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 4, 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 **?@sec-prediction**, discusses the effectiveness of the learned low-dimensional manifold in accurately representing the complex high-dimensional data. Following that, Section 3.6 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 $X = \begin{bmatrix} \mathbf{x}_1 & \mathbf{x}_2 & \cdots & \mathbf{x}_n \end{bmatrix}$, with n observations in p dimensions. The objective is to discover a low-dimensional projection $Y = \begin{bmatrix} \mathbf{y}_1 & \mathbf{y}_2 & \cdots & \mathbf{y}_n \end{bmatrix}$, 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 Introduction to UMAP

The UMAP algorithm constructs a fuzzy topological representation of the high-dimensional data and then optimizes the low-dimensional embedding to be as similar to this representation as possible (McInnes & Healy 2018).

The UMAP algorithm constructs a weighted k-nearest neighbors graph based on the local similarity of data points in the original high-dimensional space. The weighted graph is defined by a conditional probability distribution P_{ij}, which represents the probability of observing data point x_i given data point x_j as a neighbor:

$$P_{ij} = \frac{\exp(-\|x_i - x_j\|^2)}{\sum_{k\neq i} \exp(-\|x_i - x_k\|^2)}$$

Similarly, a fuzzy topological representation of the data is constructed in the low-dimensional space with conditional probability distribution Q_{ij}:

$$Q_{ij} = \frac{\exp(-\|y_i - y_j\|^2)}{\sum_{k\neq i} \exp(-\|y_i - y_k\|^2)}$$

where y_i and y_j are the corresponding low-dimensional representations of data points x_i and x_j.

UMAP minimizes the cross-entropy between the high-dimensional fuzzy topological representation P_{ij} and the low-dimensional representation Q_{ij} to find an optimal low-dimensional embedding. The loss function for UMAP is defined as:

$$L = \sum_i \sum_j P_{ij} \log \frac{P_{ij}}{Q_{ij}}$$

The algorithm uses stochastic gradient descent to optimize the positions of the data points in the low-dimensional space to minimize this loss function.

In R, the `umap` package is often used to implement UMAP and generate results.

The "n_neighbors" parameter in UMAP determines the number of nearest neighbors to consider for each data point during the construction of the fuzzy topological representation.

Higher values of “n_neighbors” capture more global structure, while lower values emphasize local structure. Choosing an appropriate value for “n_neighbors” is crucial, as it affects the balance between local and global structure in the resulting low-dimensional representation.

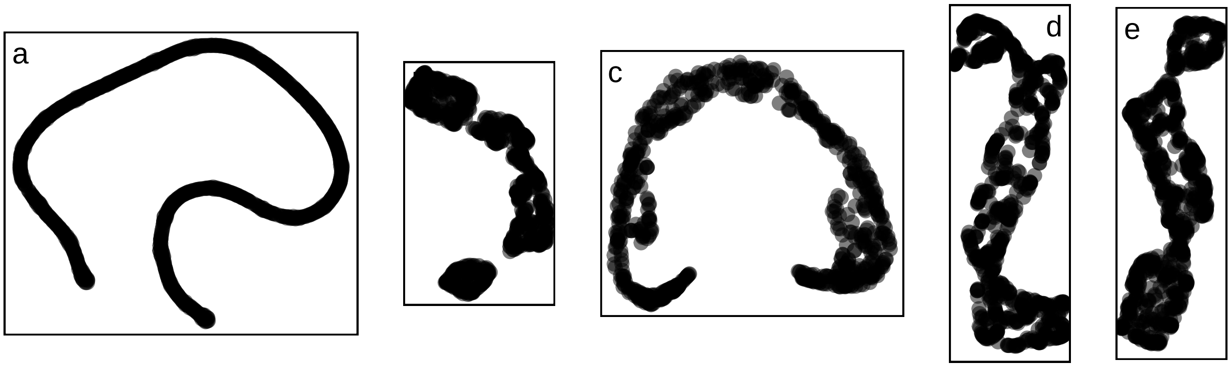


Figure 1: 2D layouts from different NLDR techniques applied the same data: (a) tSNE (perplexity = 32), (b) UMAP (n_neighbors = 50), (c) PHATE (knn = 5), (d) TriMAP (n_inliers = 5, n_outliers = 4, n_random = 3), and (e) PaCMAP (n_neighbors = 10, init = random, MN_ratio = 0.9, FP_ratio = 2). Is there a 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.

2.3 Non-parametric models

2.3.1 Hexagonal binning

Hexagonal binning is a data visualization technique that aggregates high-dimensional data into a two-dimensional representation using hexagonal regions called bins (Carr 1992). The hexagonal shape is preferred due to its circular-like appearance, providing smoother transitions between neighboring bins and efficient data aggregation. This technique is especially useful for handling large data sets as it reduces visual clutter while capturing the underlying data distribution effectively. By considering the data distribution, important features are captured, and gaps between bins are minimized, leading to a more accurate representation of the data. In our algorithm, hexagonal binning is utilized to create a low-dimensional manifold and facilitate data visualization in the original high-dimensional space.

2.3.2 Triangulation

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) finding applications in mesh generation and representing complex geometric structures efficiently and discretely. One common algorithm used for triangulation is Delaunay triangulation (Lee & Schachter 1980), 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.

In our algorithm, we are interested in triangulation as it plays a key role in providing a clear and concise representation of the data. By converting the data points into interconnected triangles, we can visualize and analyze the data in a more structured and organized manner. Triangulation helps us identify geometric relationships and patterns in the data, enabling us to make informed decisions and gain valuable insights from the information presented in the triangular form.

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.

3.1 Preprocessing steps

To tackle the complexities and noise in high-dimensional data, we apply PCA as a preprocessing step (Jolliffe & Cadima 2016, Howley et al. (2005), Indhumathi & Sathiyabama (2010)). This step helps in noise reduction by identifying principal components that represent directions of maximum variance, capturing essential patterns in the data.

3.2 Constructing the 2D model

Step 1: Computing the hexagonal grid configuration

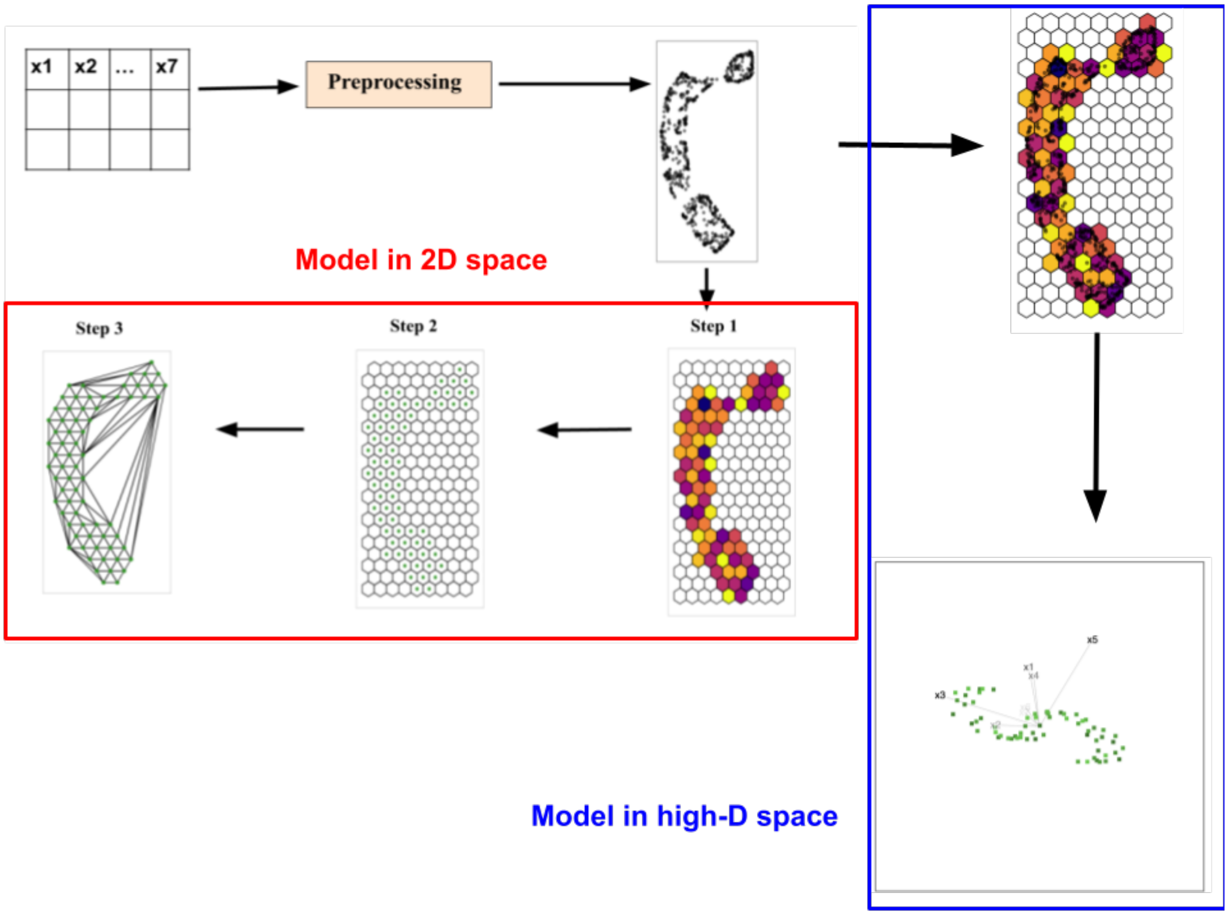


Figure 2: A flow diagram detailing the steps taken to create the low-dimensional manifold in the high dimensional space. There are two basic phases, one to generate the model in the 2D space, and other to generate the model in the high-D space.

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

The first step involves to determine the optimal number of bins (b) for creating regular hexagons in the hexagonal grid. To achieve this, we rely on the relationship between the diameter (h) and the area (A) of regular hexagons, as given by Equation 1.

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

To capture the data's structure effectively, we consider the optimal $A = 1$ (see Figure 3). Using Equation 2, we then calculate h of a regular hexagon.

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

Next, we refer the `hexbin` function in R, which provides a relationship between the diameter of the hexagon (h) and the number of bins along the x-axis (b_1) (Equation 3).

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

By utilizing the calculated h and the range of the non-linear projection component 1 (r_1), we determine the b_1 (Equation 4). The result is rounded up to the nearest whole number to ensure an appropriate number of bins that capture the variability of the data along the x-axis.

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

(b) Determine the effective shape parameter (s)

In this step, we determine the optimal shape parameter (s) for the hexagonal bins, which significantly influences their shape and arrangement within the grid. The shape parameter (s) in the hexagonal binning algorithm is defined as the ratio of the height (y) to the width (x) of the plotting region, referring to the `hexbin` function in R (see Equation 5). It determines the shape of the plotting regions and plays a vital role in generating an appropriate hexagonal grid for the data visualization.

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

To calculate the effective shape parameter (s) for our algorithm, we consider the ranges of the non-linear projection components. Denoting the range of the non-linear projection component 1 as r_1 and the range of the non-linear projection component 2 as r_2, we find the shape parameter using Equation 6.

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

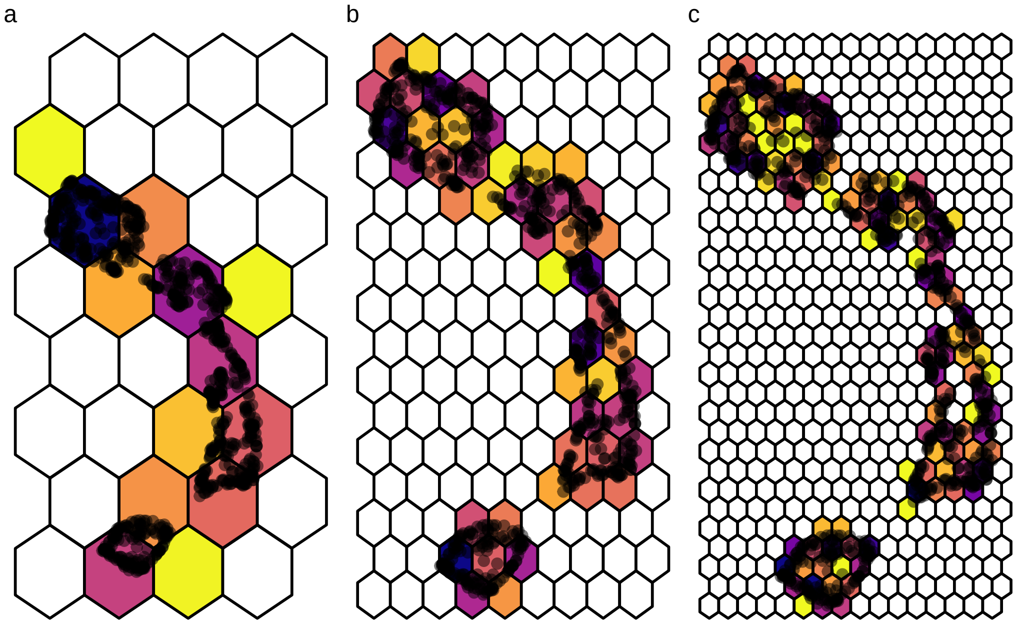


Figure 3: Hexbin plots from different number of bins for the same **s_curve_noise_umap** data: (a) b = (4, 8), s = 2.031, (b) b = (9, 20), s = 2.031, and (c) b = (16, 36), s = 2.031. The hexbins are colored based on the density of points, with yellow indicating higher point density and darker colors 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

After computing hexagonal grid configurations, the 2D emdeddings are binned into hexagon cells. As a result of hexagonal binning , the bin centroids ($C_k^{(2)} \equiv (C_{kx}, C_{ky})$) (see Figure 2 (Step 2)) are obtained.

Step 3: Triangulate bin centroids

In this step, the algorithm proceeds to triangulate the centroids (see Figure 2 (Step 3)) of the hex bins using the `tripack` package in R (Renka 1996). 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

Creating the model in the high-D space involves a two-step process. Initially, we identify observations within each hexagonal bin. After this, we calculate the average of the high-dimensional data within each hex bin. The resulting values represent the averaged high-dimensional data points ($C_k^{(p)} \equiv (C_{kx_1}, ..., C_{kx_p})$). Essentially, these averaged high-dimensional data points serve as the representation of the hex bin centroids in the high-D space.

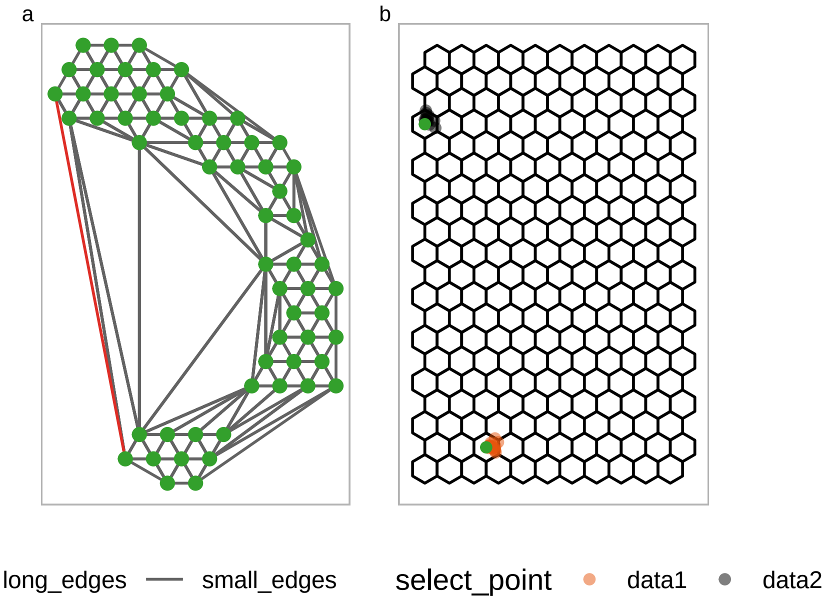


Figure 4: Visualization.

3.4 Tunning the model

3.5 Summaries of the model

3.5.1 Predicted values and residuals

In our prediction process, we employ the K-nearest neighbors (KNN) algorithm to predict the hex bin IDs. Specifically, for each new high-dimensional data point, the algorithm assigns the first neighboring hex bin ID as the predicted hex bin ID.

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:

$$\text{MSE} = \sum_{j=1}^{n} \frac{\sum_{i=1}^{p}(x_{ij} - C_{x_ij})^2}{\text{total number of bins}} \tag{7}$$

Here, b signifies the total number of bins, p denotes the number of dimensions in the high-dimensional (high-D) 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:

$$\text{AIC} = 2bp + np * log(\text{MSE}) \tag{8}$$

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

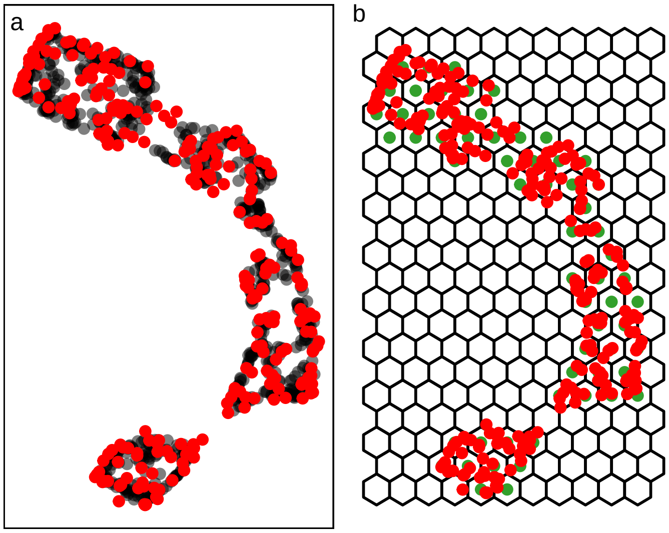


Figure 5: The .

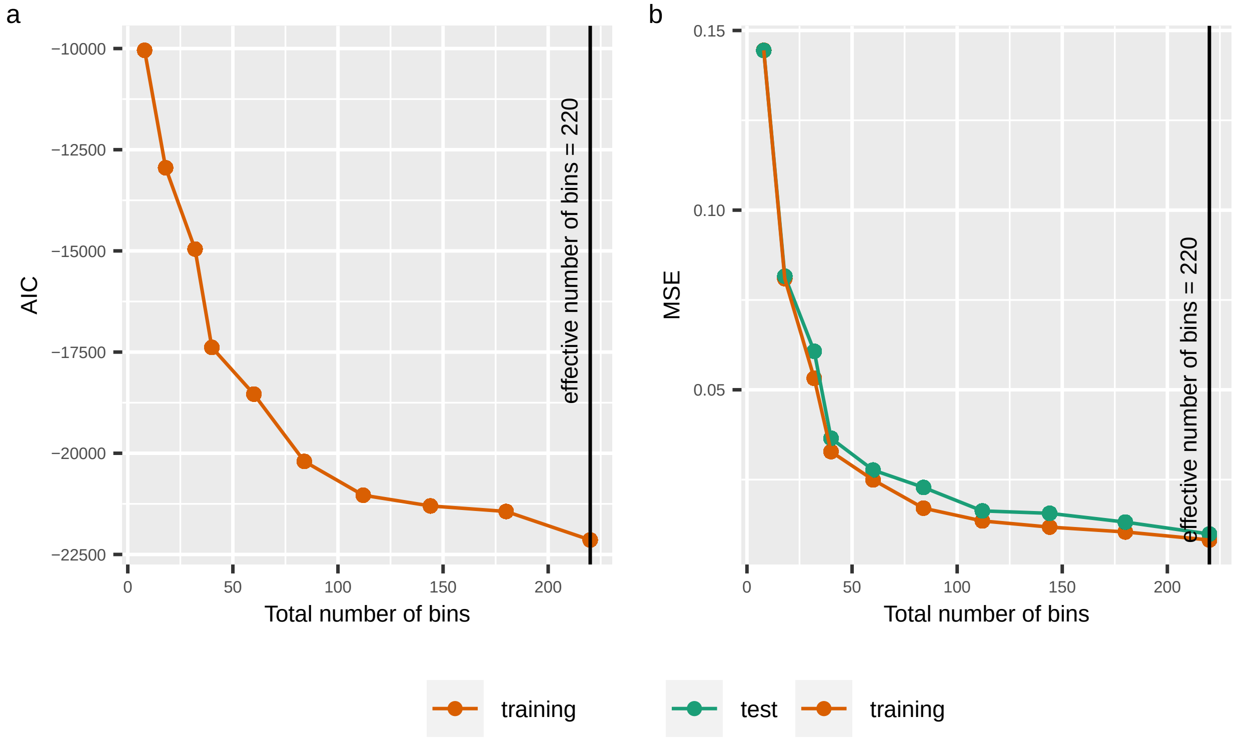


Figure 6: The .

3.5.2 Goodness of fit statistics

In this analysis, we're looking at how well our model is doing in two aspects. First, we're checking the distances in our lower-dimensional space (see Figure 7 (a)). We want to see if the distances between different types of edges are similar. If they are, it means our model is doing a good job of keeping the relationships consistent. If there are big differences, it could mean our model is struggling with certain types of edges.

Next, we're examining how our model handles the relationship between high-dimensional and low-dimensional distances for each type of edge (see Figure 7 (b)). If our data points nicely line up around the expected line, it tells us that our model is preserving the original structure well. If they're scattered, it could mean our model is having trouble with specific data points. This two-part analysis helps us really understand how our model fits the data.

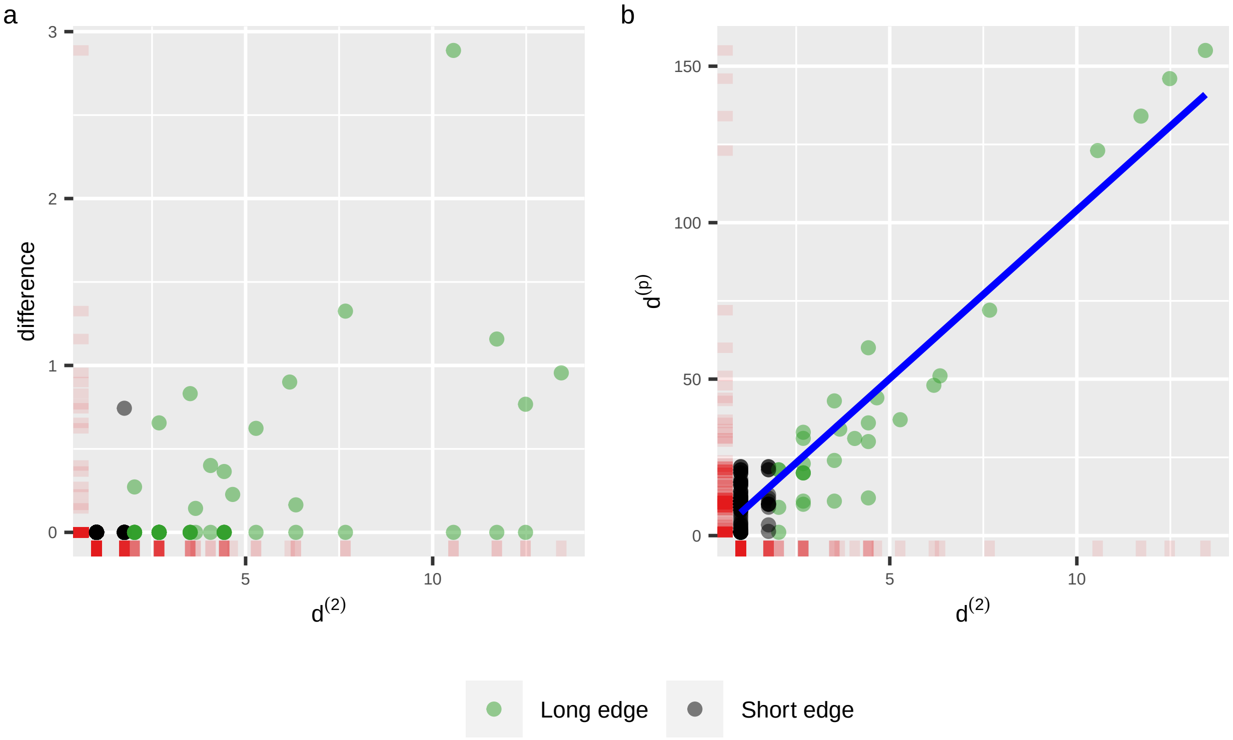


Figure 7: The two diagnostic plot assess the effectiveness of the high-D representation algorithm performed for the **s_curve_noise_umap** data set： (a) visualizing the differences between low-D distances and (b) visualizing the relationship between high-D ($d^{(p)}$) and low-D distances ($d^{(2)}$).

3.6 Simulated data examples

In this section, we demonstrate the effectiveness of our approach through the use of simulation data. These simulations are designed to have known cluster structures and underlying data geometries. We begin with a straightforward example where we generate spherical clusters with a doublet structure. This allows us to assess how well our algorithm can identify and represent these known structures in the low-dimensional space.

We then proceed to a more complex example where the data exhibits a more intricate geometry. This simulation provides a challenging test for our approach to capture and preserve the underlying data structure accurately.

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.

Example 1: Exploring spherical Gaussian clusters with doublet structure

The next data set consists of two distinct types of clusters embedded in a 10-d space. The first set of clusters includes three Gaussian clusters with equal variance. Each of these Gaussian clusters is equidistant from the others, forming a well-separated and symmetrical arrangement. The second set of clusters consists of three clusters with a doublet structure. These clusters are also embedded in the 10-d space, but each is positioned in the middle of its parent Gaussian clusters. Moreover, each doublet cluster contains an average number of points from its corresponding parent Gaussian clusters.

For analysis, we run tSNE on the training data with a perplexity setting 18. Figure 8 (a) displays the tSNE results, indicating that the sub-clusters have been correctly identified. However, the relative locations of certain clusters to each other appear distorted in the low-dimensional view, such as the red and pink clusters being far apart.

Nonetheless, we find no apparent impact on the perception of the data structure when we examine the tour view and the model (see Figure 8) through the linked video (see Figure 9). The model demonstrates accurate identification of clusters and captures the internal variations within each cluster, reflecting the inherent variance present in the data. The configurations used for the analysis are detailed in **?@tbl-table02**. Additionally, the representation of the parent clusters in the low-dimensional space appears stretched, aligning with the variance of these clusters. On the other hand, the doublet clusters are squeezed, reflecting their specific characteristics and relationship to their parent Gaussian clusters.

Example 2: Exploring data with piecewise linear structure

In our next exploration, we delve into simulated noisy tree-structured data designed to mimic branching trajectories of cell differentiation. The simulation is crafted to resemble the hierarchical clustering pattern observed in cell differentiation trajectories, where mature cells are akin to the tips of branches.

To begin our analysis, we apply principal components to the data and focus on the first ten principal components, collectively accounting for approximately 68% of the variance explained in the data set. Then, we run TriMAP with the default parameters on the training data (n_inliers: 5, n_outliers: 4, and n_random: 3). Through this process, we create a low-dimensional manifold using the configurations detailed in **?@tbl-table02**. Finally, we present the model along with the data using the tour view (Figure Figure 11), allowing us to visualize and explore the intricate relationships and structures present in the data's low-dimensional representation.

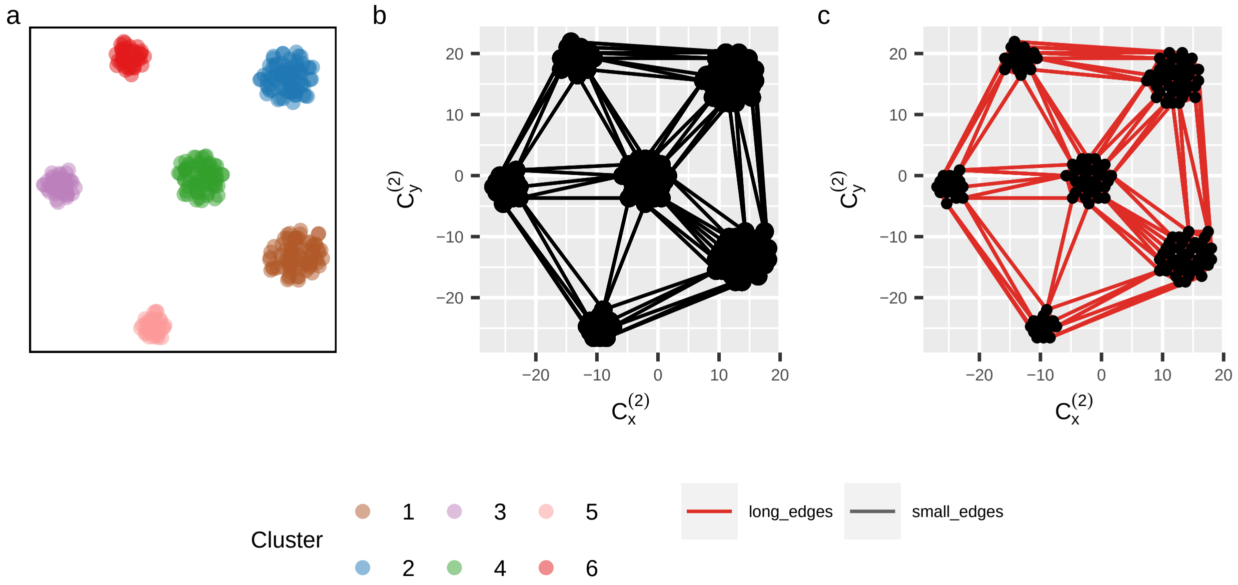
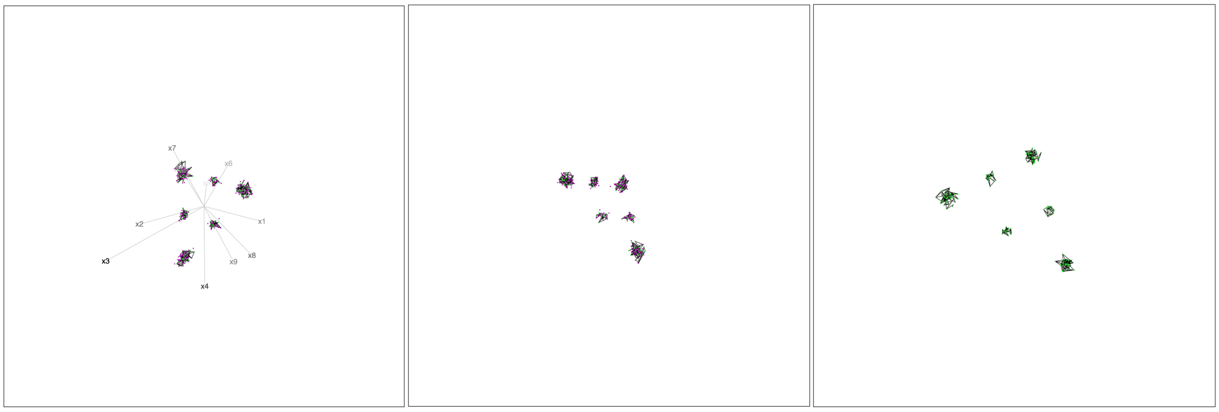


Figure 8: Visualization and refinement of low-dimensional representation for sub-clustered data: (a) 2D layout of tSNE (perplexity = 18), (b) triangular mesh, and (c) triangular mesh colored by edge type. Edges with lengths less than the benchmark value of 2 are classifiedas small, while edges with lengths greater than or equal to this value are labeled as long. The combination of tSNE, hexagonal binning, and triangulation provides the effectiveness of our approach in visualizing and refining the low-dimensional representation of the sub-clustered data.



(a) (b) (c)

Figure 9: Screen shots of the **langevitour** with the low-dimensional view applied to sub-clustered data set, a video of the tour animation is available at (https://drive.google.com/file/d/15nAbmc4UoVlUyFcGtnlnmyxrJEbK39nx/view). Combination of the **langevitour** and low-dimensional representation of the sub-clustered data provides the effectiveness of our approach.

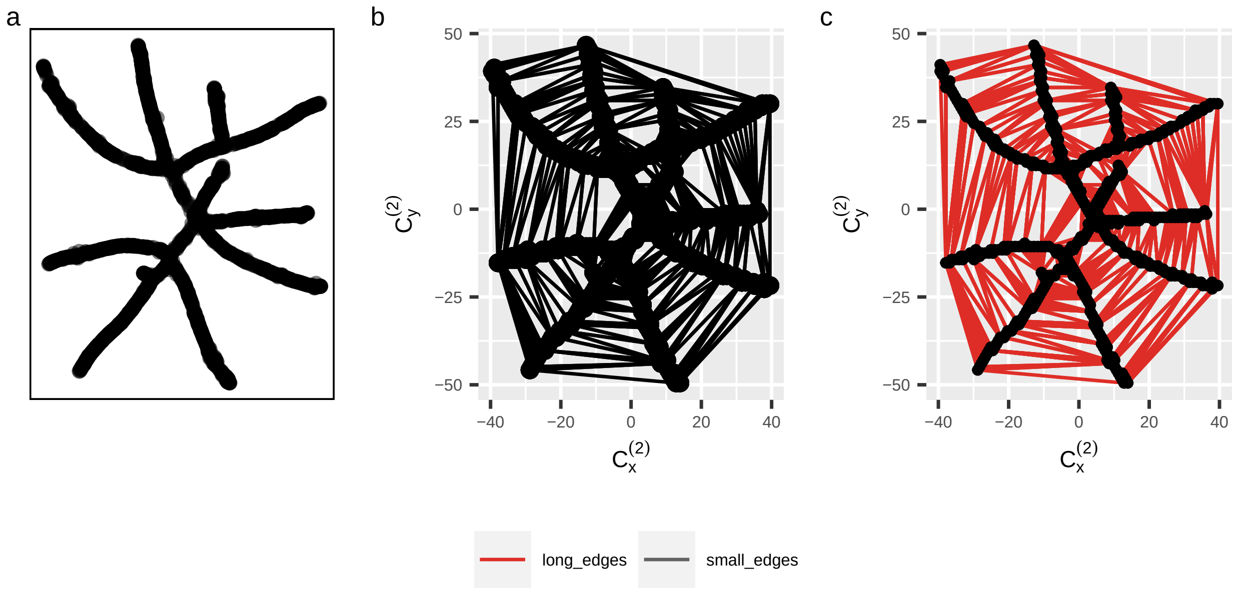


Figure 10: Visualization and refinement of low-dimensional representation for tree structured data: (a) 2D layout of TriMAP (n-inliers = 5, n-outliers = 4, n-random = 3), (b) triangular mesh, and (c) triangular mesh colored by edge type. Edges with lengths less than the benchmark value of 2 are classified as small, while edges with lengths greater than or equal to this value are labeled as long. The combination of TriMAP, hexagonal binning, and triangulation provides the effectiveness of our approach in visualizing and refining the low-dimensional representation of the tree structured data.

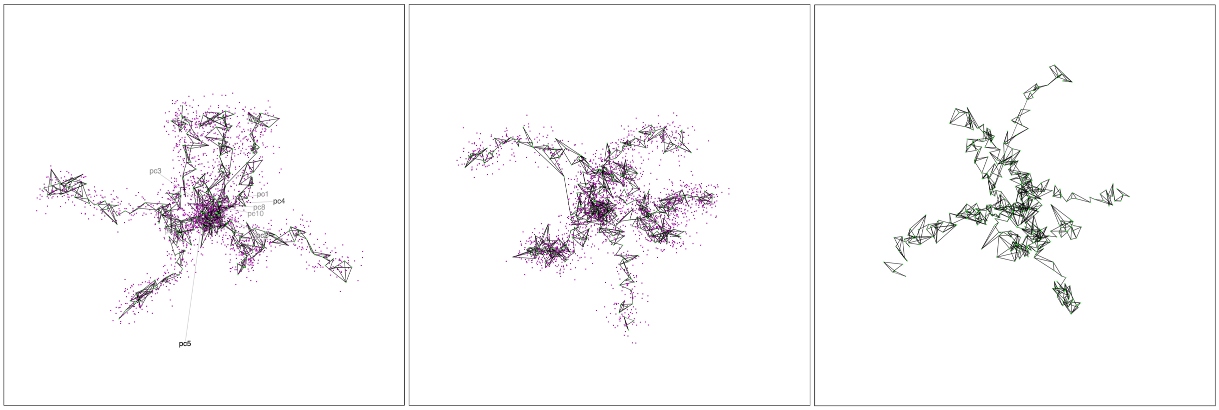


Figure 11: Screen shots of the **langevitour** with the low-dimensional view applied to tree structured data set, a video of the tour animation is available at (https://drive.google.com/file/d/1YQx4UGlzDify1mGqeDtlqTHncgqidDhq/view). Combination of the **langevitour** and low-dimensional representation of the tree structured data provides the effectiveness of our approach.

4 Examples/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 Figure 12, 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 Figure 13).

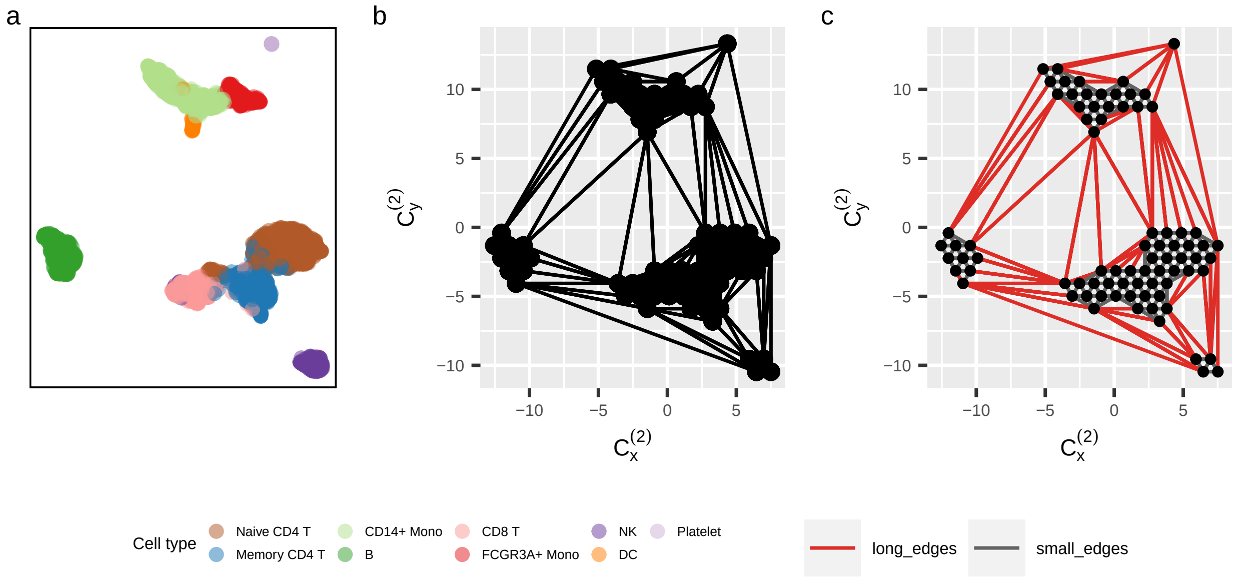
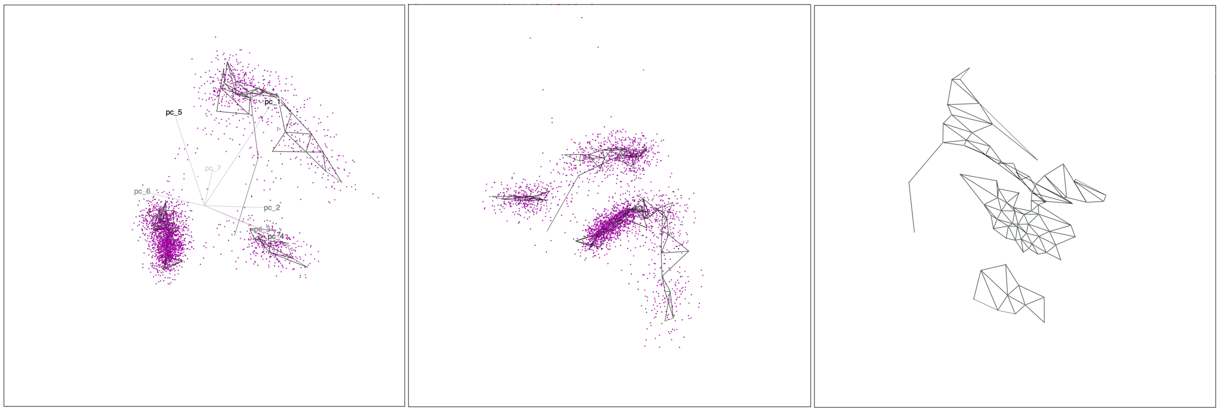


Figure 12: Visualization and refinement of low-dimensional representation for pbmc data: (a) 2D layout of UMAP (n-neigbors = 15), (b) triangular mesh, and (c) triangular mesh colored by edge type. Edges with lengths less than the benchmark value of 2 are classified as small, while edges with lengths greater than or equal to this value are labeled as long. The combination of UMAP, hexagonal binning, and triangulation provides the effectiveness of our approach in visualizing and refining the low-dimensional representation of the pbmc data.



(a)

(b)

(c)

Figure 13: Screen shots of the **langevitour** with the low-dimensional view applied to pbmc data set, a video of the tour animation is available at (https://drive.google.com/file/d/1F7Xb2EIpfUruCxYihPw2rKro4FB5Y2uL/view). Combination of the **langevitour** and low-dimensional representation of the pbmc data provides the effectiveness of our approach.

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 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.

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/*