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

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

January 8, 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) (van der Maaten & Hinton 2008), uniform manifold approximation and projection (UMAP) (McInnes & Healy 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 2022), 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). 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.

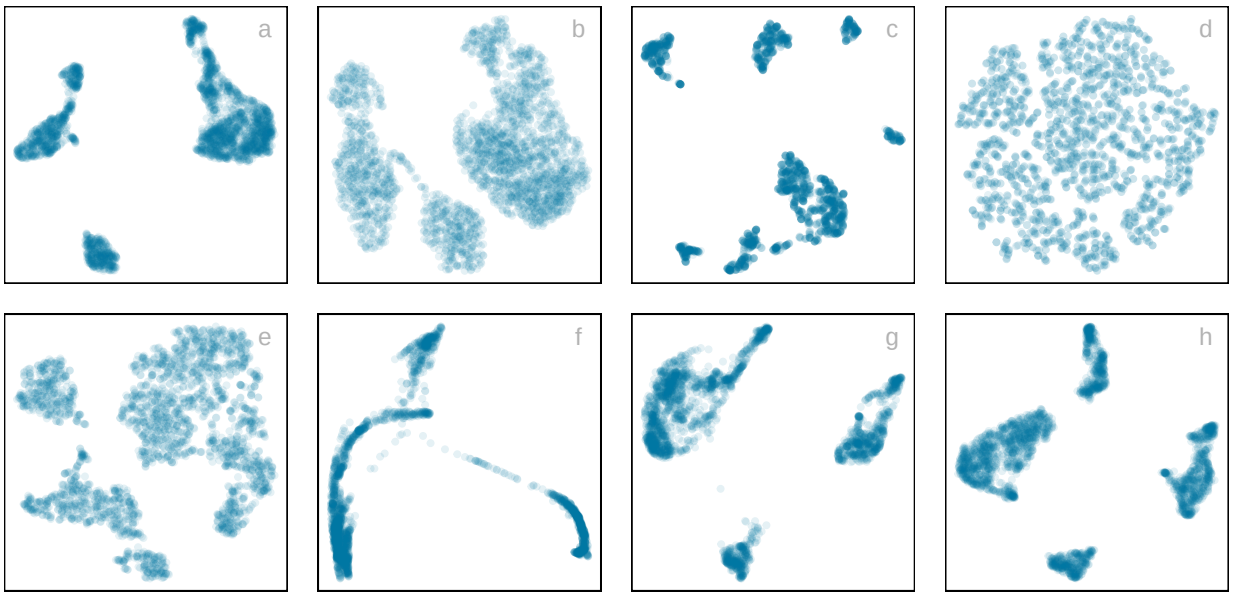


Figure 1: Eight different NLDR representations of the same data. Different techniques and different parameter choices are used. Researchers may have seen any of these in their analysis of this data, depending on their choice of method, or typical parameter choice. Would they make different decisions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

The 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 7. Limitations and future directions are provided in Section 8.

2 Background

Historically, k-D representations of p-D data have been computed using multidimensional scaling (MDS) (Borg & Groenen 2005), which includes principal components analysis (PCA) (Jolliffe 2011) as a special case. 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 multidimensional scaling (MDS) (Borg & Groenen 2005). (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 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 3, and documented below.

To illustrate the method, we use 7-D simulated data, which we call the "S-curve". It is constructed by simulating $n = 750$ observations from $\theta \sim U(-3\pi/2, 3\pi/2)$, $X_1 = \sin(\theta)$, $X_2 \sim U(0, 2)$ (adding thickness to the S), $X_3 = \text{sign}(\theta) \times (\cos(\theta) - 1)$. The remaining variables X_4, X_5, X_6, X_7 are all uniform error, with small variance. We would consider $T = (X_1, X_2, X_3)$ to be the geometric structure (true model) that we hope to capture.

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.

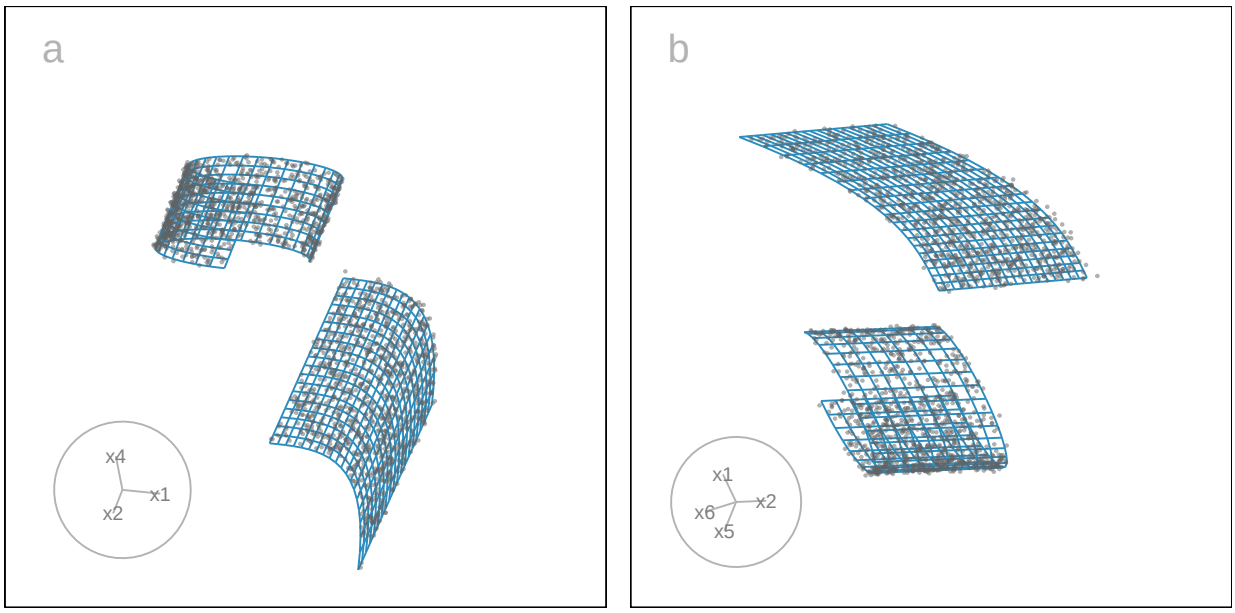


Figure 2: Two views of the true model (blue lines) in 2-D projections from 7-D, for the two C-shaped clusters data (grey points). In one C-shaped cluster, the data is spread throughout the cluster, while in the other C-shaped cluster, the data is concentrated more in one corner of the cluster. The **langevitour** software is used to view the data with a tour, and the full video is available at <>.

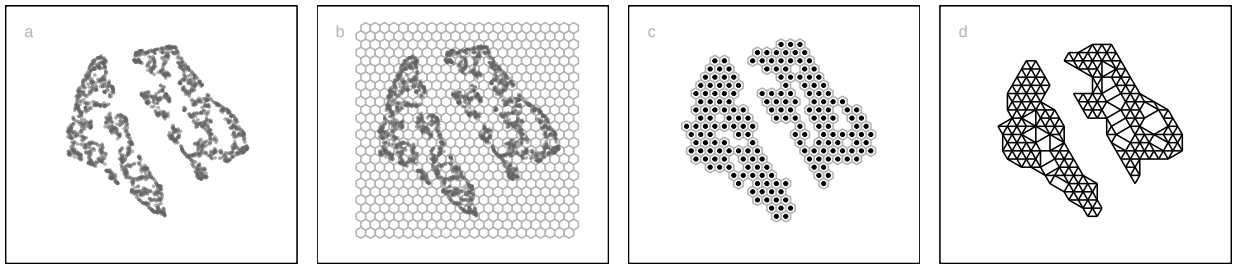


Figure 3: Key steps for constructing the model on the UMAP layout ($k = 2$): (a) data, (b) hexagon bins, (c) bin centroids, and (d) triangulated centroids. The two C-shaped clusters data is shown.

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 $y_{2,\max} = \frac{r_2}{r_1}$ (Figure 3). 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 4 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, $a_1 > \frac{1+2q}{b_1-1}$. Similarly, for the second NLDR component, $a_2 > \frac{r_2+q(1+r_2)}{(b_2-1)}$. Since $a_2 = \frac{\sqrt{(3)}}{2}a_1$ 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, $b_2 = \left\lceil 1 + \frac{2[r_2+q(1+r_2)](b_1-1)}{\sqrt{3}(1+2q)} \right\rceil$.

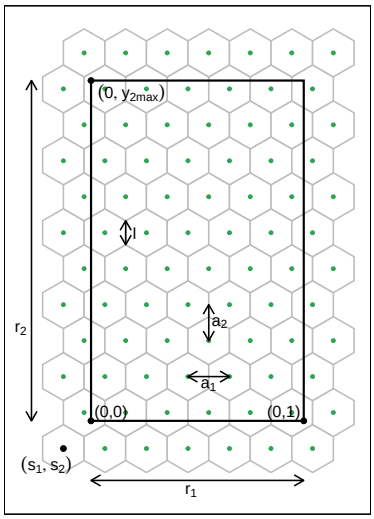


Figure 4: The components of the hexagon grid illustrating notation.

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 $u : \mathbb{R}^{n\times 2} \to \mathbb{R}^{m\times 2}$, 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 3 (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 3 (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.

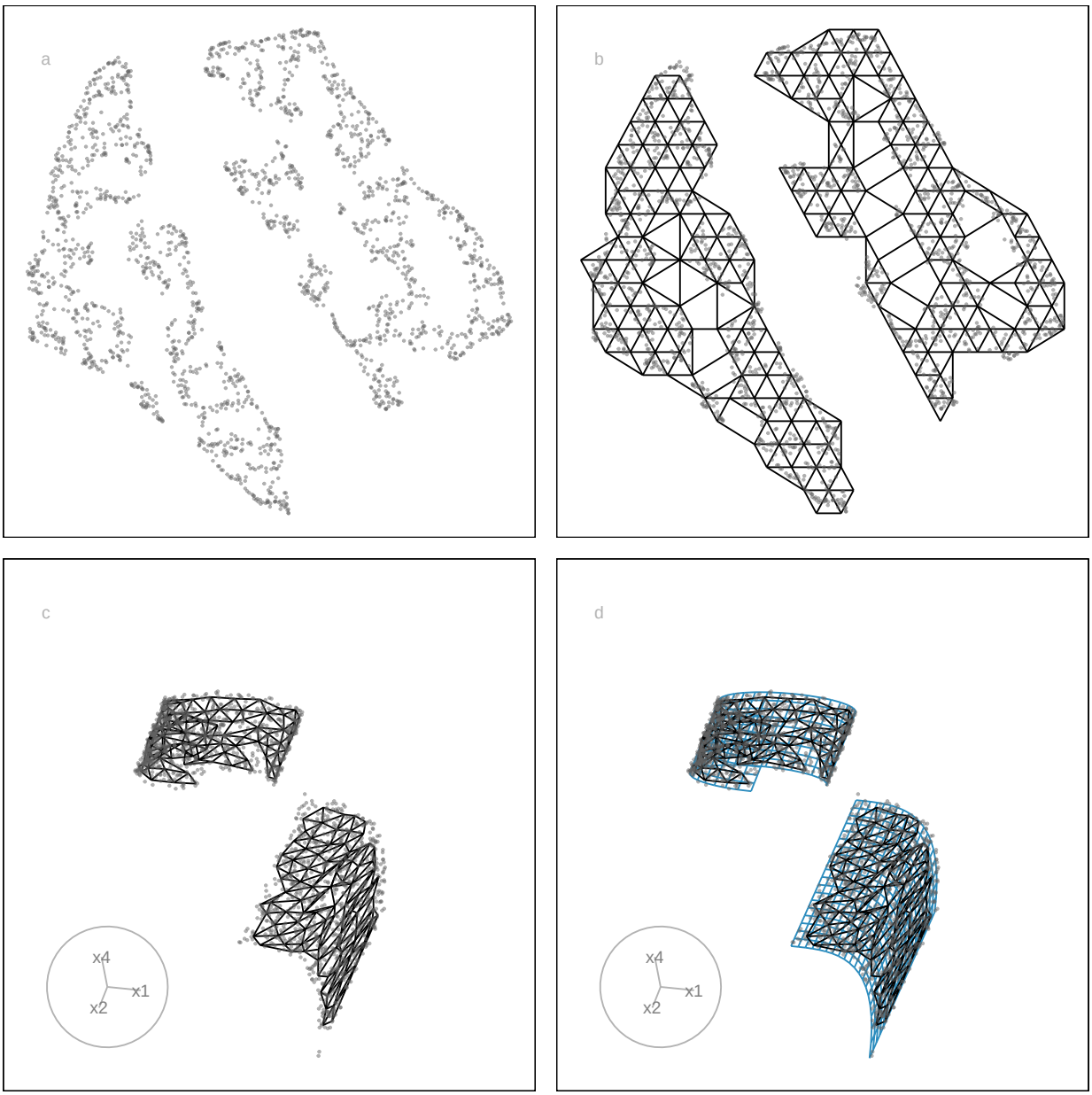


Figure 5: Model in 2-D ($a_1 = 0.12$), on the layout and two views of the fit in projections from 7-D, for the two C-shaped clusters data ($n = 2000$ and $p = 7$). One view shows the fitted model, and the other shows the fitted and true models alongside the data. The fitted model accurately represents the shape of the clusters but does not capture the edges of the structure. This illustrates a characteristic of UMAP, which compresses the data when transforming the data into 2-D. Video of the langevitour animations is available at <>.

3.4 Measuring the fit

The model here is similar to a confirmatory factor analysis model (Brown 2015), $\widehat{T}(X_1, X_2, X_3) + \mathrm{E}$. The difference between the fitted model and observed values would be considered to be residuals, and for this problem are 7-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 mean squared error (MSE) as:

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

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 $e_{hj}=\sum_{j=1}^{p}(\boldsymbol{x}_{hij}-C_{hj}^{(p)})^2$ to be the residual for each observation.

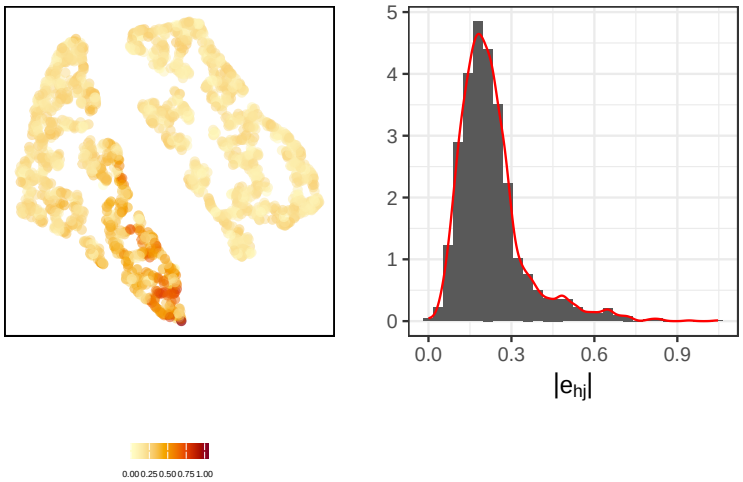


Figure 6: The 7-D model error in 2-D layout. Color indicates square root of total error, dark blue indicating high error and light indicates low error. Most large errors are distributed near the edges of the clusters.

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, and
- edge length maximum, remove long edges from 2-D representation.

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 7 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.

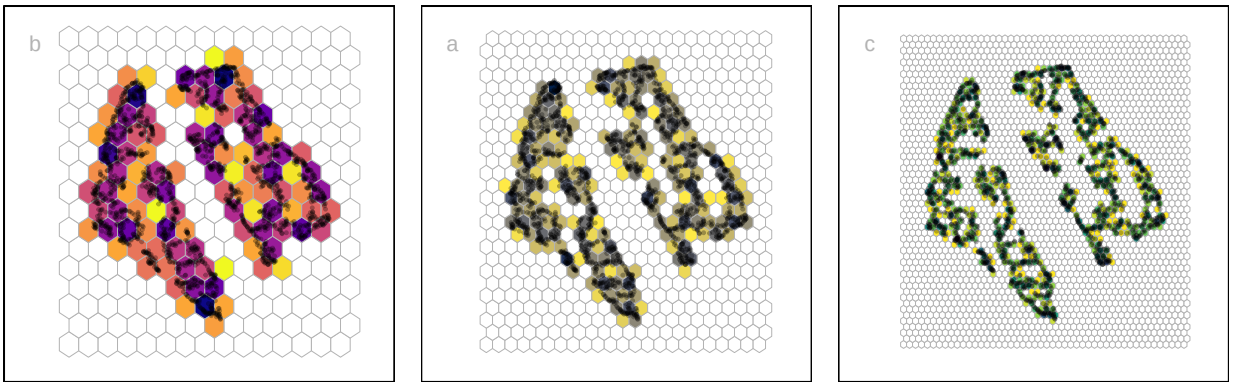


Figure 7: Hexbin density plots of tSNE layout of the two C-shaped clusters data, using three different bin inputs: (a) $b = 80$ $(10,\ 8)$, (b) $b = 130$ $(13,\ 10)$, and (c) $b = 391$ $(23,\ 17)$. Color indicates standardized counts, dark indicating high count and light indicates low count. At the smallest 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 8 (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 8 (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 8 (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.

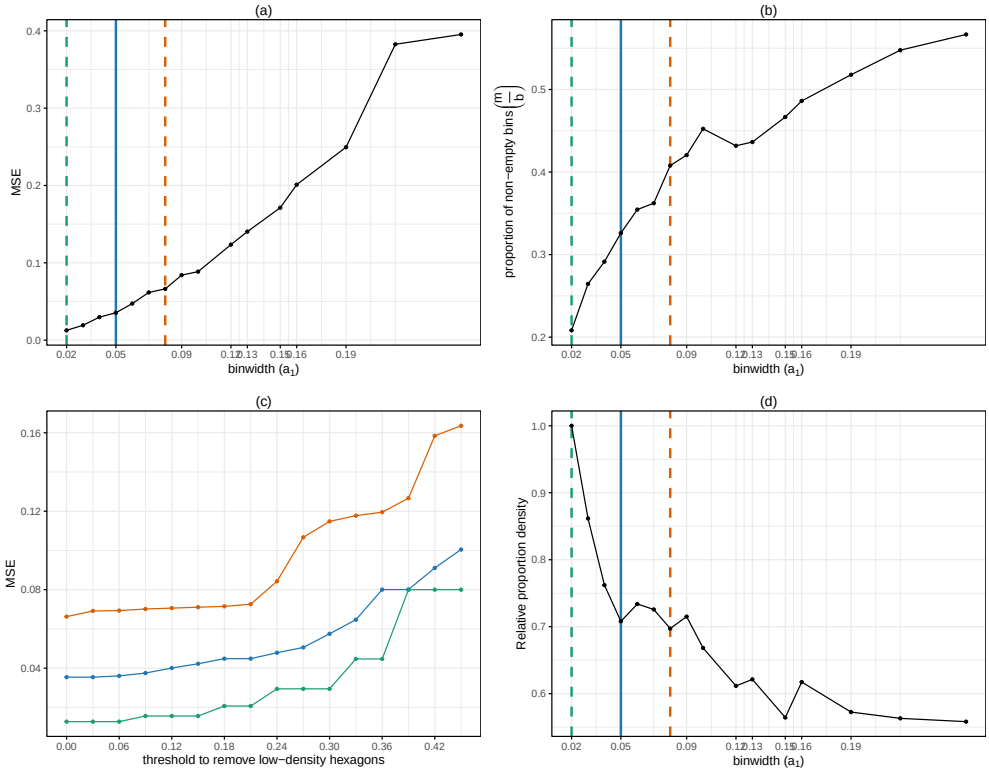


Figure 8: Various plots to help assess best hexagon bin parameters and thresholds to remove low-density bins. Both (a) and (c) show MSE, against binwidth (a_1) and threshold. A good benchmark value for these parameters is when the MSE drops and then flattens out. Three binwidth choices were made: 0.05, 0.09, and 0.13 to investigate. As the binwidth increases, the proportion of non-empty bins also increases. There are two peaks at binwidths 0.09 and 0.13. The relative proportion density decreases and levels off. Binwidths 0.05 is the favorable choices. Based on the MSE, 0.09 was 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 MSE.

3.6.4 Removing long edges

Edges define the neighbourhood structure, in order to provide a smooth 2-D representation of the fitted model. Figure 2 shows a wire frame of the true model that was used to generate the two C-shaped clusters example data. The ideal is that the representation of the fitted model, at least for this example where we know the true model, should look similar to this.

The Delaunay triangulation will ensure that all centroids are connected into a triangular mesh. For some structures, like clustered data, or highly non-linear shapes, breaks in the mesh are meaningful. When separated clusters are present the mesh should be broken across the gaps. For non-linear structures like the C-shaped curvilinear, the mesh should run unbroken along the C, but there should be no edges connecting the top of the C directly to the bottom of the C. For these reasons it is necessary to remove edges from the mesh in some applications.

The decision on edge length removal is made based on the distribution of edge lengths. In particular, a gap between values, where there a concentration of small values and then a

few larger values, likely suggests a cutoff for edge removal. Because the triangulation is typically done on the hexagon centroids, there are particular discrete edge lengths, based on bin widths. Figure 9 illustrates edge length distributions.

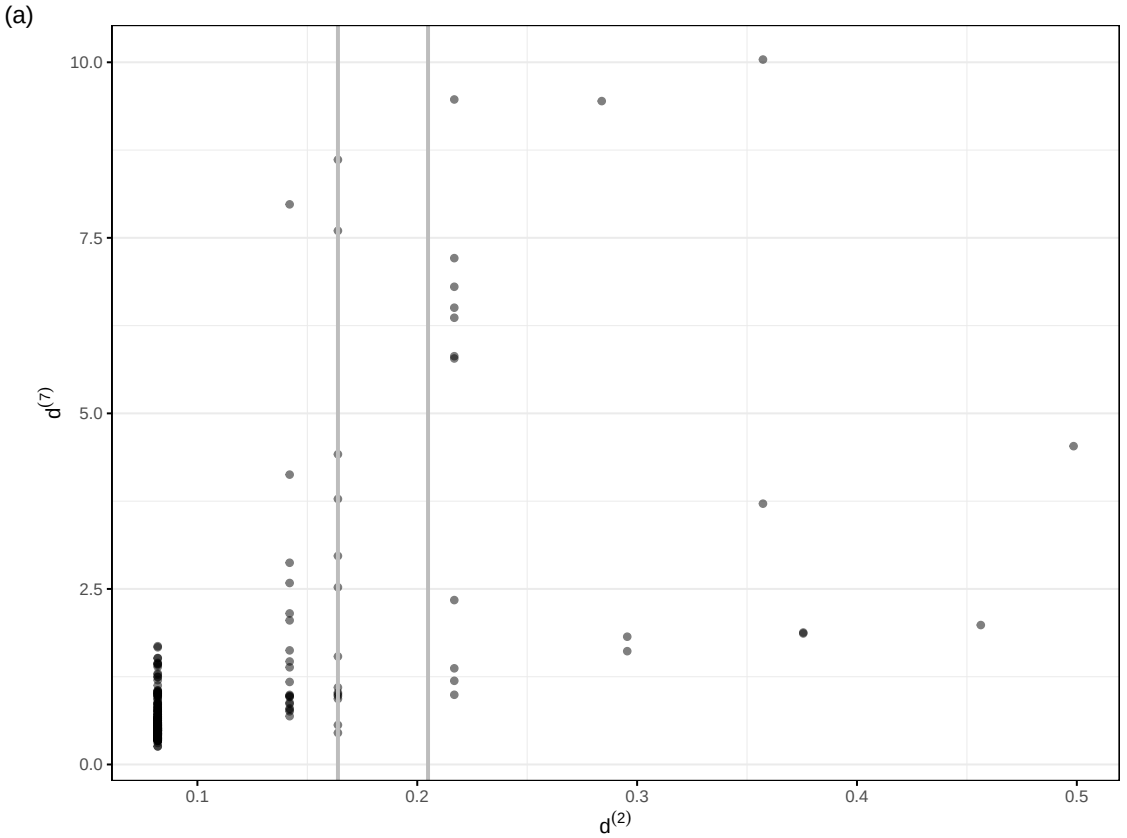
There is an additional step that is needed. When the model is lifted into p-D, if the fit is good all the edges should be relatively small in this space, too. If this is not the case, then there are several possible actions: (1) re-do the NLDR to get a more representative layout; (2) identify the edge and remove it from the model, in 2-D and p-D; (3) consider different values for the model fit, number of bins, initial bin position or removing low density bins.

4 Best fit

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, long edge 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.



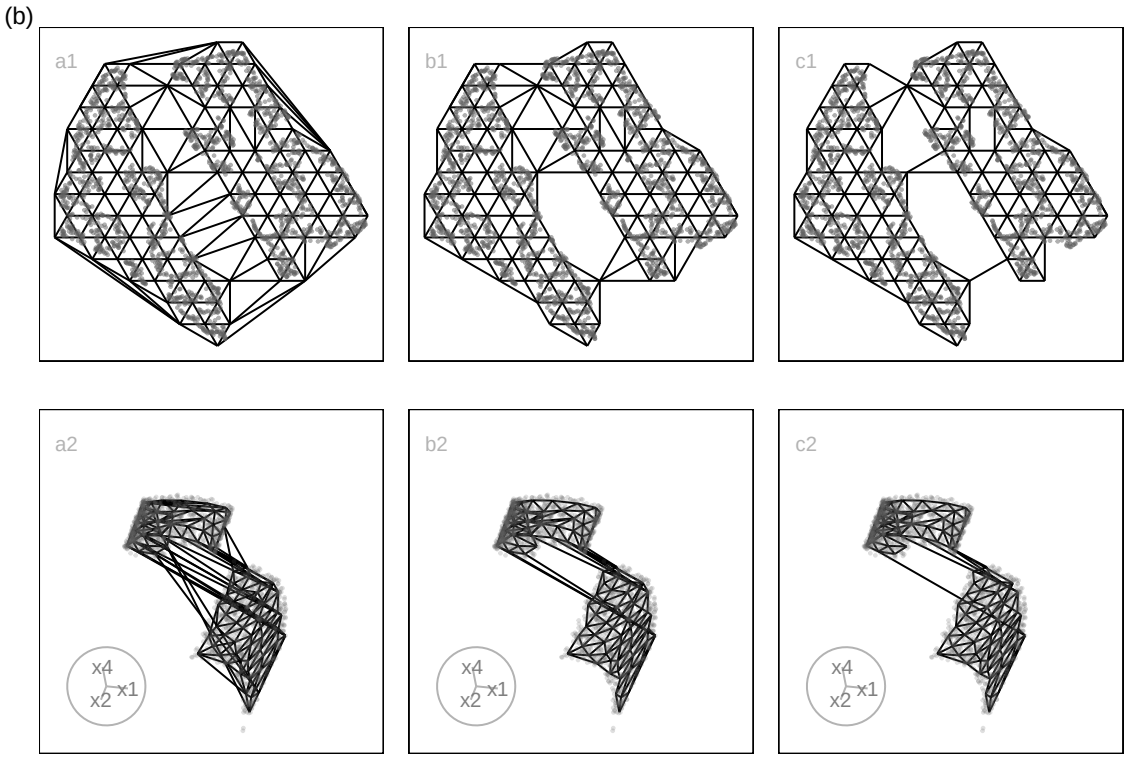


Figure 9: Distribution of edge lengths and wireframes in 2-D and 7-D with Delaunay triangulation, and two choices of long edge removal: (b) benchmark $= 2.5a_1$ and (c) benchmark $= 2a_1$, where $a_1 = 0.09$.

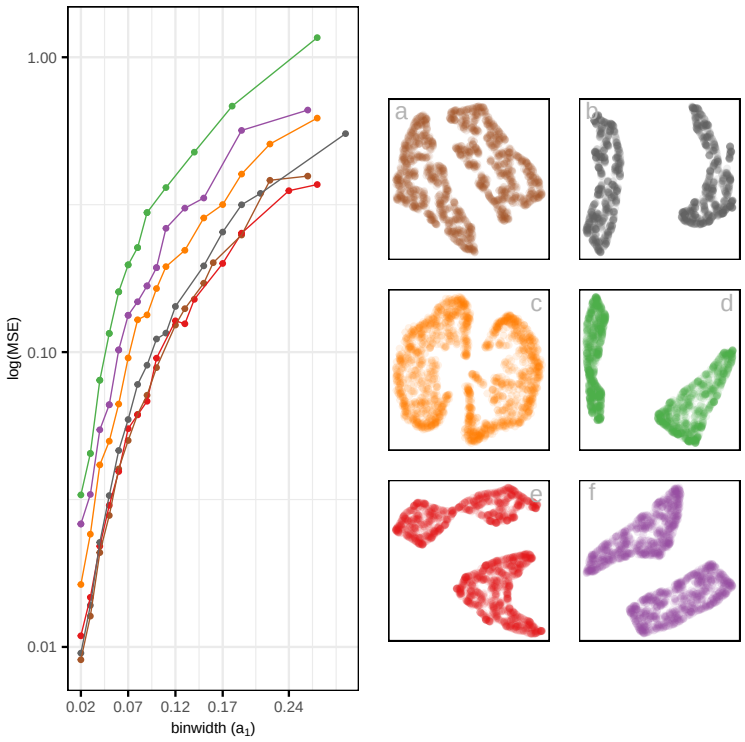


Figure 10: Assessing which of the 5 NLDR layouts on the two C-shaped clusters data is the better representation using MSE for varying binwidth (a_1). Colour used for the lines and points in the left plot and in the scatterplots represents NLDR layout (a-e). Layout d is universally poor. Layouts a that show two close clusters are universally suboptimal. Layout a is the best choice.

5 Linked plots

It's important to access the 2-D layout and the generated model overlaid on data in p-D together to understand whether it fits the points everywhere, fits better in some places, or simply mismatches the pattern. Interactivity also helps in understanding the quirks that occur with different NLDR techniques (XXXXRefer the video after finalising the two C-shaped clusters data).

6 Curiosities

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

6.1 Ordering of points

DHC: Will need to have more bins in each of the methods to illustrate this properly.

To illustrate a difference in how the methods organise points in the NLDR layout, simulated 4D data having five Gaussian clusters is used. Figure 11 a1, b1, c1 show the 2D layouts

for (a) t-SNE, (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 11 a2, b2, c2 show the fitted models in a projection of the 4D space. In this projection the difference between methods can be seen. These clusters are fully 4D in nature, so we would expect the model to be a *crumpled 2D sheet* that stretches in all four dimensions. This is what is observed for t-SNE 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 2D 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 4D doesn't reflect the dimension of each cluster. DHC: is there something in the documentation that might help to explain what has happened?

This pattern can also be observed by examining the error from the model fit (Figure 11 a3, b3, c3). With t-SNE and UMAP the error is relatively uniformly distributed, but with PaCMAP there is more error in the center of the 2D layout, reflecting that there are more points in the middle that are further from the fitted model.

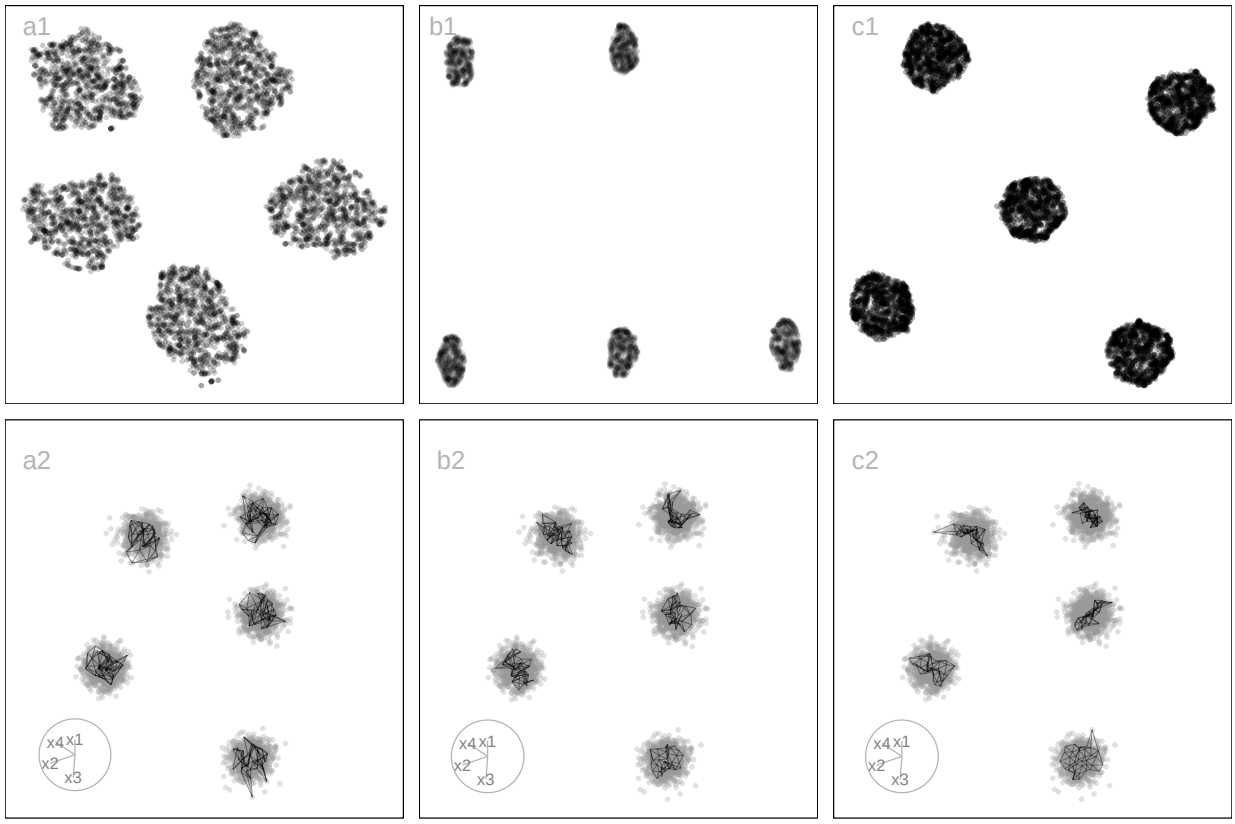


Figure 11: NLDR's organise points in the 2-D layout in different ways, possibly misleadingly, illustrated using three layouts: (a) tSNE, (b) UMAP, (c) PaCMAP. The data has five Gaussian cluster in 4-D. The bottom row of plots shows a 2-D projection from a tour on 4-D revealing the differences generated 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.

6.2 The effect of density

To illustrate how the methods organise difference number of points within the structure in the NLDR layout, simulated 7-D data having C-shaped structure is used.

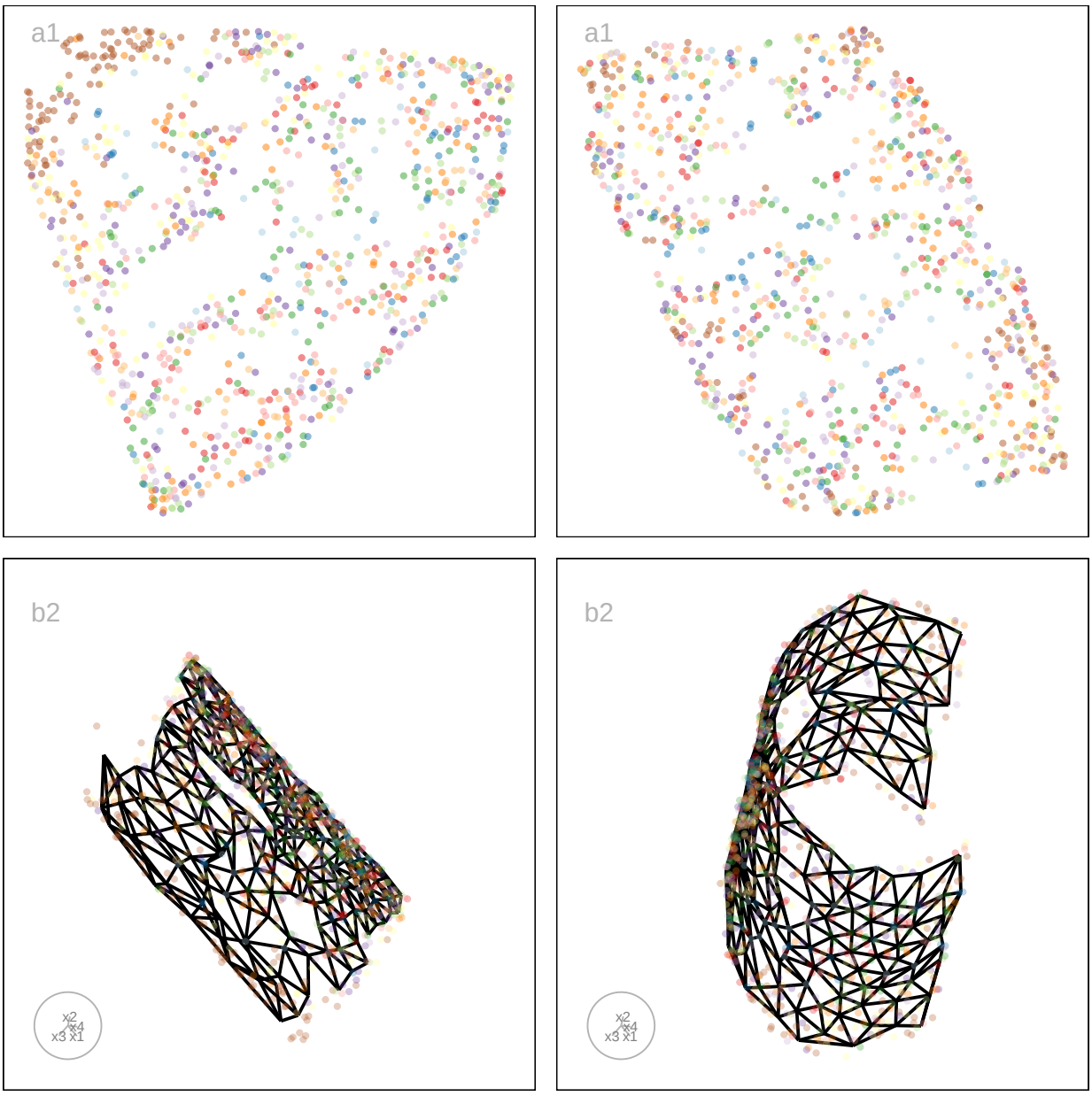


Figure 12: All 2-D layouts exhibit high error in the sparse end. The three 2-D layouts are: (a) tSNE, (b) UMAP, and (c) PaCMaP, which represent a C-shaped dense cluster colored according to the high-dimensional absolute error. Darker colors indicate high error, while lighter colors indicate low error. Regardless of the NLDR method used, the dense end of the structure shows low error, whereas the sparse end displays high error.

7 Applications

7.1 Single-cell gene expression

In the field of single-cell studies, a common analytical task involves clustering to identify groups of cells with similar expression profiles. NLDR methods are commonly used to display clusters, and help to verify the results. For example, Chen et al. (2023) illus-

trates the use of UMAP to identify clusters in Human Peripheral Blood Mononuclear Cells (PBMC3k). There are 2622 single cells. First 9 principal components are used to generate the UMAP. Figure 1 (a) is the reproduction of the published plot. The objective is to assess the published layout, and if it does not accurately represent the three clusters with small separations of the PBMC3k dataset (Figure 14 (a2)), then select a reasonable 2-D layout.

The Figure 1 (a) shows three well-separated clusters with big separations. However, as shown in Figure 14 (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 13). 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 18 (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 18 (b2)). Furthermore, the clusters shows non-linear shapes (Figure 18 (b3)).

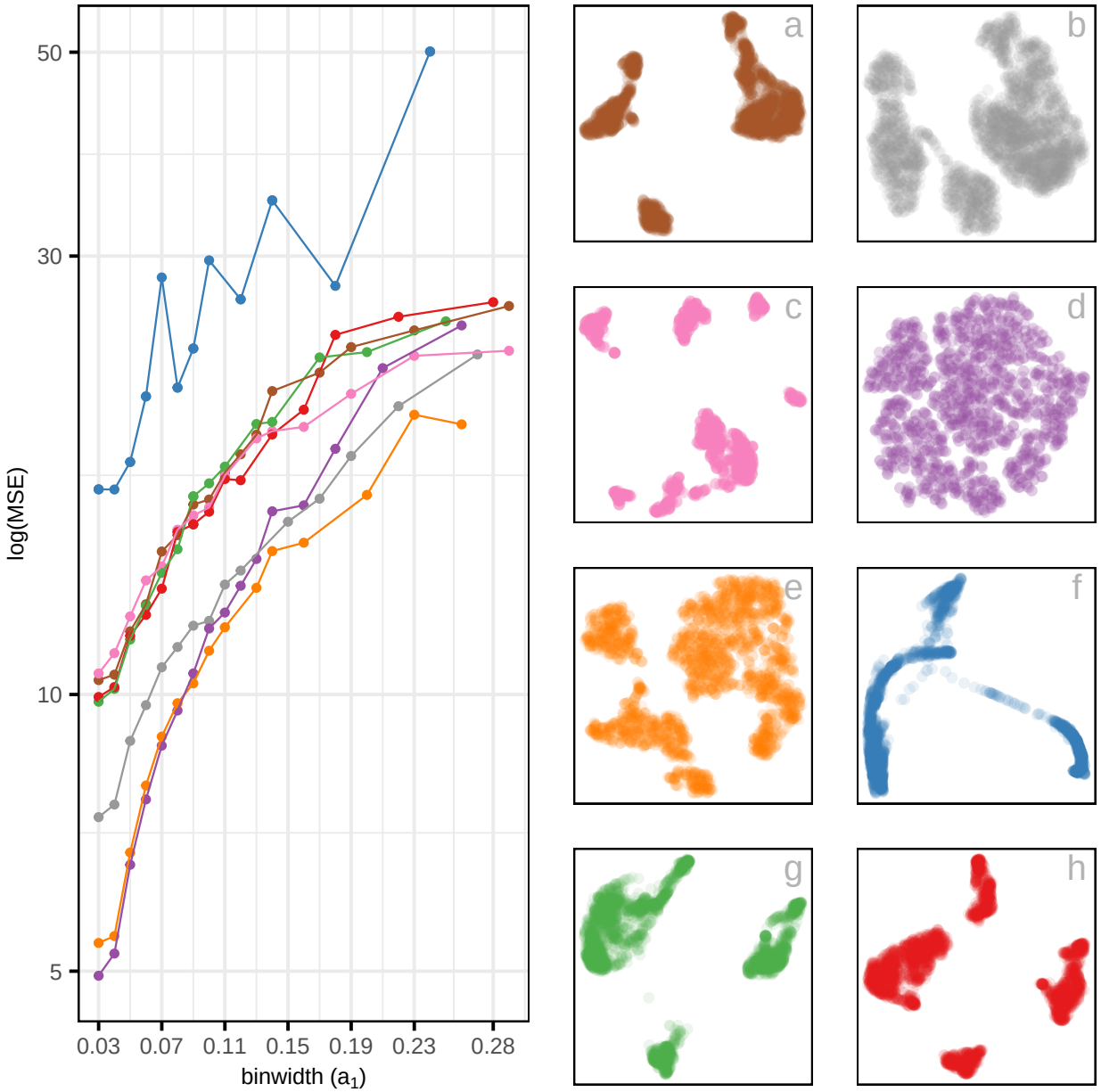


Figure 13: Assessing which of the 8 NLDR layouts on the PBMC3k data (shown in Figure 1) is the better representation using MSE for varying binwidth (a_1). Colour used for the lines and points in the left plot and in the scatterplots represents NLDR layout (a-h). Layout f is universally poor. Layouts 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.

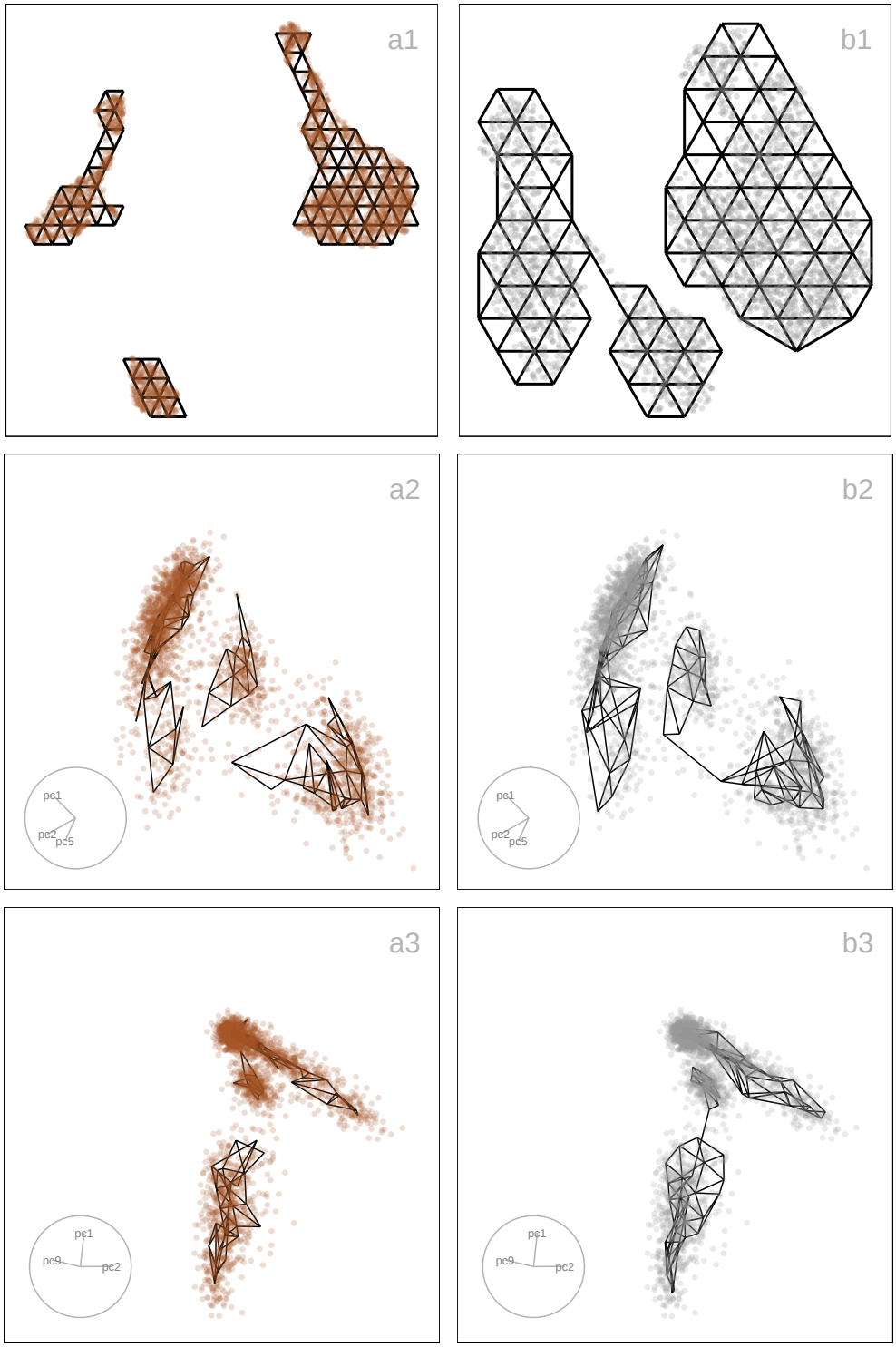


Figure 14: Model in 2-D, on the layout a and b, and two views of the fit in projections from 9-D, for the PBMC3k data ($n = 2622$ and $p = 9$). Layout a shows three-well separated clusters with big separation, while layout b shows close clusters. In 9-D, the data shows three clusters with two of them being close and one separated from the others. This is why b1, b2, and b3 shows a connected edge between the close clusters, as proven by Figure 13 as a reasonable layout. Viewing the fitted model in 9-D, helps 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.

7.1.1 Comparison with results of scDEED recommendations

In the field of single-cell studies, clustering is a common analytical task used to identify groups of cells with similar expression profiles. Non-linear dimensional reduction (NLDR) methods are frequently employed to visualize these clusters and help validate the results. However, it is well known that the 2D embeddings produced by t-SNE and UMAP may not accurately reflect the similarities among cell clusters.

To address this challenge, Xia et al. (2023) introduces a statistical method called scDEED, which detects unreliable cell embeddings generated by 2-D embedding techniques. scDEED calculates a reliability score for each cell embedding based on the similarity between the cell's 2-D embedding neighbors and its neighbors prior to embedding. By identifying embeddings with low reliability scores as dubious and those with high scores as trustworthy, scDEED minimizes the number of questionable cell embeddings. This approach also provides clear guidance for optimizing the hyperparameters of an embedding method.

To verify that the 2-D cell embeddings became better aligned with the cell types after scDEED's optimization, the Human Peripheral Blood Mononuclear Cells (PBMC) dataset was used. It contained $31,021$ cells with cell type labels, and the gene expression levels were in the unit of log-transformed UMI count per $10,000$. They focused on three sequencing methods (inDrops, DropSeq, and SeqWell) and four common cell types Cytotoxic T cell, CD4+T cell, CD14+ Monocyte, and B cell.

For illustration purposes, we only selected cells generated with inDrops ($n = 5858$ cells) and UMAP cell embeddings. Also, Xia et al. (2023) used first 50 principal components to generate the UMAP. The objective is to assess the optimized layout by scDEED, and if it does not accurately represent the three clusters with small separations of the PBMC dataset, then select a reasonable 2-D layout.

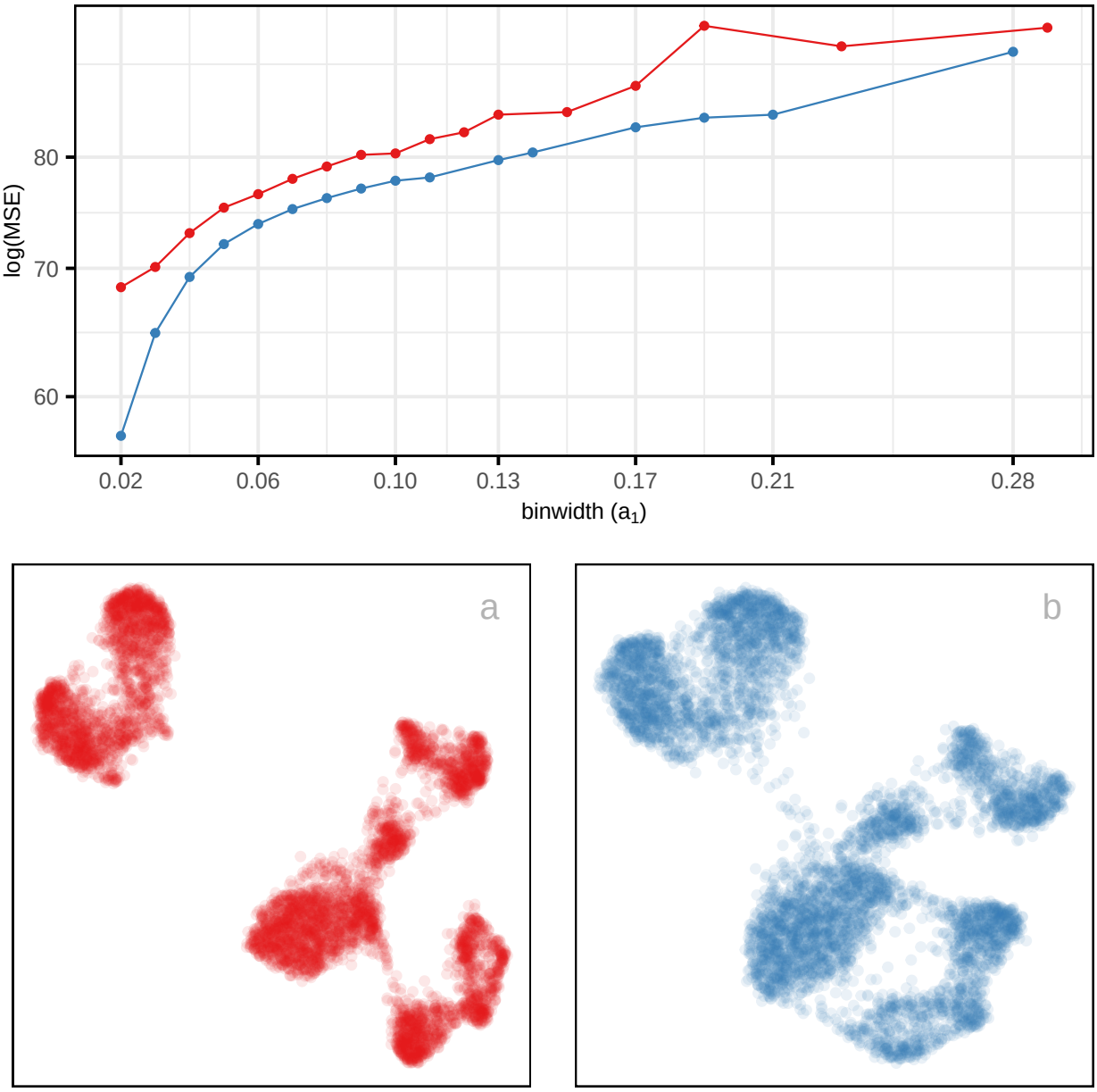


Figure 15: Assessing which of the 2 UMAP layouts with different hyperparameter settings (n_neighbors: 30, min_dist: 0.3 (red), n_neighbors: 80, min_dist: 0.5 (blue)) on the PBMC data is the better representation using MSE for varying binwidth (a_1). Colour used for the lines and points in the left plot and in the scatterplots represents UMAP layout (a, b). Layout b is universally suboptimal. Layout b is the best choice.

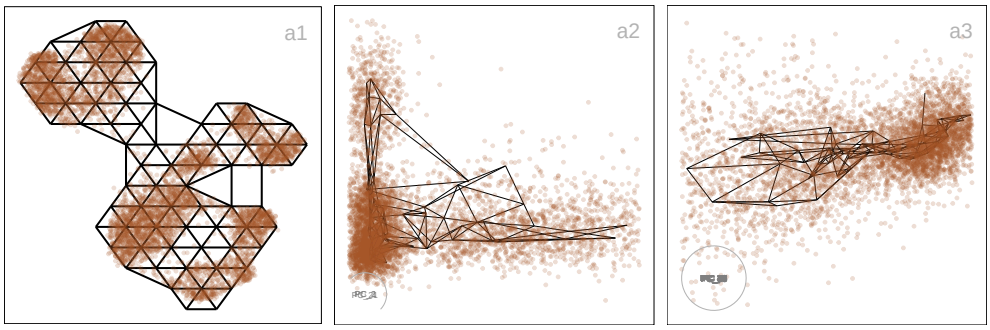


Figure 16: Model in 2-D, on the layout a, and two views of the fit in projections from 50-D, for the PBMC data ($n = 5858$ and $p = 50$). Layout shows close clusters. In 50-D, the data shows

7.2 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 18 (a)).

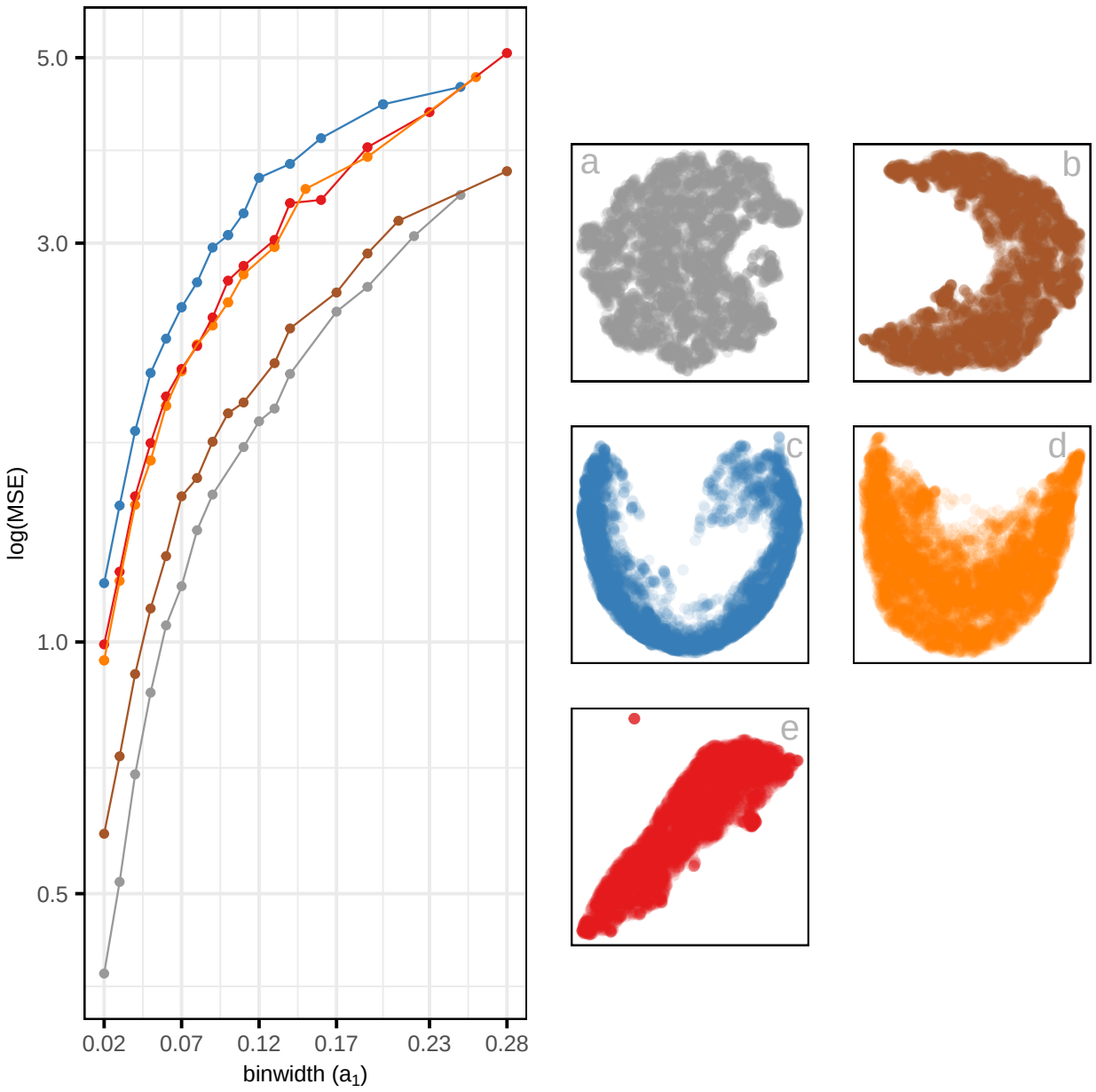


Figure 17: Assessing which of the 5 NLDR layouts on the MNIST digit 1 data is the better representation using MSE for varying binwidth (a_1). Colour used for the lines and points in the left plot and in the scatterplots represents NLDR layout (a-e). All the layouts appear to be very similar. Layout c is universally poor. Layouts a that show two close clusters are universally suboptimal. Layout a is the best choice.

The MSE for different binwidths (a_1) using tSNE, UMAP, PHATE, PaCMAP, and TriMAP with default (hyper-)parameter setting (Figure 17) were calculated. It is found that tSNE (Figure 17 (a)) 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 (**?@fig-tsne-best**).

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 (**?@fig-tsne-best**), while the small cluster contains the digit 1 images with different patterns of the digit 1, unlike the usual (Figure 19 (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 18 (a), 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 18 (b)). Furthermore, the presence of long edges indicates the existence of the small cluster located closely together (Figure 18 (c)).

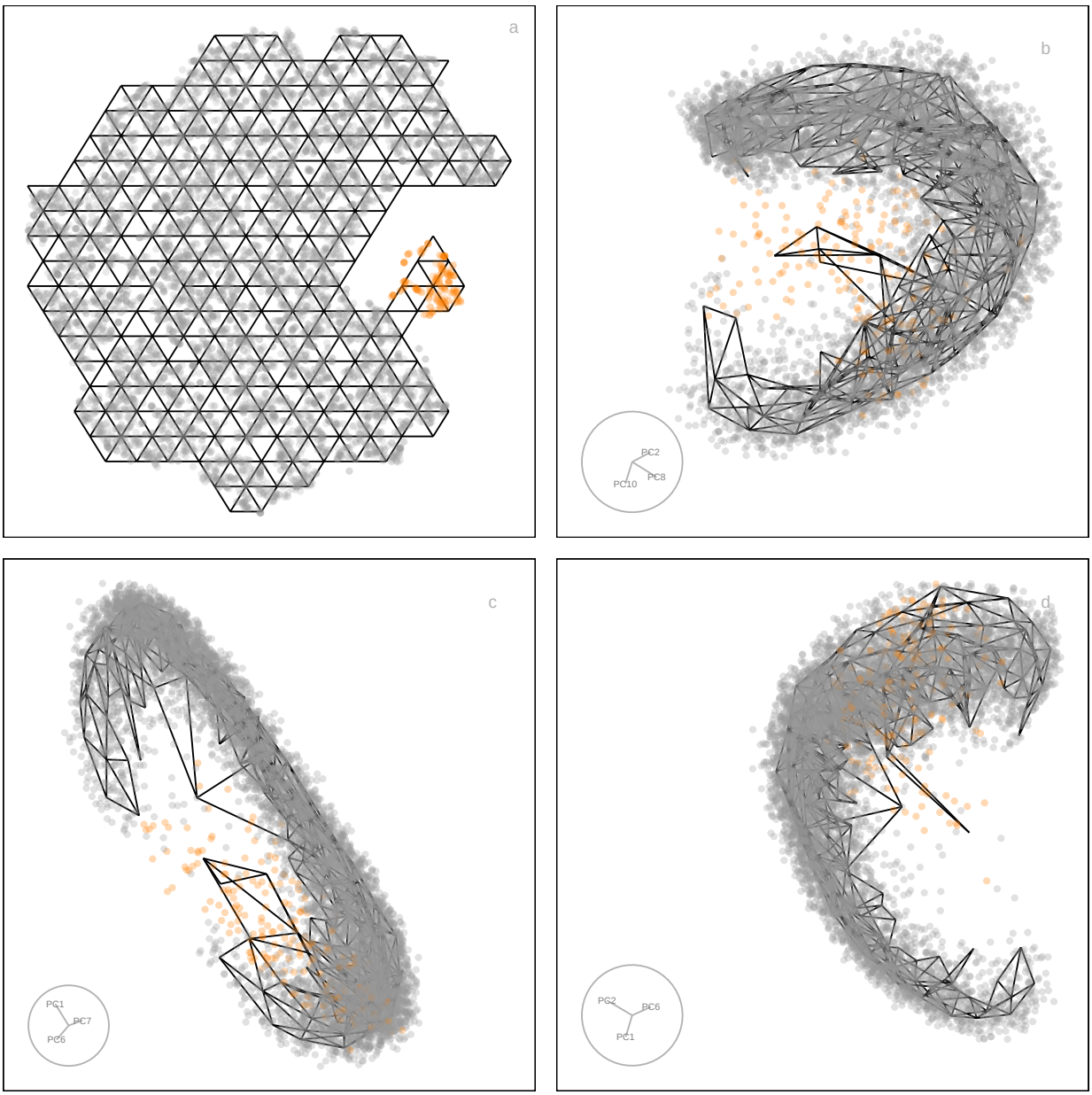


Figure 18: Model in 2-D, on the layout a and three views of the fit in projections from 10-D, for the MNIST digit 1 data ($n = 7877$ and $p = 10$). There is a big non-linear cluster and a small clusterlocated very close to one corner of the big cluster. There is a twisted pattern that is hardly visible in the static plot. Video of the langevitour animation are available at .

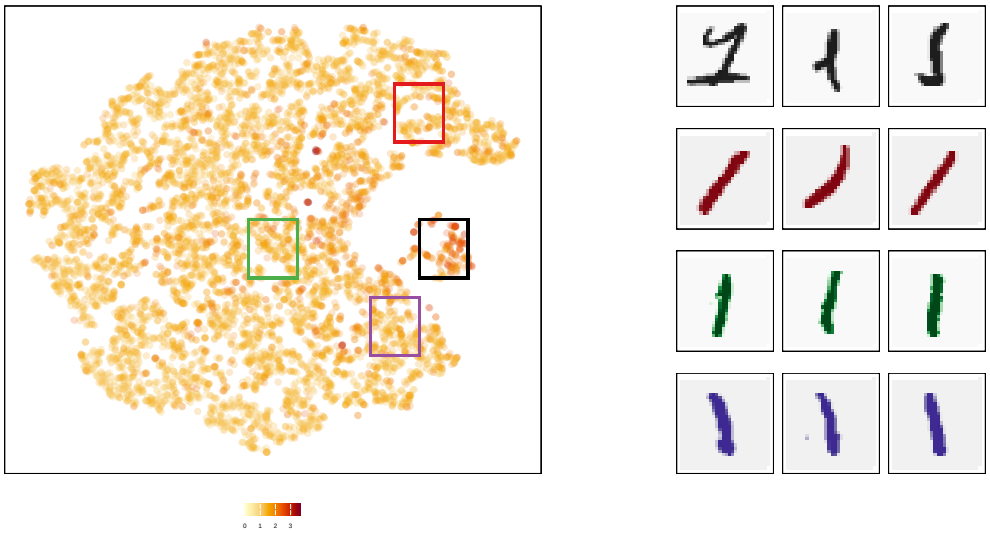


Figure 19: The 10-D model error in layout a of the MNIST digit 1 dataset show a pattern. Most low model errors are distributed along the big non-linear clutser, while most large model errors are distributed along the small cluster. Along the non-linear cluster, the angle of digit 1 changes. Some images have large errors due to their deviation from the non-linear cluster, which makes them anomalies. The images associated with large model errors shows different patterns of digit 1.

8 Discussion

This study makes several important contributions to the field of NLDR. We have developed an algorithm to evaluate the most useful NLDR method and (hyper-)parameter choices for creating an accurate 2-D layout of high-dimensional data. Our objective is to fit a model for the 2-D layout that preserves the relationships between neighboring points and turns it into a high-dimensional wireframe, which can be overlaid on the data and visualized using 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.

The effectiveness of this approach is illustrated through various examples. For instance, the two-curvy clusters example demonstrates how the model accurately fits the points, capturing both local and global structures in high-dimensional space. Our simulation case study further, five Gaussian cluster example shows that while all observed NLDR methods preserve the global structure, only tSNE effectively maintains the local structure, highlighting the specific strengths and quirks of different methods.

Human behavior often shows a desire for more certainty and a tendency to prefer well-separated views. This emphasizes the importance of clear and distinct clusters. For example, in the UMAP layout of the **PBMC3k** dataset suggested by Chen et al. (2023), three distant, well-separated clusters are shown. However, our model reveals that these clusters are actually close to each other in p-D. Additionally, the model discovers non-uniform data distribution and non-linear structures within the clusters that are not visible in the UMAP layout, demonstrating the ability of our model in uncovering hidden data characteristics.

Evaluating the error or unexplained variance is important for assessing how well the model fits the data. By examining the error for different numbers of bins, we found that tSNE with a perplexity of 30 provides a reasonable representation for the **pbmc** dataset. Connecting the closest clusters with line segments in the fitted model further supports the preservation of neighborhood relationships.

The **digit: 1** example further illustrates the model's ability to accurately capture non-linear structures and provide additional information. Key findings include a twisted pattern that compresses the structure in some projections and long line segments that detect anomalies.

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.

In conclusion, while our method effectively captures and represents high-dimensional data structures, further enhancements could involve introducing approaches to bind the data, indicate line segments beyond 2-D, and diagnose the fitted model. These improvements would help in creating a more accurate representation of the data when 2-D layout is inadequate.

9 Supplementary Materials

Appendix: The appendix includes more details about the hexagonal binning algorithm (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)

10 Acknowledgments

These `R` packages were used for the work: `tidyverse` (Wickham et al. (2019)), `png` (Urbanek (2022)), `Rtsne` (Krijthe (2015)), `umap` (Konopka (2023)), `ggplot2` (Wickham (2016)), `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 (2022)) and `pacmap` (Wang et al. (2021)). The article was created with `R` packages `rticles` (Allaire et al. (2024)), `knitr` (Xie (2014)), and `rmarkdown` (Xie et al. (2020)). The project's GitHub repository (https://github.com/JayaniLakshika/paper-nldr-vis-algorithm) contains all materials required to reproduce this article.

References

Allaire, J., Xie, Y., Dervieux, C., R Foundation, Wickham, H., Journal of Statistical Software, Vaidyanathan, R., Association for Computing Machinery, Boettiger, C., El-

sevier, Broman, K., Mueller, K., Quast, B., Pruim, R., Marwick, B., Wickham, C., Keyes, O., Yu, M., Emaasit, D., Onkelinx, T., Gasparini, A., Desautels, M.-A., Leutnant, D., MDPI, Taylor and Francis, Öğreden, O., Hance, D., Nüst, D., Uvesten, P., Campitelli, E., Muschelli, J., Hayes, A., Kamvar, Z. N., Ross, N., Cannoodt, R., Luguern, D., Kaplan, D. M., Kreutzer, S., Wang, S., Hesselberth, J. & Hyndman, R. (2024), *rticles: Article Formats for R Markdown*. R package version 0.27. https://CRAN.R-project.org/package=rticles.

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

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. (2023), 'Directly selecting differentially expressed genes for single-cell clustering analyses', *bioRxiv* . https://www.biorxiv.org/content/early/2023/07/29/2023.07.26.550670.

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.

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.

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'.

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

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

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**.

Urbanek, S. (2022), *png: Read and write PNG images*. R package version 0.1-8. https://CRAN.R-project.org/package=png.

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

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

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. (2016), *ggplot2: Elegant Graphics for Data Analysis*, Springer-Verlag New York. https://ggplot2.tidyverse.org.

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/.

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.

Xie, Y. (2014), knitr: A comprehensive tool for reproducible research in R, *in* V. Stodden, F. Leisch & R. D. Peng, eds, 'Implementing reproducible computational research', Chapman and Hall/CRC. ISBN 978-1466561595. http://www.crcpress.com/product/isbn/9781466561595.

Xie, Y., Dervieux, C. & Riederer, E. (2020), *R Markdown cookbook*, Chapman and Hall/CRC, Boca Raton, Florida. ISBN 9780367563837. https://bookdown.org/yihui/rmarkdown-cookbook.

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.