# Evaluating Dimensionality Reduction Methods
# Final Report

*Abstract*- **Working with a dataset containing n number of features becomes really hard when it comes to visualizing it and trying to establish the relationships between the features. Dimensionality reduction techniques aim at the simplification of complexity, removal of redundant data, ease to comprehend the relevant data, reduce computation time, data visualization and facilitation of clustering for algorithms that struggle with too many dimensions [1]. With linear dimensionality reduction methods like Principal Component Analysis(PCA) which use Matrix Factorization and the evolving non-linear methods like Uniform Manifold Approximation and Projection(UMAP) and t-distributed Stochastic Neighbor Embedding(tSNE) which uses graph-based mechanisms for reduction, it becomes crucial task to analyze these techniques based on various measures [1]. This project aims at reviewing the metrics for two popularly used DR models: UMAP and t-SNE and contrast them using a new metric.**

## I. Introduction

*Why DR can be advantageous?* While trying to convert datasets of high-dimensionality to low-dimensionality, we need to prioritize the rubric of finding the structures in dataset- local or global. The reduced embedding needs to be a compressed version while preserving the data structure with minimum loss. This technique is called Dimensionality Reduction and this not only makes it easier to visualize and understand the manifold, it also provides a method to reduce computational resources.

*Challenges in Dimensionality Reduction:* When we encounter a supervised learning problem, we have a baseline or reference to check the correctness of the results produced by our algorithm. However, this becomes complicated in the case of Dimensionality Reduction. Here we do not have any natural way to directly measure the quality of the output produced or to compare two methods by an objective measure like for instance modeling efficiency or classification error. This can be accounted to the fact that every method optimizes a different error function. Comparing, for instance, t-SNE and PCA by means of recovered variance or divergence in data will be a bit inconsistent, because both have different methods of reduction [2][21]. Therefore, we have themed our analysis of DR algorithms on the basis of their performance in case of specific metrics.

*When is a DR method inadequate?:* The difficulty in applying DR is that each DR method is designed to maintain certain aspects of the original data and therefore may be appropriate for one task and inappropriate for another. Thus, choosing an inadequate method may imply that much of the underlying structure remains undiscovered. Methods like Umap and t-SNE also have parameters to tune and follow different assumptions.

## II. Exploratory Data Analysis

To begin with our research in the evaluation of various DR techniques, the major task was to come up with a comprehensive metric which accurately compares DR techniques on the diverse datasets. In order to achieve this, we were required to experiment with the wide range of methods and datasets to ensure that the evaluation is reliable. We chose datasets from different fields which provide grounds for unbiased evaluation of models. Running DR methods on a wide range of datasets with high dimensionality which find purpose in distinct real-life problems helps provide the fair ground for evaluation. The datasets we started with are:

- MNIST
- Gene expression cancer RNA-Seq Data Set
- F-MNIST
- COIL-20
- CIFAR-10
- STL-10

Thus, after careful analysis of methods and variations of methods, we chose 6 different DR algorithms based upon their applications in real-world problems and how frequently they have been used for DR and further for building models. They can be classified under two broad categories:-

**Table 1: The classification of DR methods**

| Projection Based | Component Based |
|---|---|
| tSNE | PCA |
| Isomap | MDS |
| UMAP | FastICA |

*Inferences:* We had studied embeddings by various models, we found tSNE and UMAP to produce the best results: For tSNE, it can be seen shrinking the clusters which already sparse and expanding the dense clusters. The perplexity parameter value was 30 in our case. This ensured that cluster of classes did was not spread out too much. On further exploration, we found out that for t-SNE to give best result we need to set the parameters associated with it very wisely, in order to obtain good results. For UMAP, One of the best clustering and well-defined boundary was seen in case of both datasets when UMAP was used for dimension reduction. Apparently, it grouped the individual digit classes and simultaneously retained the overall overall global structure among the different digit classes – keeping 1 far from 0, and grouping triplets of 3,5,8 and 4,7,9 which can blend into one another in some cases. This shows that UMAP is a high-performance algorithm.

## III. DR Algorithms for Comparison

*t-Stochastic Neighbour Embedding(t-SNE):* One of the most robust non-linear dimensionality reduction techniques which aims to reduce the Kullback-Leibler divergence of similarities of a pair of points x and y in high dimensional space H and low dimensional space L (equation mentioned below) [10]. As the developers of tSNE state, the goal they had in mind was to capture the local structure of high-dimensional data well and at the same time, disclose the global data structure using clustering [14]. The wrapping of tSNE in dimRed package is Rtsne and it provides us with the perplexity parameter to tune which allocates the size of kernel neighbourhood [10]. One important point here to use dimRed package is that generally tSNE takes a runtime of O(n$^2$) . However using the Rtsne library, we can achieve the runtime of O(nlogn) which helps deal with the runtime issues of tSNE which we faced during the initial stages of our project [12].

$$\mathrm{KL}(H||L) = \sum_{x \neq y} h_{xy} log h_{xy} / l_{xy}$$

*Unifold Manifold Approximation (UMAP):* Currently dominates the DR techniques due to various reasons which include the following: robust method, preservation of global structure, low runtime and memory usage and supports various distance metrics. UMAP has several hyperparameters that can be tuned to suit the problem type [15]:

- *n_neighbours:* This parameter is used to restrict the size of the neighbourhood which UMAP uses to learn about the data structure. Low values mean UMAP prioritizes local structure while high values mean global structure concentration [15].
- *min_distance:* Refers to how tightly UMAP packs the points.
- *n_components:* Sets the dimensions of the reduced embedding produced.
- *metric:* states the type of distance metric used to compute distance. For example, euclidean and haversine.

## IV. Datasets

**i. Synthetic Datasets:** We have used synthetic data sets in our analysis for covering metrics the first 5 metrics as stated in the table. As these synthetic datasets are small and contain manifold in an easily visualized manner to grasp the basic properties of each DR method, we chose these. If a DR method performs well with synthetic data, it could be reasonable to expect that method to perform well on real-world data sets [20]. Here, we have used five synthetic data sets to reduce them from 3D to 2D: Swiss Roll, Swiss hole, punctured sphere, Variable Noise Helix and Twin Peaks.

These datasets have been loaded into the dimRed package and have been defined to work with the various methods and metrics and contain 2000 rows each and loaded as an object of the dimRedData class.

**ii. Real World Datasets:**

*MNIST:* Modified National Institute of Standards and Technology dataset is very popularly used in image processing contains 60,000 handwritten 28x28 images for digits 0-9. The training set contains 60,000 examples and the test set contains 10,000 examples. Its multidimensionality of 784 featured vector makes it suitable for studying the various dimensionality reduction models [5].

*Gene expression cancer RNA-Seq Data Set:* A dataset designed to search for genes of interest and with the aim of performing this study in a multivariate setting in order to realize the inter-gene relationships [4]. It contains the gene expressions of cancer patients having either of five classes of tumor. It has multivariate characteristics, 801 instances, 20531 attributes, much associated with tasks of classification and clustering. The high attribute count that makes it suitable for our end goal.

*Micromass Data Set: I*t is a dataset to explore machine learning approaches for the identification of microorganisms from mass-spectrometry data. This dataset distributes into 20 classes and hence we chose these metrics for analyzing the performance of our model. Data Set Characteristics is Multivariate with 931 instances and number of attributes is 1300 [6].

## V. Qualitative Comparison- Visualization of embedding:

Since different dimensionality reduction models aim at the preservation of different structures: local or global, we evaluated the models by plotting the reduced dimensions and compare two models on the same dataset and derive qualitative results.

*DimRed Package Visualizations:* It is a framework for Dimensionality Reduction written in R containing the techniques provided as an open-source software package that enabled us to visualize high-dimensional data and provided easy access to multiple classical and advanced DR methods using a common interface and calculate values of different metrics [10].

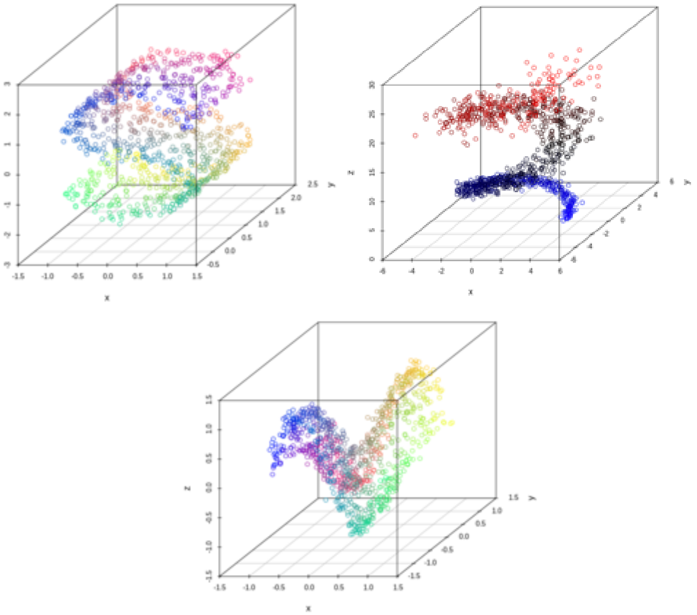


Fig 5.1 3D Representations of Datasets in dimRed

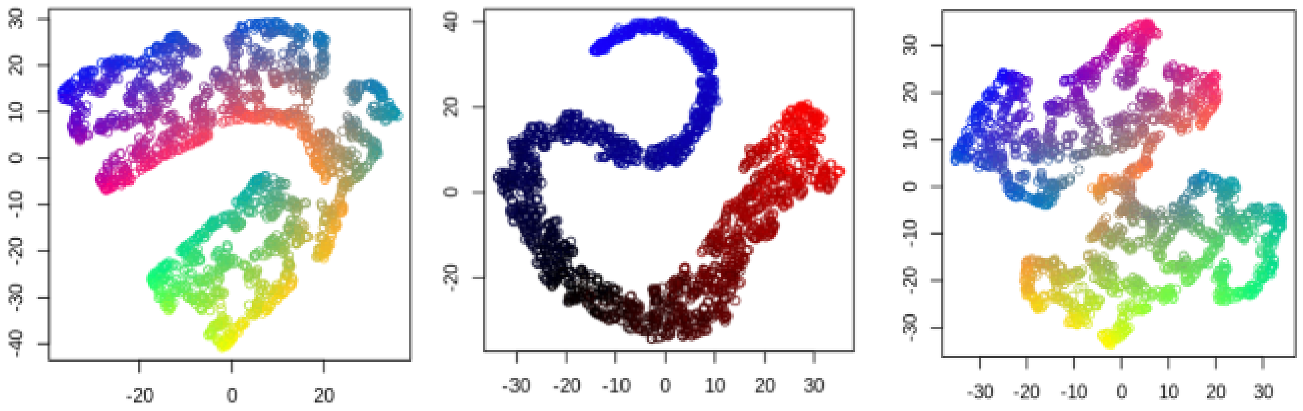


Fig 5.2: t-SNE reduction of datasets (i) 3D-S Curve (ii) Variable Noise Helix (iii) Twin Peaks

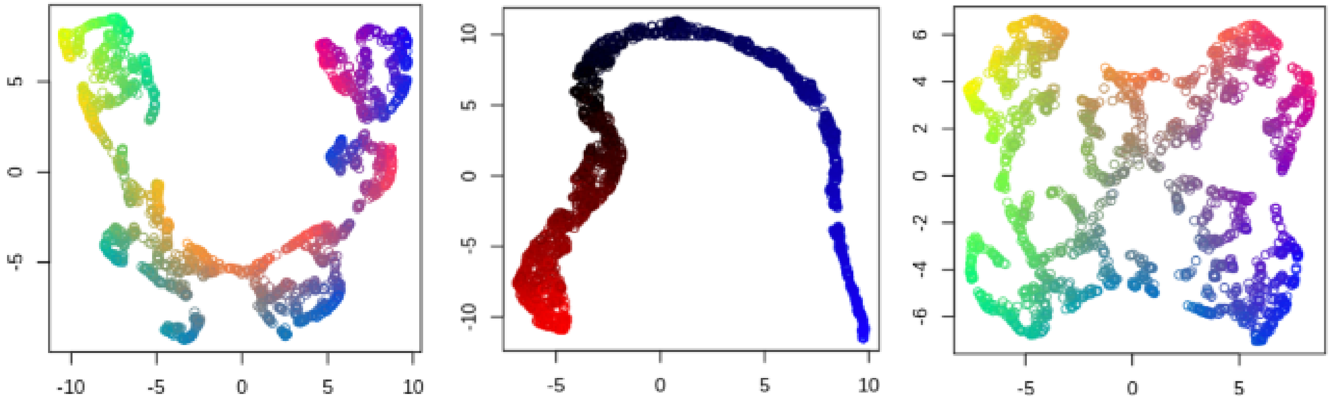


Fig 5.3: UMAP reduction of datasets (i) 3D-S Curve (ii) Variable Noise Helix (iii) Twin Peaks

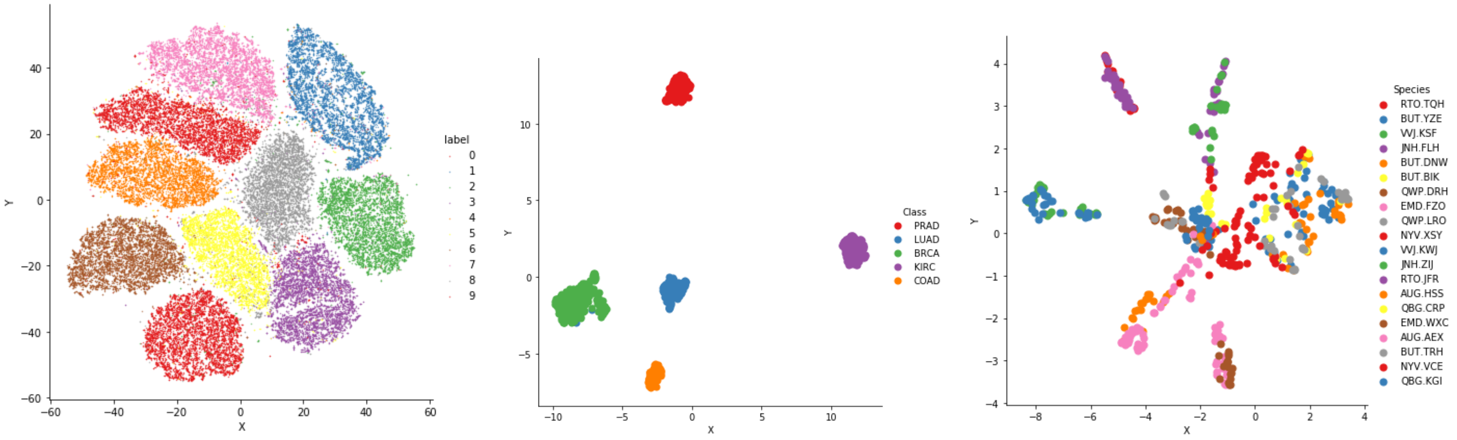


Fig. 5.4. Embedding of t-sNE on (i) MNIST (ii) RNA-Seq Data Set (iii) Micromass Data Set

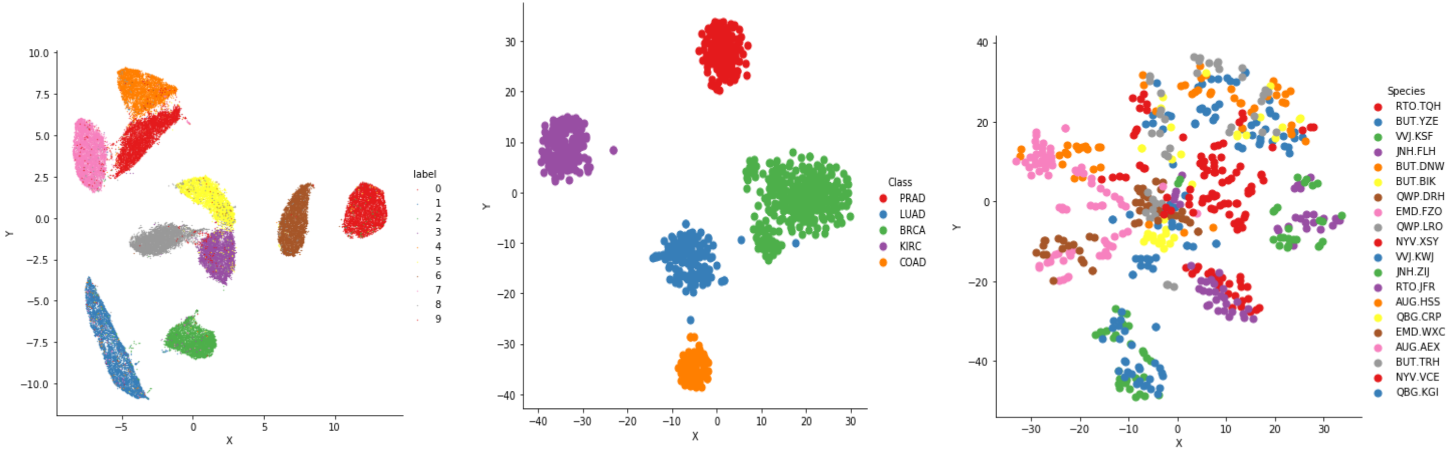


Fig 5.5. Embedding of UMAP on (i) MNIST (ii) RNA-Seq Data Set (iii) Micromass Data Set

*Observations from Fig 5.2 and 5.3:* The 3D visualization of the three datasets seems to be well represented using the dimRed package. The points are clearly shown and it fulfills the purpose of trying to understand the dataset completely. The plots justify the reason behind taking up the synthetic datasets and help in clear understanding of the metrics we will use to compare the dimensionality reduction methods.

*Observations from Fig 5.4 and 5.5:* The cluster distances for tSNE ensure that the points within the clusters are similar but cannot comprehend if two clusters are similar. We can account for the same for the following reason: Stochastic gradient descent was used in UMAP for instead of the regular gradient used in tSNE while calculating the gradient from a random subset [11].

## VI. Metric Based Quantitative Comparison

To facilitate the study of evaluating the two techniques we finalised previously-tSNE and UMAP, we developed two separate sets of metrics and used a combination of the two sets for the final measure. The two sets and their approaches have been explained in the next section and a brief overview of the methods and datasets have been given in figure.

**Table 2: An Overview of the two sets of metrics**

| Sr. No. | Name of Measure | Type of dataset used |
|---|---|---|
| 1 | Local neighborhood preservation ( Q-local ) | **Synthetic Dataset:**<br>● 3D S Curve<br>● Variable Noise Helix<br>● Twin Peaks |
| 2 | Global neighborhood preservation ( Q-global) | |
| 3 | R_NX | |
| 4 | AUC_lnK_R_NX | |
| 5 | Variation of Q_NX with dataset size | |
| 6 | Trustworthiness | **Original Real world Dataset:**<br>● MNIST - benchmark<br>● RNA-Seq Data Set - high attribute count<br>● Micromass Data Set Biomedical Dataset |
| 7 | LCMC | |
| 8 | Continuity | |
| 9 | Classification Score | |
| 10 | Execution Time for RFT Algorithm | |

### i. Metrics Section 1:

In order to cover the first set of metrics on our DR methods, we had to understand the DimRed package present in the R language. The reason for choosing this particular library was the lack of a robust library to visualize the metrics we want to cover in the language Python. At first, it took us some time to get familiar with R syntax but we caught hold of the implementation in matter of few weeks. The library has been provided for the convenience of data scientists to produce reduced embeddings using various dimensionality reduction methods efficiently and apply metrics to evaluate their performance on varied data.

Unlike supervised methods, unsupervised methods do not use a target variable for learning, it represents the data in different forms like clusters or classes for broader understanding. Unsupervised methods do not provide direct evaluation of quality of embedding like classification accuracy as different methods try to minimize errors from different perspectives [10]. The base metric is Co-ranking matrix where $r_{ij}$ represents that two points $x_i$ and $x_j$ are $r_{ij-}$ the closest neighbours in set X [10].

$$q_{kl} = |(i,j) : \hat{r_{ij}} = k \text{ and } r_{ij} = l|$$

The 6 main metrics using Co-ranking matrix as the base:

**$Q_{NX}$ (K):** This metric basically provides us a normalized quality of embedding using the number of data points in k-th nearest neighbors in the original data as well as its reduced form [10].

**Local Continuity Meta Criterion LCMC(K):** This is a very commonly used criterion when it comes to checking the $Q_{NX}$ (K) values for random embeddings. We will be using this metric in our next set of metrics. A well defined $k_{max}$ is an added advantage to our experimental analysis as we will see how it plays an important role in understanding the n_neighbours values [10].

**$k_{max}$ :** It provides the best possible representation of the manifold of original data structure in 2D [10].

**$Q_{local}$ and $Q_{global}$:** As the names suggest, these two measures quantify the local and global distances. For understanding these two measures, let us take up Isomap to understand the variation of **$Q_{local}$** k nearest neighbours. Choosing **$Q_{local}$** prioritizes local distance preservation. We experimented with the variable Noise Helix dataset and found these results. After visualising the Q_local results with k, we find that at k = 23, we find the maximum quality when considering the local distance preservation. After plotting the embeddings for k=5, k=kmax, k=500, we found the following embeddings. We plotted the graph for k=0 to 40 so as to visualize the changes, after 40, the value of quality remained somewhat constant and made it hard to see the variation, refer fig 6.1 and 6.2.

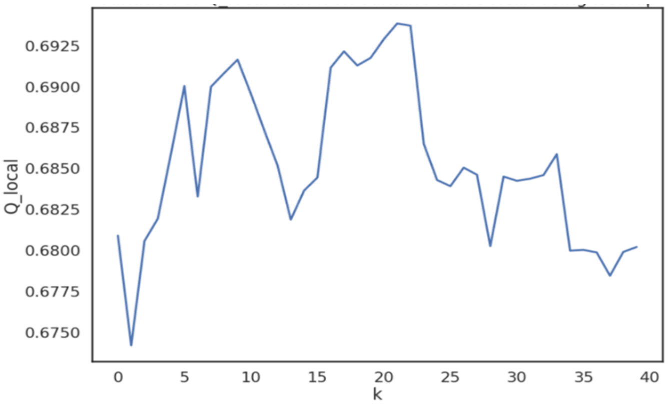


Fig. 6.1 Variation of Q_local with k on Variable noise Helix

**$R_{NX}$(k):** $R_{NX}$(k) is the normalized value of LCMC(k) and provides a meaningful measure of quality. We are using the values of $R_{NX}(k)$ to understand the performance of models with respect to the area under the curve which is explained below. The value of 1 means that the embedding is good and near to 0 means it is a random embedding [10].

**$AUC_{\ln\ k}$ ( $R_{NX}(k)$):** In simple terms, this measure can be compared to the area under the curve while finding accuracy of any machine learning model. However, here it inputs k at a log scale and it provides good results for DR methods preserving local distances as we will see in the following section [10]

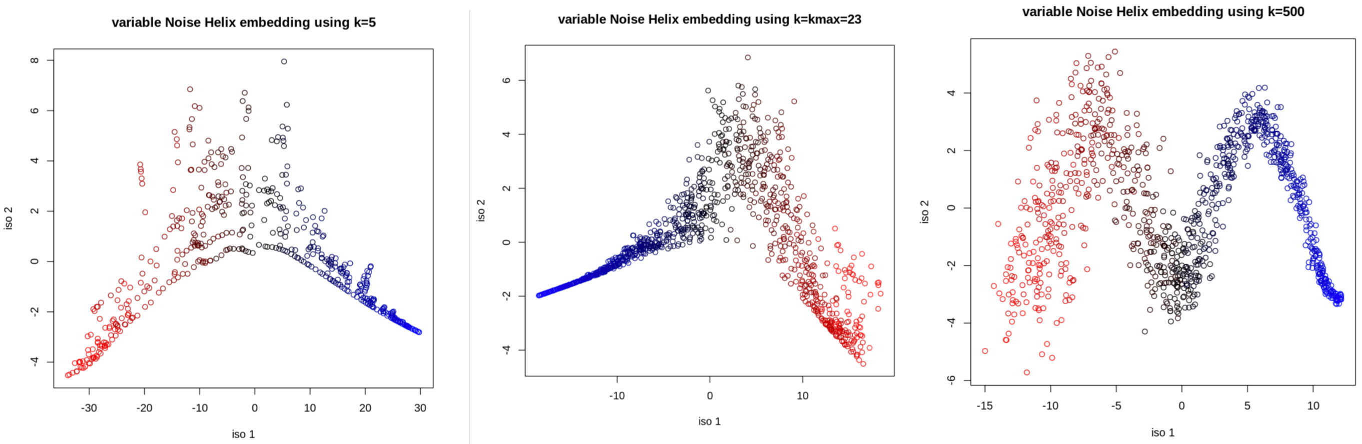


Fig 6.2: Comparison: Embeddings using different values of k (i). The value of k is too small and we can observe holes in the embedding as the manifold's inner composition is preserved but the recovery will be inaccurate. (ii) This value of k seems to perfectly represent the manifold structure and provides better reconstruction. (iii)The value k=500 seems to be too large and the quality value decreases as seen from the graph.

*Variations of R_NX with K and AUC values:* R_NX provides a holistic view of the metric by giving us the variation over both global and local performances of embedding methods used. Using the plot_R_NX function to visualise the quality of embedding with respect to different neighbour sizes. So, we plotted the results for all three datasets and found out the AUC values as mentioned in the legend. We obtained similar results for all three datasets where tSNE takes over UMAP when the number of neighbours is low and UMAP provides a higher R_NX value when the number of neighbours increases. Thus, we can conclude that UMAP preserves the global data structure and tSNE preserves the local data structure fig 6.4.

*Variation of Q_NX with increasing dataset size:* From fig 6.3 visualizing the values of $Q_{NX}$ with increasing datasize, i.e. as the number of rows increase in the dataset, we find a common distribution of values for t-SNE and UMAP over our datasets. As discussed above, we know that $Q_{NX}$ refers to the number of data points that occur in the k-nearest neighbours in the original dataset and the embedding [16]. We observe that for t-SNE, the values of quality increase till a certain local maxima and probably decrease to a minima after. However, for the maximum number of rows, the values are highest. Thus, we can say that to produce good quality embeddings, either look for that local maxima or go with the highest possible rows. Here, we have considered the variation over 2000 rows. Since t-SNE is high on time and memory usage as concluded after our secondary stage analysis, we decided to find the local minima of number of rows and compute efficiently.

For UMAP, we find a contrasting variation. We find that as we increase the size of the dataset, we observe a steady increase to a point and after which, the quality either remains constant after or decreases to some extent. As expected and observed in t-SNE, for a greater number of rows, the quality is the highest. We can conclude that if required to run the UMAP reduction on a high dimensional dataset, we should find that local maxima after which $Q_{NX}$ remains constant and run the model efficiently reducing time and memory resources.

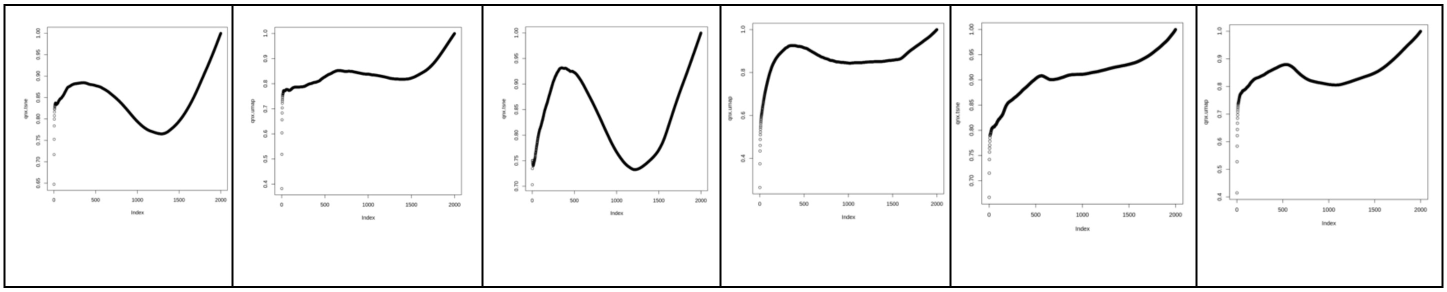


Fig. 6.3 Variation on (i) 3D S Curve (ii) Variable Noise Helix (ii) Twin Peak with t-SNE and UMAP respectively

**Table 3: Metrics: Q_local, Q_global, Mean_R_NX, AUC**

| Datasets | Q_local | | Q_global | | Mean $R_{NX}$ | | AUC | |
|---|---|---|---|---|---|---|---|---|
| | t-SNE | UMAP | t-SNE | UMAP | t-SNE | UMAP | t-SNE | UMAP |
| **3D S Curve** | 0.78 | 0.71 | 0.34 | 0.35 | 0.48 | 0.56 | 0.79 | 0.65 |
| **Variable Noise Helix** | 0.73 | 0.69 | 0.28 | 0.32 | 0.55 | 0.61 | 0.70 | 0.62 |
| **Twin Peaks** | 0.76 | 0.72 | 0.36 | 0.34 | 0.62 | 0.57 | 0.74 | 0.65 |
| **Σ value / 3** | ***0.756*** | 0.706 | 0.326 | ***0.336*** | 0.55 | ***0.58*** | ***0.743*** | 0.64 |

*Q_local, Q_global, Mean_$R_{NX}$, AUC observations:* In table 3, we computed the Q_local, Q_global, Mean_R_NX, AUC values for the three datasets for t-SNE and UMAP. To develop a fair evaluation metric of our own, we computed the mean on the three datasets for each method and plotted the results (Section VII). We observe that t-SNE defeats UMAP for metric Q_local and reverse for Q_global. This can be accounted by the fact that t-SNE very well maintains the closer distances and UMAP concentrates more on maintaining the structure as a whole.

From table 3, Mean_R_NX value for UMAP is higher than that for t-SNE as R_NX is high for higher values of neighbourhood members chosen signifies global gradient preservation. This is further clarified by the $AUC_{1/k}$ value which is higher for t-SNE.

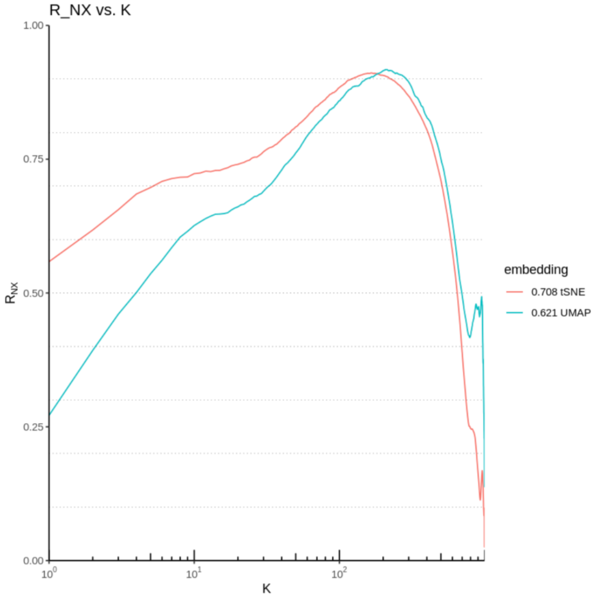


Fig 6.4: R_NX v/s K on variable noise Helix dataset

### ii. Metrics Section 2:

**Methodology:**

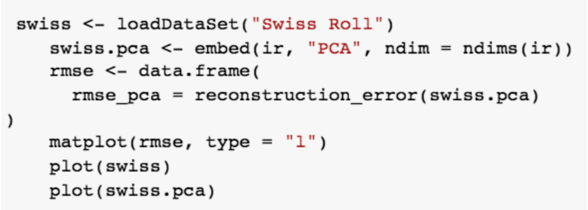
1. Standardizing all datasets using Sklearn - standardscaler.
2. Data Scaling using PCA
3. Fitting the Machine Learning Model on the reduced embedding
4. Analysing the values of different metrics.

**Ensembling with t-SNE and UMAP with PCA:**

Prior to working on original data with UMAP and t-SNE, we had scaled the high dimensional datasets - CIFAR10, RNA-Seq, FMNIST and MNIST to 100 features using PCA during working on our progress report. Similarly, we scaled our new Micromass Data Set as well. This step highly required so as allow us work with t-SNE, because its computation time is very high. t-SNE does not scale efficiently when it comes to datasets in tens of thousands of data points and more than 30-50 features and struggles mightily with millions of data points and 50+ features. PCA while preserving the important data structure and PCA efficiently separates information from noise. To keep it fair with UMAP as well, we used PCA scaling in this case also. This was kind of the ensembling that we proposed to do, in our project plan. An ensemble of LDR + NLDR techniques. Not only, it proved advantageous in terms of time but also produced embeddings, which gave higher classification score when tested over the K Neighbour Classifier.

*Why PCA scaling was apt ?* One of the fairest measures of an algorithm that performs dimensionality reduction is how well it performs the inverse mapping and is able to reconstruct the original data from the reduced embedding [10]. This is measured using the reconstruction error, i.e., reconstructing the original data from a limited number of dimensions. Though not many methods provide such forward and inverse mappings to permit reconstruction. Fortunately, PCA provides it and reconstructs the data very well, thus aiding our motive. For instance consider the plots below for original 3D representation of Swiss Roll Dataset and it reconstructed 2D, followed by the Rmse curve that falls down to zero after having extracted the entire dataset [7]. Refer fig 6.5

*Code Snippet- dimRed Package in R:* reconstruction_error() function is a fair assessment of the quality of embedding of the two dimensional embedding and the dimRed package provides the inverse mapping for the DR methods.



```
swiss <- loadDataSet("Swiss Roll")
   swiss.pca <- embed(ir, "PCA", ndim = ndims(ir))
   rmse <- data.frame(
     rmse_pca = reconstruction_error(swiss.pca)
)
   matplot(rmse, type = "l")
   plot(swiss)
   plot(swiss.pca)
```

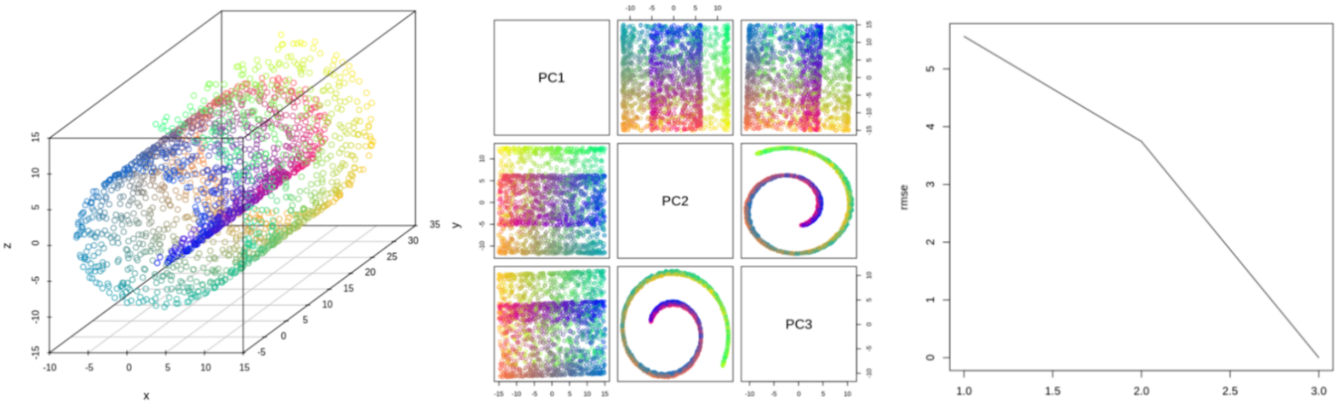


Fig 6.5 (i) 3D Swiss Roll (ii) PCA Reconstruction Visualization on Swiss Roll (ii) Decreasing Rmse

**Table 4: Metrics: Classification Score, Execution Time(sec), Memory (bytes), Trustworthiness, LCMC, Continuity**

| Datasets | **Classification Score** | | **Execution Time(sec)** | | **Memory (bytes)** | | **Trustworthiness** | | **LCMC** | | **Continuity** | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | t-SNE | UMAP | t-SNE | UMAP | t-SNE | UMAP | t-SNE | UMAP | t-SNE | UMAP | t-SNE | UMAP |
| **MNIST** | 0.134 | 0.937 | 147 | 12 | 490336 | 270336 | 0.716 | 0.92 | 0.383 | 0.954 | 0.873 | 0.451 |
| **RNA-Seq** | 0.73 | 1.00 | 1 | 4 | 2699 | 110592 | 0.736 | 0.89 | 0.342 | 0.923 | 0.865 | 0.333 |
| **Micromass** | 0.63 | 0.86 | 157 | 62 | 598832 | 32004 | 0.583 | 0.638 | 0.223 | 0.732 | 0.543 | 0.223 |

*Trustworthiness and Continuity:* They are essentially complementary factors [17]. On projecting data to lower dimension we lose original proximity information in between the points [10]. Some data points wrongly tend to get closer and the extent to which happens is measured by trustworthiness. Continuity is just opposite, it tells which proximity that was present in the original direction is also retained in the reduced data [18].

Continuity values for all datasets was very high in case of t-SNE. We accounted for the ability of t-SNE to retain local information better as compared to UMAP. However, UMAP achieved higher values of trustworthiness ~0.8 clearly stating that global information takes care of wrongly bringing data points together.

*Classification Score and Execution Time for KNN and RFT Classifier*: We fed our reduced data into 2 classifier, earlier we used KNNs, this time to bring variation we used RFTs. Because KNNs were significantly slower on larger datasets we used RFT which used averaging to improve the predictive accuracy and control over-fitting. Classification Score for UMAP embedded data was observed to be way higher than t-SNE embedded dataset. This clearly speaks about the precision and unerring nature of UMAP. Along, with the very small running time it is definitely a state of the art model in present era.

*Memory:* t-SNE consumed a lot of memory as opposed to UMAP. We researched and ran t-SNE for different values of perplexity and discovered a high correlation between Memory consumption and perplexity. After increasing perplexity value over 35, memory increased but performance was unstable.

## VII. Results

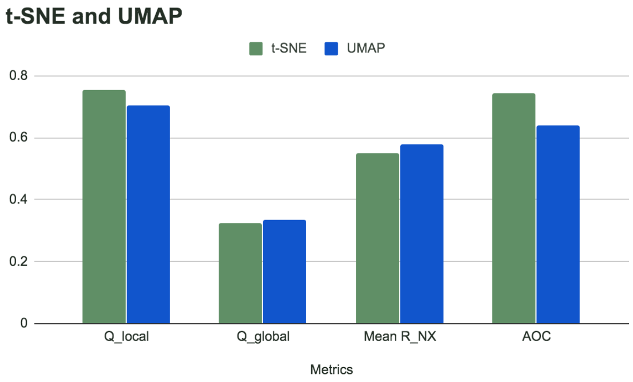


Fig 7.1: Metric set 1s: Q_local, Q_global, Mean_R_NX, AUC

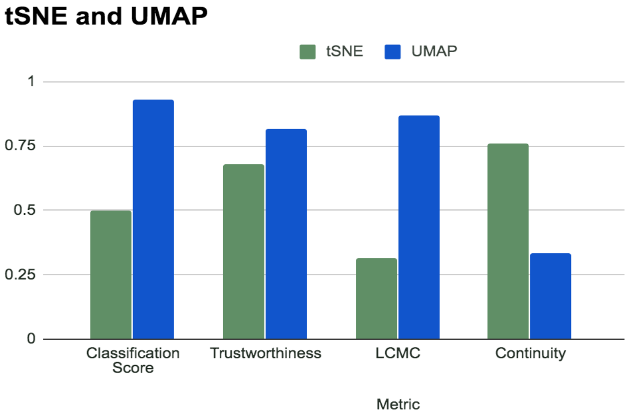


Fig 7.2: Metrics set 2: Classification Score, LCMC, Continuity

*Equations (1 and 2)*:

$$m1 = [\frac{Q_{local} * K_{max} + Q_{global} * K_{max}}{2}] + 2 * mean_{Rnx} + AUC_{1/k}$$

$$m2 = 2 * (Trustworthiness + LCMC + Continuity) + 2 * ClassificationScore + [\frac{Runtime + memory}{2}]$$

**Normalized average Index (Our Metric) :** After the computation of the above mentioned metrics, we researched about the various combinations of metrics available and how to develop our own metric using approaches like weighted average, min bubble up, max bubble up and majority rules [8]. Since our strategy involves two separate sets of metrics, we carried out a thorough mathematical analysis of the all the quantitative metrics as mentioned below and computed two metrics $m_1$ and $m_2$ as weighted sums. First, we normalized all metric values as required by our metric. The metrics $Q_{local}$ and $Q_{global}$ are both dependent on the values of $K_{max}$, so to interpret the collective quality of embedding $Q_{nx}$, we took weighted mean using $K_{max}$ as the weights to compute an aggregate $Q_{nx}$ and assigned it a weight of 1. As found by experimental analysis similar to section Embeddings using different values of k, $K_{max}$ values are 31 for t-SNE and 19 for UMAP. Next, mean_R_NX and $AUC_{1/k}$ seemed to be a crucial standalone metric so we assigned a weight of 2 to it and $AUC_{1/k}$ a weight of 1. Finally, we calculated $m_1$ using the following formula (1). Next to compute a combination of metrics from section 2, we found that Trustworthiness and Continuity are complementary to each other and LCMC also shares the same idea of predicting the quality of dimensionality reduction by the tendency of points to stay together in the reduced dimensions. Thus, we clubbed them together as an aggregate TCL metric and gave it a weight of 2. Classification score is another pillar of this set of metrics and we gave it a weight of 2 considering its relevance towards our main motive and runtime for running the classification algorithm a weight of 1 as it is not that relevant from our perspective. Next, we clubbed runtime and memory together as they usually form the space-time trade-off and took an average along with a weight of 1. Finally, we took the weighted sums as $m_2$ (2).

*Observations:*

| metric | UMAP score | t-SNE score |
|---|---|---|
| m1 | 11.802 | 9.614 |
| m2 | 5.89 | 4.495 |
| mean | **8.846** | 7.0545 |

*Conclusion:*

We tried to evaluate all the DR methods in an unbiased manner by the selection of metrics which evaluate the quality of embedding using independent rubrics. Though t-SNE performs well with methods that directly deal with the local preservation of data structure like the $Q_{local}$ and AUC measures, we find that the runtime and memory issues while running on large datasets like MNIST and CIFAR10 are serious issues to be considered while using t-SNE as the standalone reduction technique. However, using ensemble models like PCA+t-SNE could be used to deal with these issues as we stated in our analysis (Metrics section 2). When dealing with UMAP, we observed an ease in application to large datasets as well as great classification score after reduction. Also, the tuning of hyperparameters like n_neighbours, min_distance, n_components and metric provide a fair playground for testing the quality of $Q_{NX}$ as illustrated in fig 6.2 Thus, we conclude that according to our research and metric, UMAP is a reliable and robust method of Dimensionality Reduction.

We can conclude by stating the fact that while encountering a problem that requires local data structure preservation at some cost of computational issues, one might consider the use of t-SNE or an ensemble model using it. Further, when required to preserve the global gradient and high computational efficiency at the cost of loss of local data structure, we could consider using UMAP. This is just an example of how to choose the apt method of DR from the vast range of linear and non-linear DR methods introduced and will be introduced in future. The selection of the right DR technique affects the resource usage as well as prediction accuracy of a machine learning model to a great extent.

## VIII. Future Work

Due to time constraint, we have not been able to find the result of running Neural Network based models of our embedding. This might have given some interesting insights into how unsupervised models behave on reduced data. For attaining a holistic view of the quality of metrics we covered, the essential set of metrics on different possible datasets. In future we will try to cover Metrics Set1 and Metrics Set2 on same datasets of different types and sizes. We intend to continue our research on further evaluations of DR algorithms based on the points stated above. We will also try to incorporate other metrics of dimRed package like Cophenetic Correlation, reconstruction error and total correlation. As reconstruction error is a very crucial metric in understanding the reconstruction accuracy of a DR method. We tried implementing this metric using the dimRed package, however, since it does not provide an inverse mapping for tSNE and UMAP. We have visualized the results for PCA.

Even though we tried to incorporate metrics of varied types and datasets from different backgrounds, we still plan on improving our method to consider other aspects of evaluation as mentioned in the following section.

## IX. References

[1] https://nbisweden.github.io/excelerate-scRNAseq/session-dim-reduction/lecture_dimensionality_reduction.pdf
[2] https://data-flair.training/blogs/dimensionality-reduction-tutorial/
[3] https://github.com/gdkrmr/dimRed/tree/master/R
[4] Dua, Dheeru and Graff, Casey, "{UCI} Machine Learning Repository," [Online]. Available: http://archive.ics.uci.edu/ml.
[5] "MNIST dataset," [Online]. Available: http://yann.lecun.com/exdb/mnist/ .
[6] https://archive.ics.uci.edu/ml/datasets/Micromass
[7] http://www.cs.cmu.edu/~02317/slides/lec_6.pdf
[8] https://www.ibm.com/support/knowledgecenter/en/SSEP7J_10.1.1/com.ibm.swg.ba.cognos.ug_mm.10.1.1.doc/t_sc_metrics.html
[9] dimRed functions: https://cran.r-project.org/web/packages/dimRed/dimRed.pdf
[10] dimRed and coRanking—Unifying Dimensionality Reduction in R Guido Kraemer Markus Reichstein Miguel D. Mahecha May 7, 2019
[11] https://towardsdatascience.com/how-exactly-umap-works-13e3040e1668
[12] https://towardsdatascience.com/how-to-tune-hyperparameters-of-tsne-7c0596a18868
[13] https://www.nature.com/articles/nbt0308-303
[14] https://lvdmaaten.github.io/publications/papers/JMLR_2008.pdf
[15] https://buildmedia.readthedocs.org/media/pdf/umap-learn/latest/umap-learn.pdf
[16] http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.148.2808&rep=rep1&type=pdf
[17] https://pdfs.semanticscholar.org/23da/c0c4f81472c0261c51669e1d244d52f0c21d.pdf
[18] https://pdfs.semanticscholar.org/23da/c0c4f81472c0261c51669e1d244d52f0c21d.pdf
[19] https://arxiv.org/abs/1403.2877
[20]http://web.stanford.edu/~yaoliu/docs/lopa.pdf
[21] https://www.datacamp.com/community/tutorials/introduction-t-sne