Visualising How Non-linear Dimension Reduction Warps Your Data

Jayani P.G. Lakshika
Econometrics & Business Statistics, Monash University
and
Dianne Cook
Econometrics & Business Statistics, Monash University
and
Paul Harrison
MGBP, BDInstitute, Monash University
and
Michael Lydeamore
Econometrics & Business Statistics, Monash University
and
Thiyanga S. Talagala
Statistics, University of Sri Jayewardenepura

January 14, 2024

Abstract

Non-Linear Dimension Reduction (NLDR) techniques have emerged as powerful tools to visualize high-dimensional data. However, their complexity and parameter choices may lead to distrustful or misleading results. To address this challenge, we propose a novel approach that combines the tour technique with a low-dimensional manifold generated using NLDR techniques, hexagonal binning, and triangulation. This integration enables a clear examination of the low-dimensional representation in the original high-dimensional space. Our approach not only preserves the advantages of both tours and NLDR but also offers a more intuitive perception of complex data structures and facilitates accurate data transformation assessments. The method and example data sets are available in the **quollr** R package.

Keywords: high-dimensional data, dimension reduction, triangulation, hexagonal binning, low-dimensional manifold, manifold learning, tour, data vizualization

1 Introduction

High-dimensional (high-D) data is widespread in many fields including ecology and bioinformatics (Guo et al. 2023), in part because of new data collection technologies (Johnstone & Titterington 2009, Ayesha et al. 2020). Working with high-dimensional data poses considerable challenges due to the difficulty in visualizing beyond two dimensions (Jia et al. 2022). High-dimensional data also presents difficulties for model fitting (Johnstone & Titterington 2009), both computationally and interpretation, each of which benefits from visualization.

To create visual representations of high-dimensional data, it is common to apply dimension reduction techniques. Linear methods such as principal component analysis (PCA) (F.R.S. 1901) have been used for many years, and remain popular. Non-linear methods such as multi-dimensional scaling (MDS) (Torgerson 1967) have also been routinely used. In the past decade, there has merged many new techniques non-linear dimension reduction (NLDR), such as t-distributed stochastic neighbor embedding (tSNE) (van der Maaten & Hinton 2008), uniform manifold approximation and projection (UMAP) (McInnes & Healy 2018), designed to capture the complex and non-linear relationships present within high-dimensional data (Johnstone & Titterington 2009).

However, projecting high-dimensional data has limitations, such as information loss and potential distortion of essential structures and patterns (Jia et al. 2022, Venna et al. (2010)). The choice of technique and parameters further impacts the accuracy of the visualization, necessitating careful consideration for meaningful interpretation (see Figure 1).

Interactive and dynamic graphics systems have also been developed over the years to enable visualizing high dimensions. One method, called a tour (Asimov 1985), shows a sequence of linear projections is shown as a movie, allowing exploration without warping the space (Lee et al. 2021). Interactive tools like **XGobi** and **GGobi** have been successful in incorporating tours for exploring high-dimensional data (Swayne et al. 1998). The R package **tourr** (Wickham et al. 2011) further enhances tour visualization within R, although it may face limitations in frame rate and interactive features compared to **GGobi**.

To overcome these limitations, the R package **detourr** (Hart & Wang 2022) has been developed, leveraging a Javascript widget via htmlwidgets (Vaidyanathan et al. 2023) to achieve higher frame rates and enhanced interactivity. Additionally, the R package **langevitour** (Paul Harrison 2022) utilizes Langevin Dynamics to generate a continuous path of projections, eliminating the need for interpolation between projections for animation. The tour technique has proven valuable in exploring statistical model fits (Wickham et al. 2015) and factorial experimental designs (Buja et al. 1996). Augmenting the results of non-linear dimensional reduction methods with the tour, as demonstrated in the **liminal** R package (Lee et al. 2020), further enhances data exploration.

While tours (Asimov 1985) preserve space without warping (Lee et al. 2021), they require integrating multiple low-dimensional views mentally to perceive high-dimensional structures. To address this challenge, we propose a novel approach by combining the tour technique with a low-dimensional manifold. This manifold is created through the synergistic use of Non-Linear Dimension Reduction (NLDR) techniques, hexagonal binning, and triangulation. By merging these techniques, our approach offers a comprehensive and efficient means to visualize and explore high-dimensional data while retaining the advantages of

both tours and NLDR. This integration facilitates a more intuitive perception of complex data structures and empowers analysts with a robust tool for assessing the accuracy of data transformations. The implementation of our approach is available as an R package called **quollr**.

The outline of this paper is as follows. The Section 2 provides an detailed overview of dimension reduction methods, triangulation, and tours. Building upon this foundation, the Section 3 delves into the proposed algorithm, **quollr**, and its implementation details. In Section 3.6, discusses the effectiveness of the learned low-dimensional manifold in accurately representing the complex high-dimensional data. Following that, Section 3.7 presents simple examples from simulations to illustrate the functionality of the algorithm. Subsequently, Section 4 showcases real-world applications of **quollr** on different data sets, particularly in single-cell RNA-seq data. These applications reveal insights into the performance and trustworthiness of NLDR algorithms. We analyze the results to identify situations where NLDR techniques may lead to misleading interpretations. Finally, **?@sec-conclusions** concludes by summarizing the findings and emphasizing the significance of the proposed approach in tackling the challenges of high-dimensional data visualization.

2 Background

2.1 Dimension Reduction

Consider the high-dimensional data a rectangular matrix X, where $X = \begin{bmatrix} \mathbf{x}_1 & \mathbf{x}_2 & \cdots & \mathbf{x}_n \end{bmatrix}^\top$, with n observations in p dimensions. The objective is to discover a low-dimensional projection $Y = \begin{bmatrix} \mathbf{y}_1 & \mathbf{y}_2 & \cdots & \mathbf{y}_n \end{bmatrix}^\top$, represented as an $n \times d$ matrix, where $d \ll p$. The reduction process seeks to remove noise from the original data set while retaining essential information.

There are two main categories of dimension reduction techniques: linear and non-linear methods. Linear techniques involve a linear transformation of the data, with one popular example being PCA. PCA performs an eigen-decomposition of the sample covariance matrix to obtain orthogonal principal components that capture the variance of the data (F.R.S. 1901). However, linear methods may not fully capture complex non-linear relationships present in the data.

In contrast, NLDR techniques generate the low-dimensional representation Y from the high-dimensional data X, often using pre-processing techniques like k-nearest neighbors graph or kernel transformations. Multidimensional Scaling (MDS) is a class of NLDR methods that aims to construct an embedding Y in a low-dimensional space, approximating the pair-wise distances in X (Torgerson 1967). Variants of MDS include non-metric scaling (Kruskal 1964) and Isomap, which estimate geodesic distances to create the low-dimensional representation (Silva & Tenenbaum 2002). Other approaches based on diffusion processes, like diffusion maps (Coifman et al. 2005) and the PHATE algorithm (Moon et al. 2019), also fall under NLDR methods.

A challenge with NLDR methods is selecting and tuning appropriate parameters. One specific technique we focus on is Pairwise Controlled Manifold Approximation (PaCMAP).

Similar considerations apply to related methods like tSNE (van der Maaten & Hinton 2008), UMAP (McInnes & Healy 2018), and TrMAP (Amid & Warmuth 2022).

It is important to note that methods like PCA and auto-encoders (Rumelhart et al. 1986) provide a reverse mapping from the low-dimensional space back to the high-dimensional space, enabling data reconstruction. However, many non-linear methods, including tSNE, prioritize visualization and exploration over reconstruction. Their focus is on capturing complex structures that may not be easily represented in the original space, making a straightforward reverse mapping challenging.

2.1.1 Non-linear dimension reduction techniques

Non-linear dimension reduction techniques play a crucial role in the analysis and visualization of high-dimensional data, where the complexities of relationships among variables may not be adequately captured by linear methods. Among these techniques, tSNE stands out for its emphasis on preserving pairwise distances. By minimizing the divergence between probability distributions in both the high and low-dimensional spaces, t-SNE effectively reveals intricate structures and patterns within the data. Its application is widespread in tasks requiring the visualization of clusters and local relationships, though it does require careful consideration of the perplexity parameter for optimal results.

UMAP is another powerful non-linear technique that strikes a balance between preserving local and global structures. Constructing a fuzzy topological representation using a weighted k-nearest neighbors graph, UMAP optimizes the low-dimensional embedding to resemble this representation. Known for its efficiency and scalability, UMAP is versatile across various scales of relationships in the data, although parameter sensitivity, particularly concerning the choice of neighbors, must be taken into account.

For trajectory data, PHATE provides specialized capabilities. It models the affinity between data points, simulating a heat diffusion process to capture developmental processes, particularly in single-cell genomics. While PHATE excels in revealing trajectory structures and offering insights into cellular development, it necessitates careful parameter tuning due to its specialized nature.

TriMAP adopts a unique approach by approximating the data manifold through the construction of a triangulated graph representation. This technique efficiently captures both global and local structures by representing the data as a network of triangles. TriMAP's strength lies in its ability to efficiently capture complex structures, albeit with sensitivity to parameter choices, including the number of neighbors.

In contrast, PaCMAP introduces supervised learning to create a low-dimensional representation while preserving pair-wise relationships. Constructing a graph based on pair-wise distances, PaCMAP optimizes an embedding using a customizable loss function. Particularly notable is PaCMAP's flexibility in incorporating class labels or additional information to guide the embedding process, offering users a means to customize its behavior and performance.

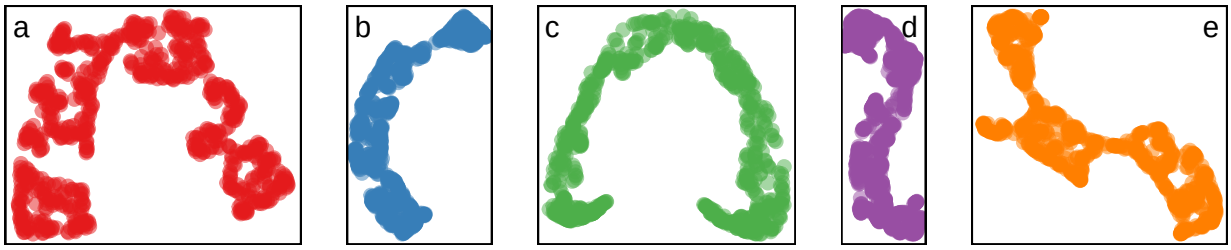

Figure 1: 2D layouts from different NLDR techniques applied the same data: (a) tSNE (perplexity = 27), (b) UMAP (n_neighbors = 50), (c) PHATE (knn = 5), (d) TriMAP (n_inliers = 5, n_outliers = 4, n_random = 3), and (e) PaCMAP (n_neighbors = 10, init = random, MN_ratio = 0.9, FP_ratio = 2). Is there a best representation of the original data or are they all providing equivalent information?

2.2 Linear overviews using tours

A tour is a powerful visualization technique used to explore high-dimensional data by generating a sequence of projections, typically into two dimensions. There are two main types of tours: the Grand Tour and the Guided Tour. The Grand Tour explores the data's shape and global structure by using random projections (Asimov 1985). In contrast, the Guided Tour focuses on specific patterns by moving towards interesting projections defined by a predefined index function (Cook et al. 1995).

The process begins with a real data matrix X containing n observations in p dimensions. It generates a sequence of $p \times d$ orthonormal projection matrices (bases), usually 1 or 2 dimensions. For each pair of orthonormal bases A_t and A_{t+1}, a geodesic path is interpolated to create smooth animation between projections.

In the Grand Tour, new orthonormal bases are randomly chosen to explore the d-dimensional subspace. The data is often sphered via principal components to reduce dimensionality. The Guided Tour uses a predefined index function to generate a sequence of 'interesting' projections. The resulting tour continuously visualizes the projected data $Y_t = XA_t$ as it interpolates between successive bases.

While both tours can be used to visualize data, examples often focus on using the Grand Tour to observe global structures. However, software like **langevitour** can visualize both types of tours, providing flexibility for exploring high-dimensional data with various objectives.

3 Methodology

Our algorithm comprises two main phases: (1) generate the model in the 2D space, and (2) generate the model in the high-D space. These two phases are described in details in this section.

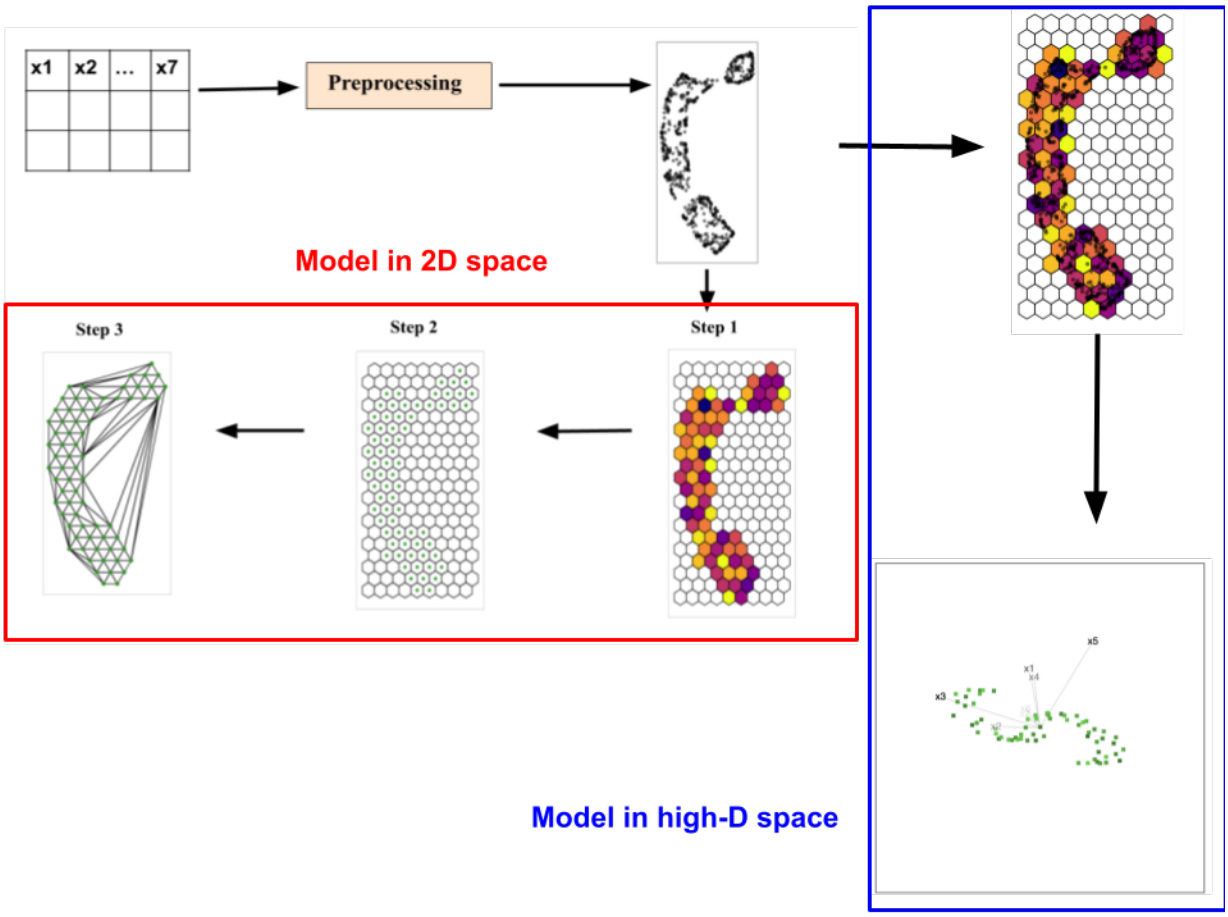

Figure 2: A flow diagram detailing the steps taken to create the low-dimensional manifold in the high dimensional space. There are two basic phases, one to generate the model in the 2D space, and other to generate the model in the high-D space.

3.1 Preprocessing steps

In order to reduce the computational complexity associated with performing NLDR techniques to high-D data, and as a initial step of noise reduction of high-D data, PCA (Jolliffe & Cadima 2016, Howley et al. (2005), Indhumathi & Sathiyabama (2010)) is applied as a preprocessing step. Subsequently, the identified principal components, representing directions of maximum variance, are used to construct the model.

3.2 Constructing the 2D model

Step 1: Computing the hexagonal grid configuration

The hexagonal grid, formed through hexagonal binning (Carr et al. 1987, Carr (1992)), serves as a type of bivariate histogram employed to visualize the structure of high-D data. Hexagons, being one of only three regular polygons capable of tessellating a plane (Carr et al. 2013), possess both symmetry of nearest neighbors and the maximum number of sides for a regular tessellation of the plane. This unique combination makes hexagons more efficient in covering the plane compared to other regular tessellations. Additionally, hexagons exhibit lower visual bias when displaying densities, setting them apart from other regular tessellations (Carr et al. 2023). In our algorithm, hexagonal binning is used as the initial step of constructing the 2D model and the total number of bins (b) is the crucial parameter that defines the granularity of the hexagonal grid.

(a) Determine the number of bins along the x-axis (b_1)

First, the number of bins along the x-axis (b_1) is computed using the relationship between the diameter (h) and the area (A) of regular hexagons (see Equation 1).

$$\mathrm{A} = \frac{\sqrt{3}}{2} h^2 \tag{1}$$

To construct regular hexagons, $A = 1$ (see Figure 3) use as the default setting. Then, the diameter (h) of the regular hexagons is calculated (see Equation 2).

$$\mathrm{h} = \sqrt{\frac{2}{\sqrt{3}} A} \tag{2}$$

Carr et al. (2013) mentioned about the relationship between the diameter (h) of regular hexagons and the height (y) of the plotting region. According to our algorithm, the height (y) of the plotting region is the the range of 2D embedding component 1 (r_1) (see Equation 3).

$$h = \frac{r_1}{b_1} \tag{3}$$

After rearranging the Equation 3 as Equation 4, b_1 is computed. The b_1 value is rounded up to the nearest whole number to have an integer value.

$$b_1 = \frac{r_1}{h} \tag{4}$$

(b) Determine the shape parameter (s)

In this step, we determine the shape parameter (s) for the hexagonal bins, which significantly influences their shape and arrangement within the grid. The s in the hexagonal binning algorithm is defined as the ratio of the height (y) to the width (x) of the plotting region as defined in Equation 5.

$$s = \frac{y}{x} \tag{5}$$

The shape parameter (s) of our algorithm is calculated as the ratio of the ranges of 2D embedding components, where r_1 and r_2 represent the ranges of 2D embedding components 1 and component 2, respectively (see Equation 6).

$$s = \frac{r_2}{r_1} \tag{6}$$

(c) Determine the number of bins along the y-axis (b_2)

Next, the number of bins along the y-axis is computed based on the number of bins along the x-axis (b_1) and the shape parameter (s) (see Equation 7) (Carr et al. 2013).

$$b_2 = 2 * \left(\frac{(b_1 \times s)}{\sqrt{(3)}} + 1.5001 \right) \tag{7}$$

(d) Determine the total number of bins (b)

The total number of bins is determined by multiplying the number of bins along the x-axis (b_1) with the number of bins along the y-axis (b_2) (see Equation 8).

$$b = b_1 \times b_2 \tag{8}$$

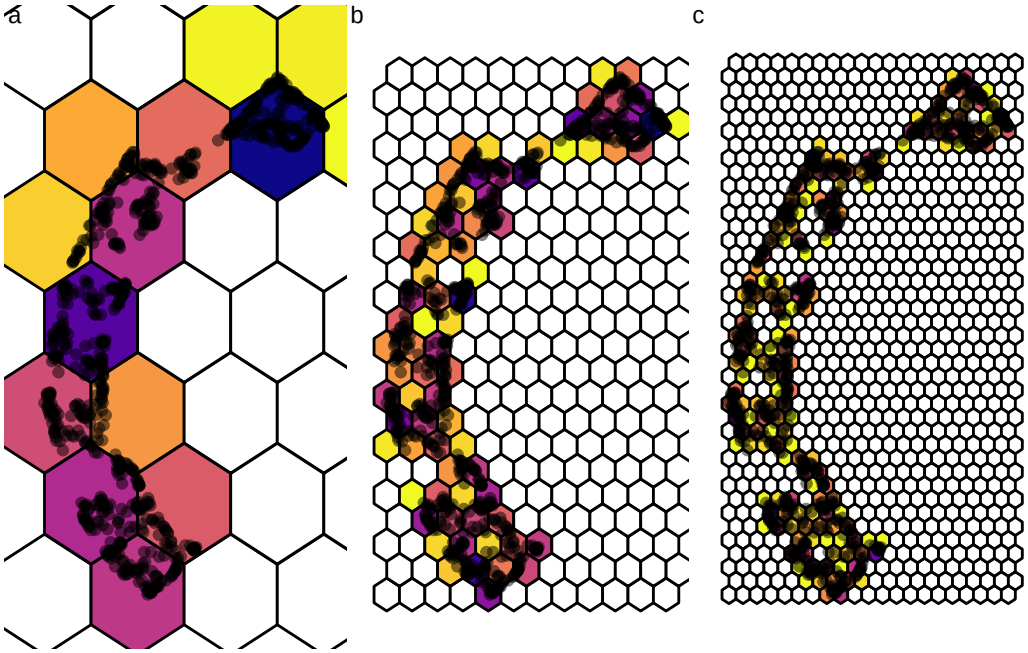

Figure 3: Hexbin plots from different number of bins for the **UMAP** projections of **S-curve** data: (a) b = 32 (4, 8), s = 1.643542, (b) b = 264 (12, 22), s = 1.643542, and (c) b = 840 (21, 40), s = 1.643542. The hexbins are colored based on the density of points, with darker colors indicating higher point density and yellow color representing lower point density within each bin. Does a value of number of bins exist to effectively represent the low-dimensional data?

Step 2: Obtain bin centroids

As a result of hexagonal binning for high-D data, all the high-D data are clustered into hexagons. In this step, the bin centroids ($C_k^{(2)} \equiv (C_{ky_1}, C_{ky_2})$ (see Figure 2 Step 2) are obtained (Carr et al. 2013).

Step 3: Triangulate bin centroids

In this step, the algorithm proceeds to triangulate the hexagonal bin centroids (see Figure 2 (Step 3)). Triangulation is a fundamental process in computational geometry and computer graphics that involves dividing a set of points in a given space into interconnected triangles (Lloyd 1977). One common algorithm used for triangulation is Delaunay triangulation (Lee & Schachter 1980, Renka (1996)), where points are connected in a way that maximizes the minimum angles of the resulting triangles, leading to a more regular and well-conditioned triangulation.

Since we are working with the centroids of regular hexagonal bins, the resulting mesh will predominantly comprise equal-sized regular triangles. However, the triangulation also helps span any gaps that may exist between clusters of points, allowing for a more complete and interconnected representation of the data.

3.3 Lifting the model into high dimensions

3.3.1 Lifting the triangular mesh points into high dimensions

Step1: Cluster 2D points to hexagons

Expanding upon the information regarding hexagonal binning discussed in Step 1 of Section 3.2, the primary objective in this step is to determine the 2D embedding points associated with each hexagon. As the initial process of hexagonal binning, the 2D points are clustered into their respective hexagonal bins. By mapping this information with the hexagonal bin centroids ($C_k^{(2)} \equiv (C_{ky_1}, C_{ky_2})$) that obtained in Step 2 of the 2D model building (see Section 3.2), we can find which 2D points are assigned to which data set (see Figure 4 (a)).

Step2: Cluster high dimensional points to hexagons

Following the step 1, the main focus in this step to determine the corresponding high dimensional points for each hexagon. Every 2D embedding point serves as a projection of a data point belonging to high dimensional space. By using this mapping between the high dimensional data and their corresponding projections, the high dimensional points allocated to each hexagons are determined (see video linked in Figure 4).

Step3: Compute the mean within hexagon

Having identified the high-dimensional points associated with hexagons, the final step involves computing the mean within each hexagonal bin. This implies calculating the average of the high-dimensional data points located within each hex bin. These averaged high-dimensional data points, denoted as $C_k^{(p)} \equiv (C_{kx_1}, ..., C_{kx_p})$, serve as the representative coordinates for the hex bin centroids within the expansive high-dimensional space (see video linked in Figure 4).

3.3.2 Lifting the 2D triangular mesh into high dimensions

Based on the insights gained in Step 3 of Section 3.2, where connected edges in the 2D triangular mesh were identified, this step involves lifting these connections into the high dimensions. Having the mappings of all 2D triangular mesh points into high dimensions, the points connected in 2D are also connected in high dimensional space (see video linked in Figure 4).

3.4 Tunning the model

In our model tuning process, we strategically adjust three key parameters to optimize the performance and accuracy of our approach. They are (i) the total number of bins (b), (ii) a benchmark value to remove low-density hexagons, and (iii) a benchmark value to remove long edges.

3.4.1 Total number of bins

Adjusting the parameter b_1 provides control over the total number of bins b, allowing us to fine-tune the hexagonal grid.

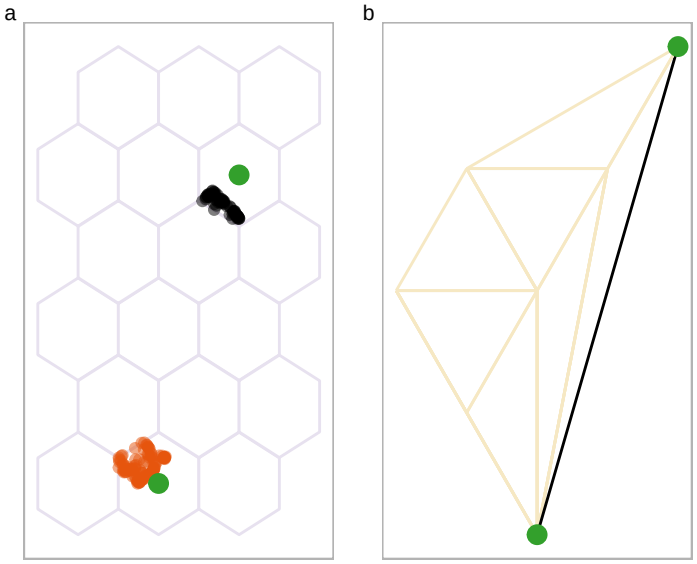

Figure 4: How the 2D model lift into high dimensions? (a) visualize the points and the hexagonal bin centroids related 2nd and 15th hexagons, (b) visualization of the edge connected the 2nd and 15th hexagons (colored in red) in the triangular mesh. A video of tour animation is available at https://www.youtube.com/watch?v=vBvM30Plt24.

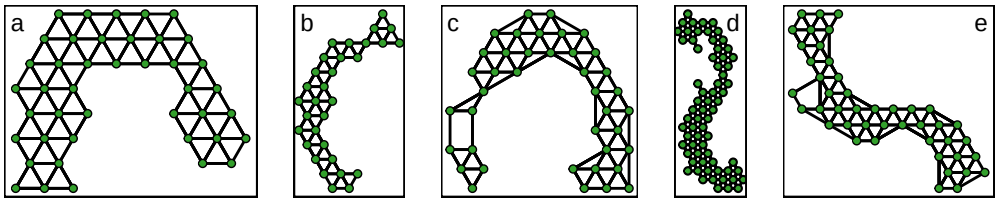

Figure 5: Model.

3.4.2 Benchmark value to remove low-density hexagons

Addressing low-density hexagons is a systematic process to handle sparsely represented data in certain regions. For each hex bin, we identify the six nearest hex bins using an equal 2D distance metric. Then, we calculate the mean density, as outlined in the equations:

$$\text{standard count} = \frac{\text{count}}{\text{max count}} \tag{9}$$

$$\text{mean density} = \frac{\text{standard count}}{6} \tag{10}$$

The standard count is derived from the number of observations in the hex bins. By examining the distribution of mean densities and designating the first quartile as the benchmark value, hex bins with mean densities below this benchmark are removed. This process ensures the elimination of regions with insufficient data density, focusing on areas with more significant data representation and preserving the overall structure in the low-dimensional space.

3.4.3 Benchmark value to remove long edges

The removal of long edges is a critical step to create a smoother representation by iteratively eliminating hexagons with excessive distances between centroids. This process eliminates outliers and noise while preserving essential local relationships within the data. To achieve this, distances between vertices are sorted, and unique distance values are extracted. By setting a threshold based on the largest difference between consecutive distance values, long edges are identified and removed. This refinement step contributes to enhancing the quality of the triangular mesh, ensuring a more accurate representation of the data structure.

3.5 Model summaries

3.5.1 Predicted values and residuals

The approach involves employing the K-nearest neighbors (KNN) algorithm to identify the nearest hexagonal bin centroid in the 2D space. Subsequently, the coordinates of this centroid are assigned as the low-dimensional predicted values for the test data in 2D space. It is noteworthy that traditional NLDR methods, such as tSNE, often lack a direct predict function, making our approach valuable for generating predicted values in the absence of such functionalities.

The concept of "residuals" is pivotal in evaluating the accuracy of representing bin centroids in high dimensions. To quantify this accuracy, we introduce an error metric, which measures the sum of squared differences between the high-dimensional data (x_{ij}) and the predicted bin centroid data in high-dimensional space ($C_{x_i j}$) across all bins and dimensions. Mathematically, this error is expressed as:

$$\text{Error} = \sum_{j=1}^{n} \sum_{i=1}^{p} (x_{ij} - C_{x_i j})^2 \tag{11}$$

Here, n represents the number of bins, p represents the dimensions, x_{ij} is the actual high-dimensional data, and C_{x_ij} is the predicted bin centroid data in high dimensions.

The error metric outlined above provides valuable insights into the overall accuracy of our predictive model. By quantifying the squared deviations between the actual and predicted values across all bins and dimensions, we gain a comprehensive understanding of how well our method captures and represents the underlying structure of the data in the reduced 2D space. This assessment is crucial for evaluating the efficacy of our NLDR technique in preserving the essential information present in the original high-dimensional data.

3.5.2 Goodness of fit statistics

Moving on to the assessment of prediction accuracy, we calculate the Mean Squared Error (MSE). The MSE helps measure the average squared differences between the actual high-dimensional data (x_{ij}) and the predicted bin centroid data in high-D (C_{x_ij}) values across all bins. Mathematically, this is expressed as:

$$\text{MSE} = \sum_{j=1}^{n} \frac{\sum_{i=1}^{p}(x_{ij} - C_{x_ij})^2}{\text{total number of bins}} \tag{12}$$

Here, b signifies the total number of bins, p denotes the number of dimensions in the high-dimensional data, and n represents the number of observations.

Additionally, we gauge the model's performance using the Akaike Information Criterion (AIC), calculated by the formula:

$$\text{AIC} = 2bp + np * log(\text{MSE}) \tag{13}$$

These metrics, MSE and AIC, collectively offer valuable insights into the model's predictive performance, considering both accuracy and complexity in the predictions.

3.6 Prediction

In this context, "prediction" denotes the 2D embedding generated for the NLDR technique. The methodology encompasses identifying the nearest averaged high-D points in the high-D space for the test data, by computing high-D Euclidean distances. As the averaged high-D point corresponds to the lifting of the 2D model, determining its nearest counterpart allows us to map its hexagonal bin centroid coordinates. Consequently, these centroid coordinates serve as the assigned low-dimensional predicted values for the test data in the 2D space.

Some NLDR techniques, such as tSNE, often lack a direct prediction, making our approach valuable for generating predicted values in the absence of such functionalities.

3.7 Simulated data example

In this section, we showcase the effectiveness of our methodology using simulated data. The dataset comprises five spherical Gaussian clusters in 4-d, with each cluster containing

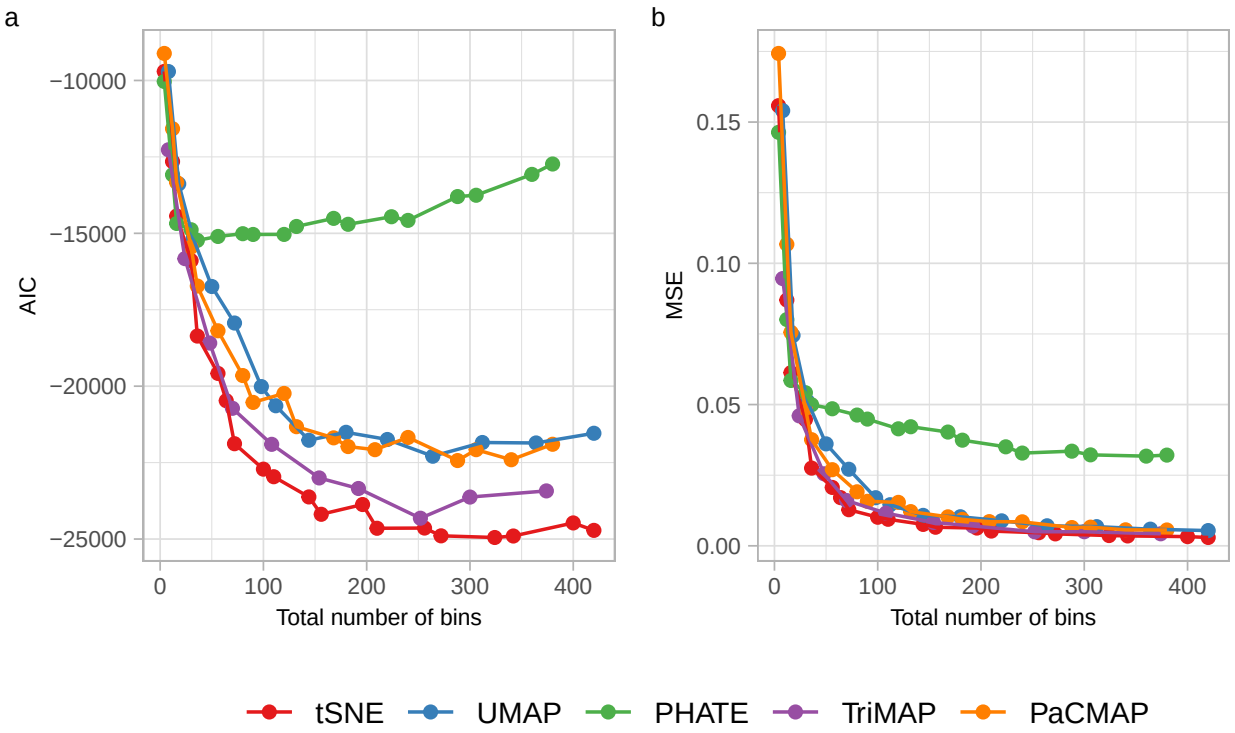

Figure 6: Goodness of fit statistics from different NLDR techniques applied to training S-curve dataset. What is the best NLDR technique to represent the original data in 2D?

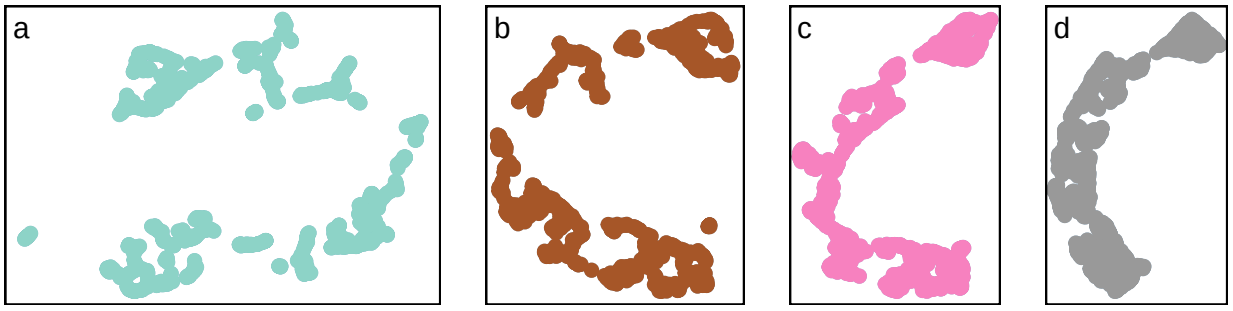

Figure 7: 2D layouts from UMAP applied for the S-curve data: (a) UMAP (n_neighbors = 7), (b) UMAP (n_neighbors = 15), (c) UMAP (n_neighbors = 32), (d) UMAP (n_neighbors = 50). Is there a best parameter choice in representing UMAP or are they all providing equivalent information?

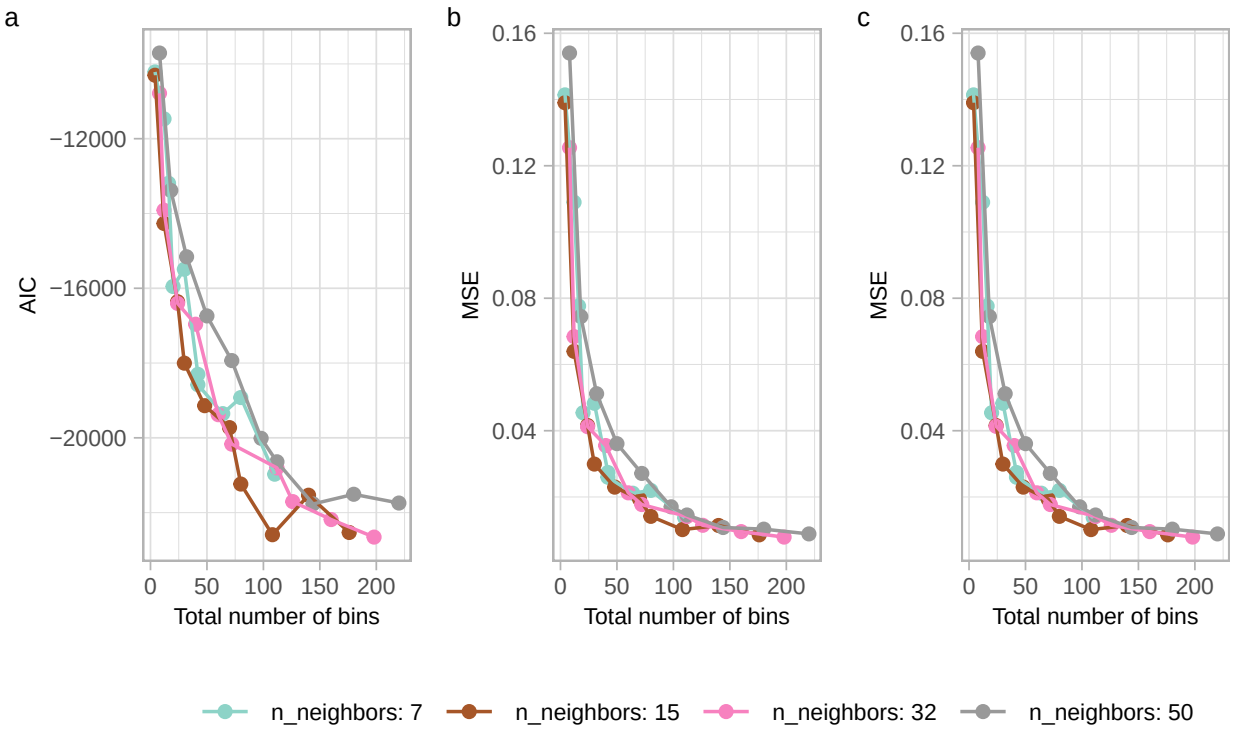

Figure 8: Goodness of fit statistics from different n_neighbors parameter of UMAP applied to training S-curve dataset. What is the best parameter choice in UMAP to represent the original data in 2D?

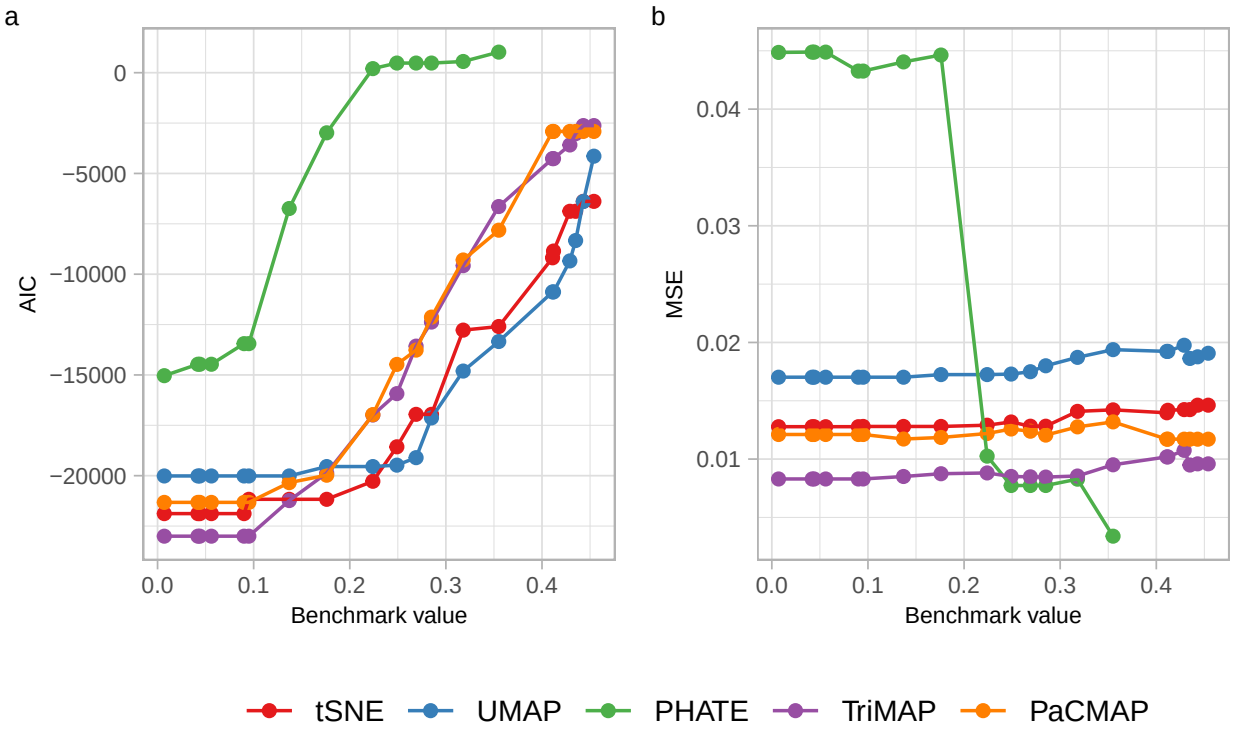

Figure 9: Goodness of fit statistics from different NLDR techniques applied to training S-curve dataset with different benchmark values to remove the low-density heaxgons. What is the effective benchark valuw to remove the low-density heaxgons?

an equal number of points and consistent within variation.

We *strongly* recommend viewing the linked videos for each study while reading. Links to the videos are available in the figures for each example. The videos show the visual appearance of the **langevitour** interface with low-dimensional view and how we can interact with the tour via the controls.

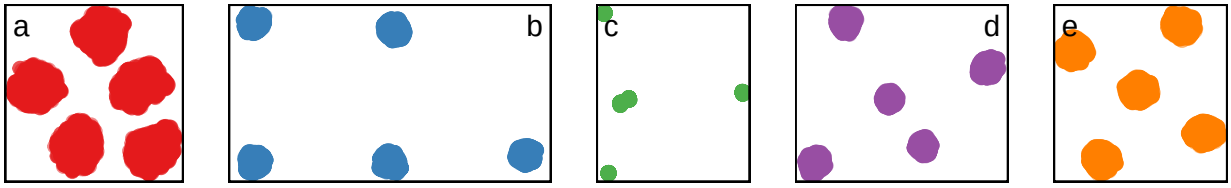

Figure 10: 2D layouts from different NLDR techniques applied the same data: (a) tSNE (perplexity = 61), (b) UMAP (n_neighbors = 15), (c) PHATE (knn = 5), (d) TriMAP (n_inliers = 5, n_outliers = 4, n_random = 3), and (e) PaCMAP (n_neighbors = 10, init = random, MN_ratio = 0.9, FP_ratio = 2). Is there a best representation of the original data or are they all providing equivalent information?

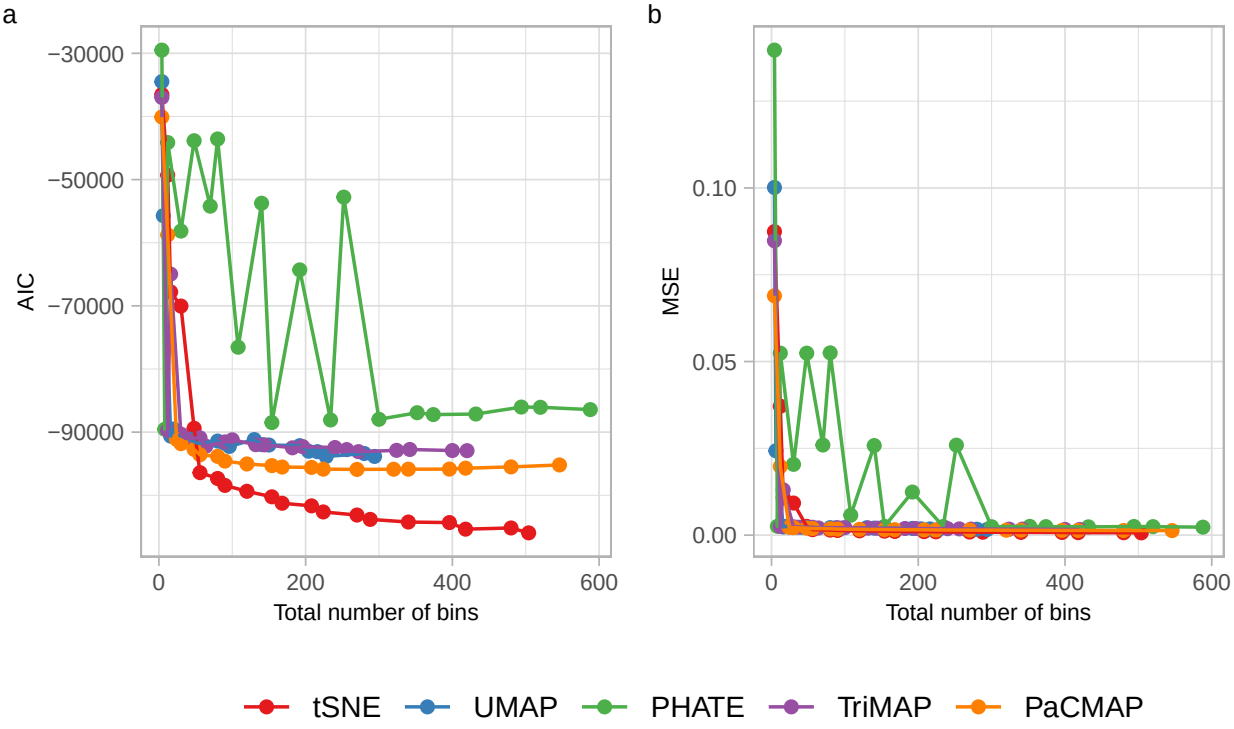

Figure 11: Goodness of fit statistics from different NLDR techniques applied to training five spherical Gaussian cluster dataset. What is the best NLDR technique to represent the original data in 2D?

4 Applications

4.1 Single-cell RNA-seq data of human

In the field of single-cell studies, a common analysis task involves clustering to identify groups of cells with similar expression profiles. Analysts often turn to NLDR techniques to verify and identify these clusters and explore developmental trajectories (e.g., example 1). In clustering workflows, the main objective is to verify the existence of clusters and subsequently identify them as specific cell types by examining the expression of "known" marker genes. In this context, a "faithful" embedding should ideally preserve the topology of the data, ensuring that cells corresponding to the same cell type are situated close to the high-dimensional space.

To begin our analysis, we installed the Peripheral Blood Mononuclear Cells (pbmc) data set obtained from 10x Genomics using the `SeuratData` R package (Satija et al. 2019), which facilitates the distribution of data sets in the form of Seurat objects (Hao et al. 2021). This data set contains 13,714 features across 2,700 samples within a single assay. The active assay is RNA, with 13,714 features representing different gene expressions. After loading the data set, we obtained the principal components (PCs) and assessed the variance explained by each PC. Based on this evaluation, we selected seven PCs, representing approximately 50% of the variance in gene expression, for further analysis.

Next, we employed the UMAP technique with default parameter settings. As illustrated in **?@fig-pbmc**, the cell types B and Platelet are well-separated in the UMAP layout. Moreover, CD14+ Mono, FCGR3A+ Mono, and DC form a distinct cluster, while Naive CD4 T, NK, Memory CD4 T, and CD8 T are grouped together in another cluster. The values utilized to construct the smooth low-dimensional manifold are presented in **?@tbl-table02**. The linked video, demonstrating the tour with the model, showcases the generation of a smooth surface for this application, enabling a comprehensive exploration of the data's structure and relationships (see **?@fig-pbmc_sc**).

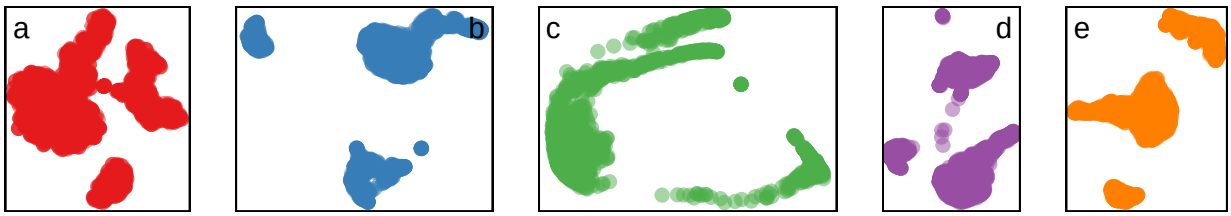

Figure 12: 2D layouts from different NLDR techniques applied for the training PBMC dataset: (a) tSNE (perplexity = 30), (b) UMAP (n_neighbors = 15), (c) PHATE (knn = 5), (d) TriMAP (n_inliers = 5, n_outliers = 4, n_random = 3), and (e) PaCMAP (n_neighbors = 10, init = random, MN_ratio = 0.9, FP_ratio = 2). Is there a best representation of the original data or are they all providing equivalent information?

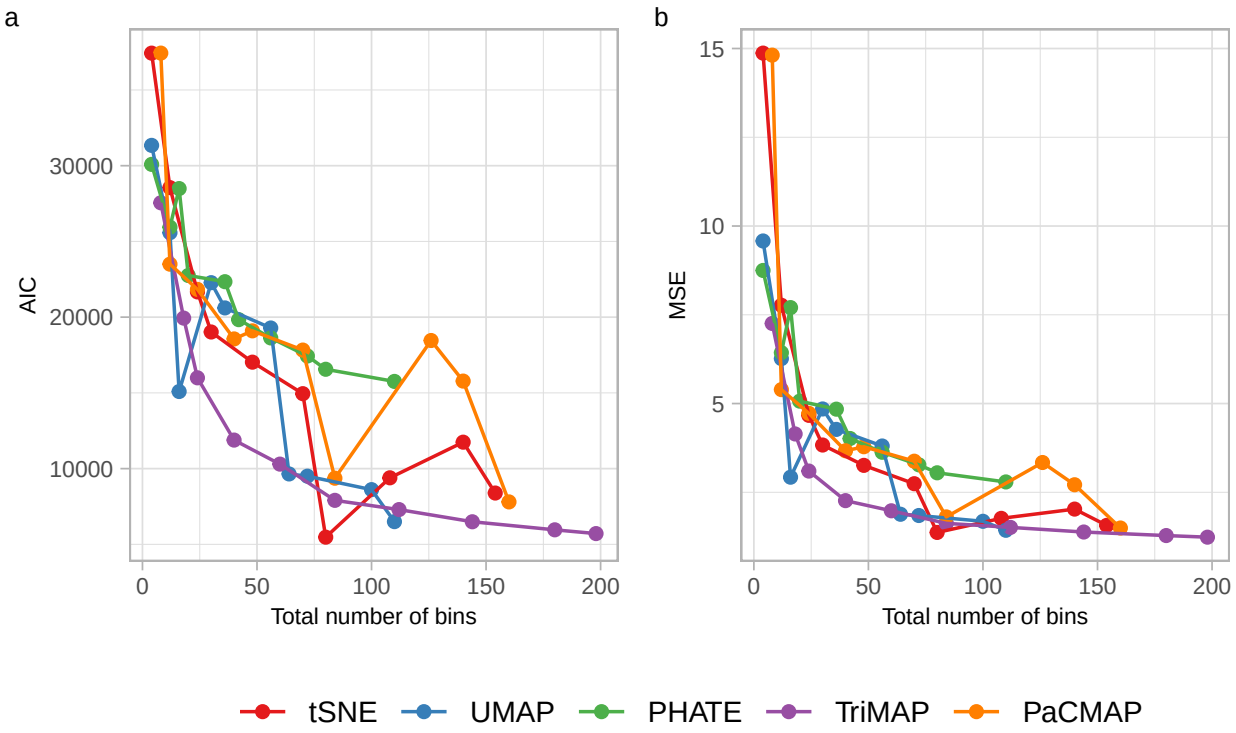

Figure 13: Goodness of fit statistics from different NLDR techniques applied to training PBMC dataset. What is the best NLDR technique to represent the original data in 2D?

4.2 Single-Cell Tagged Reverse Transcription sequencing data of mouse

The Zeisel mouse brain dataset, obtained through Spatial Transcriptomics (STRT-Seq). Within this dataset, information is collected from a substantial 2,816 individual mouse brain cells. Each of these cells acts as a molecular snapshot, capturing the distinctive genetic activity within various cell types. This diversity spans neurons, glial cells, and other essential components of the brain, offering a comprehensive view of the cellular tapestry.

What makes this dataset particularly valuable is its ability to shed light on the spatial distribution of cells. Researchers can explore how gene expression patterns vary across different regions of the mouse brain, unlocking insights into the functional specialization of these regions and the intricate networks that underpin neural processes.

5 Discussion

Our research introduces a comprehensive framework that leverages tours for interactive exploration of high-dimensional data coupled with a low-dimensional manifold, facilitated by the `quollr` R package. Regardless of the Non-Linear Dimension Reduction (NLDR) technique in use, our approach demonstrates effectiveness through simulation examples, particularly in the iterative removal of long edges for a smoother representation and capturing cluster variance.

In the example with doublets, our method successfully captures the tweak within each

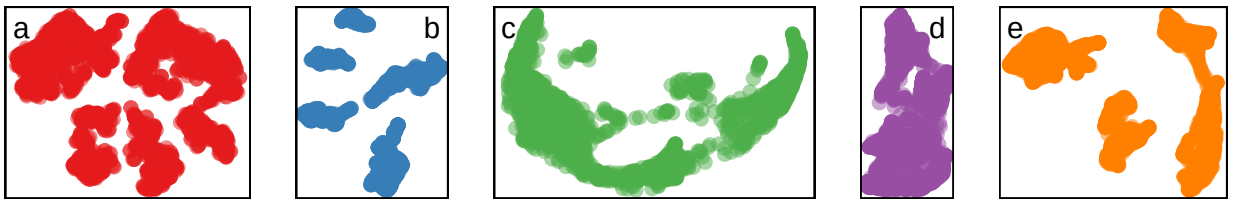

Figure 14: 2D layouts from different NLDR techniques applied for the training Zeisel mouse brain dataset: (a) tSNE (perplexity = 30), (b) UMAP (n_neighbors = 15), (c) PHATE (knn = 5), (d) TriMAP (n_inliers = 5, n_outliers = 4, n_random = 3), and (e) PaCMAP (n_neighbors = 10, init = random, MN_ratio = 0.9, FP_ratio = 2). Is there a best representation of the original data or are they all providing equivalent information?

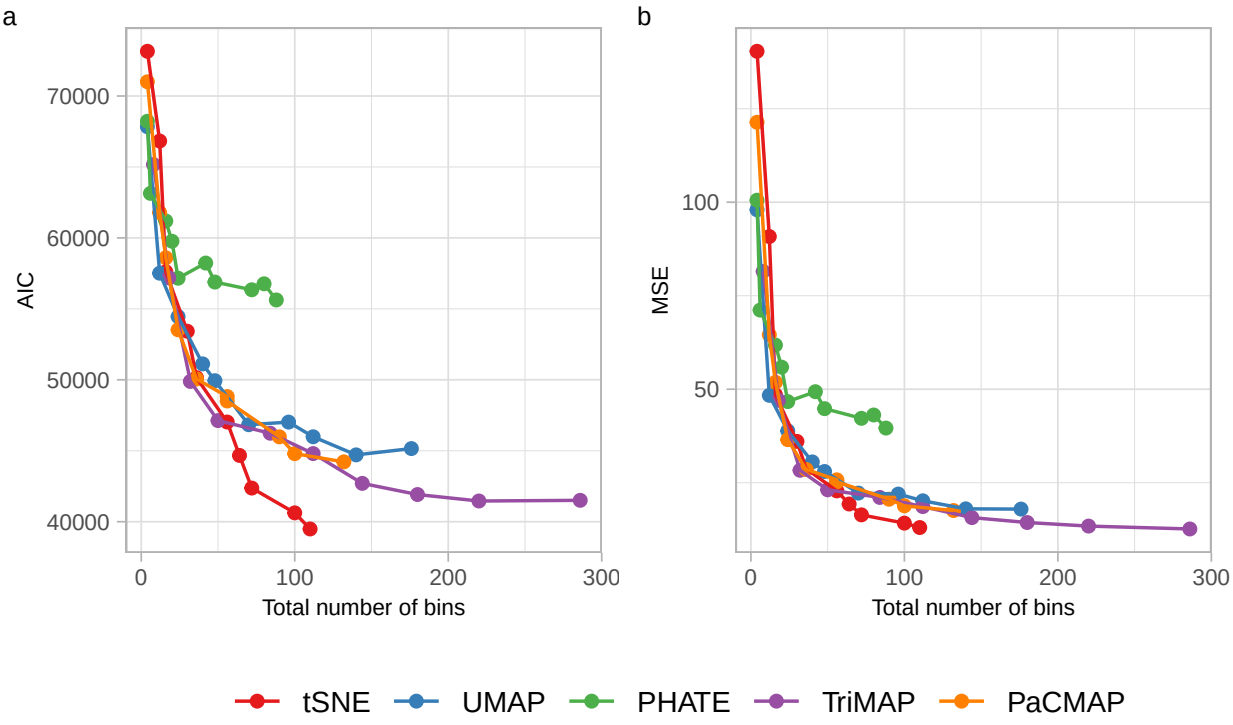

Figure 15: Goodness of fit statistics from different NLDR techniques applied to training Zeisel mouse brain dataset. What is the best NLDR technique to represent the original data in 2D?

cluster, indicating the variance present within them. However, the model may not appear smooth in high-dimensional space due to considerable noise when the data has a piecewise linear geometry, such as the tree simulation.

The practical application of our framework, as showcased with the UMAP view, enables visual inspection of well-separated clusters. Furthermore, the combined tour and model provide a robust assessment of whether UMAP preserves the data structure and accurately transforms the data.

The advantages of our approach include its versatility across various NLDR techniques and the ability to generate interactive visualizations for detailed exploration. The tour provides an intuitive way to navigate and comprehend high-dimensional data while assessing the accuracy of dimensionality reduction.

However, one limitation is that the approach may be less effective in cases with significant noise, as seen in the tree simulation example. Additionally, while our method aids in visual verification, quantifying the accuracy of embeddings might require further evaluation metrics.

In conclusion, our framework presents a powerful tool for researchers and analysts in single-cell studies to assess their embeddings by visually inspecting them alongside the original data. By leveraging the advantages of tours and low-dimensional manifolds, our approach offers valuable insights into the data transformation process, empowering users to make informed decisions in analyzing high-dimensional data. Future work could enhance the method's robustness in the presence of noise and explore additional evaluation metrics for quantifying embedding accuracy.

References

Amid, E. & Warmuth, M. K. (2022), 'Trimap: Large-scale dimensionality reduction using triplets'.

Asimov, D. (1985), 'The grand tour: A tool for viewing multidimensional data', *SIAM Journal on Scientific and Statistical Computing* **6**(1), 128–143.
URL: *https://doi.org/10.1137/0906011*

Ayesha, S., Hanif, M. K. & Talib, R. (2020), 'Overview and comparative study of dimensionality reduction techniques for high dimensional data', *Information Fusion* **59**, 44–58.

Buja, A., Cook, D. & Swayne, D. F. (1996), 'Interactive high-dimensional data visualization', *Journal of Computational and Graphical Statistics* **5**(1), 78–99.
URL: *http://www.jstor.org/stable/1390754*

Carr, D. B. (1992), Looking at large data sets using binned data plots.

Carr, D. B., Littlefield, R. J., Nicholson, W. L. & Littlefield, J. S. (1987), 'Scatterplot matrix techniques for large n', *Journal of the American Statistical Association* **82**(398), 424–436.
URL: *http://www.jstor.org/stable/2289444*

Carr, D., Olsen, A. & White, D. (2013), 'Hexagon mosaic maps for display of univariate and bivariate geographical data', *Cartography and Geographic Information Systems* **19**, 228–236.

Carr, D., ported by Nicholas Lewin-Koh, Maechler, M. & contains copies of lattice functions written by Deepayan Sarkar (2023), *hexbin: Hexagonal Binning Routines*. R package version 1.28.3.
URL: *https://CRAN.R-project.org/package=hexbin*

Coifman, R., Lafon, S., Lee, A., Maggioni, M., Nadler, B., Warner, F. & Zucker, S. (2005), 'Geometric diffusions as a tool for harmonic analysis and structure definition of data: Diffusion maps', *Proceedings of the National Academy of Sciences of the United States of America* **102**, 7426–31.

Cook, D., Buja, A., Cabrera, J. & Hurley, C. (1995), 'Grand tour and projection pursuit', *Journal of Computational and Graphical Statistics* **4**(3), 155–172.
URL: *https://www.tandfonline.com/doi/abs/10.1080/10618600.1995.10474674*

F.R.S., K. P. (1901), 'Liii. on lines and planes of closest fit to systems of points in space', *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* **2**(11), 559–572.
URL: *https://doi.org/10.1080/14786440109462720*

Guo, B., Huuki-Myers, L. A., Grant-Peters, M., Collado-Torres, L. & Hicks, S. C. (2023), 'escheR: Unified multi-dimensional visualizations with Gestalt principles', *bioRxiv* . Publisher: Cold Spring Harbor Laboratory _eprint: https://www.biorxiv.org/content/early/2023/06/08/2023.03.18.533302.full.pdf.
URL: *https://www.biorxiv.org/content/early/2023/06/08/2023.03.18.533302*

Hao, Y., Hao, S., Andersen-Nissen, E., III, W. M. M., Zheng, S., Butler, A., Lee, M. J., Wilk, A. J., Darby, C., Zagar, M., Hoffman, P., Stoeckius, M., Papalexi, E., Mimitou, E. P., Jain, J., Srivastava, A., Stuart, T., Fleming, L. B., Yeung, B., Rogers, A. J., McElrath, J. M., Blish, C. A., Gottardo, R., Smibert, P. & Satija, R. (2021), 'Integrated analysis of multimodal single-cell data', *Cell* .
URL: *https://doi.org/10.1016/j.cell.2021.04.048*

Hart, C. & Wang, E. (2022), *detourr: Portable and Performant Tour Animations*. R package version 0.1.0.
URL: *https://casperhart.github.io/detourr/*

Howley, T., Madden, M., O'Connell, M.-L. & Ryder, A. (2005), The effect of principal component analysis on machine learning accuracy with high dimensional spectral data, pp. 209–222.

Indhumathi, R. & Sathiyabama, S. (2010), 'Reducing and clustering high dimensional data through principal component analysis', *International Journal of Computer Applications* **11**(8), 1–4.

Jia, W., Sun, M., Lian, J. & Hou, S. (2022), 'Feature dimensionality reduction: a review', *Complex & Intelligent Systems* **8**(3), 2663–2693.
URL: *https://doi.org/10.1007/s40747-021-00637-x*

Johnstone, I. M. & Titterington, D. M. (2009), 'Statistical challenges of high-dimensional data', *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* **367**(1906), 4237–4253.
URL: *https://royalsocietypublishing.org/doi/abs/10.1098/rsta.2009.0159*

Jolliffe, I. T. & Cadima, J. (2016), 'Principal component analysis: a review and recent developments', *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* **374**(2065), 20150202.
URL: *https://royalsocietypublishing.org/doi/abs/10.1098/rsta.2015.0202*

Kruskal, J. B. (1964), 'Nonmetric multidimensional scaling: a numerical method', *Psychometrika* **29**(2), 115–129.

Lee, D. T. & Schachter, B. J. (1980), 'Two algorithms for constructing a Delaunay triangulation', *International Journal of Computer & Information Sciences* **9**(3), 219–242.
URL: *https://doi.org/10.1007/BF00977785*

Lee, S., Cook, D., da Silva, N., Laa, U., Wang, E., Spyrison, N. & Zhang, H. S. (2021), 'A review of the state-of-the-art on tours for dynamic visualization of high-dimensional data'.

Lee, S., Laa, U. & Cook, D. (2020), 'Casting multiple shadows: High-dimensional interactive data visualisation with tours and embeddings'.
URL: *https://arxiv.org/abs/2012.06077*

Lloyd, E. L. (1977), On triangulations of a set of points in the plane, *in* '18th Annual Symposium on Foundations of Computer Science (sfcs 1977)', pp. 228–240.

McInnes, L. & Healy, J. (2018), 'Umap: Uniform manifold approximation and projection for dimension reduction', *ArXiv* **abs/1802.03426**.

Moon, K. R., van Dijk, D., Wang, Z., Gigante, S. A., Burkhardt, D. B., Chen, W. S., Yim, K., van den Elzen, A., Hirn, M. J., Coifman, R. R., Ivanova, N. B., Wolf, G. & Krishnaswamy, S. (2019), 'Visualizing structure and transitions in high-dimensional biological data', *Nature Biotechnology* **37**, 1482 – 1492.

Paul Harrison (2022), 'langevitour: smooth interactive touring of high dimensions, demonstrated with scRNA-Seq data', *bioRxiv* p. 2022.08.24.505207.
URL: *http://biorxiv.org/content/early/2022/08/26/2022.08.24.505207.abstract*

Renka, R. J. (1996), 'Algorithm 751: Tripack: A constrained two-dimensional delaunay triangulation package', *ACM Trans. Math. Softw.* **22**(1), 1–8.
URL: *https://doi.org/10.1145/225545.225546*

Rumelhart, D. E., Hinton, G. E. & Williams, R. J. (1986), 'Learning representations by back-propagating errors', *Nature* **323**, 533–536.

Satija, R., Hoffman, P. & Butler, A. (2019), *SeuratData: Install and Manage Seurat Datasets*. http://www.satijalab.org/seurat, https://github.com/satijalab/seurat-data.

Silva, V. & Tenenbaum, J. (2002), 'Global versus local methods in nonlinear dimensionality reduction', *Advances in neural information processing systems* **15**.

Swayne, D. F., Cook, D. & Buja, A. (1998), 'Xgobi: Interactive dynamic data visualization in the x window system', *Journal of Computational and Graphical Statistics* **7**(1), 113–130.
URL: *http://www.jstor.org/stable/1390772*

Torgerson, W. S. (1967), *Theory and methods of scaling*, Wiley New York.

Vaidyanathan, R., Xie, Y., Allaire, J., Cheng, J., Sievert, C. & Russell, K. (2023), *htmlwidgets: HTML Widgets for R*. R package version 1.6.2.
URL: *https://CRAN.R-project.org/package=htmlwidgets*

van der Maaten, L. & Hinton, G. E. (2008), 'Visualizing data using t-sne', *Journal of Machine Learning Research* **9**, 2579–2605.

Venna, J., Peltonen, J., Nybo, K., Aidos, H. & Kaski, S. (2010), 'Information Retrieval Perspective to Nonlinear Dimensionality Reduction for Data Visualization', *Journal of Machine Learning Research* **11**(13), 451–490.
URL: *http://jmlr.org/papers/v11/venna10a.html*

Wickham, H., Cook, D. & Hofmann, H. (2015), 'Visualizing statistical models: Removing the blindfold', *Statistical Analysis and Data Mining: The ASA Data Science Journal* **8**(4), 203–225.
URL: *https://onlinelibrary.wiley.com/doi/abs/10.1002/sam.11271*

Wickham, H., Cook, D., Hofmann, H. & Buja, A. (2011), 'tourr: An r package for exploring multivariate data with projections', *Journal of Statistical Software* **40**(2), 1—18.
URL: *http://www.jstatsoft.org/v40/i02/*