Looking at Non-Linear Dimension Reductions as Models in the Data Space

Jayani P. Gamage
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

May 14, 2025

Abstract

Non-linear dimension reduction (NLDR) techniques such as tSNE, and UMAP provide a low-dimensional representation of high-dimensional data (p-D) by applying a non-linear transformation. NLDR often exaggerates random patterns, sometimes due to the samples observed. But NLDR views have an important role in data analysis because, if done well, they provide a concise visual (and conceptual) summary of p-D distributions. The NLDR methods and (hyper)parameter choices can create wildly different representations, making it difficult to decide which is best, or whether any or all are accurate or misleading. To help assess the NLDR and decide on which, if any, is the most reasonable representation of the structure(s) present in the p-D data, we have developed an algorithm to show the 2-D NLDR model in the p-D space, viewed with a tour, a movie of linear projections. From this, one can see if the model fits everywhere, or better in some subspaces, or completely mismatches the data. Also, we can see how different methods may have similar summaries or quirks.

Keywords: high-dimensional data vizualization, non-linear dimension reduction, tour

1 Introduction

Non-linear dimension reduction (NLDR) is popular for making a convenient low-dimensional (k-D) representation of high-dimensional (p-D) data ($k < p$). Recently developed methods include t-distributed stochastic neighbor embedding (tSNE) (Maaten & Hinton 2008), uniform manifold approximation and projection (UMAP) (McInnes et al. 2018), potential of heat-diffusion for affinity-based trajectory embedding (PHATE) algorithm (Moon et al. 2019), large-scale dimensionality reduction Using triplets (TriMAP) (Amid & Warmuth 2019), and pairwise controlled manifold approximation (PaCMAP) (Wang et al. 2021). However, the representation generated can vary dramatically from method to method, and with different choices of parameters or random seeds made using the same method (Figure 1).

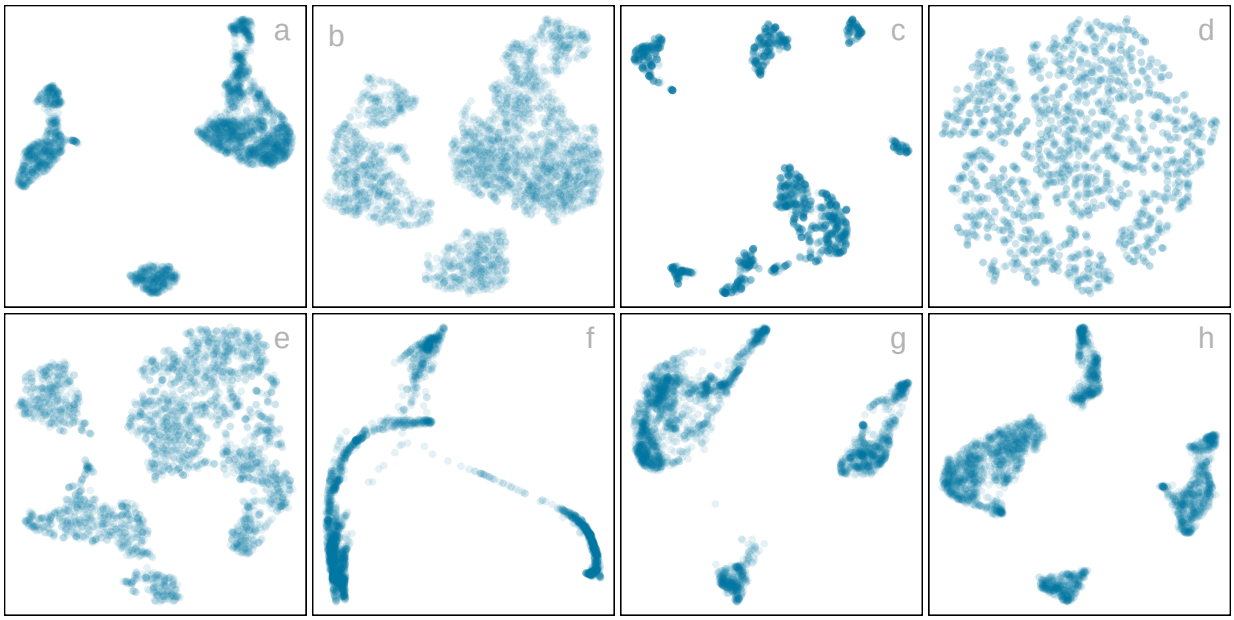

Figure 1: Eight different NLDR representations of the same data. Different techniques and different parameter choices are used. Researchers may have seen any of these in their analysis of this data, depending on their choice of method, or typical parameter choice. Would they make different decisions downstream in the analysis depending on which version seen? Which is the most accurate representation of the structure in high dimensions?

The dilemma for the analyst is then, **which representation to use**. The choice might result in different procedures used in the downstream analysis, or different inferential conclusions. The research described here provides new visual tools to aid with this decision.

The paper is organized as follows. Section 2 provides a summary of the literature on NLDR, and high-dimensional data visualization methods. Section 3 contains the details of the new methodology, including simulated data examples. Two applications illustrating the use of the new methodology for bioinformatics and image classification are in Section 6. Limitations and future directions are provided in Section 7.

2 Background

Historically, k-D representations of p-D data have been computed using multidimensional scaling (MDS) (Kruskal 1964), which includes principal components analysis (PCA) (Jolliffe 2011) as a special case. (A contemporary comprehensive guide to MDS can be found in Borg & Groenen (2005).) The k-D representation can be considered to be a layout of points in k-D produced by an embedding procedure that maps the data from p-D. In MDS, the k-D layout is constructed by minimizing a stress function that differences distances between points in p-D with potential distances between points in k-D. Various formulations of the stress function result in non-metric scaling (Saeed et al. 2018) and isomap (Silva & Tenenbaum 2002). Challenges in working with high-dimensional data, including visualization, are outlined in Johnstone & Titterington (2009).

Many new methods for NLDR have emerged in recent years, all designed to better capture specific structures potentially existing in p-D. Here we focus on five currently popular

techniques, tSNE, UMAP, PHATE, TriMAP and PaCMAP. tNSE and UMAP can be considered to produce the k-D minimizing the divergence between two distributions, where the distributions are modeling the inter-point distances. PHATE, TriMAP and PaCMAP are examples of diffusion processes (Coifman et al. 2005) spreading to capture geometric shapes, that include both global and local structure.

The array of layouts in Figure 1 illustrate what can emerge from the choices of method and parameters, and the random seed that initiates the computation. Key structures interpreted from these views suggest: (1) highly **separated clusters** (a, b, e, g, h) with the number ranging from 3-6; (2) **stringy branches** (f), and (3) **barely separated clusters** (c, d) which would **contradict** the other representations.

It happens because these methods and parameter choices provide different lenses on the interpoint distances in the data.

The alternative approach to visualizing the high-dimensional data is to use linear projections. PCA is the classical approach, resulting in a set of new variables which are linear combinations of the original variables. Tours, defined by Lee et al. (2021), broaden the scope by providing movies of linear projections, that provide views the data from all directions. Lee et al. (2021) provides an review of the main developments in tours. There are many tour algorithms implemented, with many available in the R package `tourr` (Wickham et al. 2011), and versions enabling better interactivity in `langevitour` (Harrison 2023) and `detourr` (Hart & Wang 2022). Linear projections are a safe way to view high-dimensional data, because they do not warp the space, so they are more faithful representations of the structure. However, linear projections can be cluttered, and global patterns can obscure local structure. The simple activity of projecting data from p-D suffers from piling (Laa

et al. 2022), where data concentrates in the center of projections. NLDR is designed to escape these issues, to exaggerate structure so that it can be observed. But as a result NLDR can hallucinate wildly, to suggest patterns that are not actually present in the data.

The solution is to use the tour to examine how the NLDR is warping the space. This approach follows what Wickham et al. (2015) describes as *model-in-the-data-space*. The fitted model should be overlaid on the data, to examine the fit relative the spread of the observations. While this is straightforward, and commonly done when data is 2-D, it is also possible in p-D, for many models, when a tour is used.

Wickham et al. (2015) provides several examples of models overlaid on the data in p-D. In hierarchical clustering, a representation of the dendrogrom using points and lines can be constructed by augmenting the data with points marking merging of clusters. Showing the movie of linear projections reveals shows how the algorithm sequentially fitted the cluster model to the data. For linear discriminant analysis or model-based clustering the model can be indicated by $(p-1)$-D ellipses. It is possible to see whether the elliptical shapes appropriately matches the variance of the relevant clusters, and to compare and contrast different fits. For PCA, one can display the k-D plane of the reduced dimension using wireframes of transformed cubes. Using a wireframe is the approach we take here, to represent the NLDR model in p-D.

3 Method

3.1 What is the NLDR model?

At first glance, thinking of NLDR as a modeling technique might seem strange. It is a simplified representation or abstraction of a system, process, or phenomenon in the real world. The p-D observations are the realization of the phenomenon, and the k-D NLDR layout is the simplified representation. From a statistical perspective we can consider the distances between points in the k-D layout to be variance that the model explains, and the (relative) difference with their distances in p-D is the error, or unexplained variance. We can also imagine that the positioning of points in 2-D represent the fitted values, that will have some prescribed position in p-D that can be compared with their observed values. This is the conceptual framework underlying the more formal versions of factor analysis (Jöreskog 1969) and MDS. (Note that, for this thinking the full p-D data needs to be available, not just the interpoint distances.)

We define the NLDR as a function g: $\mathbb{R}^{n\times p} \rightarrow \mathbb{R}^{n\times k}$, with (hyper-)parameters $\boldsymbol{\theta}$. The parameters, $\boldsymbol{\theta}$, depend on the choice of g, and can be considered part of model fitting in the traditional sense. Common choices for g include functions used in tSNE, UMAP, PHATE, TriMAP, PaCMAP, or MDS, although in theory any function that does this mapping is suitable.

With our goal being to make a representation of this 2-D layout that can be lifted into high-dimensional space, the layout needs to be augmented to include neighbour information. A simple approach would be to triangulate the points and add edges. A more stable approach is to first bin the data, reducing it from n to $m \leq n$ observations, and connect

Notation	Description
n, p, k	number of observations, variables, embedding dimension, respectively
$\boldsymbol{X}, \boldsymbol{x}$	p-dimensional data (population, sample)
$\boldsymbol{y}$	k-dimensional layout
P	orthonormal basis, generating a *d-dimensional* linear projection of p-dimensional data
T	true model
g	functional mapping from p-D to k-D, especially as prescribed by NLDR
$\boldsymbol{\theta}$	(Hyper-) parameters for NLDR method
r	ranges of the embedding components
$C^{(j)}$	j-dimensional bin centers
(b_1, b_2)	number of bins in each direction
(a_1, a_2)	binwidths, distance between centroids in each direction
$(s_1,\ s_2)$	starting coordinates of the hexagonal grid
q	buffer to ensure hexgrid covers data, proportion of data range, 0-1
m	number of non-empty bins
b	number of hexagons in the grid
h	hexagonal id
l	side length
A	area

Table 1: Summary of notation for describing new methodology.

the bin centroids. We recommend using a hexagon grid because it better reflects the data distribution and has less artifacts than a rectangular grid. This process serves to reduce some noisiness in the resulting surface shown in p-D. The steps in this process are shown in Figure 2, and documented below.

To illustrate the method, we use 7-D simulated data, which we call the "non-linear clusters". It is constructed by simulating two clusters, each consisting of 1000 observations. The C-shaped cluster is generated from $\theta \sim U(-3\pi/2, 0)$, $X_1 = \sin(\theta)$, $X_2 \sim U(0, 2)$ (adding thickness to the C), $X_3 = \text{sign}(\theta) \times (\cos(\theta) - 1)$, $X_4 = \cos(\theta)$. The other cluster is from $X_1 \sim U(0, 2)$, $X_2 \sim U(0, 3)$, $\gamma \sim U(0, 0.5)$, $X_3 = -(X_1^3 + X_2) + \gamma$, and $X_4 \sim U(0, 2)$. We would consider $T = (X_1, X_2, X_3, X_4)$ to be the geometric structure (true model) that we hope to capture.

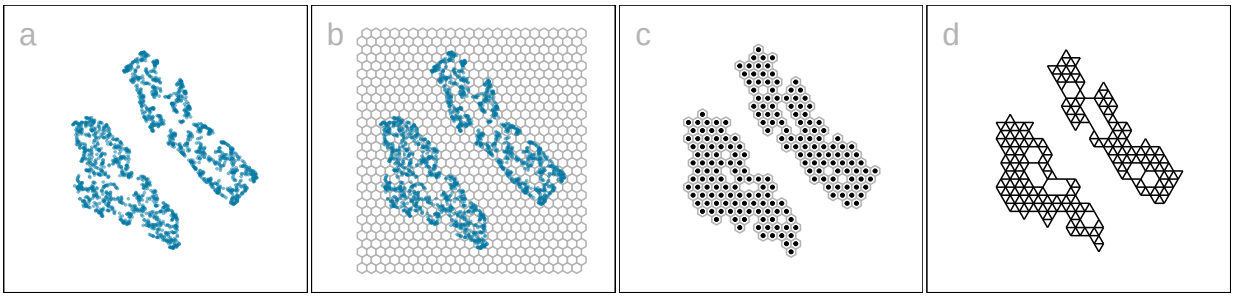

Figure 2: Key steps for constructing the model on the tSNE layout ($k = 2$): (a) data, (b) hexagon bins, (c) bin centroids, and (d) triangulated centroids. The two non-linear clusters data is shown.

3.2 Algorithm to represent the model in 2-D

3.2.1 Scale the data

Because we are working with distances between points, starting with data having a standard scale, e.g. [0, 1], is recommended. The default should take the aspect ratio produced by the

NLDR $(r_1, r_2, ..., r_k)$ into account. When $k = 2$, as in hexagon binning, the default range is $[0, y_{i,\max}], i = 1, 2$, where $y_{1,\max} = 1$ and $y_{2,\max} = r_2/r_1$ (Figure 2). If the NLDR aspect ratio is ignored then set $y_{2,\max} = 1$.

3.2.2 Computing hexagon grid configuration

Although there are several implementations of hexagon binning (Carr et al. 1987), and a published paper (Carr et al. 2023), surprisingly, none has sufficient detail or components that produce everything needed for this project. So we described the process used here. Figure 3 illustrates the notation used.

The 2-D hexagon grid is defined by its bin centroids. Each hexagon, H_h $(h = 1, \ldots, b)$ is uniquely described by centroid, $C_h^{(2)} = (c_{h1}, c_{h2})$. The number of bins in each direction is denoted as (b_1, b_2), with $b = b_1 \times b_2$ being the total number of bins. We expect the user to provide just b_1 and we calculate b_2 using the NLDR ratio, to compute the grid.

To ensure that the grid covers the range of data values a buffer parameter (q) is set as a proportion of the range. By default, $q = 0.1$. The buffer should be extending a full hexagon width (a_1) and height (a_2) beyond the data, in all directions. The lower left position where the grid starts is defined as (s_1, s_2), and corresponds to the centroid of the lowest left hexagon, $C_1^{(2)} = (c_{11}, c_{12})$. This must be smaller than the minimum data value. Because it is one buffer unit, q below the minimum data values, $s_1 = -q$ and $s_2 = -qr_2$.

The value for b_2 is computed by fixing b_1. Considering the upper bound of the first NLDR component, $a_1 > (1 + 2q)/(b_1 - 1)$. Similarly, for the second NLDR component,

$$a_2 > \frac{r_2 + q(1 + r_2)}{(b_2 - 1)}.$$

Since $a_2 = \sqrt{3}a_1/2$ for regular hexagons,

$$a_1 > \frac{2[r_2 + q(1 + r_2)]}{\sqrt{3}(b_2 - 1)}.$$

This is a linear optimization problem. Therefore, the optimal solution must occur on a vertex. Therefore,

$$b_2 = \left\lceil 1 + \frac{2[r_2 + q(1 + r_2)](b_1 - 1)}{\sqrt{3}(1 + 2q)} \right\rceil.$$

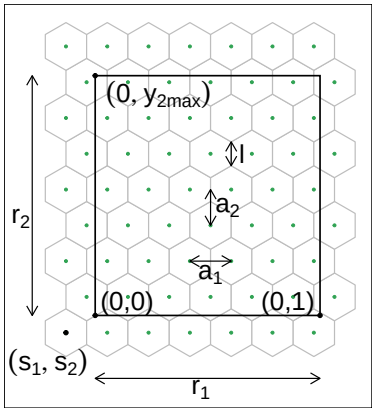

Figure 3: The components of the hexagon grid illustrating notation.

3.2.3 Binning the data

Observations are grouped into bins based on their nearest centroid. This produces a reduction in size of the data from n to m, where $m \leq b$ (total number of bins). This can be defined using the function $u : \mathbb{R}^{n\times 2} \rightarrow \mathbb{R}^{m\times 2}$, where

$u(i) = \arg\min_{j=1,\ldots,b} \sqrt{(y_{i1} - C_{j1}^{(2)})^2 + (y_{i2} - C_{j2}^{(2)})^2}$, mapping observation i into $H_h = \{i|u(i) = h\}$.

By default, the bin centroid is used for describing a hexagon (as done in Figure 2 (c)), but any measure of center, such as a mean or weighted mean of the points within each hexagon, could be used. The bin centers, and the binned data, are the two important components needed to render the model representation in high dimensions.

3.2.4 Indicating neighborhood

Delaunay triangulation (Lee & Schachter 1980, Gebhardt et al. 2024) is used to connect points so that edges indicate neighbouring observations, in both the NLDR layout (Figure 2 (d)) and the p-D model representation. When the data has been binned the triangulation connectd centroids. The edges preserve the neighborhood information when the model is lifted into p-D.

When shapes are non-linear in the NLDR layout, some edges could be long. It can also happen that distant centroids can be connected, particularly if clustering is present, which can result in long line segments. In order to generate a smooth surface in 2-D, these long line segments should be removed when tuning the model fit.

3.3 Rendering the model in p-D

The last step is to lift the k-D model into p-D by computing p-D vectors that represent bin centroids. We use the p-D mean of the points in H_h to map the centroid $C_h^{(2)} = (c_{h1}, c_{h2})$ to a point in p-D. Let the p-D mean be

$$C_h^{(p)} = \frac{1}{n_h} \sum_{i=1}^{n_h} x_i, h = 1, \ldots, b; n_h > 0.$$

Furthermore, line segments that exist in the k-D model generate line segments in p-D by connecting the p-D means of the corresponding k-D bin centroids. If additional long edges need to be removed, compute the edges in p-D and pruned any detected long edges to improve the accuracy. Once pruned, re-plot the 2-D view to ensure it accurately captures the data.

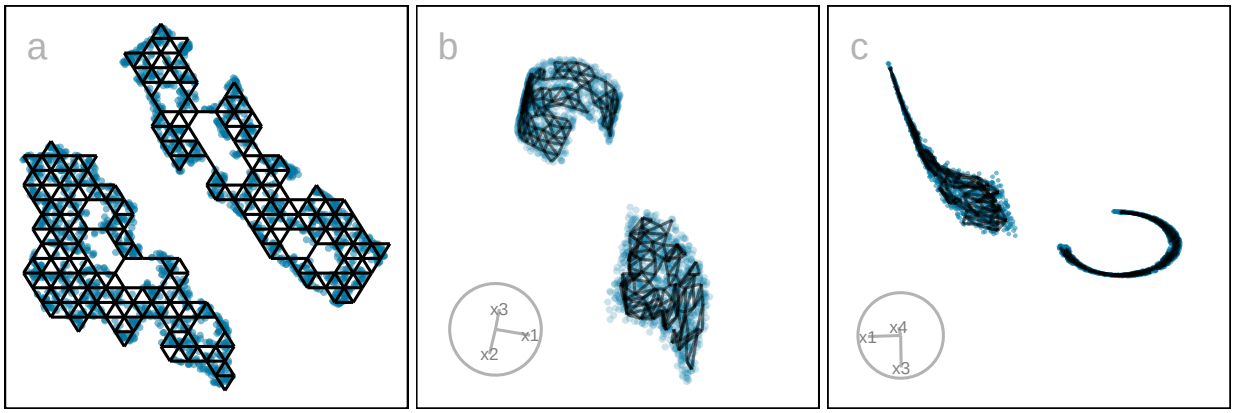

Figure 4: Lifting the fitted model into p-D. Starting from a default tSNE layout (a) of non-linear clusters data ($n = 2000$ and $p = 4$), from which the 2-D wireframe is constructed (a) and then lifted to 4-D (b, c). The fit is reasonbly tight with the data, although it does not fully spread the full width. This highlights a key characteristic of tSNE, which compresses the data when applying tSNE. Furthermore, the sparse space observed in the middle of the tSNE layout is a result of (hyper-)parameter choice. Video of the langevitour animation is available at <>.

3.4 Measuring the fit

The model here is similar to a confirmatory factor analysis model (Brown 2015), $\widehat{T}(X_1, X_2, X_3, X_4) + \mathrm{E}$. The difference between the fitted model and observed values would be considered to be residuals, and for this problem are 4-D.

Observations are associated with their bin center, $C_h^{(p)}$, which are also considered to be the

fitted values. These can also be denoted as $\widehat{X}$.

The error is computed by taking the squared p-D Euclidean distance, corresponding to computing the root mean squared error (RMSE) as:

$$\sqrt{\frac{1}{n}\sum_{h=1}^{b}\sum_{i=1}^{n_h}\sum_{j=1}^{p}(\boldsymbol{x}_{hij}-C_{hj}^{(p)})^2} \tag{1}$$

where n is the number of observations, b is the number of bins, n_h is the number of observations in h^{th} bin, p is the number of variables, $\boldsymbol{x}_{hij}$ is the j^{th} dimensional data of i^{th} observation in h^{th} hexagon. We can consider $e_{hj} = \sqrt{\sum_{j=1}^{p}(\boldsymbol{x}_{hij}-C_{hj}^{(p)})^2}$ to be the residual for each observation.

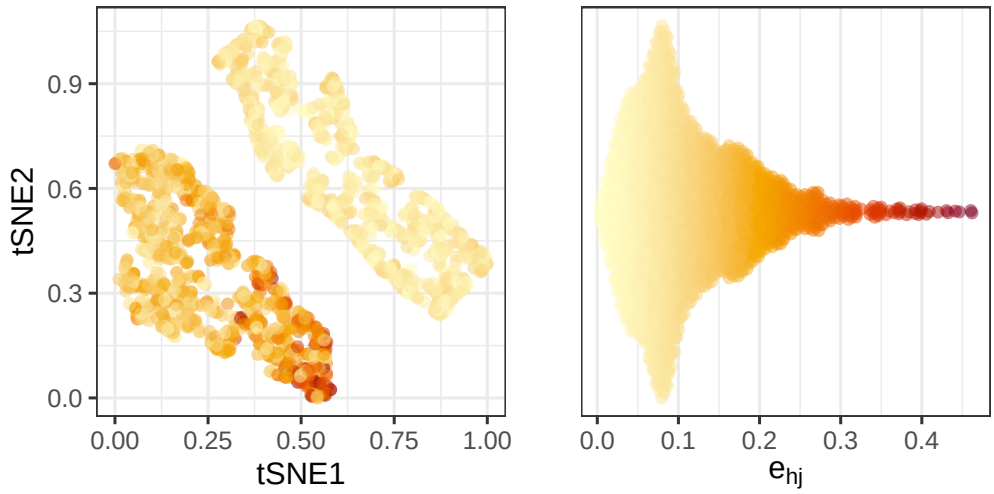

Figure 5: The 4-D model error in 2-D layout. Color indicates error (e_{hj}), dark colour indicating high error and light indicates low error. Most large errors are distributed near the sparse end of the non-linear cluster with dense corner.

3.5 Prediction into 2-D

A new benefit of this fitted model is that it allows us to now predict a new observation's value in the NLDR, for any method. The steps are to determine the closest bin centroid

in p-D, $C_h^{(p)}$ and predict it to be the centroid of this bin in 2-D, $C_h^{(2)}$. This can be written as, let $z(i) = \arg\min_{j=1,\ldots,b} \sqrt{\sum_{v=1}^{p}(x_{iv} - C_{jv}^{(p)})^2}$, then the new observation i falls in the hexagon, $H_h = \{i|z(i) = h\}$ and the corresponding k-D bin centroids, $C_h^{(2)} = (c_{h1}, c_{h2})$.

3.6 Tuning

The model fitting can be adjusted using these parameters:

- hexagon bin parameters
  - bottom left bin position $(s_1,\ s_2)$,
  - the total number of bins (b),
- bin density cutoff, to remove low-density hexagons.

Default values are provided for each of these, but it is expected that the user will examine the MSE for a range of choices. Choosing these parameters according to MSE can be automated but it is recommended that the user examine the resulting model representation by overlaying it on the data in p-D. The next few subsections describe the calculation of default values, and the effect that different choices have on the model fit.

3.6.1 Hexagon bin parameters

The values $(s_1,\ s_2)$ define the position of the centroid of the bottom left hexagon. By default, this is at $s_1 = -q, s_2 = -qr_2$, where q is the buffer bound the data. The choice of these values can have some effect on the distribution of bin counts. Figure 6 illustrates

this. The distribution of bin counts for s_1 varying between $-0.1 - 0.0$ is shown. Generally, a more uniform distribution among these possibilities would indicate that the bins are reliably capturing the underlying distribution of observations.

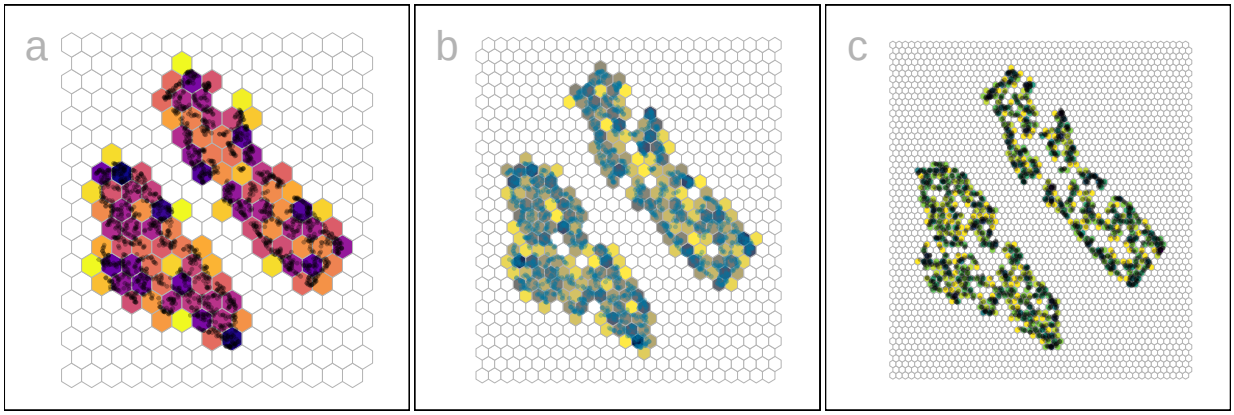

Figure 6: Hexbin density plots of tSNE layout of the non-linear cluster data, using three different bin inputs: (a) $b = 240/98$ (15, 16), (b) $b = 720/215$ (24, 30), and (c) $b = 2496/549$ (48, 52). Color indicates standardized counts, dark indicating high count and light indicates low count. At the smallest bin size, the data structure is discontinuous, suggesting that there are too many bins. Using the MSE of the model fit in 7-D helps decide on a useful choice of number of bins.

The default number of bins $b = b_1 \times b_2$ is computed based on the sample size, by setting $b_1 = n^{1/3}$, consistent with the Diaconis-Freedman rule (Freedman & Diaconis 1981). The value of b_2 is determined analytically by b_1, q, r_2. Values of b_1 between 2 and $b_1 = \sqrt{\frac{n}{r_2}}$ are allowed. Figure 7 (a) shows the effect of different choices of b_1 on the MSE of the fitted model.

3.6.2 Measurement of capturing the data shape in 2-D

The area of a hexagon is defined as $A = \frac{3\sqrt{3}}{2}l^2$ where l is the side length of the hexagon. If we know a_1 and a_2, l can be computed (see appendix). The density of a hexagon grid is calculated as $\frac{\sum_{i=1}^{h} n_h}{A}$ and the proportion is $\frac{\sum_{i=1}^{h} n_h}{A \times b}$. The baseline proportion is the

proportion density at the smallest possible value of a_1. The relative proportion density is the ratio of the observed proportion density to the baseline proportion density.

3.6.3 Removal of low density bins

By default, when assessing the choice of b_1, the total number of bins is measured by the number of **non-empty** bins. This more accurately reflects the hexagon grid relative the MSE than the full number of bins in the grid. It may also be beneficial to remove low count bins also, in the situation where data is clustered or stringy, where the observed data is sparse. In order to decide if this is necessary, you would examine the distribution of bin counts, or the density which puts the counts on a standard scale. If there is something of a gap at low values, this would suggest a potential value to use as a cutoff. Alternatively, one could choose to remove based on a percentile, the bins with density in the lowest 5% of all bins, for example. Figure 7 (c) illustrates the effect on the model representation of removing bins below different percentages. Generally, we would urge caution in removing low count bins.

The benchmark value for removing low-density hexagons ranges between 0 and 1. When analyzing how these benchmark values influence model performance, it's essential to observe the change in MSE as the benchmark value increases (Figure 7 (c)). The MSE shows a gradual decrease as the benchmark value goes from 1 to 0. Evaluating this rate of increase is important. If the increment is not considerable, the decision might lean towards retaining low-density hexagons.

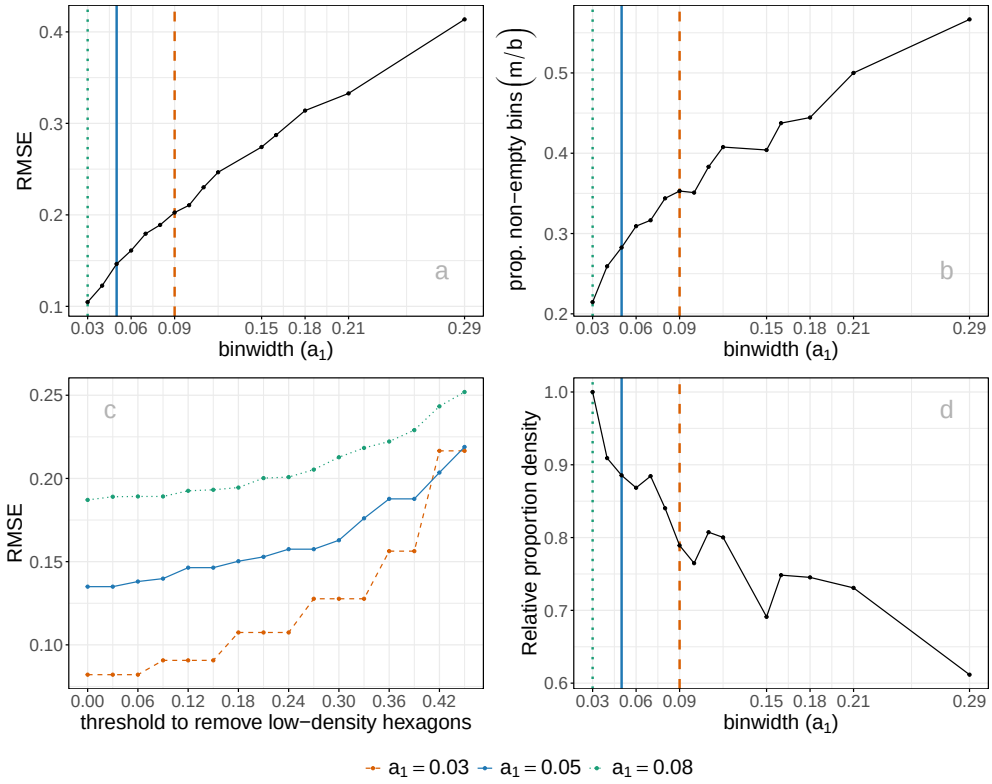

Figure 7: Various plots to help assess best hexagon bin parameters (a, b, d) and thresholds to remove low-density bins (c). Both (a) and (c) show RMSE, against binwidth (a_1) and threshold. A good benchmark value for these parameters is when the RMSE drops and then flattens out. Three binwidth choices were made: 0.03 (orange dashed), 0.05 (blue solid), and 0.09 (green dotted) to investigate. As the binwidth increases, the proportion of non-empty bins also increases (b). The relative proportion density decreases and levels off (d). Binwidth 0.05 is chosen as the initial best binwidth for further analysis. There is no need to remove the low-density hexagons because as shown in (b), there is no considerable drop in RMSE.

3.7 Linked plots

Diagnosing the model while locating points in the 2-D layout and displaying the generated model overlaid on data in the p-D is important.

The 2-D layout and the langevitour view with the model are linked together via rectangular brushes; when a brush is active, points will be highlighted in the adjacent view. Because the langevitour is dynamic, brush events that become active will pause the animation, so

that a user can interrogate the current view. The interface is constructed as a **browsable HTML widget** specifically designed for interactive data analysis.

To understand how well the model fits the points whether it fits well, works better in some positions, or fails to match the overall pattern. It is important to link and brush the points with high model error in the p-D error plot, 2-D layout, and the generated model overlay on the data in p-D.

4 Best fit, or at least avoiding an inaccurate representation

Figure 8 shows a summary of plots that can be used to help assess the strength of the fit, and compare the fits for different representations. What does it mean to be a best fit for this problem? There probably isn't a best fit, but there are wrong representations. The goal is to help users decide on a useful and appropriate low-dimensional representation of the high-dimensional data.

Deciding on the best fit relies on several elements:

- the choice of NLDR method, and the parameters used to create it, and
- model fit parameters: bin size, low density bin removal.

Comparing the MSE to obtain the best fit is suitable if one starts from the same NLDR representation. In theory, because the MSE is computed on p-D measuring the fit between model and data it might still be useful to compare different NLDR representations. A good

NLDR representation should produce a good fit, producing a low MSE if the model fits the data well. However, it technically might be quite variable.

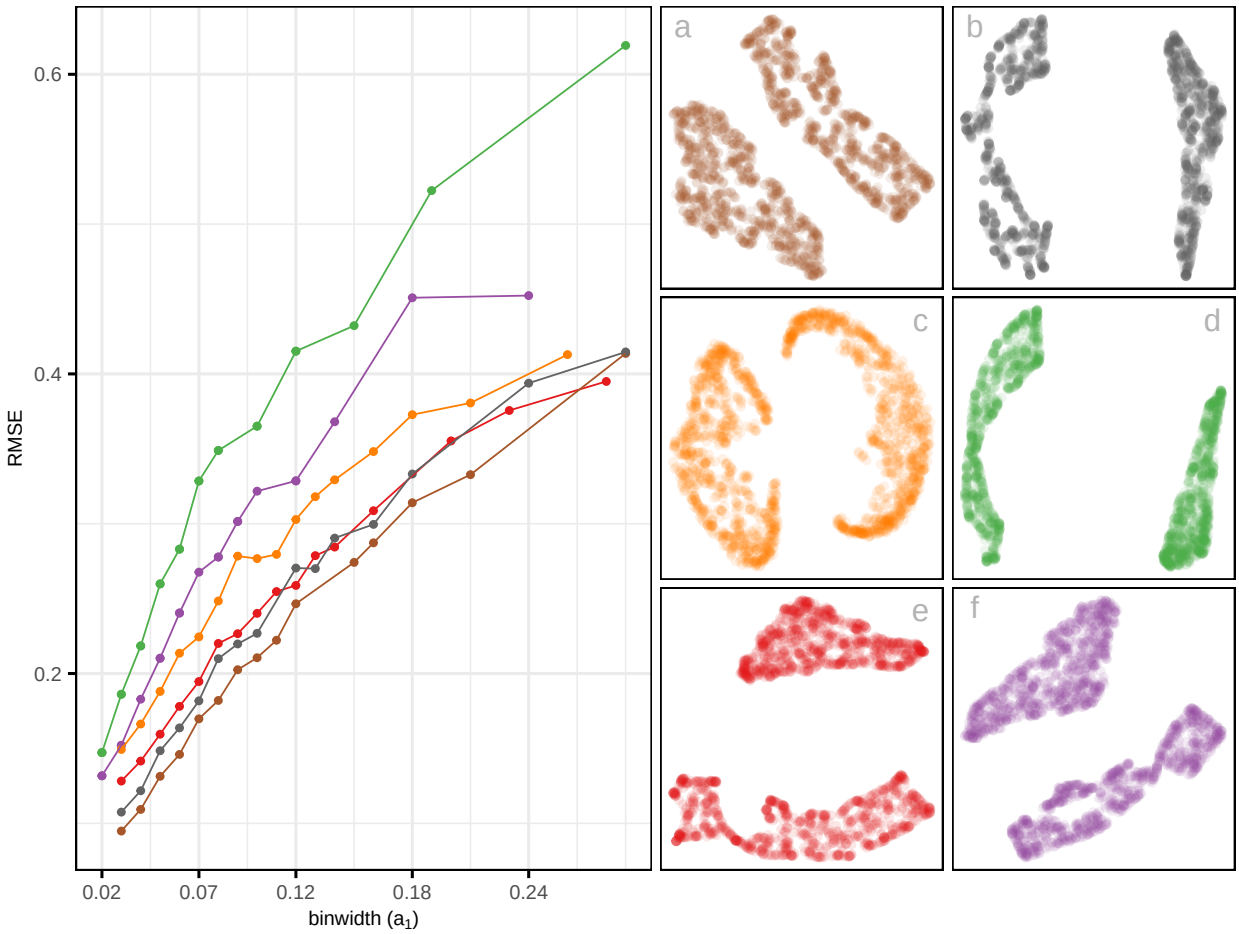

Figure 8: Assessing which of the 6 NLDR layouts on the two non-linear clusters data is the better representation using RMSE for varying binwidth (a_1). Colour used for the lines and points in the left plot and in the scatterplots represents NLDR layout (a-f). Layout d is universally poor. Layouts a, b, e that show two close clusters are universally suboptimal. Layout b with little separation performs well at tiny binwidth (where most points are in their own bin) and poorly as binwidth increases. Layout e has small separation with oddly shaped clusters. Layout a is the best choice.

5 Curiosities

With the drawing of the model in the data, several interesting differences between NLDR methods can be observed.

5.1 Ordering of points

To illustrate a difference in how the methods organise points in the NLDR layout, simulated 4-D data having five Gaussian clusters is used. Figure 9 a1, b1, c1 show the 2-D layouts for (a) tSNE, (b) UMAP, and (c) PaCMAP, respectively. The default hyper-parameters are used. All three methods show the five clusters, with varying degrees of separation.

The models are fitted to these layouts, but we focus on a single cluster to illustrate the curious detail. Figure 9 a2, b2, c2 show the fitted models in a projection of the 4-D space. In this projection the difference between methods can be seen. These clusters are fully 4-D in nature, so we would expect the model to be a *crumpled* 2-D *sheet* that stretches in all four dimensions. This is what is observed for tSNE and UMAP. The curious detail is that the model for PaCMAP is closer to a *pancake* in shape! What this means is that there has to be some ordering of points in the 2-D PaCMAP layout that induces the flat model, likely that the points are organised by the global principal components. So the layout of the model in 4-D doesn't reflect the dimension of each cluster. This can be happen because PaCMAP balances three components (near, mid-range, and far) rather than focusing solely on near neighbors. Mid-range neighbors ensure that transitions between clusters or groups are preserved and less concern with local preservation.

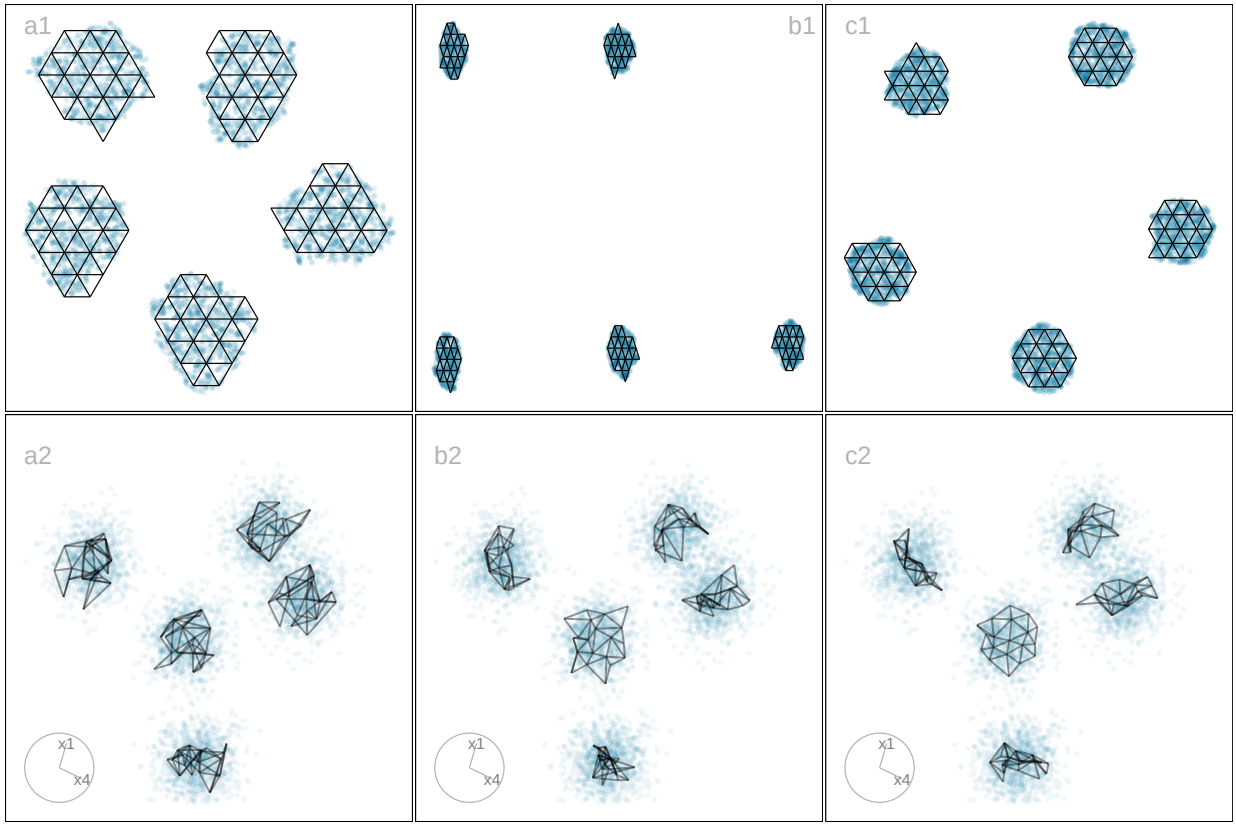

Figure 9: NLDR's organise points in the 2-D layout in different ways, possibly misleadingly, illustrated using three layouts: (a) tSNE, (b) UMAP, (c) PaCMAP. The data has five Gaussian cluster in 4-D. The bottom row of plots shows a 2-D projection from a tour on 4-D revealing the differences generated by the layouts on the model fits. We would expect the model fit to be like that in (a2) where it is distinctly is separate for each cluster but like a hairball in each. This would indicate the distinct clusters, each being fully 4-D. With (c2), the curiousity is that the model is a 2-D pancake shape in 4-D, indicating that there is some ordering of points done by PaCMAP, posisbly along some principal component axes. Videos of the langevitour animations are available at https://youtu.be/oQxEb4wRdHI, https://youtu.be/JW49csPpDx4, and XXX respectively.

5.2 The effect of density

To illustrate how tSNE organize different number of points within the structure in the tSNE layout, simulated 4-D data having C-shaped structure is used. Figure 10 b shows 2-D layouts for tSNE. The default hyper-parameters are used. The 2-D layout is colored according to the 4-D model error (Figure 10 a). In Figure 10 a, the large model error

points are clustered in the top corner, forming a separate cluster with high model errors.

The model is fitted to this layout. Figure 10 c shows the 2-D projection of the fitted model. In this projection, the distribution of 4-D model errors can be observed. The dense C-shaped structure shows high model errors at the sparse end. In the dense C-shaped structure, the high model errors occur because there are fewer data points positioned within the bins at the sparse end of the structure.

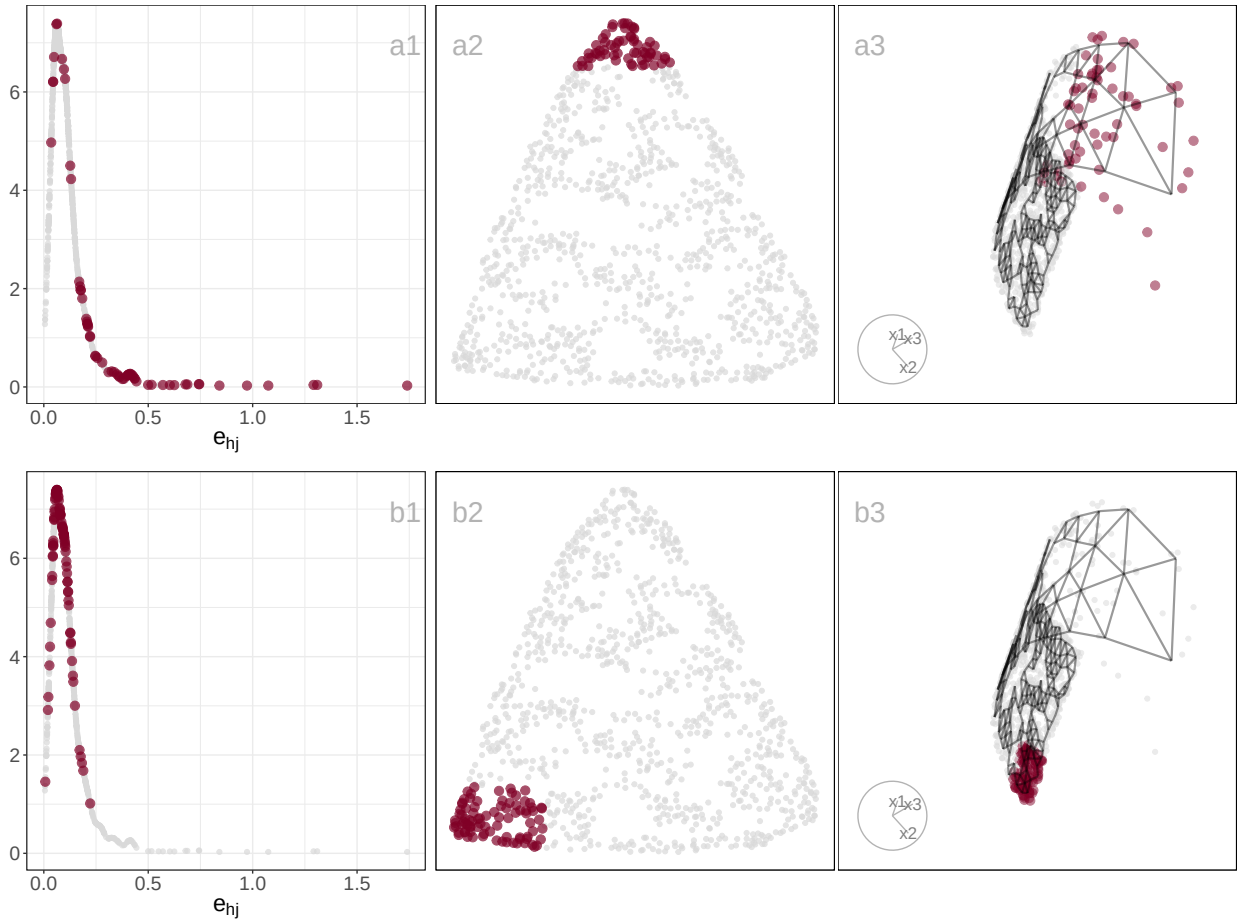

Figure 10: The sparse end of the structure shows a high model error, while the dense end shows a low model error. The tSNE layout of the C-shaped structure, with dense points (a1-a3) are colored according to their 4-D model error. Darker colors represent a high error, while lighter colors indicate a low error. a2, a3 and b2, b3 present the results from linked plots that brushed the high 4-D error points in 2-D and 4-D. It helps in identifying the high 4-D model errors that occur due to the sparse end. Videos of the linked plots are available at <>.

6 Applications

To showcase the robustness of our method when applied to data with different characteristics, this section presents two case studies. The first application focuses on single-cell data, a domain where the high dimensionality, sparsity, and heterogeneity of gene expression measurements pose significant analytical challenges. Our approach facilitates the identification of the most accurate NLDR layout, enabling the discovery of cell populations with similar expression profiles. The second case study explores the MNIST hand-written digits dataset, which serves as a benchmark for evaluating the preservation of local structures, specifically with digit 1.

6.1 Single-cell gene expression

Single-cell data refers to the characteristics of individual cells within a population (Haque et al. (2017)). These data also provides insights at the level of individual cells, enabling deeper understanding of gene expression.

Clustering of single-cell data is used to identify groups of cells with similar expression profiles. NLDR often used to summarise the discovered clusters, and help to understand the results. The purpose of this example is to *illustrate how to use our method to help decide on an appropriate NLDR layout that accurately represents the data.* The cluster results of Human Peripheral Blood Mononuclear Cells (PBMC3k) in Chen et al. (2024) are examined. There are 2622 single cells, with 1000 gene expressions. Following their pre-processing, UMAP was performed using nine principal components. Figure 1 (a) is the reproduction of the published plot. The question is whether this accurately represents the

cluster structure in the data. Our method can help here, and also help to provide a better 2-D layout, as needed.

The Figure 1 (a) shows three well-separated clusters with big separations. However, as shown in Figure 12 (a2), there is no big separation between three clusters in 9-D. Therefore, the suggested UMAP representation (Figure 1 (a)) does not accurately represent the structure of PBMC3k dataset.

As a result, it is necessary to find an appropriate layout for the dataset. MSE for different binwidths (a_1) using tSNE, UMAP, PHATE, PaCMAP, and TriMAP with various (hyper-)~parameter settings were computed (Figure 11). Layouts c, d, and e, which show small separations between clusters, are universally optimal. However, layout d performs well with smaller binwidths and poorly with larger binwidths. On the other hand, layout e performs well with larger binwidths. Layout c was selected for further analysis due to its better (hyper-)parameter selection for the same method of the published plot.

The visualization of the selected layout in the 9-D shows edges between the clusters (Figure 12 (b2)). This supports the presence of small separations between clusters. Additionally, the data points are not uniformly distributed across the clusters, resulting in dense areas (Figure 12 (b2)). Furthermore, the clusters shows non-linear shapes (Figure 12 (b3)).

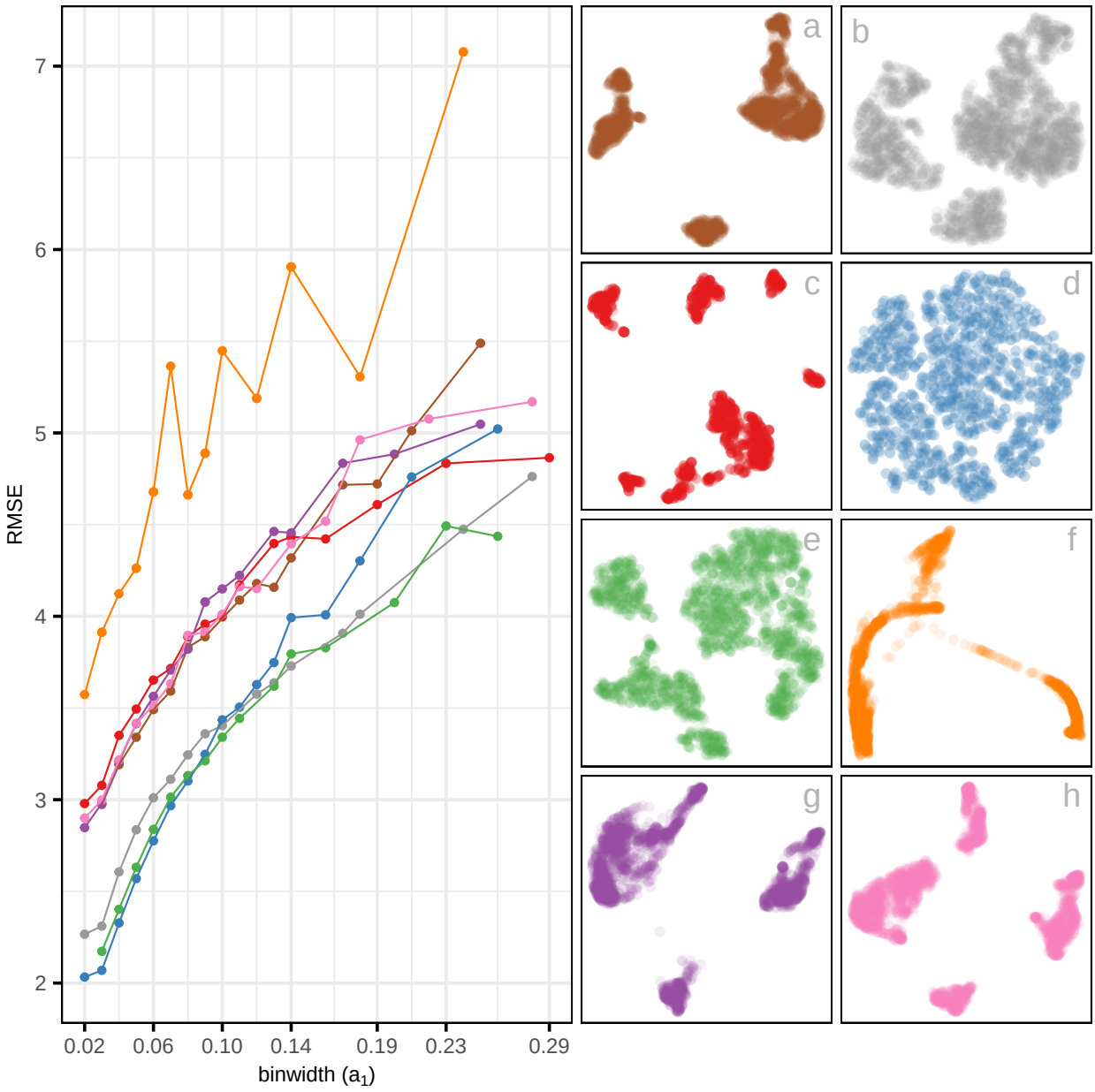

Figure 11: Assessing which of the 8 NLDR layouts on the PBMC3k data (shown in Figure 1) is the better representation using RMSE for varying binwidth (a_1). Colour used for the lines and points in the left plot and in the scatterplots represents NLDR layout (a-h). Layout f is universally poor. Layouts a, c, g, h that show large separations between clusters are universally suboptimal. Layout d with little separation performs well at tiny binwidth (where most points are in their own bin) and poorly as binwidth increases. The choice of best is between layouts b and e, that have small separations between oddly shaped clusters. Layout e is the best choice.

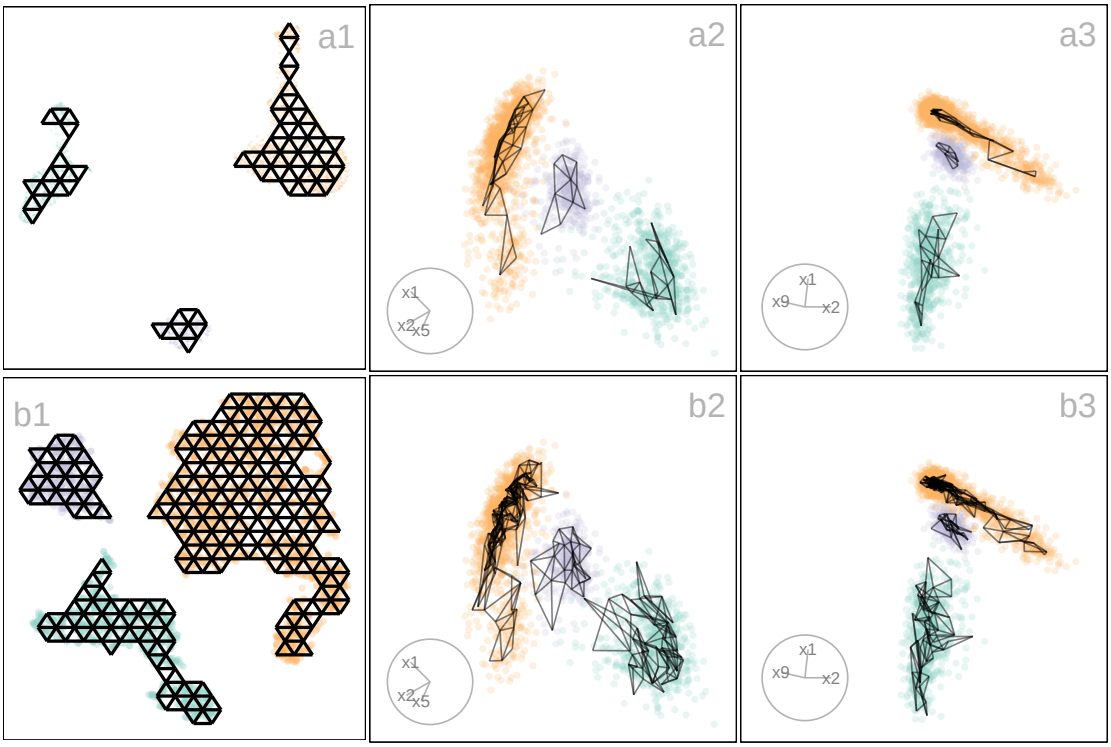

Figure 12: Compare the published 2-D layout (Figure 11 a) and the 2-D layout selected (Figure 11 e) by MSE plot (Figure 11) from the UMAP and tSNE with default hyper-parameters. The PBMC3k data ($n = 2622$) has three close clusters in 9-D. Two 2-D projection from a tour on 9-D of the model fit with Figure 11 a ($a_1 = 0.06$, $b = 616/86(22, 28)$) shows three-well separated clusters with big separations. On the other hand, the model fit with Figure 11 e ($a_1 = 0.06$, $b = 704/227(22, 32)$) shows three-well separated clusters with small separations. Therefore, Figure 11 e is more reasonable than Figure 11 a. Also, the fitted models in 9-D help to see some unobserved patterns of the data: (i) dense points, and (ii) non-linear clusters. Videos of the langevitour animations are available at https://youtu.be/0cKX_HG_n0k and https://youtu.be/KhJvsRtaX04 respectively.

6.2 Hand-written digits

The digit 1 of the MNIST dataset consists of 7877 grayscale images of handwritten digits (LeCun & Cortes 2010). Before further analysis, PCA was used to preprocess the data, where the first 10 principal components, explaining 83% of the total variation, were selected. The objective is to select a reasonable 2-D layout, representing the non-linear structure of the digit 1 dataset in 10-D (Figure 14 (a)).

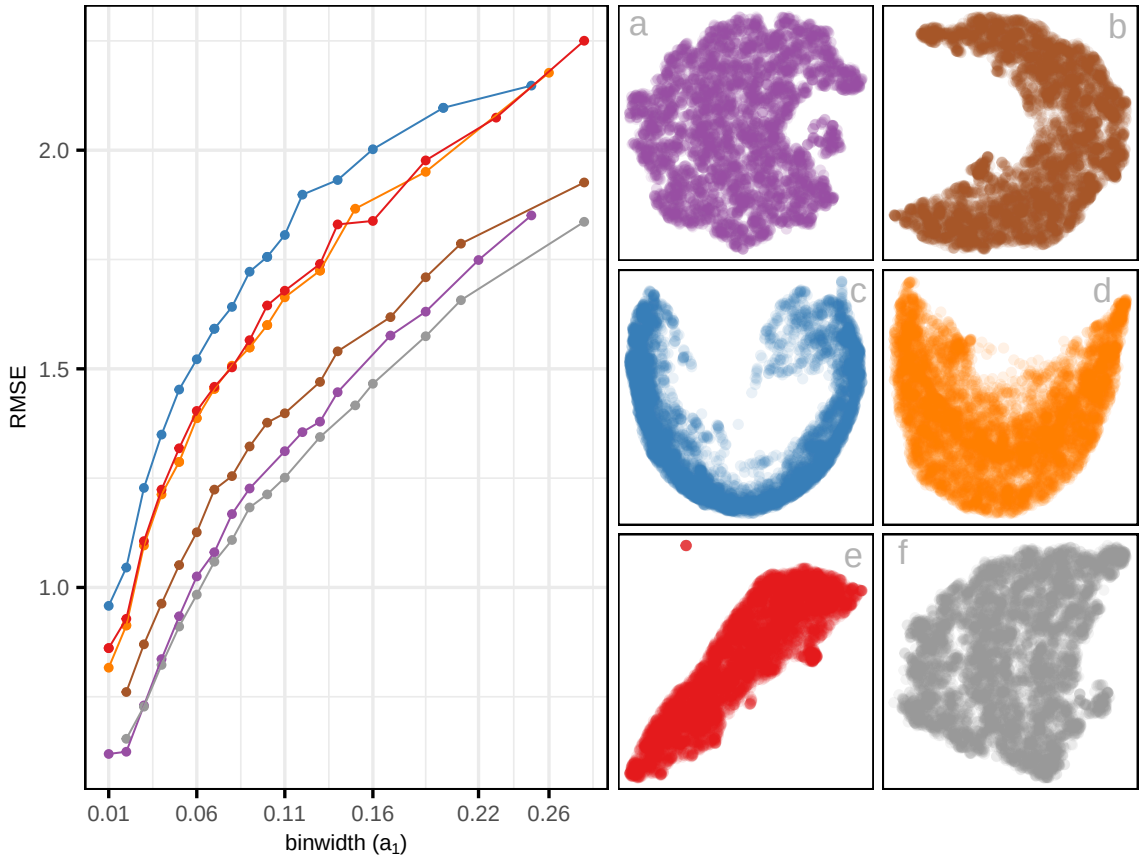

Figure 13: Assessing which of the 6 NLDR layouts on the MNIST digit 1 data is the better representation using RMSE for varying binwidth (a_1). Colour used for the lines and points in the left plot and in the scatterplots represents NLDR layout (a-f). All the layouts appear to be very similar. Layout c is universally poor. Layouts a, f that show two close clusters are universally suboptimal. Layout a with little separation performs well at tiny binwidth (where most points are in their own bin) and poorly as binwidth increases. Layout f is the best choice.

The MSE for different binwidths (a_1) using UMAP, PHATE, PaCMAP, TriMAP with default (hyper-)parameter setting and tSNE with default (hyper-)parameter setting and perplexity 89 (Figure 13) were calculated. It is found that tSNE with perplexity 89 (Figure 13 (f)) provide the most reasonable representation for the digit 1 dataset, showing universally best. However, 2-D representation shows a big non-linear cluster and a small cluster with a small gap.

In the case of digit 1 data, there should not be any clusters unless anomalies exist, indicating

different digit 1 patterns. The angle of the digit 1 images varies along this non-linear clustering structure (Figure 15), while the small cluster contains the digit 1 images with different patterns of the digit 1, unlike the usual (Figure 15 (c)). This provides the evidence for two close clusters.

By visualizing the model generated for tSNE in 10-D helps to assess the 2-D layout. As shown in Figure 14 (a1), the model provides the evidence for the non-linear structure of the digit 1 data in 10-D. The model shows some quirks. The model's twisted pattern provides evidence for the 10-D data structure, which is not observed by 2-D layout (Figure 14 (a2)).

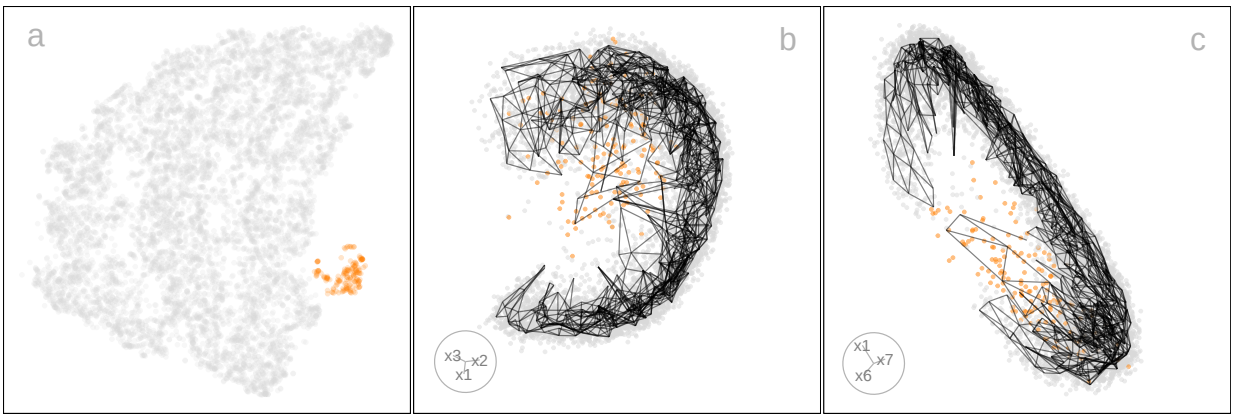

Figure 14: The tSNE layout of the MNIST digit 1 data shows a big non-linear cluster (grey) and a small cluster (orange) located very close to the one corner of the big cluster in 2-D (a). The MNIST digit 1 data ($n = 7877$ and $p = 10$) has a non-linear structure in 10-D. Two 2-D projections from a tour on 10-D reveal that the closeness of the clusters in 10-D and the twisted pattern of the model fit with tSNE ($a_1 = 0.041$, $b = 659/585$ (30, 48)). Video of the langevitour animation is available at <>. The brushing feature in the linked plots helps in visualizing the closeness of the small cluster to the big cluster. Videos of the linked plots are available at <>.

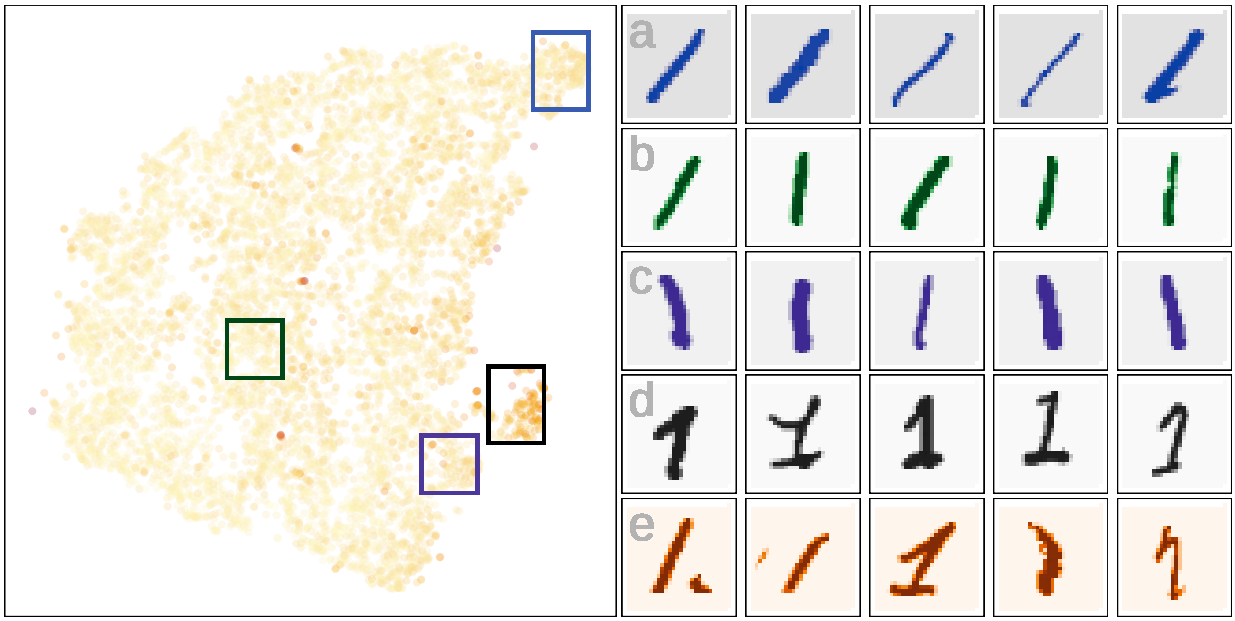

Figure 15: The 10-D model error in 2-D layout of the MNIST digit 1 data shows a pattern. Most low model errors are distributed along the big non-linear cluster, while most large model errors are distributed along the small cluster. The images associated with large model errors shows different patterns of digit 1, some inside (e) the non-linear structure and others outside (d). Along the non-linear cluster, the angle of digit 1 changes (a-c).

7 Discussion

We have developed an algorithm to assess the NLDR method and (hyper-)parameter choice(s) and choose the most accurate representation of the structure(s) present in the p-D data. Starting from 2-D layout, the fitted model, as represented by the positions of points in k-D, and turn it into a p-D wireframe to overlay on the data, viewing it with a tour. This approach is defined as *model-in-data-space*. Viewing a model in the data space is an ideal way to examine the fit.

With the model fit, several observations can be made. The simulated 4-D data, which contains five Gaussian clusters, was analyzed. Both tSNE and UMAP demonstrated the fitted model as a *crumpled* 2-D *sheet* that stretches across all four dimensions, highlighting

its ability to preserve local structure. In contrast, the flat representation generated by PaCMAP indicates its failure to capture the variation within the clusters. This highlights how different NLDR methods organize the data points differently.

Another usability of our approach is how NLDR methods behave with different density of the data structure. The tSNE layout of simulated 4-D C-shaped data with different densities are used. The high 4-D model errors are scattered to a corner of the 2-D layout of dense C-shaped structure. In the dense C-shaped structure, the high model errors occur because there are fewer data points positioned within the bins at the sparse end of the structure.

Our algorithm can also be useful in practice to decide on a reasonable NLDR layout that accurately represents the data, as described with a single-cell gene expression data, **PBMC3k**. The published results of UMAP with default (hyper-)parameter setting by Chen et al. (2024) were evaluated. The data showed three close clusters. But the published layout showed three well-separated clusters. Therefore, by changing the NLDR method to tSNE with default (hyper-)parameters, we found a better layout, which is more representative the data. Additionally, the model discovers non-uniform data distribution and non-linear structures within the clusters that are not visible in the 2-D layout, demonstrating the ability of our model in uncovering hidden data characteristics. Also, as explained in Appendix, with recently introduced approach, scDEED also proved the layout suggested by our layout is better than the published one.

The digit 1 of **Hand-written digits** data shows how the model accurately capture the non-linear structure. Not only that, the 10-D model error help to find the anomalies of digit 1 images. Also, twisted pattern of the model evident that how tSNE tries to maintain

the local structure by positioning the similar patterns of digit 1 close to each other.

Diagnose is important to find where the 2-D points located in p-D. One approach is to use interactivity. By linking the 2-D layout with the fitted model in p-D and employing brushing techniques, we can identify which points fit well and which do not. Additionally, linking the 2-D error plot with the fitted model in p-D can help pinpoint where mismatches occur.

Predicting new observations in k-D is particularly valuable due to the limitations of some NLDR methods, like tSNE, which don't provide a straightforward method for prediction. As a result, our approach offers a solution that capable of generating predicted k-D embedding regardless of the NLDR method employed, effectively addressing this functional gap.

Furthermore, to extend layouts beyond k-D, when 2-D is clearly inadequate, k-means can be used as an alternative to bin centroids.

There are many avenues for future work. First, the development of evaluation metrics is important for NLDR. We introduced a qualitative approach. But having a quantitative measure also will be useful for various application domains. Second, one could use our approach to create an approach for tuning (hyper-)parameters. Another direction for future work is to introduce more approaches to diagnose the fitted model. Finally, one can think of designing a new NLDR method by following the binning in high-dimensions.

8 Supplementary Materials

Appendix: The appendix includes more details about the hexagonal binning algorithm and scDEED method (appendix.pdf, Portable Document Format file).

R package `quollr`: The R package `quollr` containing codes to fit, and visualize the model. (need to add quollr .zip file, GNU zipped tar file)

Direct links to videos for viewing online are available in Table 2.

data	URL
PBMC3k	link1

Table 2: Example videos.

9 Acknowledgments

These `R` packages were used for the work: `tidyverse` (Wickham et al. (2019)), `Rtsne` (Krijthe (2015)), `umap` (Konopka (2023)), `patchwork` (Pedersen (2024)), `colorspace` (Zeileis et al. (2020)), `langevitour` (Harrison (2023)), `conflicted` (Wickham (2023)), `reticulate` (Ushey et al. (2024)), `kableExtra` (Zhu (2024)). These `python` packages were used for the work: `trimap` (Amid & Warmuth (2019)) and `pacmap` (Wang et al. (2021)). The article was created with `R` packages `quarto` (Allaire & Dervieux (2024)). The project's GitHub repository (https://github.com/JayaniLakshika/paper-nldr-vis-algorithm) contains all materials required to reproduce this article.

References

Allaire, J. & Dervieux, C. (2024), *quarto: R Interface to Quarto Markdown Publishing System*. R package version 1.4.4. https://CRAN.R-project.org/package=quarto.

Amid, E. & Warmuth, M. K. (2019), 'Trimap: Large-scale dimensionality reduction using triplets', *ArXiv* **abs/1910.00204**. https://api.semanticscholar.org/CorpusID:203610264.

Borg, I. & Groenen, P. J. F. (2005), *Modern Multidimensional Scaling Theory and Applications*, Springer, New York.

Brown, T. A. (2015), *Confirmatory Factor Analysis for Applied Research*, Guilford publications.

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. http://www.jstor.org/stable/2289444.

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. https://CRAN.R-project.org/package=hexbin.

Chen, Z., Wang, C., Huang, S., Shi, Y. & Xi, R. (2024), 'Directly selecting cell-type marker genes for single-cell clustering analyses', *Cell Reports Methods* **4**(7), 100810. https://www.sciencedirect.com/science/article/pii/S2667237524001735.

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.

Freedman, D. A. & Diaconis, P. (1981), 'On the histogram as a density estimator:l2 theory', *Probability Theory and Related Fields* **57**, 453–476. https://doi.org/10.1007/BF01025868.

Gebhardt, A., Bivand, R. & Sinclair, D. (2024), *interp: Interpolation Methods*. R package version 1.1-6. https://CRAN.R-project.org/package=interp

Haque, A., Engel, J., Teichmann, S. A. & Lönnberg, T. (2017), 'A practical guide to single-cell rna-sequencing for biomedical research and clinical applications', *Genome Medicine* **9**, 1–12.

Harrison, P. (2023), 'langevitour: Smooth interactive touring of high dimensions, demonstrated with scrna-seq data', *The R Journal* **15**, 206–219. https://doi.org/10.32614/RJ-2023-046.

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

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. https://royalsocietypublishing.org/doi/abs/10.1098/rsta.2009.0159.

Jolliffe, I. (2011), *Principal Component Analysis*, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 1094–1096. https://doi.org/10.1007/978-3-642-04898-2_455.

Jöreskog, K. G. (1969), 'A general approach to confirmatory maximum likelihood factor analysis', *Psychometrika* pp. 183–202. https://doi.org/10.1007/BF02289343.

Konopka, T. (2023), *umap: Uniform Manifold Approximation and Projection*. R package version 0.2.10.0. https://CRAN.R-project.org/package=umap.

Krijthe, J. H. (2015), *Rtsne: T-Distributed Stochastic Neighbor Embedding using Barnes-Hut Implementation*. R package version 0.16. https://github.com/jkrijthe/Rtsne.

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

Laa, U., Cook, D. & Lee, S. (2022), 'Burning sage: Reversing the curse of dimensionality in the visualization of high-dimensional data', *J. Comput. Graph. Stat.* **31**(1), 40–49. https://doi.org/10.1080/10618600.2021.1963264.

LeCun, Y. & Cortes, C. (2010), 'Mnist handwritten digit database'. http://yann.lecun.com/exdb/mnist/.

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

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

McInnes, L., Healy, J., Saul, N. & Großberger, L. (2018), 'Umap: Uniform manifold approximation and projection', *Journal of Open Source Software* **3**(29), 861. https://doi.org/10.21105/joss.00861.

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.

Pedersen, T. L. (2024), *patchwork: The Composer of Plots*. R package version 1.2.0. https://CRAN.R-project.org/package=patchwork.

Saeed, N., Nam, H., Haq, M. I. U. & Muhammad Saqib, D. B. (2018), 'A survey on multidimensional scaling', *ACM Comput. Surv.* **51**(3). https://doi.org/10.1145/3178155.

Silva, V. & Tenenbaum, J. (2002), 'Global versus local methods in nonlinear dimensionality reduction', *Advances in Neural Information Processing Systems* **15**.

Ushey, K., Allaire, J. & Tang, Y. (2024), *reticulate: Interface to Python*. R package version 1.38.0. https://CRAN.R-project.org/package=reticulate.

Wang, Y., Huang, H., Rudin, C. & Shaposhnik, Y. (2021), 'Understanding how dimension reduction tools work: An empirical approach to deciphering t-sne, umap, trimap, and pacmap for data visualization', *Journal of Machine Learning Research* **22**(201), 1–73. http://jmlr.org/papers/v22/20-1061.html.

Wickham, H. (2023), *conflicted: An Alternative Conflict Resolution Strategy*. R package version 1.2.0. https://CRAN.R-project.org/package=conflicted.

Wickham, H., Averick, M., Bryan, J., Chang, W., McGowan, L. D., François, R., Grolemund, G., Hayes, A., Henry, L., Hester, J., Kuhn, M., Pedersen, T. L., Miller, E., Bache, S. M., Müller, K., Ooms, J., Robinson, D., Seidel, D. P., Spinu, V., Takahashi, K., Vaughan, D., Wilke, C., Woo, K. & Yutani, H. (2019), 'Welcome to the tidyverse', *Journal of Open Source Software* **4**(43), 1686.

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. 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. http://www.jstatsoft.org/v40/i02/.

Zeileis, A., Fisher, J. C., Hornik, K., Ihaka, R., McWhite, C. D., Murrell, P., Stauffer, R. & Wilke, C. O. (2020), 'colorspace: A toolbox for manipulating and assessing colors and palettes', *Journal of Statistical Software* **96**(1), 1–49.

Zhu, H. (2024), *kableExtra: Construct Complex Table with kable and Pipe Syntax*. R package version 1.4.0. https://CRAN.R-project.org/package=kableExtra.