacks of tSNE are its inability to preserve global structure and its long runtime. UMAP is a more recent unsupervised nonlinear dimensionality reduction method similar to tSNE that is argued by its author Leland McInnes to fix many of the pitfalls of tSNE. UMAP has shown to be far faster than tSNE for high dimensional data. The reduced runtime is due mainly to UMAP not applying normalization to either high or low dimensional probabilities ­­[CITE towards data science], a key component of the tSNE algorithm. Also, UMAP utilizes Stochatic Gradient Descent (SGD) as opposed to the regular Gradient Descent (GD) used in tSNE. This change not only reduces runtime, but also reduces the memory usage of UMAP. Furthermore, UMAP can preserve global structure by using cross entropy as the cost function instead of KL-divergence like tSNE does.

Another approach to nonlinear dimensionality reduction is the use of autoencoders (AEs), a type of neural network that learns to reconstruct data through a low dimensional bottleneck. An AE consists of an encoder network that learns a low dimensional representation of the original data, and a decoder network that learns to recreate the original data from the latent space. AEs have been wildly successful in computer vision, and were able to learn low dimensional representations of high dimensional image data and accurately reconstruct new images from latent spaces. Because of their neural network architecture, AEs come with several advantages over other dimensionality reduction techniques. Firstly, AEs are highly parallelizable, dramatically speeding up the training process to learn latent spaces. Furthermore, the encoder network can be extracted from the AE after training and used to rapidly generate low dimensional representations of new samples outside the training dataset. Online training, continual improvement, and transfer learning can all be performed with AEs as well, which make them well suited to a variety of applications. AEs are also highly customizable as the designer is able to modify the network to account for any underlying assumptions or structure of the data in order to generate more accurate reconstructions from the latent representations.

In the past few years, work has been published to investigate how AEs can be used to process and analyze single cell data. Deep Count Autoencoder (DCA) is an AE published by Eraslan et al. in 2019 to denoise scRNA-seq data[cite]. DCA accounts for the distribution of raw counts, sparsity, and other aspects of scRNA-seq data using a negative binomial noise model, and can be used for denoising and data imputation on millions of samples[cite]. In another preprint published by Zhang in 2019, a variational autoencoder (VAE) that used maximum mean ­discrepancy instead of KL divergence as an objective function was able to generate superior embeddings in terms of information retained in latent space and reconstruction error when compared to a vanilla VAE[cite]. Zhang argues that MMD-VAE would thus be a better option for single-cell data analysis. Considering the recent interest in the use of AEs on single cell data, we sought to investigate how various AE models compare to unsupervised dimensionality reduction algorithms like PCA, tSNE, and UMAP. We focused our attention on simple and general models (feed-forward and vanilla VAE) to get an understanding of how changing AE architecture affects the overall embeddings.

Data

We analyzed the performance of different dimensionality reduction algorithms on two mass cytometry (CyTOF) datasets. The first dataset, samusik01, was a dataset also analyzed by Becht et al. This dataset is the first bone marrow sample analyzed by Samusik et al. in “Automated mapping of phenotype space with single-cell data” [cite]; the sample was taken from C57BL/6 mice and consists of over 86,000 events, 38 parameters, and 24 different cell populations. Samusik01 data was obtained in a zipped format from the following link: <https://web.stanford.edu/~samusik/Panorama%20BM%201-10.zip>.

The second dataset, levine, is from the paper “Data-Driven Phenotypic Dissection of AML Reveals Progenitor-like Cells that Correlate with Prognosis” by Levine et al. The levine dataset consists of protein expression levels from healthy human bone marrow mononuclear cells (BMCs) from two healthy individuals. The whole dataset contains over 250,000 events, 32 parameters, and 14 different cell populations along with cell annotations from the authors, but for our analysis we filtered out the unassigned cells leaving just over 100,000 samples. Levine data was obtained in a zipped format from the following link: <https://github.com/lmweber/benchmark-data-Levine-32-dim/tree/master/data>.

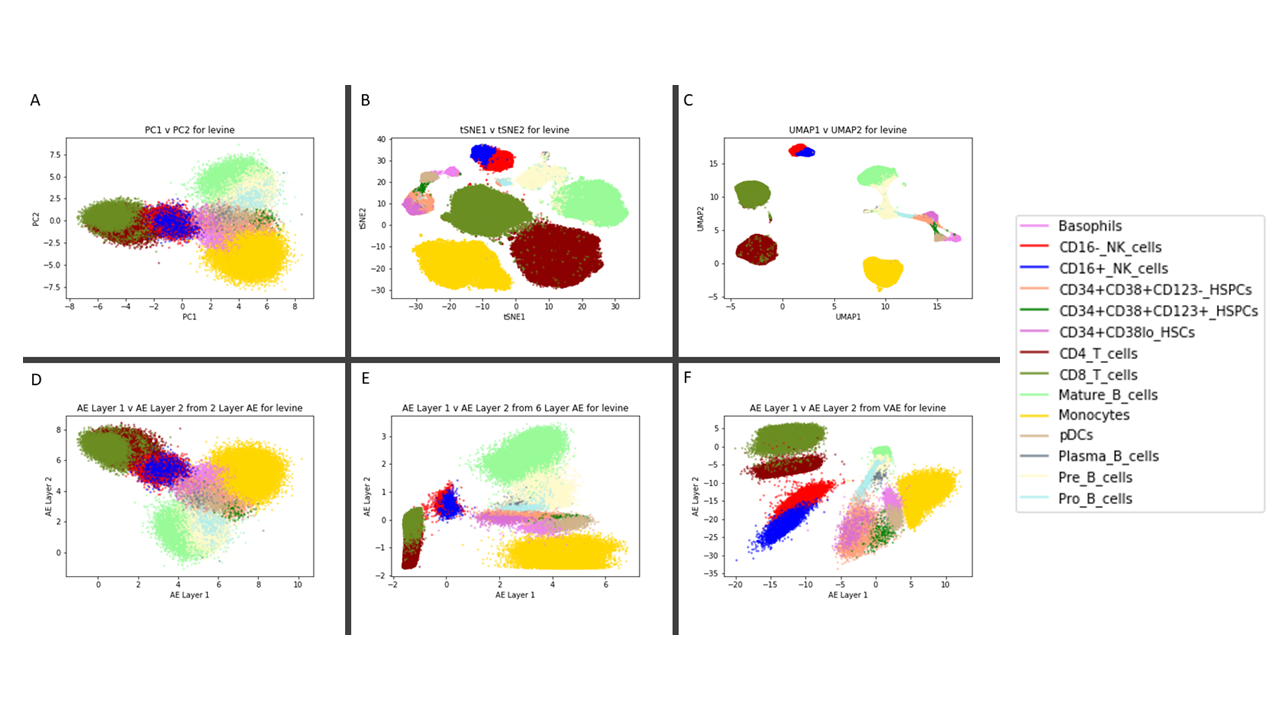
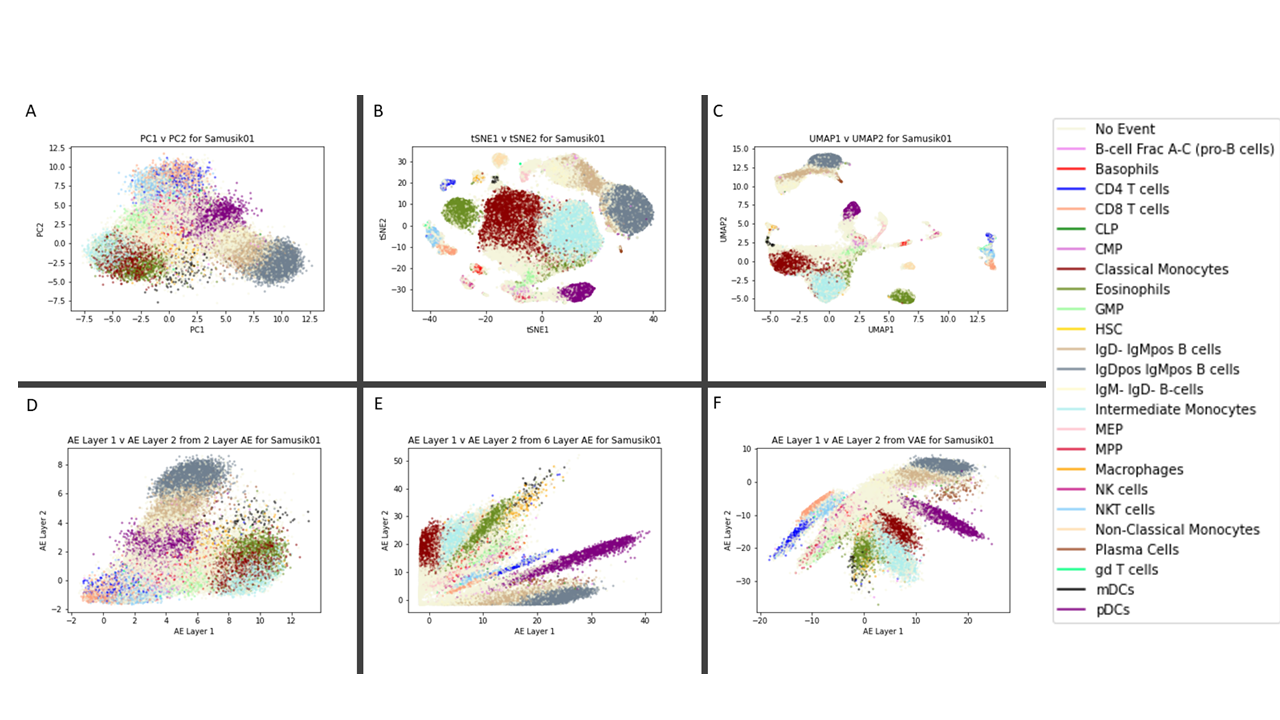
Methods

Before embeddings were generated, data was first pre-processed by selecting only the dimensions relevant for study and applying a hyperbolic arcsine transformation as is standard for mass cytometry data [cite?]. After preprocessing, dimensionality reduction was performed using PCA, tSNE, UMAP, and three different AE models. PCA and tSNE were performed using the implementations available in scikit-learn [cite], and UMAP was performed using the umap-learn implementation [cite]. The three AE models included a two-layer feedforward model (FF2), a six-layer feedforward model (FF6), and a variational autoencoder (VAE). AE models were built with Keras and were trained on either samusik01 or levine data until convergence. FF2 and FF6 were trained to minimize the reconstruction error (mean squared error) of the data, whereas VAE minimized both reconstruction error and the KL divergence of the encoder and decoder probability distributions [citation?].

The embeddings generated by each model were evaluated on the separation of data classes, the average normalized mutual information of k-means clustering performed on the embeddings of data subsamples and k-means clustering performed on total datasets, the preservation of correlation of coordinates in subsamples versus in the embedding of the full dataset, the preservation of pairwise distances, and the runtime to generate the embeddings. Separation of data classes was measured using the same method as [OG paper]. A random forest classifier was trained to predict cell type from each type of embedding, and accuracy on held out data in a 5-fold cross validation setting was recorded. To validate the robustness of the separation of data classes, we also trained and tested a support vector machine (SVM) classifier and a K-nearest neighbors (KNN) classifier in the same way. To further measure the preservation of local structure, k-means clustering (with k = 10) was performed on the embeddings of five data subsamples of size 20,000 as well as on the embedding of the full dataset and the average normalized mutual information was calculated. This is a slight deviation from the method used in Becht et al as the parameters had to be adjusted to match the smaller dataset sizes in this analysis. The reproducibility of embeddings on a large scale was measured using the same method as [OG paper]. The correlation of coordinates on random subsamples of varying sizes versus embeddings of the full datasets were measured while accounting for symmetries across the axes. Preservation of pairwise distances was measured by correlating pairwise distances of random samples taken from original data and pairwise distances of those same samples in the embeddings. Finally, time to generate embeddings for various sample sizes, ranging from 100 samples to 50,000 samples, was measured for each algorithm.

Results

Figure 1. Latent space visualizations of Samusik01 and Levine datasets. Top six images are embeddings of the samusik01 dataset, and bottom six images are embeddings of the levine dataset. Embeddings are organized as such: A - PCA, B - tSNE, C - UMAP, D - FF2, E - FF6, F - VAE.



On the samusik01 dataset, we were able to replicate the UMAP embeddings as visualized in the Becht et al. paper. However, tSNE embeddings could not be replicated as well, which is likely a result of tSNE’s sensitivity to initial conditions. Looking qualitatively at both the samusik01 and levine datasets, UMAP appears to create embeddings where cells of a particular type are tightly clustered together, whereas tSNE generates embeddings with large blob-like clusters. UMAP and tSNE both cluster similar cell types together (i.e., classical & intermediate monocytes in samusik01, CD16+ & CD16- NK cells in levine), but they differ in their inter-cluster distances. UMAP appears to have larger inter-cluster distances for different cell types whereas tSNE inter-cluster distances do not appear to be meaningful; this result is most apparent in the levine embeddings. Both algorithms markedly outperform PCA in their ability to separate out different cell types, which highlights the nonlinear structure of mass cytometry single cell data.

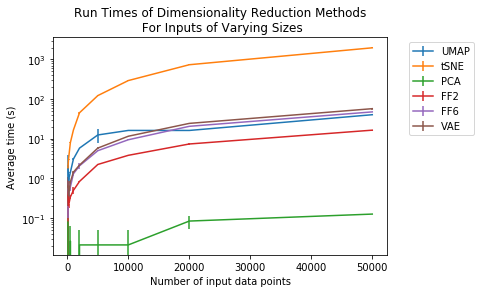


Figure 2. Runtimes of Dimensionality Reduction Algorithms. Runtimes of each algorithm is plotted for various input sizes on a logarithmic scale.

For the three AEs, a general trend appears to be that the more complex the model, the better it is able to separate the data. FF2 embeddings look as if the PCA embeddings were rotated and/or flipped. This is somewhat expected as a simple two-layer model would not have enough complexity to capture any underlying nonlinear structure of the data, and so it learns a linear approximation of the data. However, FF6 and VAE are able to separate the data classes much better than FF2 and PCA, indicating that they are learning some nonlinear structure. VAE appears to perform the best out of all the AEs, which is logical considering it not only minimizes reconstruction error but also the KL divergence of the encoder and decoder distributions. This allows VAE to actually learn a probability distribution underlying the data instead of an arbitrary mapping between high dimensional space and a 2D coordinate. VAE’s power is best illustrated by the embeddings for CD16+ and CD16- NK cells in levine – FF2 is unable to separate them, FF6 struggles as well, but VAE does a much better job. It is important to note that none of the AEs generated embeddings as tight or clean as tSNE and UMAP. This is most likely because the models we used were still too simplistic – it is possible that more complex AE models that account for underlying assumptions in single cell data would perform much better.

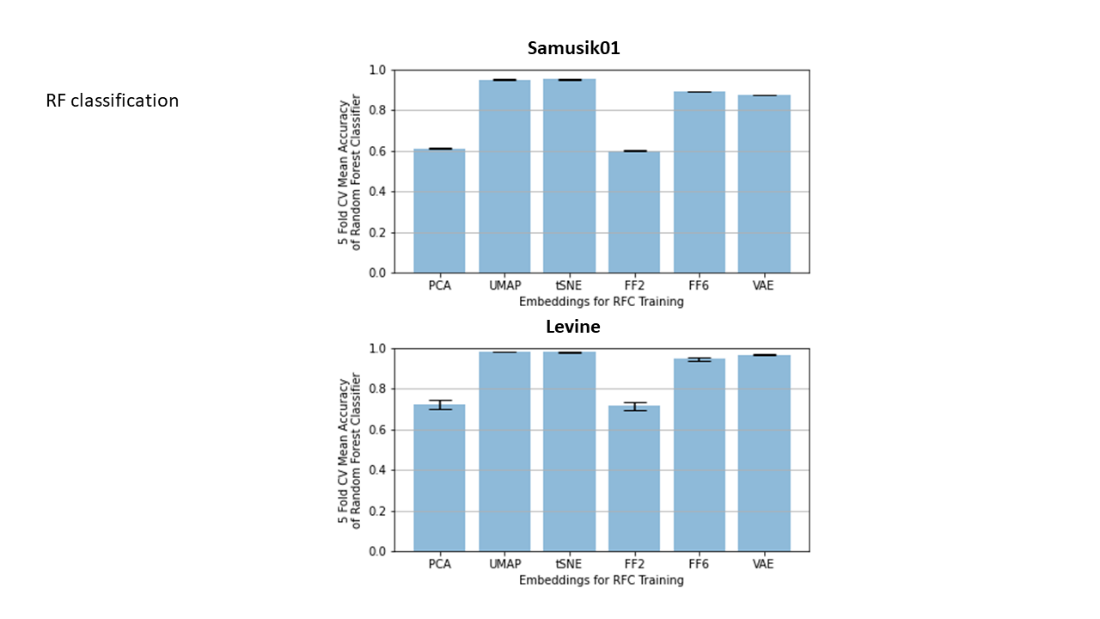


Figure 3. Random Forest Classifier Accuracy on Embeddings. RF accuracy of label prediction is plotted for each embedding.

One of the main advantages of AEs over other dimensionality reduction algorithms is their high degree of parallelism. In the Becht et al. paper, the authors make a point of showing that a major benefit of UMAP is that it scales well with a large number of input data points. We were able to recreate this result in Figure 2, where UMAP is able to scale much better than tSNE with increasing input size. It is also important to note that the AEs scale as well as UMAP with a large number of samples, and increasing the complexity of the model from 2 to 6 layers saw only a slight increase in training time. Furthermore, this speed was achieved with only multiprocessing on a CPU – with GPU acceleration, training times for deep learning architectures may be much shorter than UMAP runtime.

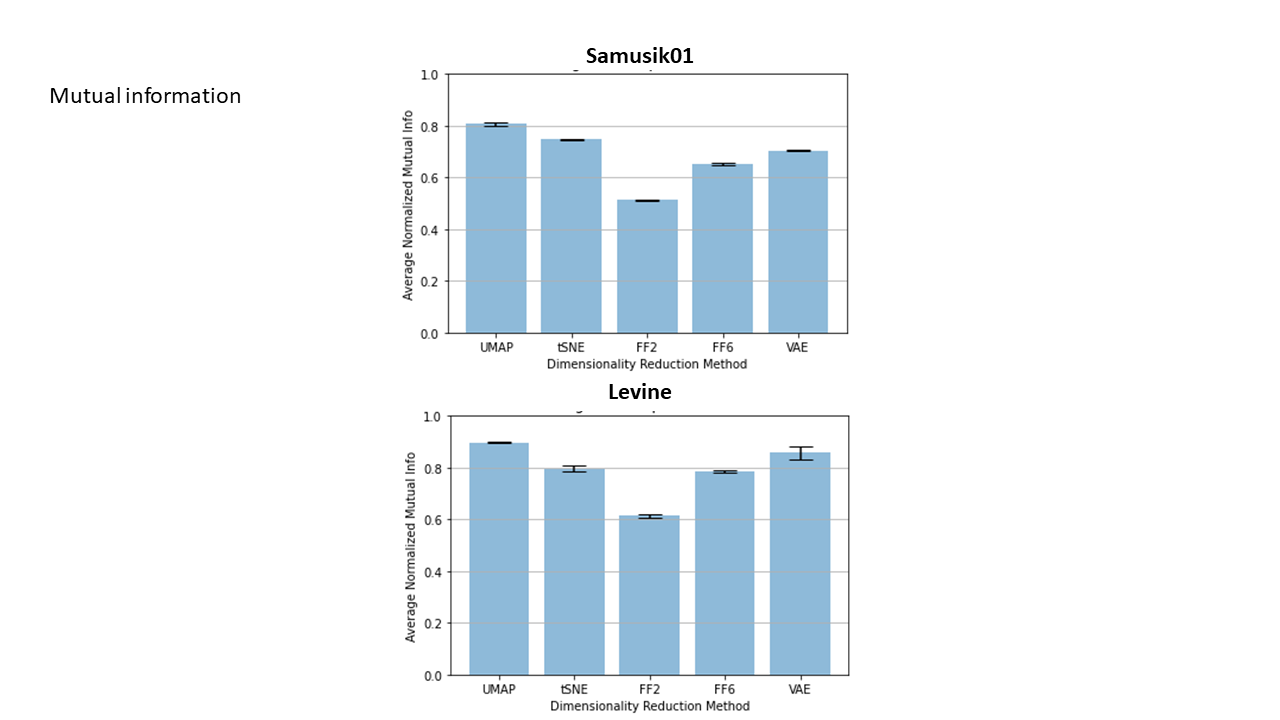


Figure 4. Normalized Mutual Information for each embedding. Average normalized mutual information of k-means clustering (k = 10) performed on the embeddings of data subsamples and k-means clustering (k = 10) performed on total datasets. The average across five random subsamples of size 20,000 is shown, with vertical bars representing s.d.

The authors in the Becht et al. paper argue that one of the main benefits of UMAP over tSNE is the reproducibility of its embeddings on a local scale. We were able to recreate this result in Figure 4, where each algorithm appears to produce embeddings of subsamples consistent with embeddings of the full datasets with normalized mutual information coefficients between 0.6 and 0.9 (excluding FF2). UMAP still performs the best on both datasets, with tSNE ranked second on the first dataset and ranked third on the second dataset. The AEs performed surprisingly well with this metric, with the VAE ranked third on the first dataset and ranked second on the second dataset, passing even tSNE. These results are very promising and suggest that with more sophisticated autoencoders, it is possible that we see performance comparable to that of UMAP.

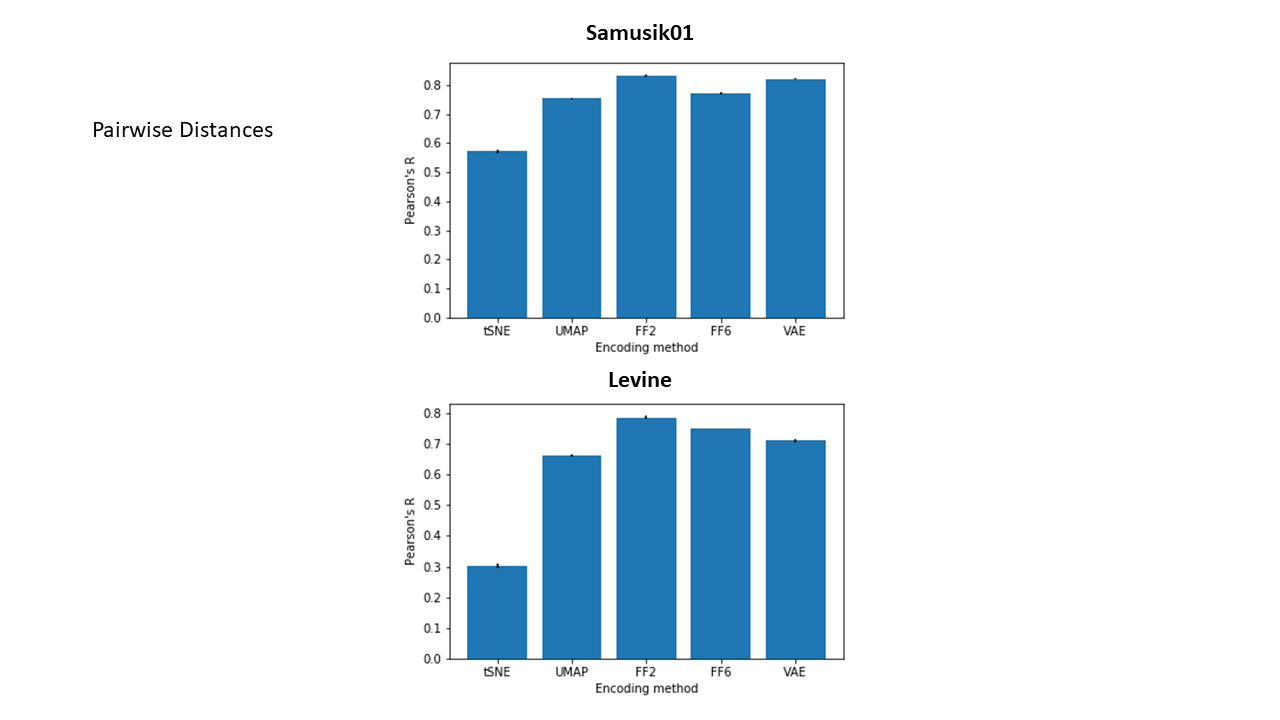


Figure 5. Correlation of pair-wise distances in original dataset and embeddings. The Pearson correlation coefficient computed over the pairs of pairwise distances (1000 pts) is shown for each embedding, with vertical bars representing s.d.

Another one of the main advantages of AEs over other dimensionality reduction algorithms is their ability to preserve global data structure. By comparing the distances between random pairs of points in the original datasets and in the embeddings in Figure 5, we see that all three AEs perform better than both tSNE and UMAP on both datasets. The performances of tSNE and UMAP match those shown in the paper by Becht et al., with slight deviations most likely attributed to the use of Samusik01 instead of the full Samusik dataset. For the three AEs, a general trend appears to be the more complex the model, the less the global data structure is preserved. This makes sense as we saw earlier that the embeddings generated by FF2 resemble the embeddings generated by PCA, and we know that PCA emphasizes the preservation of global structure far more than either tSNE or UMAP.

Lastly, one of the main advantages of the AEs over other dimensionality reduction methods is the high level of reproducibility of large-scale structure in the embeddings. In the Becht et al. paper, the authors argue that a major benefit of UMAP over tSNE is the high level of correlation of coordinates in subsamples versus in the embedding of the full dataset. We were able to replicate this result in Figure 6, where UMAP does significantly better than tSNE in the second dataset, and slightly better in the first dataset overall. The performance of the two methods is slightly closer in Samusik01 than Becht et al. shows in their paper for the full Samusik dataset, but this may be attributed to using just the first sample instead of the full dataset, as well as the effect that initial state has on the embeddings. It is also important to note that the AEs perform better overall on the first dataset than both tSNE and UMAP, and slightly worse than UMAP and better than tSNE on the second dataset. However, we also see much more consistent correlations over different sample sizes and far smaller standard deviations for the AEs as compared to UMAP and tSNE.

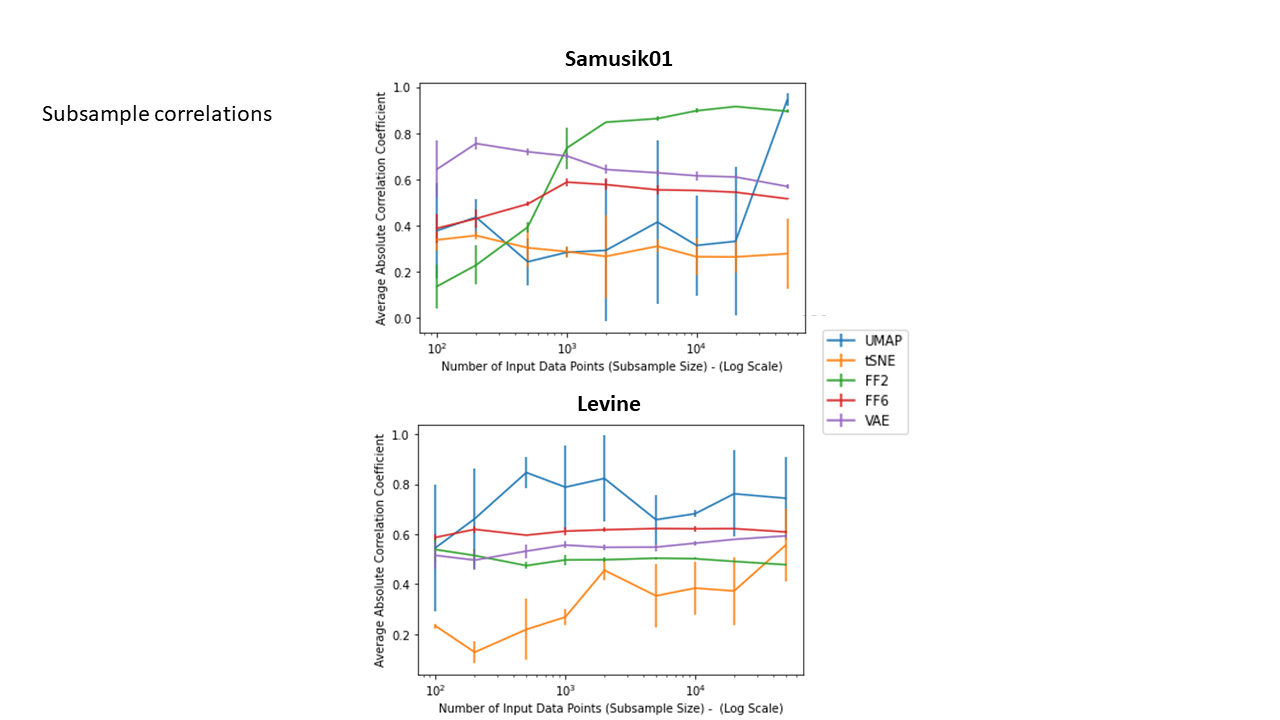


Figure 6. Correlation of subsample embeddings to whole dataset embeddings. Average unsigned Pearson correlation coefficients of the points' coordinates in the embedding of subsamples versus in the embedding of the full dataset for varying input sizes on a logarithmic scale

Conclusions

References

<https://towardsdatascience.com/how-exactly-umap-works-13e3040e1668>

Course Project Statements and Contributions