# 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

June 25, 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. 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, data visualization, tour, statistical graphics, exploratory data analysis, unsupervised learning

# 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).

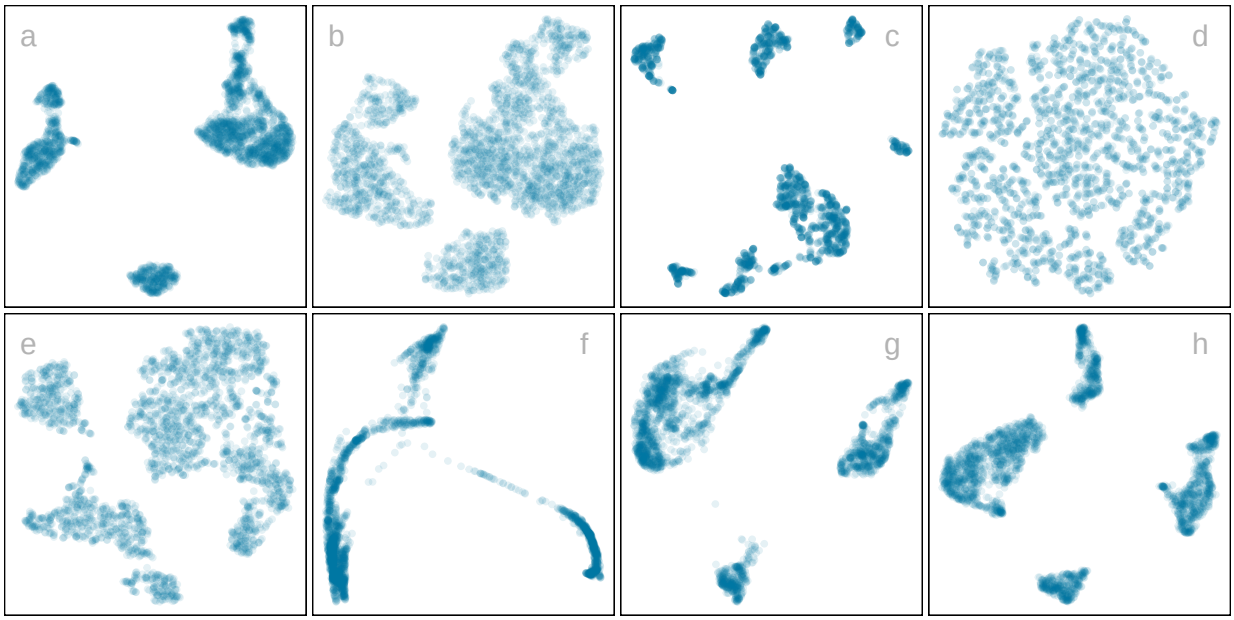


Figure 1: Eight different NLDR representations of the same data. Different methods and different parameter choices are used. Researchers may have seen any of these in their analysis of this data, depending on their choice of method, or typical parameter choice. Would they make different decisions downstream 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. In Section 4, we describe how to assess the best fit and identify the most accurate 2-$D$ layout based on the proposed model diagnostics. Curiosities and unexpected patterns discovered in NLDR results by examining the model in the data space are discussed in Section 5. 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. Links to the langevitour animation videos showing the 2-$D$ projections are provided in Table 1.

## 2 Background

Historically, low-dimensional ($k$-$D$) representations of high-dimensional ($p$-$D$) data have been computed using multidimensional scaling (MDS) (Kruskal 1964), which includes principal components analysis (PCA) (for an overview see Jolliffe (2011)). (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. Both tNSE and UMAP can be considered to produce the $k$-$D$ representation by minimizing the divergence between two distributions, where the distributions are modeling the inter-point distances. PHATE are examples of diffusion processes spreading to capture geometric shapes, that include both global and local structure. (See Coifman et al. (2005) for an explanation of diffusion processes.) TriMAP and PaCMAP

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. These contradictions arise 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 Asimov (1985), broaden the scope by providing movies of linear projections, that provide views the data from all directions. (See Lee et al. (2021) for a review of tour methods.) There are many tour algorithms imple-

mented, 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.

Our proposed solution is to use the tour to examine how the NLDR is warping the space. It 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 model (a $k$-$D$ 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. Typically, $k = 2$, which is used for the rest of this paper. From a statistical perspective we can consider the distances between points in the 2-$D$ 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 $g$: $\mathbb{R}^{n\times p} \rightarrow \mathbb{R}^{n\times 2}$, with hyper-parameters $\boldsymbol{\theta}$. These 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 neighbor 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 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 "nonlinear clusters". It is constructed by simulating two clusters, each consisting of 1000 observations. The C-shaped cluster is generated from $\theta \sim U(\text{-}3\pi/2, 0)$, $X_1 = \sin(\theta)$, $X_2 \sim U(0, 2)$ (adding thickness to the C), $X_3 = \text{sign}(\theta) \times (\cos(\theta) - 1)$, $X_4 = \cos(\theta)$. The other cluster is from $X_1 \sim U(0, 2)$, $X_2 \sim U(0, 3)$, $X_3 = \text{-}(X_1^3 + X_2)$, 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.

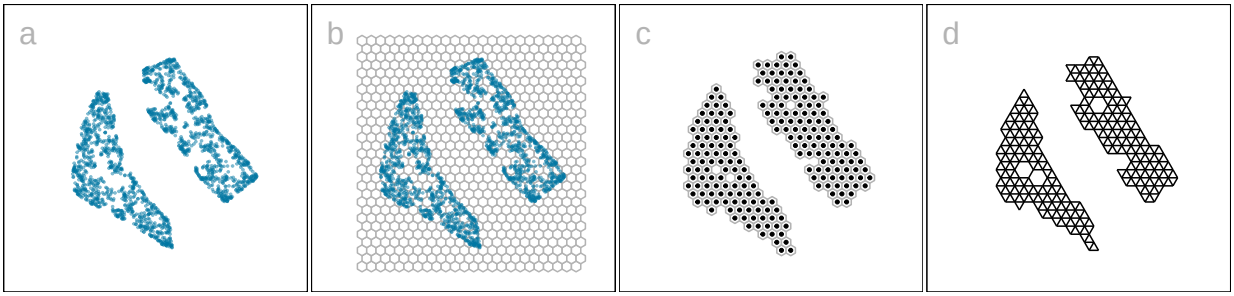


Figure 2: Key steps for constructing the model on the tSNE layout ($k = 2$): (a) data, (b) hexagon bins, (c) bin centroids, and (d) triangulated centroids. The two nonlinear clusters data is shown.

## 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 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,

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

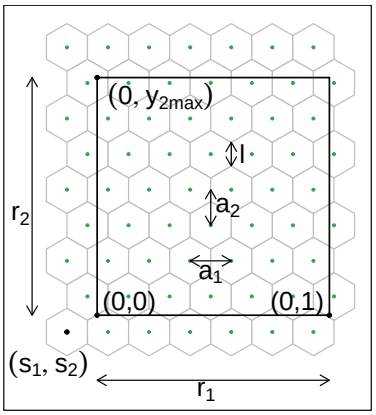


Figure 3: The components of the hexagon grid illustrating notation.

### 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

$$u(i) = \arg\min_{j=1,\ldots,b} \sqrt{(y_{i1} - C_{j1}^{(2)})^2 + (y_{i2} - C_{j2}^{(2)})^2},$$

maps 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 neighboring observations, in both the NLDR layout (Figure 2 (d)) and the $p$-$D$ model representation. When the data has been binned the triangulation connects centroids. The edges preserve the neighborhood information from the 2-$D$ representation when the model is lifted into $p$-$D$.

### 3.3 Rendering the model in $p$-$D$

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

$$C_{hj}^{(p)} = \frac{1}{n_h} \sum_{i=1}^{n_h} x_{hij}, \quad h = 1, \ldots, b; \; j = 1, \ldots, p; \; n_h > 0.$$

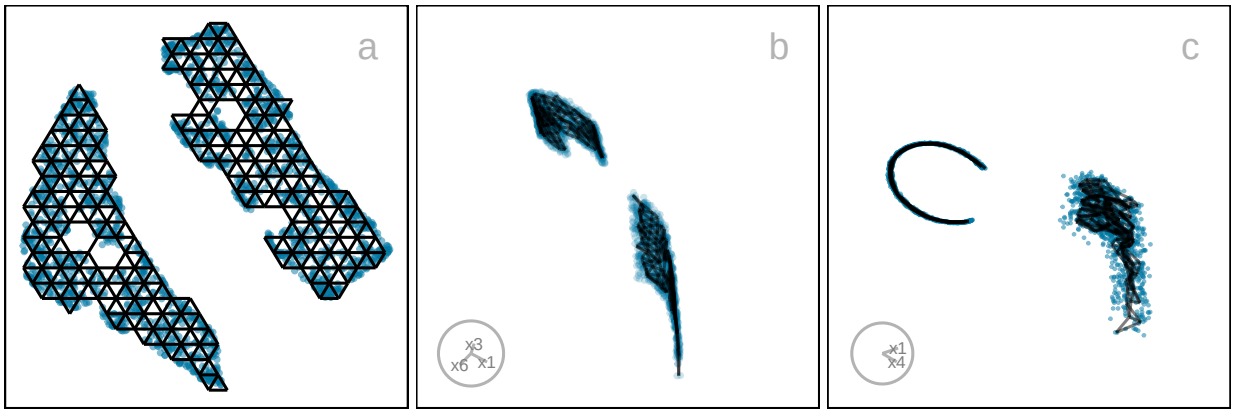


Figure 4: Lifting the 2-$D$ fitted model into $p$-$D$. Two projections of the $p$-$D$ fitted model overlaying the data are shown in b, c. The fit is reasonably tight with the data in one cluster (top one in b), but slightly less so in the other cluster probably because it is in 3-$D$. Notice also that, in the 2-$D$ layout the two clusters have internal gaps which creates a model with some holes. This lacy pattern happens regardless of the hyper-parameter choice, but this doen't severely impact the $p$-$D$ model representation.

### 3.4 Measuring the fit

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

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:

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

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

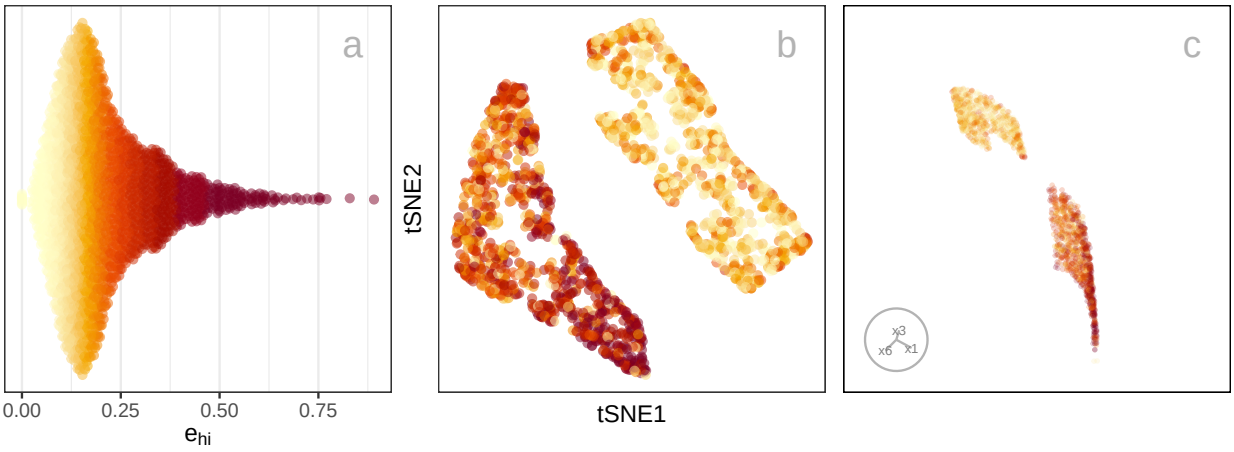


Figure 5: Examining the distribution of residuals in a jittered dotplot (a), 2-$D$ NLDR layout (b) and a tour of 4-$D$ data space (c). Color indicates error ($e_{hi}$), dark color indicating high error and light indicates low error. Most large errors are distributed in one cluster (bottom one in c) and most small errors are distributed in the other cluster.

## 3.5 Prediction into 2-$D$

A new benefit of this fitted model is that it allows us to now predict the NLDR value of a new observation, $x'$, 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-$D$, $C_h^{(2)}$.

## 3.6 Tuning

The model fitting can be adjusted using these parameters:

- hexagon bin parameters
  - bottom left bin position $(s_1,\ s_2)$,
  - the number of bins in the horizontal direction $(b_1)$, which controls the number of bins the vertical direction $(b_2)$, total number of bins $(b)$, and total number of non-empty bins $(m)$.
- bin density cutoff, to possibly remove low-density hexagons.

Default values are provided for each of these, but it is expected that the user will examine the RMSE for a range of choices. Choosing these parameters according to RMSE 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 which is seen in Figure 6. The distribution of bin counts for $s_1$ varying between -0.1 – 0.0. Generally, a more uniform distribution among these possibilities would indicate that the bins are reliably capturing the underlying distribution of observations.

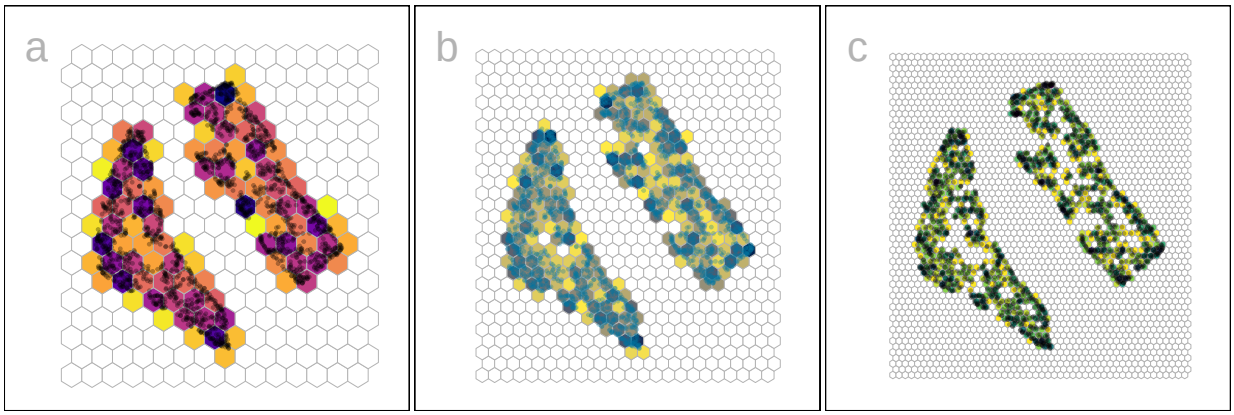


Figure 6: Hexbin density plots of tSNE layout of the nonlinear cluster data, using three different bin specifications $(b_1, b_2, b, m)$: (a) 15, 16, 240, 98, (b) 24, 30, 720, 215, and (c) 48, 52, 2496, 549. Color indicates standardized counts, dark indicating high count and light indicates low count. At the smallest bin size, the data structure is discontinuous, suggesting that there are too many bins. Using the RMSE 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 $b_1, q, r_2$ (Equation 1). Values of $b_1$ between 2 and $b_1 = \sqrt{n/r_2}$ are allowed. Figure 7 (a) shows the effect of different choices of $b_1$ on the RMSE of the fitted model.

### 3.6.2 Handling of low density bins

Standardised bin counts is computed as $w_h = n_h/n, \quad h = 1, ... m$, and $n$ is the number of observations. Density is computed as $d_h = w_h/A$ where $A$ is the area of the hexagon. As a measure of the denseness of a single grid, we can compute the average bin density, $\bar{d_h}$. This can be used to compare grids.

These quantities can be used to assess and compare the models resulting from different binwidths:

- RMSE should decrease when binwidth decreases, as the binning gets closer to observations being in their own bin. But a big drop in RMSE would indicate the lower binwidth is substantially better than the larger one.

- Bins with no observations or a small number might be dropped from the model. This will create the wireframe not extending into sparse areas, or allowing for holes in the data to be better captured. Note that, all observations are used for the RMSE calculation, though. RMSE can be examined relative to dropping a fraction of the low density bins.

- The proportion of non-empty bins is interesting to examine across different binwidths. A good binning should have just the right amount of bins to neatly cover the shape of the data, and no more or less.

- Lastly, an ideal distribution of the density of a grid is uniform, in the sense that if each bin captures a similar number of observations, then it has just the right number

of bins to neatly cover the shape of the data. To examine this, the average bin density is compared against the range of binwidths.

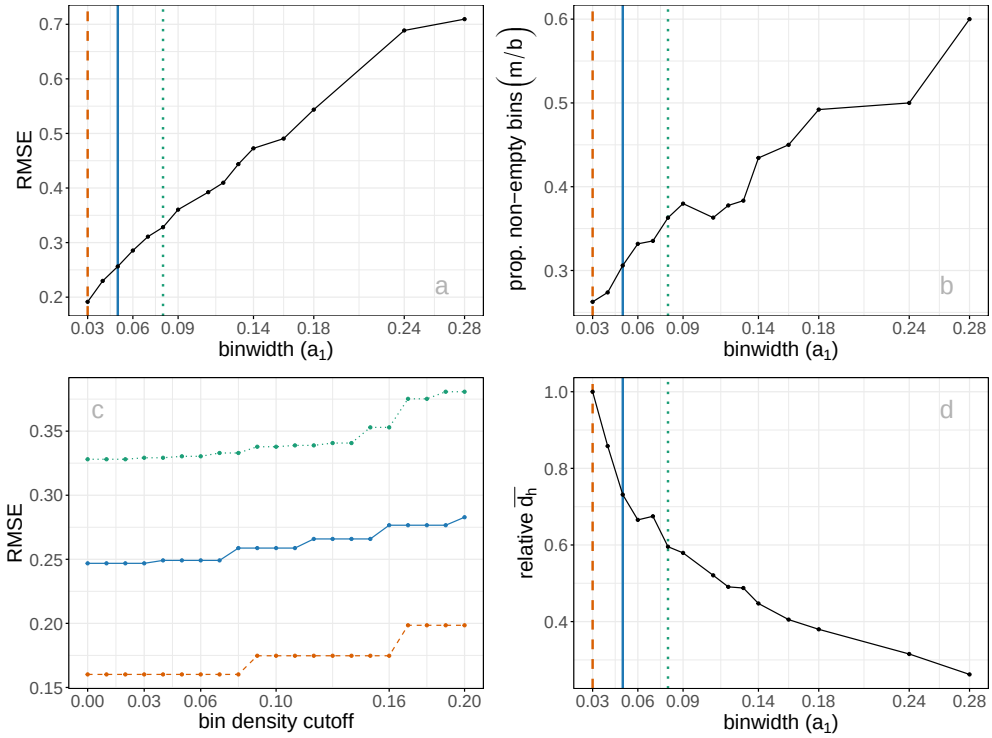


Figure 7: Various plots to help assess best hexagon bin parameters (a, b, d) and thresholds to remove low-density bins (c). Both (a) and (c) show RMSE, against binwidth ($a_1$) and bin density cutoff. A good benchmark value for these parameters is when the RMSE drops and then flattens out. Three binwidth choices were made: 0.03 (orange dashed), 0.05 (blue solid), and 0.08 (green dotted) to investigate. As the binwidth increases, the proportion of non-empty bins also increases (b). The relative $\bar{d}_h$ 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 Assessing the best fit, and hence most accurate 2-$D$ layout

Figure 8 illustrates the approach to compare the fits for different representations and assess the strength of any fit. What does it mean to be a best fit for this problem? Analysts use an NLDR layout to display the structure present in high-dimensional data in a convenient 2-$D$ display. It is a competitor to linear dimension reduction that can better represent nonlinear associations such as clusters. However, these methods can hallucinate, suggesting patterns that don't exist, and grossly exaggerate other patterns. Having a layout that best fits the high-dimensional structure is desirable but more important is to identify bad representations so they can be avoided. The goal is to help users decide on a the most useful and appropriate low-dimensional representation of the high-dimensional data.

A particular pattern that we commonly see is that analysts tend to pick layouts with clusters that have big separations between them. When you examine their data in a tour, it is almost always that we see there are no big separations, and actually often the suggested

clusters are not even present. While we don't expect that analysts include animated gifs of tours in their papers, we should expect that any 2-$D$ representation adequately indicates the clustering that is present, and honestly show lack of separation or lack of clustering when it doesn't exist. It is important for analysts to have tools to select the accurate representation not the pretty but wrong representation.

To compare and assess a range of representations an analyst needs:

- a selection of NLDR representations made with a range of parameter choices and possibly different methods (tSNE, UMAP, …).
- a range of model fits made by varying bin size and low density bin removal.
- calculated RMSE for each layout, when it is transformed into the $p$-$D$ space.

Comparing the RMSE to obtain the best fit is appropraite if the same NLDR method is used. However, because the RMSE is computed on $p$-$D$ data it measures the fit between model and data so it can also be used to compare the fit of different NLDR methods. A lower RMSE indicates a better NLDR representation.

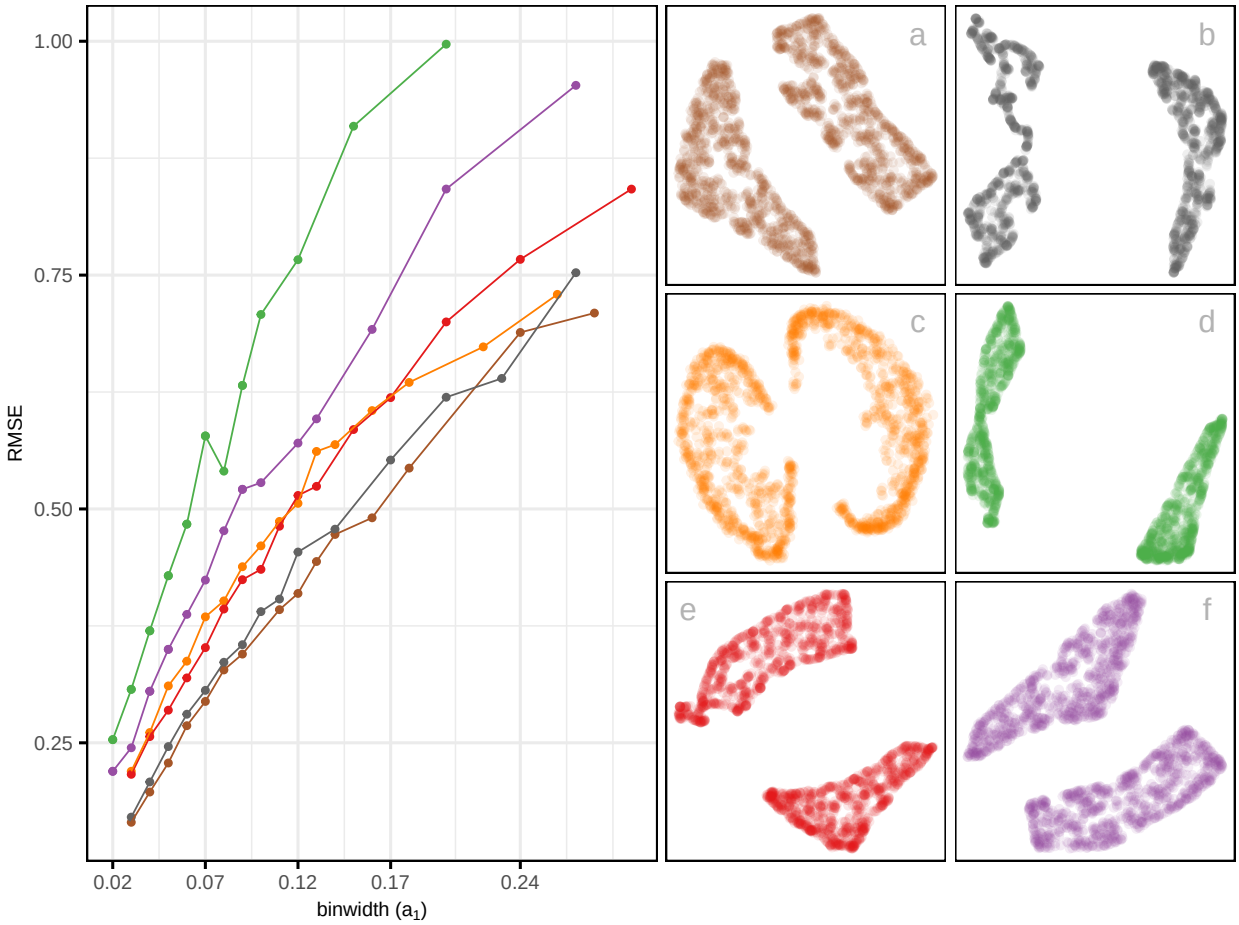


Figure 8: Assessing which of the 6 NLDR layouts on the two nonlinear clusters data is the better representation using RMSE for varying binwidth ($a_1$). Color used for the lines and points in the left plot and in the scatterplots represents NLDR layout (a-f). Layout d is universally poor. Layouts a, 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 about NLDR results discovered by examining the model in the data space

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

## 5.1 Some methods appear to order points in the layout

When examining the 2-$D$ model representations it appears to be flatter or like a pancake with some methods, especially PaCMAP, when the data is structure is higher than two dimensional. A simple example of this can be seen with data simulated to contain five 4-$D$ Gaussian clusters. Each cluster is essentially a ball in 4-$D$, so there is no 2-$D$ representation, rather the model in each cluster should resemble a crumpled sheet of paper that fills out 4-$D$.

Figure 9 a1, b1, c1 show the 2-$D$ layouts for (a) tSNE, (b) UMAP, and (c) PaCMAP, respectively. The default hyper-parameters for each method are used. In each layout we can see an accurate representation where all five clusters are visible, although with varying degrees of separation.

The models are fitted to each these layouts. Figure 9 a2, b2, c2 show the fitted models in a projection of the 4-$D$ space, taken from a tour. These clusters are fully 4-$D$ in nature, so we would expect the model to be a *crumpled sheet* that stretches in all four dimensions. This is what is mostly observed for tSNE and UMAP. The curious detail is that the model for PaCMAP is closer to a *pancake* in shape in every cluster! This single projection only shows this in three of the five clusters but if we examine a different projection the other clusters exhibit the pancake also. While we don't know what exactly causes this, it is likely due to some ordering of points in the 2-$D$ PaCMAP layout that induces the flat model. One could imagine that if the method used principal components on all the data, that it might induce some ordering that would produce the flat model. If this were the reason, the pancaking would be the same in all clusters, but it is not: The pancake is visible in some clusters

in some projections but in other clusters it is visible in different projections. It might be due to some ordering by nearest neighbors in a cluster. The PaCMAP documentation doesn't provide any helpful clues. That this happens, though, makes the PaCMAP layout inadequate for representing the high-dimensional data.

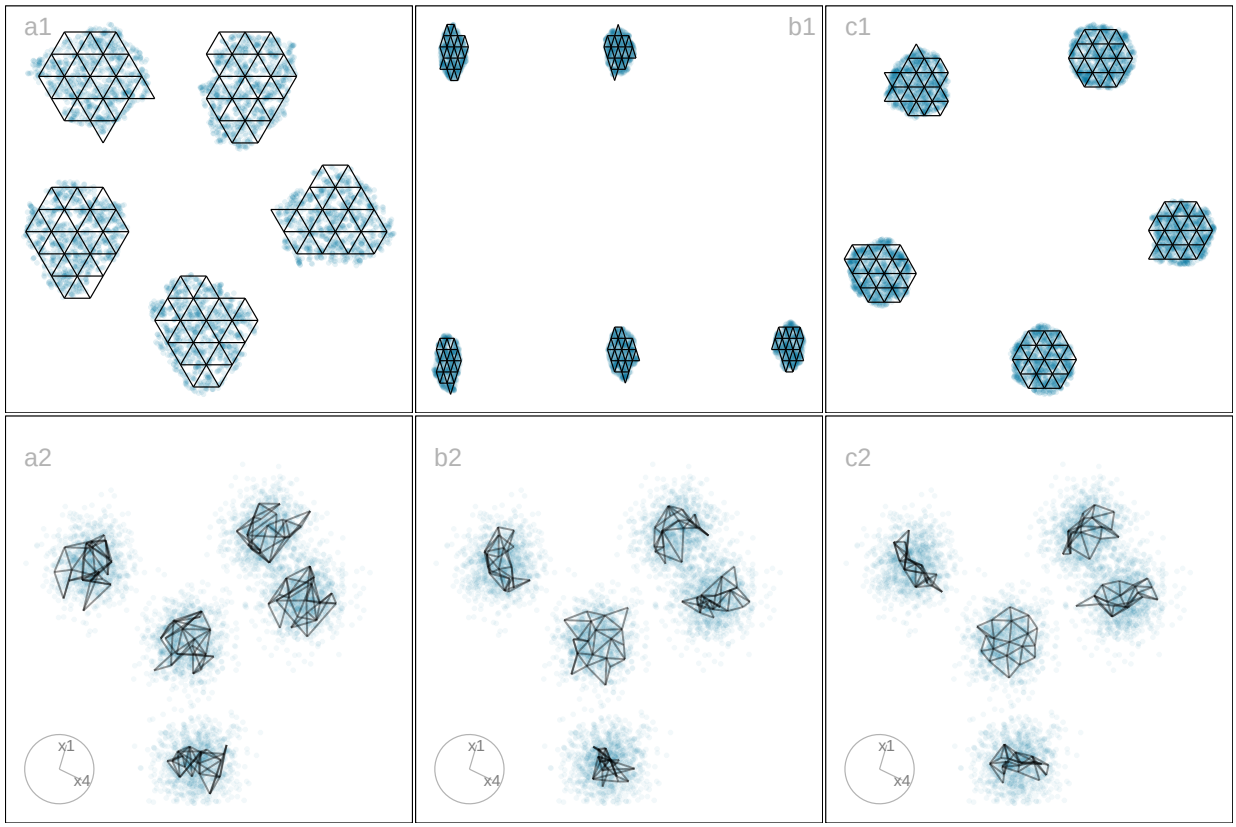


Figure 9: NLDR's organise points in the 2-$D$ layout in different ways, possibly misleadingly, illustrated using three layouts: (a) tSNE, (b) UMAP, (c) PaCMAP. The data has five Gaussian clusters in 4-$D$. The bottom row of plots shows a 2-$D$ projection from a tour on 4-$D$ revealing the differences 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.

## 5.2 Low density results in a squishing of the 2-$D$ representation in all methods

Differences in density can arise by sampling at different rates in different subspaces of $p$-$D$. For example, the data shown in Figure 10 all lies on a 2-$D$ curved sheet in 4-$D$, but one end of the sheet is sampled densely and the other very sparsely. It was simulated to illustrate the effect of the density difference on layout generated by an NLDR, illustrated using the tSNE results.

Figure 10 (a2, b2) shows a 2-$D$ layout for tSNE created using the default hyper-parameters. One would expect to see a rectangular shape if the curved sheet is flattened, but the layout is triangular. The other two displays show the residuals as a dot density plot (a1, b1), and a 2-$D$ projection of the data and the model from 4-$D$ (a3, b3). Using linked brushing between the plots, we can highlight points in the tSNE layout, and examine where they fall in the original 4-$D$. The darker (maroon) points indicate points that have been highlighted by linking. In row a, the points at the top of the triangle are highlighted, and we can see these correspond to higher residuals, and also to all points at the low density end of the curved sheet. In row b, points at the lower left side of the triangle are highlighted which corresponds to smaller residuals and one corner of the sheet at the high density end of the curved sheet.

The tSNE behaviour is to squeeze the low density area of the data together into the layout. This is common in other NLDR methods also, which means analysts need to be aware that if their data is not sampled relatively uniformly, apparent closeness in the 2-$D$ may correspond to sparseness in $p$-$D$.

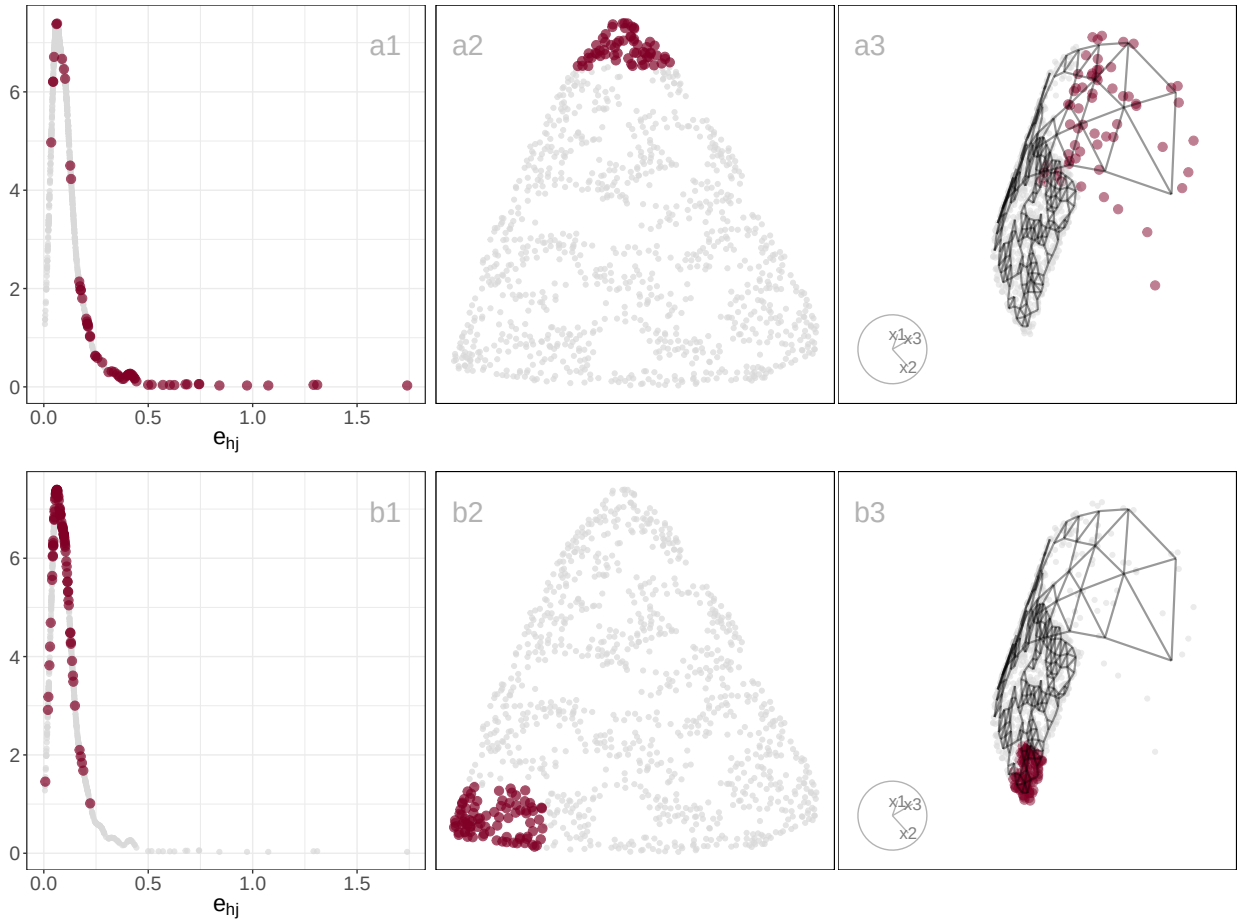


Figure 10: Exploring the effect of density on the NLDR layout using a C-shaped structure with different density at each end. The top part of the triangular shape, shown with linked brushing, identify high model error corresponding to the sparse end of the structure (a1-a3). On the other hand, one of the dense corners of the structure shows low model error and is highlighted in the right part of the triangular shape (b1-b3). The tSNE layouts (a2, b2) are colored according to selected points. a2, a3 and b2, b3 present the results from linked plots that brushed the 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.

# 6 Applications

To illustrate the approach we use two examples: PBMC3k data (single cell gene expression) where an NLDR layout is used to represent cluster structure present in the $p$-$D$ data, and MNIST hand-written digits where NLDR is used to represent essentially a low-dimensional nonlinear manifold in $p$-$D$.

## 6.1 PBMC3k

This is a benchmark single-cell RNA-Seq data set collected on Human Peripheral Blood Mononuclear Cells (PBMC3k) as used in 10x Genomics (2016). Single-cell data measures the gene expression of individual cells in a sample of tissue (see for example, Haque et al. (2017)). This type of data is used to obtain an understanding of cellular level behavior and heterogeneity in their activity. Clustering of single-cell data is used to identify groups of cells with similar expression profiles. NLDR is often used to summarize the cluster structure. Usually, NLDR does not use the cluster labels to compute the layout, but uses color to represent the cluster labels when it is plotted.

In this data there are 2622 single cells and 1000 gene expressions (variables). Following their pre-processing, different NLDR techniques were performed on the first nine principal components. Figure 1 shows this data using a variety of methods, and different hyper-parameters. You can see that the result is wildly different depending on the choices. Layout a is a reproduction of the layout published in Chen et al. (2024). This layout suggests that the data has three very well separated clusters, each with an odd shape. The question is whether this accurately represents the cluster structure in the data, or whether they should have chosen b or c or d or e or f or g or h. This is what our new method can help with – to decide which is the more accurate 2-$D$ representation of the cluster structure in the $p$-$D$ data.

Figure 11 shows RMSE across a range of binwidths ($a_1$) for each of the layouts in Figure 1. The layouts were made using tSNE, UMAP, PHATE, PaCMAP, and TriMAP with various hyper-parameter settings. Lines are color coded to match the color of the layouts shown on

the right. Lower RMSE indicates the better fit. Using a range of binwidths shows how the model changes, with possibly the best model being one that is universally low RMSE across all binwidths. It can be seen that layout f is sub-optimal with universally higher RMSE. Layout a, the published one, is better but it is not as good as layouts b, d, or e. With some imagination layout d perhaps shows three barely distinguishable clusters. Layout e shows three, possibly four, clusters that are more separated. The choice reduces from eight to these two. Layout d has slightly better RMSE when the $a_1$ is small, but layout e beats it at larger values. Thus we could argue that layout e is the most accurate representation of the cluster structure, of these eight.

To further assess the choices, we need to look at the model in the data space, by using a tour to show the wireframe model overlaid on the data in the 9-$D$ space (Figure 12). Here we compare the published layout (a) versus what we argue is the best layout (e). The top row (a1, a2, a3) correspond to the published layout and the bottom row (e1, e2, e3) correspond to the optimal choice according to our procedure. The middle and right plots show two projections. The primary difference between the two models is that, the model of layout e does not fill out to the extent of the data but concentrates in the center of each point cloud. Both suggest that three clusters is a reasonable interpretation of the structure, but layout e more accurately reflects the separation between them, which is small.

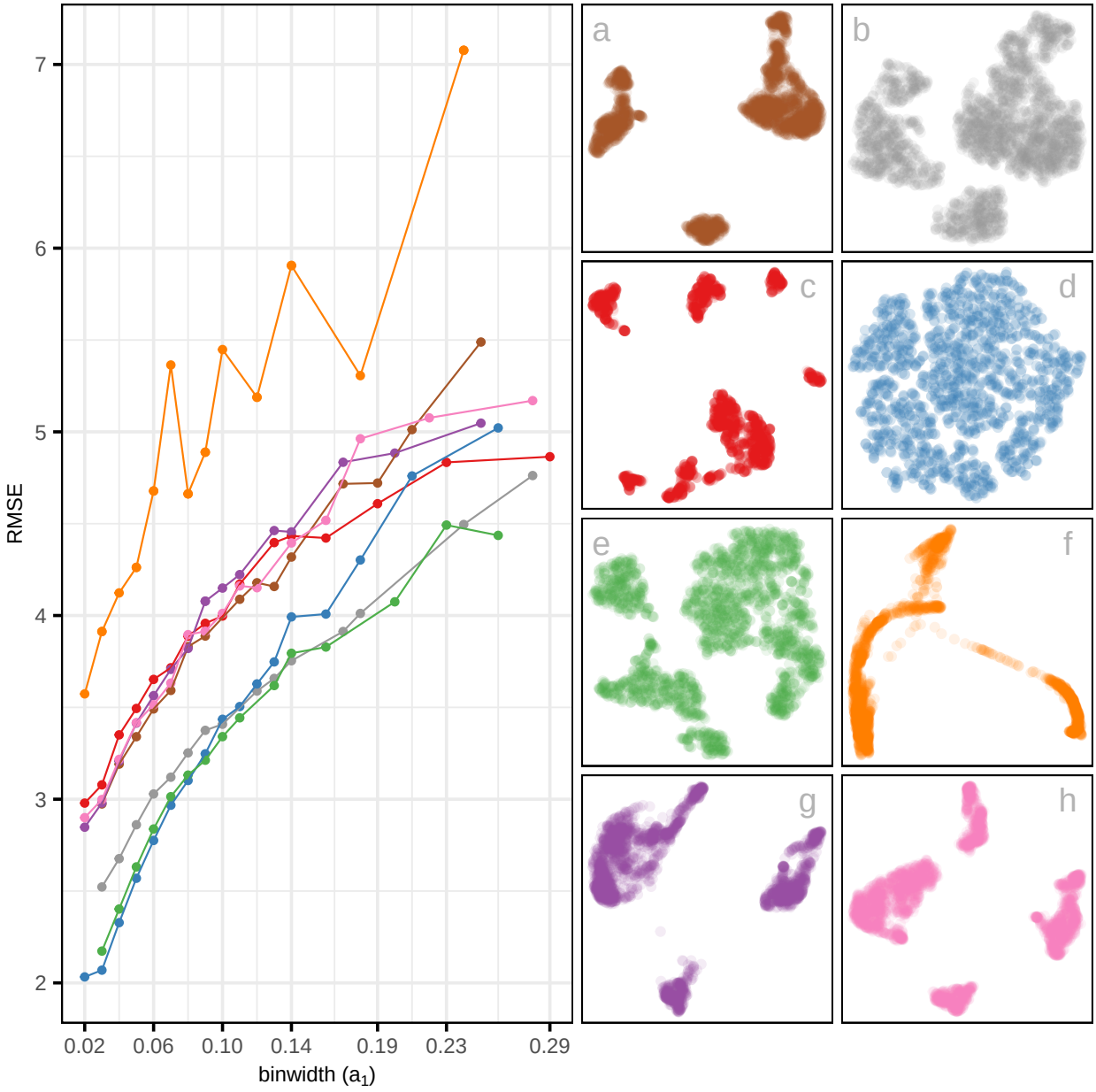


Figure 11: Assessing which of the 8 NLDR layouts on the PBMC3k data (shown in Figure 1) is the better representation using RMSE for varying binwidth ($a_1$). Color used for the lines and points in the left plot and in the scatterplots represents NLDR layout (a-h). Layout f is universally poor. Layouts 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.

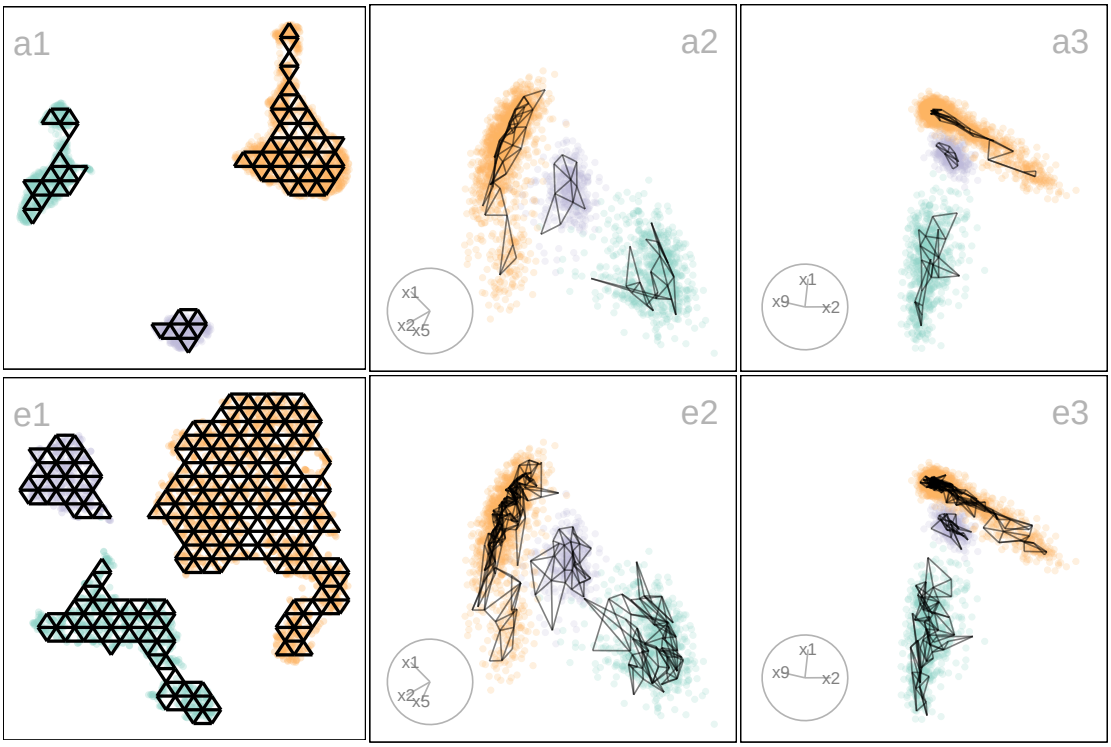


Figure 12: Compare the published 2-$D$ layout (a) made with UMAP and the 2-$D$ layout selected by RMSE plot (e) made by tSNE. The two plots on the right show projections from a tour, with the models overlaid. The published layout suggested three very separated clusters, but this is not present in the data. While there may be three clusters they are not well-separated. The difference in model fit also indicates this: the published layout a does not spread out fully into the point cloud like the model generated from layout e. This supports the choice that layout e is the better representation of the data, because it does not exaggerate separation between clusters.

## 6.2 MNIST hand-written digits

The digit "1" of the MNIST dataset (LeCun et al. 1998) consists of 7877 grayscale images of handwritten "1"s. Each image is $28 \times 28$ pixels which corresponds to 784 variables. The first 10 principal components, explaining 83% of the total variation, are used. This data is essentially lies on a nonlinear manifold in the high dimensions, defined by the shapes that "1"s make in when sketched, meaning that some pixels are almost always dark and some almost always light. We expect that the best layout captures this type of structure and

does not exhibit distinct clusters.

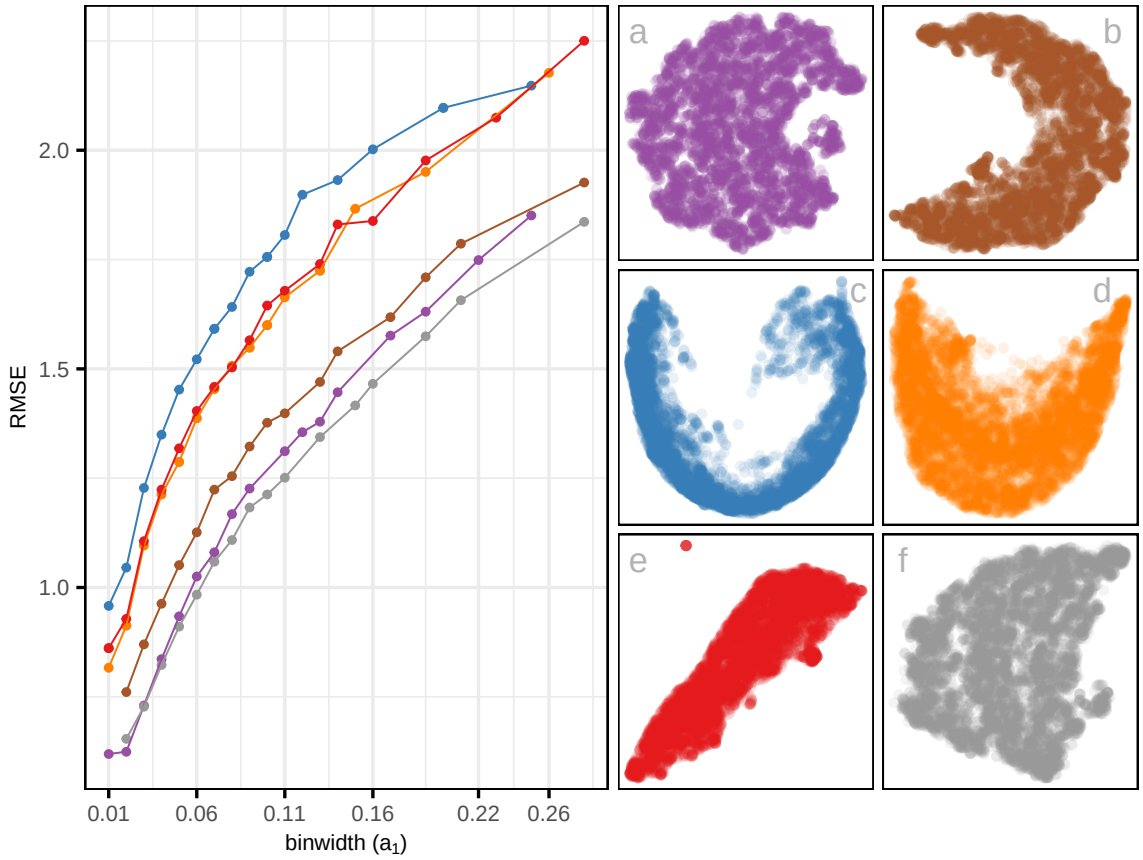


Figure 13: Assessing which of the 6 NLDR layouts of the MNIST digit 1 data is the better representation using RMSE for varying binwidth ($a_1$). Colour is used for the lines and points in the left plot to match the scatterplots of the NLDR layouts (a-f). Layout c is universally poor. Layouts a, f that show a big cluster and a small circular cluster are universally optimal. Layout a performs well at tiny binwidth (where most points are in their own bin) and not as well as f with larger binwidth, thus layout f is the best choice.

Figure 13 compares the fit of six layouts computed using UMAP (b), PHATE (c), TriMAP (d), PaCMAP (e) with default hyper-parameter setting and two tSNE runs, one with default hyper-parameter setting (a) and the other changing perplexity to 89 (f). The layouts are reasonably similar in that they all have the observations in a single blob. Some (b, c) have a more curved shape than others. Layout e is the most different having a linear shape, and a single very large outlier. Both a and f have a small clump of points perhaps slightly disconnected from the other points, in the lower to middle right.

The layout plots are colored to match the lines in the RMSE vs binwidth ($a_1$) plot. Layouts a, b and f fit the data better than c, d, e, and layout f appears to be the bets fit. Figure 14 shows this model in the data space in two projections from a tour. The data is curved in the 10-$D$ space, and the fitted model captures this curve. The small clump of points in the 2-$D$ layout is highlighted in both displays. These are almost all inside the curve of the bulk of points and are sparsely located. The fact that they are packed together in the 2-$D$ layout is likely due to the handling of density differences by the NLDR.

The next step is to investigate the 2-$D$ layout to understand what information is learned from this representation. Figure 15 summarizes this investigation. Plot a shows the layout with points colored by their residual value - darker color indicates larger residual and poor fit. The plots b, c, d, e show samples of hand-written digits taken from inside the colored boxes. Going from top to bottom around the curve shape we can see that the "1"s are drawn with from right slant to a left slant. The "1"s in d (black box) tend to have the extra up stroke but are quite varied in appearance. The "1"s shown in the plots labelled e correspond to points with big residuals. They can be seen to be more strangely drawn than the others. Overall, this 2-$D$ layout shows a useful way to summarize the variation in way "1"s are drawn.

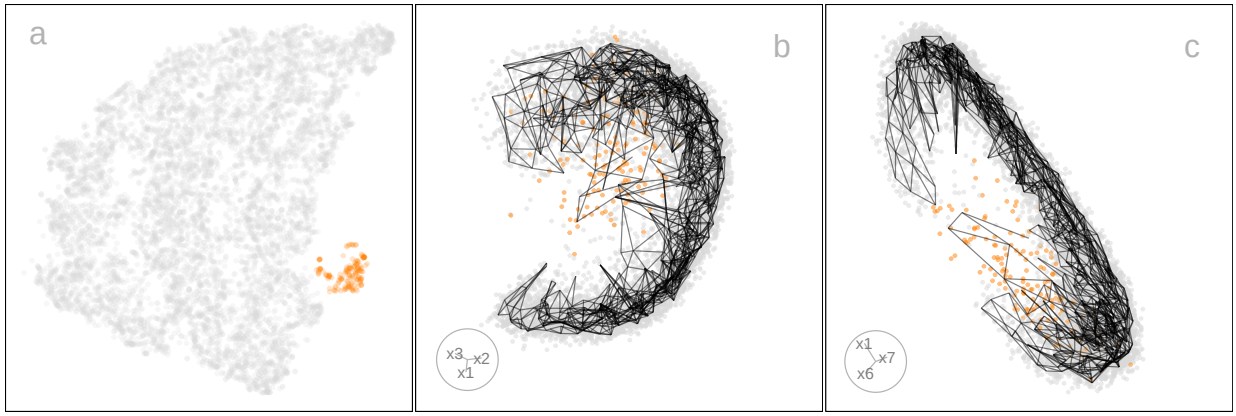


Figure 14: The tSNE layout of the MNIST digit 1 data shows a big nonlinear cluster (grey) and a small cluster (orange) located very close to the one corner of the big cluster in 2-$D$ (a). The MNIST digit 1 data has a nonlinear structure in 10-$D$. Two 2-$D$ projections from a tour on 10-$D$ reveal that the closeness of the clusters in 10-$D$ and the twisted pattern of the model fit with tSNE. The brushing feature in the linked plots helps in visualizing the closeness of the small cluster to the big cluster.

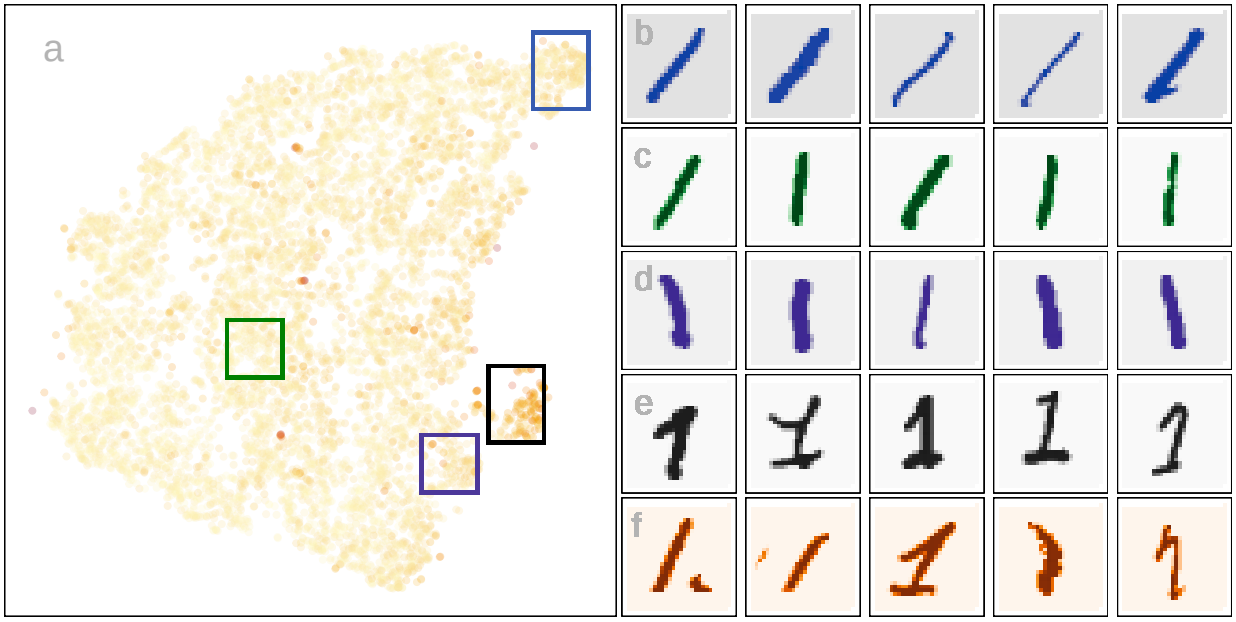


Figure 15: The 10-$D$ model error in 2-$D$ layout of the MNIST digit 1 data shows a pattern. Most low model errors are distributed along the big nonlinear cluster, while most large model errors are distributed along the small cluster. The images associated with large model errors shows different patterns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

# 7 Discussion

We have developed an approach to help assess and compare NLDR layouts, generated by different methods and hyper-parameter choice(s). It depends on conceptualizing the 2-$D$ layout as a model, allowing for the creation of a wireframe representation of the model that can be lifted into $p$-$D$. The fit is assessed by viewing the model in the data space, computing residuals and RMSE. Different layouts can be compared using the RMSE, and provides quantitative and objective methods for deciding on the most suitable NLDR layout to represent the $p$-$D$ data. It also provides a way to predict the values of new $p$-$D$ observations in the 2-$D$, which could be useful for implementing uncertainty checks such as using training and testing samples.

Two examples illustrating usage are provided: the PBMC3k data where the NLDR is summarizing clustering in $p$-$D$ and hand-written digits illustrating how NLDR represents an intrinsically lower dimensional nonlinear manifold. We demonstrated that the published layout of the PBMC3k is inaccurate, because it grossly exaggerates separation between clusters, and even suggests separation when there is none. This is common when layouts are chosen subjectively – often a preference for the "prettiest". Our approach provides a way to objectively choose the layout and hopefully avoids the use of misleading layouts in the future. In the hand-written digits we illustrate how our model fit statistics show that a flat disc layout is superior to the curved shaped layouts, and how to identify oddly written "1"s using the residuals of the fitted model.

Additional exploration of metrics to summarize the fit could be a new direction for the work. The difficulty is capturing nonlinear fits, for which Euclidean distance can be sub-optimal.

We have used a very simple approach based on clustering methods, Euclidean distances to nearest centroid, which can approximate nonlinear patterns. Other cluster metrics would be natural choices to explore.

This new method also reveals some interesting curiosities about NLDR procedures. The fitted model appears as a "pancake" in some data where clusters are regularly shaped and high-dimensional, for some methods but not others, which is odd. One can imagine that if algorithms are initiated using principal components then some ordering of points along the major axes might generate this pattern. Alternatively, if local distances dominate the algorithm then is might be possible to see this pattern with well-separated regular clusters. We also demonstrated that there is a tendency for NLDR algorithms to be confused by different density in the data space, and some patterns in the layout are due to density differences rather than nonlinear associations between variables.

Most NLDR methods only provide a 2-$D$ but if a $k$-$D$ ($k > 2$) layout is provided the approach developed here could be extended. Binning into cubes could be done in 3-$D$ or higher, relatively easily, and used as a the basis for a wireframe of the fitted model. Barber et al. (1996) and the software Laurent (2023) have algorithms for convex hulls, which $p$-$D$ which serve as an inspiration. A simpler approach using $k$-means clustering to provide centroids could also be possible, but the complication would be to determine how to connect the centroids into an appropriate wireframe.

The new methodology is accompanied by an R package called quollr, so that it is readily usable and broadly accessible. The package has methods to fit the model, compute diagnostics and also visualize the results, with interactivity. We have primarily used the langevitour software (Harrison 2023) to view the model in the data space, but other tour

software such as tourr (Wickham et al. 2011) and detourr (Hart & Wang 2022) could be also used.

# 8 Supplementary Materials

All the materials to reproduce the paper can be found at https://github.com/JayaniLakshika/paper-nldr-vis-algorithm. The Appendix includes more details about the hexagonal binning algorithm and a comparison to the results of the newly reported scDEED (Xia et al. 2023) statistic.

The R package `quollr`, available on CRAN and at https://jayanilakshika.github.io/quollr/, provides software accompaying this paper to fit the wireframe model representation, compute diagnostics, visualize the model in the data with langevitour and link multiple plots interactively. Direct links to videos for viewing online are available in Table 1.

| Figure | URL |
|---|---|
| 4 | youtu.be/AlLbrUOJ_1E |
| 5 | youtu.be/KmZdDtEMmUY |
| 9 | youtu.be/I-kxCwVfqiQ, youtu.be/gD1P01FUPyU, youtu.be/MxJ_srOFQNk |
| 10 | youtu.be/-KsQHOrII2A |
| 12 | youtu.be/3VfK3M2gnZM, youtu.be/Es84bwQcndU |
| 14 | youtu.be/sUcGd57Swdg, youtu.be/QiklCjELUxo |

Table 1: Videos of the langevitour animations and the linked plots.

# 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).

# References


10x Genomics (2016), '3k PBMCs from a Healthy Donor, Universal 3' Gene Expression dataset analysed using Cell Ranger 1.1.0'. Accessed: 2025-06-09. https://www.10xgenomics.com/datasets/3-k-pbm-cs-from-a-healthy-donor-1-standard-1-1-0.

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 preprint arXiv:1910.00204* .

Asimov, D. (1985), 'The Grand Tour: A Tool for Viewing Multidimensional Data', *SIAM Journal of Scientific and Statistical Computing* **6**(1), 128–143.

Barber, C. B., Dobkin, D. P. & Huhdanpaa, H. (1996), 'The Quickhull Algorithm for Convex Hulls', *ACM Trans. Math. Softw.* **22**(4), 469—483. https://doi.org/10.1145/235815.235821.

Borg, I. & Groenen, P. J. F. (2005), *Modern Multidimensional Scaling*, 2nd edn, Springer, New York, NY. https://doi.org/10.1007/0-387-28981-X.

Brown, T. A. (2015), *Confirmatory Factor Analysis for Applied Research*, 2nd edn, The Guilford Press, New York, NY, US.

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., Lewin-Koh, N., Maechler, M. & Sarkar, D. (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. R., Lafon, S., Lee, A. B., Maggioni, M., Nadler, B., Warner, F. & Zucker, S. W. (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**(21), 7426–7431. https://www.pnas.org/doi/abs/10.1073/pnas.0500334102.

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), 75. https://doi.org/10.1186/s13073-017-0467-4.

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* **34**(2), 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. https://doi.org/10.1007/BF02289694.

Laa, U., Cook, D. & Lee, S. (2022), 'Burning Sage: Reversing the Curse of Dimensionality in the Visualization of High-Dimensional Data', *Journal of Computational and Graphical Statistics* **31**(1), 40–49. https://doi.org/10.1080/10618600.2021.1963264.

Laurent, S. (2023), *cxhull: Convex Hull*. R package version 0.7.4. https://github.com/stla/cxhull.

LeCun, Y., Cortes, C. & Burges, C. J. C. (1998), 'The MNIST Database of Handwritten Digits'. Accessed: 2025-06-09. 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', *arXiv preprint arXiv:2104.08016* .

Maaten, L. V. D. & Hinton, G. E. (2008), 'Visualizing Data Using t-SNE', *Journal of Machine Learning Research* **9**(11), 2579–2605. https://api.semanticscholar.org/CorpusID:5855042.

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. https://doi.org/10.1038/s41587-019-0336-3.

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**, 721–728. https://proceedings.neurips.cc/paper_files/paper/2002/file/5d6646aad9bcc0be55b2c82f69750387-Paper.pdf.

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. https://doi.org/10.21105/joss.01686.

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/.

Xia, L., Lee, C. & Li, J. J. (2023), 'scDEED: A Statistical Method for Detecting Dubious 2D Single-cell Embeddings', *bioRxiv* . https://www.biorxiv.org/content/early/2023/04/25/2023.04.21.537839.

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. https://www.jstatsoft.org/index.php/jss/article/view/v096i01.

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.