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

August 14, 2024

Abstract

Non-linear dimension reduction (NLDR) techniques such as tSNE, and UMAP provide a low-dimensional representation of high-dimensional (p-D) data using non-linear transformation. The methods and parameter choices can create wildly different representations, making it difficult to decide which is best, or whether any or all are accurate or misleading. 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. To help evaluate the NLDR we have developed an algorithm to show the 2-D NLDR model in the p-D space, viewed with a tour. One can see if the model fits everywhere or better in some subspaces, or completely mismatches the data. It is used to evaluate which 2-D layout is the best representation of the p-D distribution and see how different methods may have similar summaries or quirks.

Keywords: high-dimensional data, dimension reduction, hexagon binning, low-dimensional representation, tour, data vizualization, model-in-the-data-space

1 Introduction

Non-linear dimension reduction (NLDR) is popular for making a convenient low-dimensional (k-D) representation of high-dimensional (p-D) data. 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.

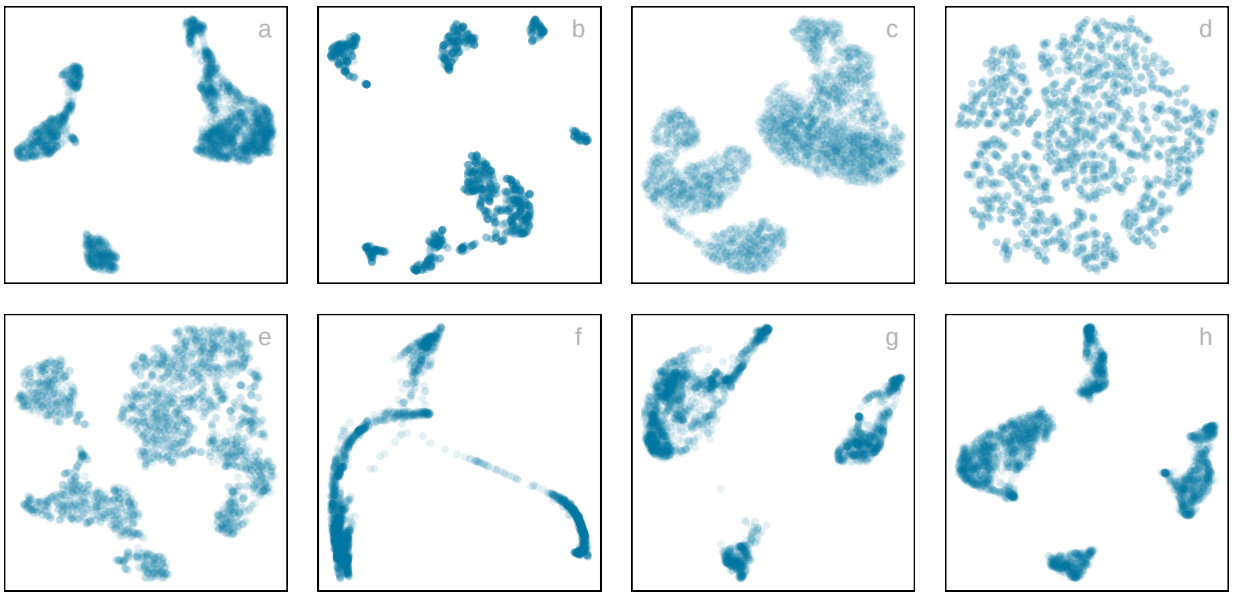


Figure 1: Six different NLDR representations of the same data. Different techniques and different parameter choices are used. Researchers may have seen any of these in their analysis of this data, depending on their choice of method, or typical parameter choice. Would they make different decisions downstream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

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

2 Background

Historically, k-D representations of p-D data have been computed using multidimensional scaling (MDS) (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

Table 1: Summary of notation for describing new methodology.

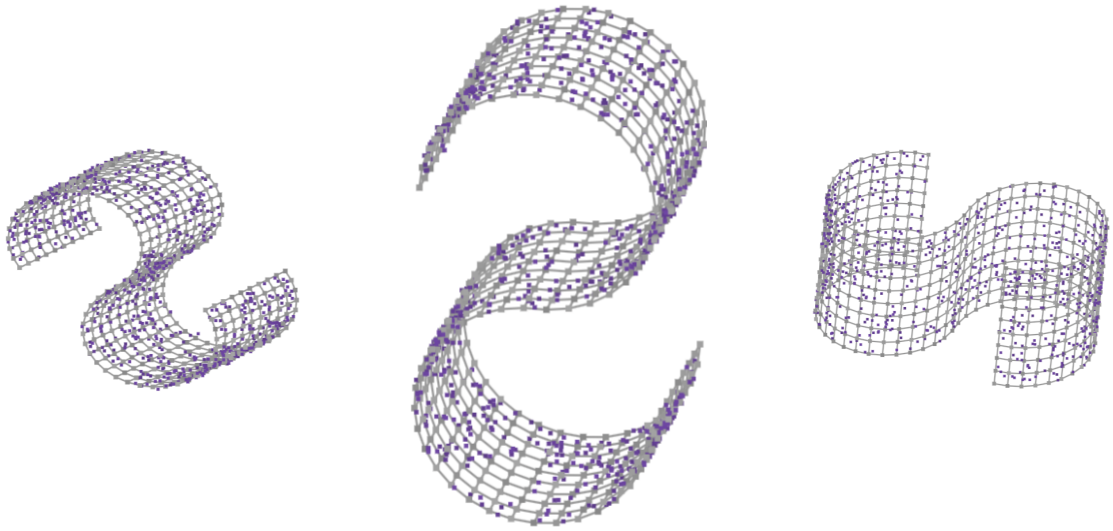


Figure 2: Three views of the true model (grey points and lines) in 2-D projections from 7-D, for the S-curve data (purple points). The data is spread along the S shape, and does not vary much from this curve. (The **langevitour** software is used to view the data with a tour, and the full video is available at https://youtu.be/I5GL23vLiw0).

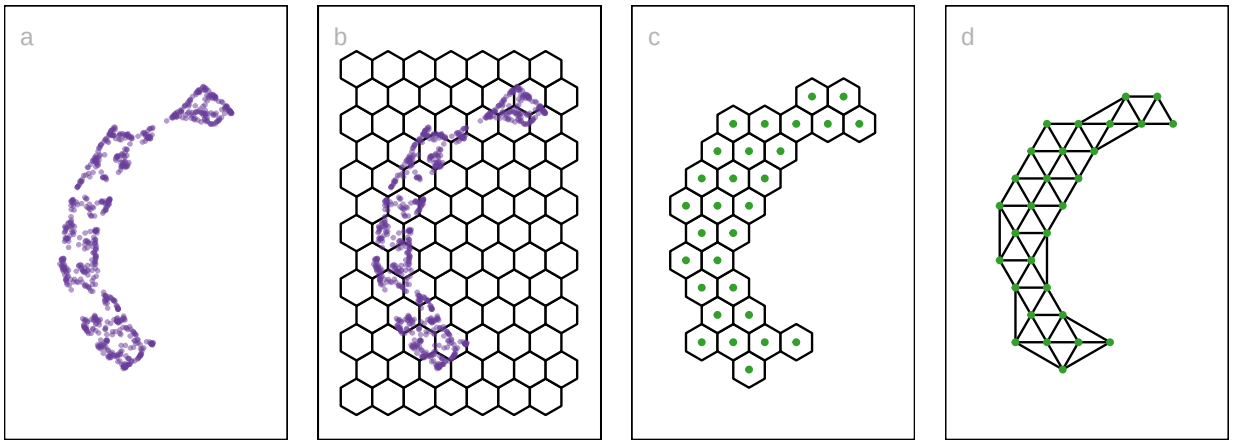


Figure 3: Key steps for constructing the model on the UMAP layout ($k = 2$): (a) data, (b) hexagon bins, (c) bin centroids, and (d) triangulated centroids. The S-curve data is shown.

3.2 Algorithm to represent the model in 2D

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

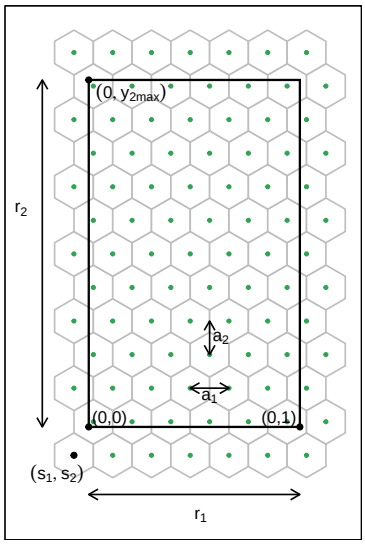


Figure 4: The components of the hexagon grid illustrating notation.

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 num-

ber of bins). This can be defined using the function $u : \mathbb{R}^{n\times 2} \rightarrow \mathbb{R}^{m\times 2}$, where $u(i) = \arg\min_{j=1,\ldots,b} \sqrt{(y_{i1} - C_{j1}^{(2)})^2 + (y_{i2} - C_{j2}^{(2)})^2}$, mapping observation i into $H_h = \{i|u(i) = h\}$.

By default, the bin centroid is used for describing a hexagon (as done in Figure 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.

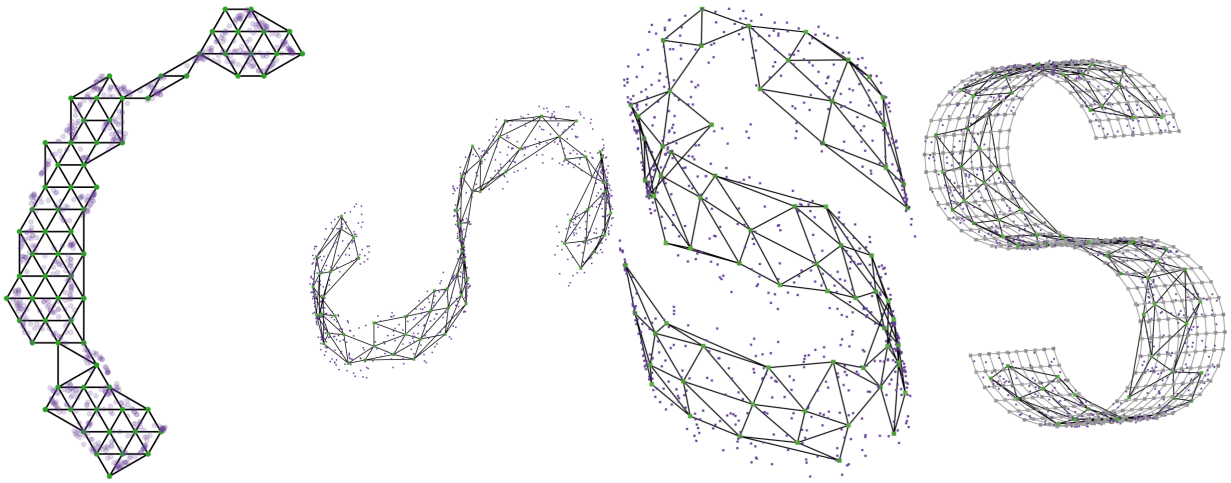


Figure 5: Model in 2-D, on the UMAP layout, and three views of the fit in projections from 7-D, for the S-curve data ($(s_1,\ s_2) = (-0.160,\ -0.263)$, $b = 405$ $(15,\ 27)$, $m = 70$, benchmark value to remove low density hexagons is 0.250, and benchmark value to remove large edges is 0.159). MSE is 0.0462. The model closely fits the shape, but it doesn't fully fill out the width of the S, which means that it does not adequately capture the surface (The **langevitour** software is used to view the data with a tour, and the full video is available at https://youtu.be/i7F7xpN1Hz8).

3.4 Measuring the fit

The model here is similar to a confirmatory factor analysis model (Brown 2015), $\widehat{T}(X_1, X_2, X_3) + \text{E}$. 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.

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, $C_h^{(2)} = (c_{h1}, c_{h2})$.

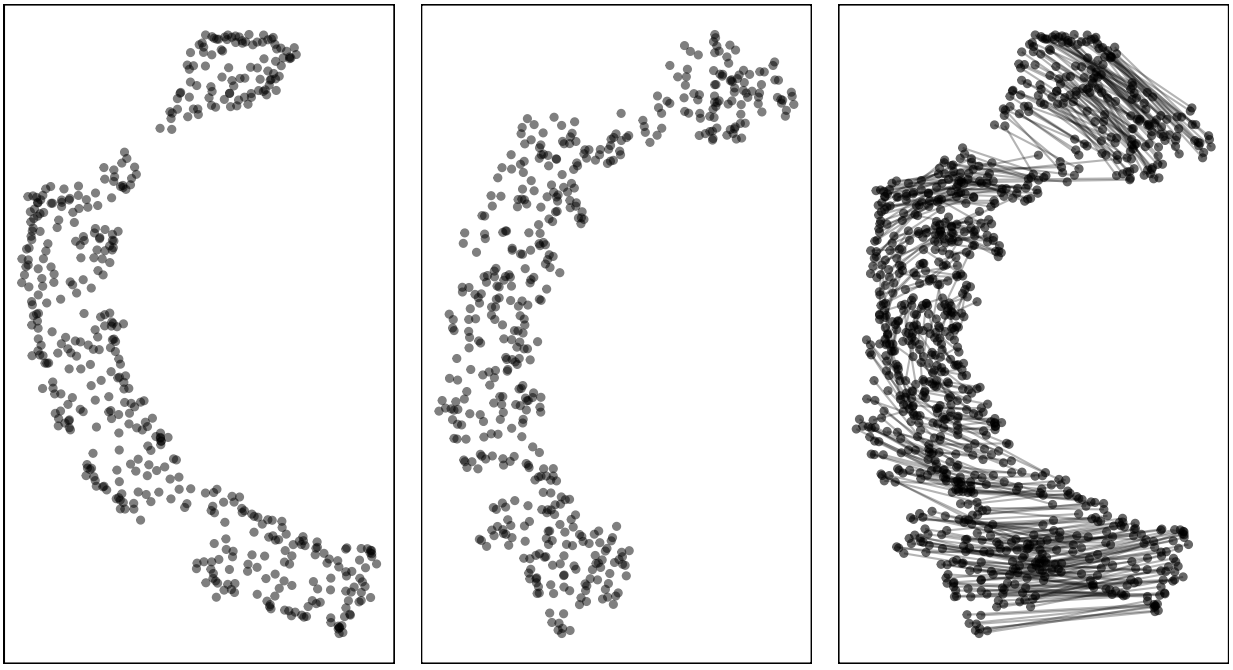


Figure 6

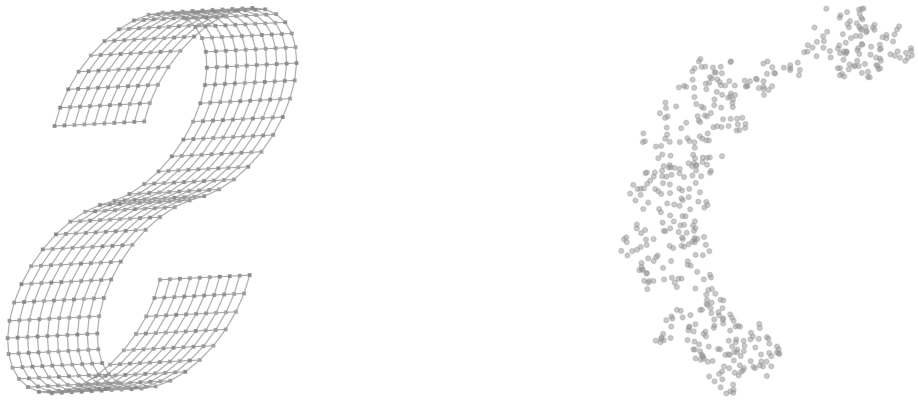


Figure 7: A view of the true model in projections from 7-D, and predictions of the true model in 2-D, for the S-curve data. The predictions fits the UMAP layout which means that it capture the geometry of S-curve with UMAP.

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 sound the data. The choice of these values can have some effect on the distribution of bin counts. Figure 9 (a) 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.

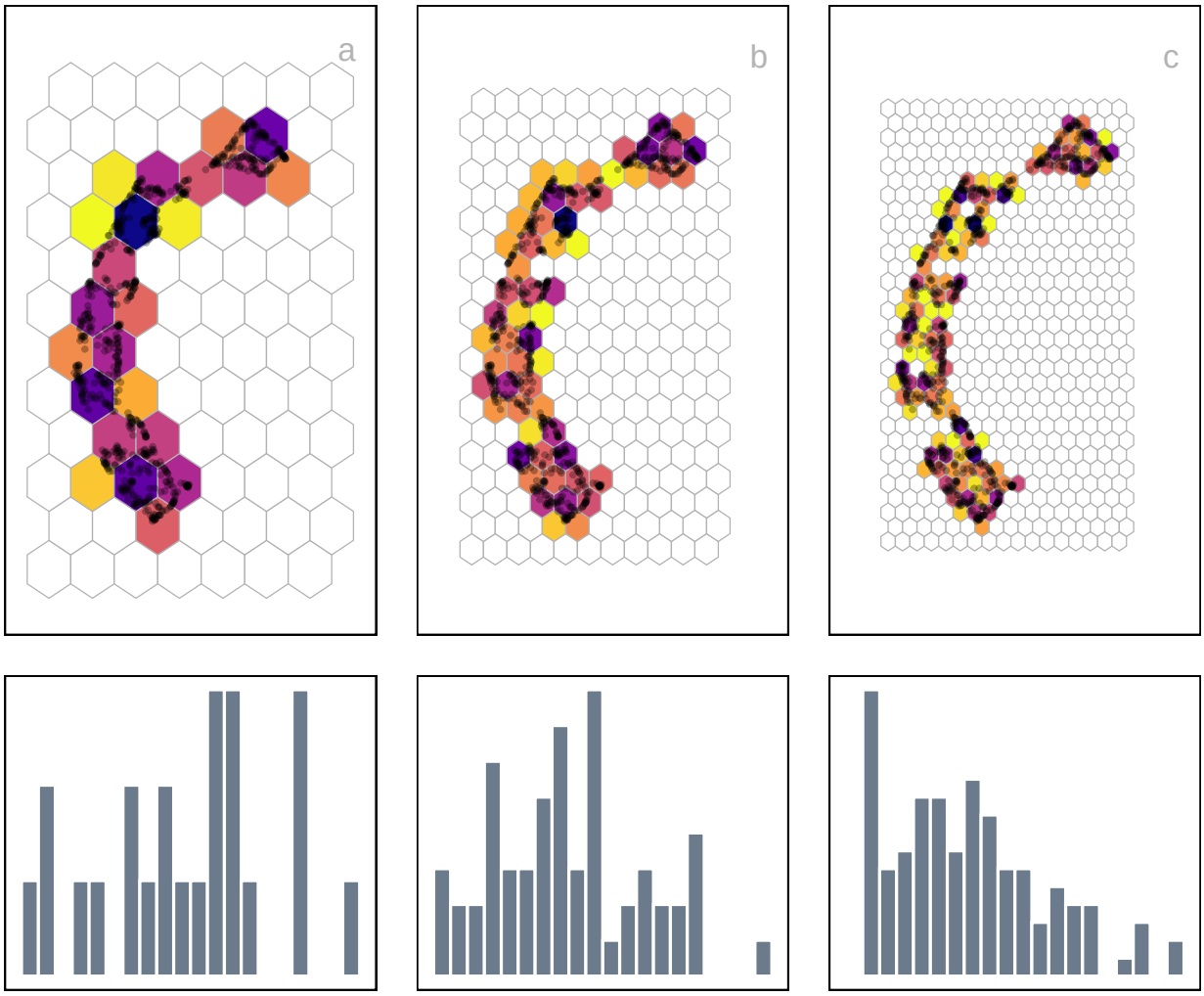


Figure 8: Hexbin density plots of UMAP layout of the S-curve data, using three different bin inputs: (a) $b = 91$ $(7,\ 13)$ $(q = 0.12)$, (b) $b = 220$ $(11,\ 20)$ $(q = 0.07)$, and (c) $b = 527$ $(17,\ 31)$ $(q = 0.05)$. Color indicates standardized counts, dark indicating high count and light indicates low count. At the smallest bin size the data segregates into two separate groups, suggesting this is too many bins. Using the MSE of the model fit in p-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 $b_1 = \frac{n}{2}$ are allowed. Figure 9 (b) shows the effect of different choices of b_1 on the MSE of the fitted model.

3.6.2 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 9 (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 9). 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.

3.6.3 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 S-curve 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 S-curve, the mesh should run unbroken along the S, but there should be no edges connecting the top of the S directly to the bottom of the S. 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 3 (d) 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.

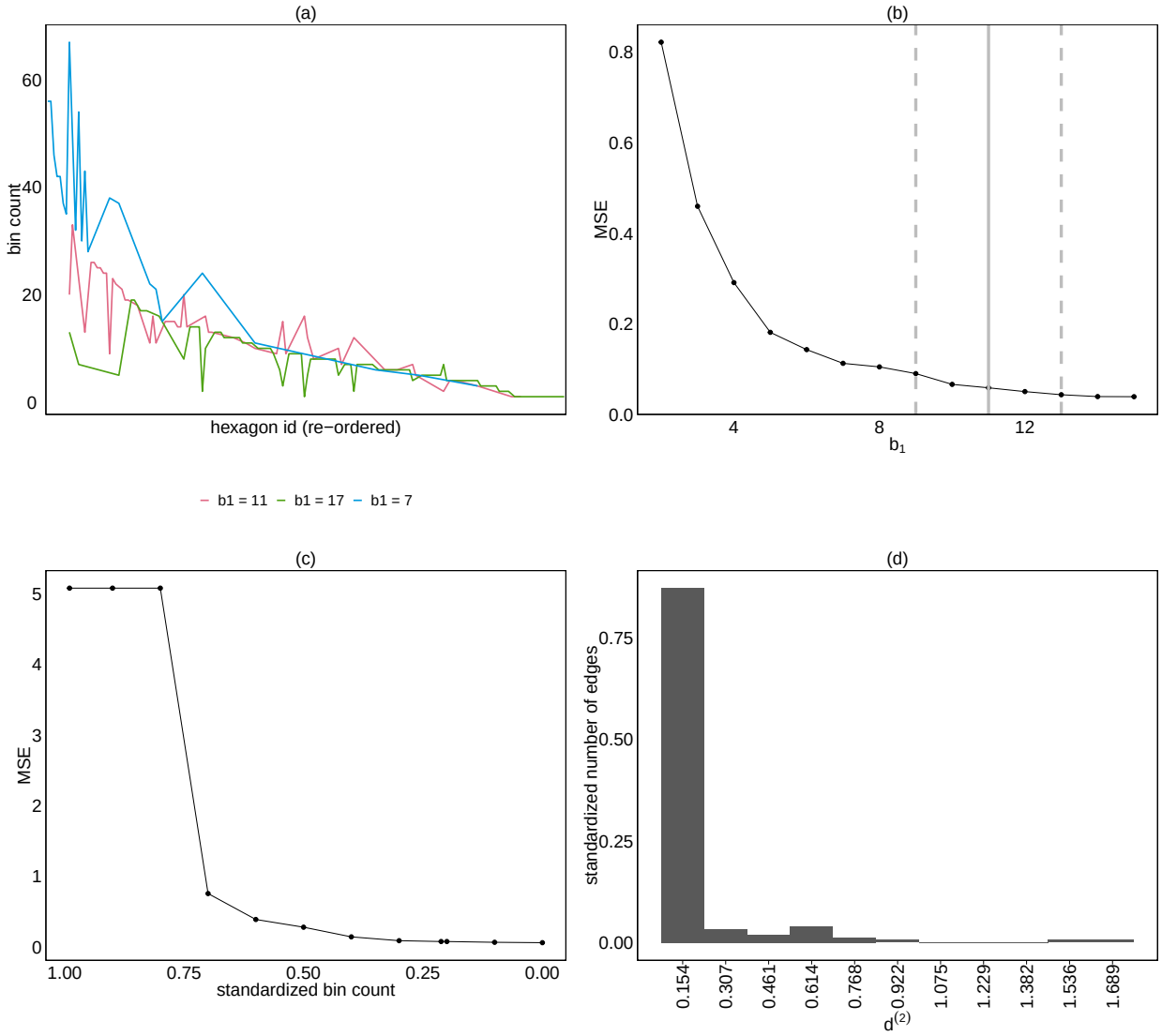


Figure 9: Various plots to help assess best hexagon bin parameters, thresholds to remove low density bins and large edges. Both (b) and (c) show MSE, against number of bins along the x-axis and standardised count. A good benchmark value for these parameters is when the MSE drops and then flattens out. Plot (a) shows the distribution of stadardised counts of hexagons. Plot (d) shows the distribution of 2-D Euclidean distances between bin centroids, with a good benchmark value for removing large edges would being the distance that shows the first large decrease.

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.

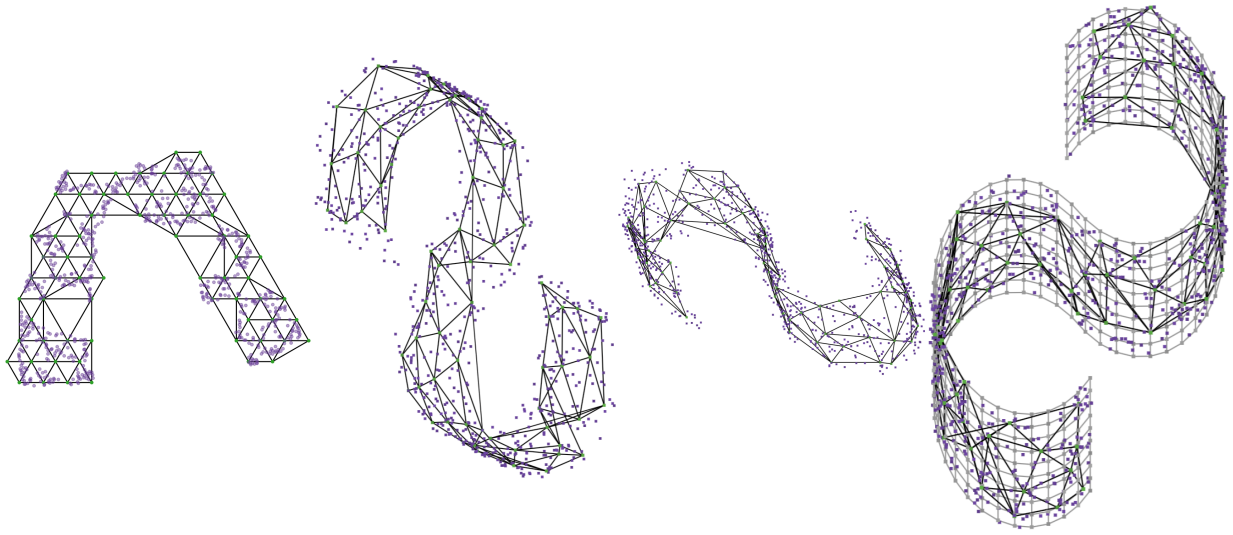


Figure 10: Model in 2-D, on the tSNE layout, and three views of the fit in projections from 7-D, for the S-curve data ($(s_1,\ s_2) = (-0.170,\ -0.137)$, $b = 256$ $(16,\ 16)$, $m = 70$, benchmark value to remove low density hexagons is 0.250, and benchmark value to remove large edges is 0.180). MSE is 0.0432. The model closely fits the shape, but it has some twists (The **langevitour** software is used to view the data with a tour, and the full video is available at https://youtu.be/ZuLdp89qJ6g).

5 A curious difference between tSNE, UMAP and PaCMAP revealer

In this section, the effectiveness of the algorithm is described using a simulated dataset. The dataset consists of five spherical Gaussian clusters in 4-D, with each cluster containing an equal number of points and the same within-cluster variation.

The 2-D layouts generated by tSNE, UMAP, and PaCMAP show five well-separated clusters which evident these methods effectively preserve the global structure. In tSNE (Figure 11 (a)), these clusters appear closely. UMAP arranges all clusters in a parallel manner, with three aligned in one line and the other two in a separate line (Figure 12 (a)). In contrast, PaCMAP shows one central cluster and the remaining four spread out in different directions (Figure 13 (a)).

The tSNE and UMAP shows *filled out* clusters which provide evidence that these methods preserve the local structure (Figure 11 (c) and Figure 12 (c)). On the other hand, PaCMAP shows *flat* shapes clusters in the model and evident that PaCMAP fail to capture the within-cluster variation (Figure 13 (c)).

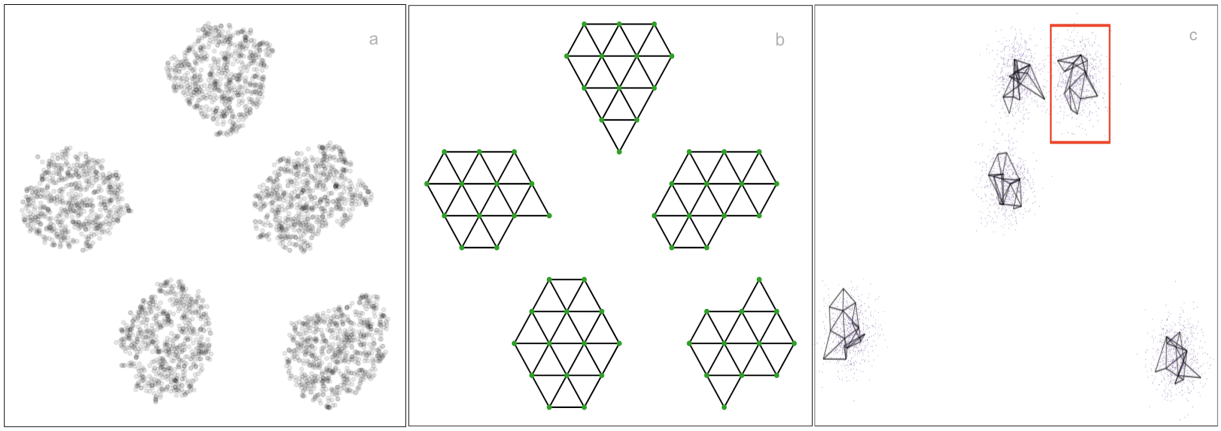


Figure 11: The tSNE layout, model in 2-D, and a view of the fit in projections from 4-D, for the five Gaussian cluster data. The model fits the separation and tries to *filled out* the clusters. (The **langevitour** software is used to view the data with a tour, and the full video is available at https://youtu.be/RASEE7N5MbM).

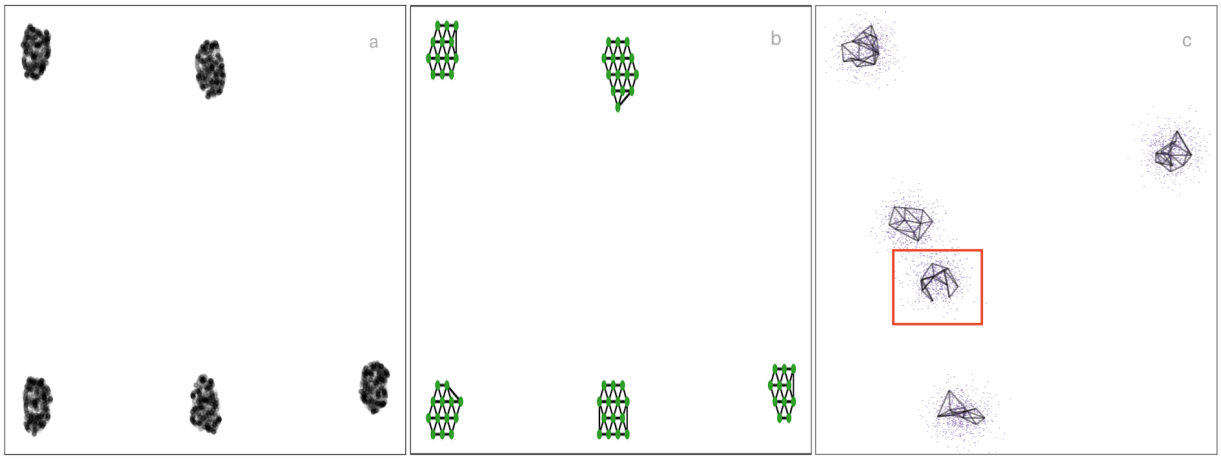


Figure 12: The UMAP layout, model in 2-D, and a view of the fit in projections from 4-D, for the five Gaussian cluster data. The model fits the separation and tries to *filled out* the clusters, but not as much as tSNE. (The **langevitour** software is used to view the data with a tour, and the full video is available at https://youtu.be/iG4bCPkJilw).

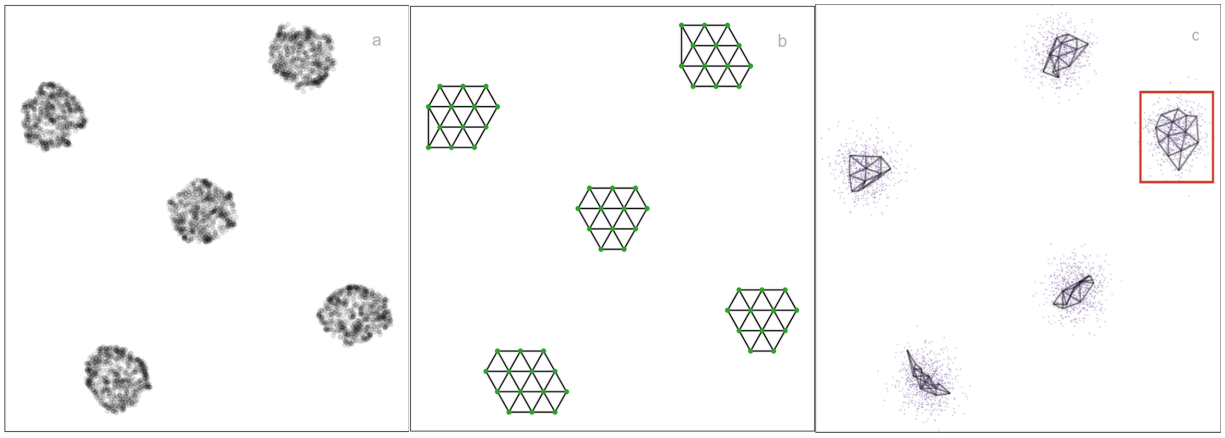


Figure 13: The PaCMAP layout, model in 2-D, and a view of the fit in projections from 4-D, for the five Gaussian cluster data. The model fits the separation and shows *flat* shaped clusters. (The **langevitour** software is used to view the data with a tour, and the full video is available at https://youtu.be/z07cKXi8EJQ).

6 Applications

6.1 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 is commonly used to display clusters, and help to verify the results. For example, Chen et al. (2023) illustrates the use of UMAP to identify clusters in Human Peripheral Blood Mononuclear Cells (PBMC3k). Figure 14 is a reproduction of the published plot. It shows three well-separated clusters.

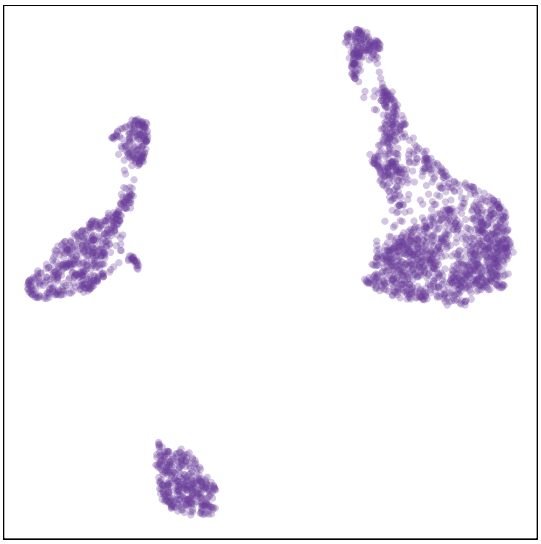


Figure 14: Reproduction of plot published in Chen et al. (2023) showing a 2-D layout from UMAP applied for the PBMC3k dataset. The (hyper-)parameter settings, beyond the defaults, are 30 nearest neighbors and minimum distance equal to 0.3. We use our model-in-the-data-space to assess whether this is an accurate representation of structure present in the high-dimensional data, or if it is misleading.

To determine whether the UMAP representation with the (hyper-)parameter choice suggested by Chen et al. (2023) preserves the original data structure, we visualize the model constructed with UMAP overlaid on the p-D data. The Figure 14 shows three well-separated clusters. However, as shown in Figure 16, there is no big separation between three clusters in p-D. Therefore, the suggested UMAP representation (Figure 14) does not accurately represent the structure(s) present in PBMC3k dataset. But, when visualizing the model-in-th-data-space some unobserved structures can be seen. Some clusters have non-linear continuity patterns and high-density patches (Figure 16).

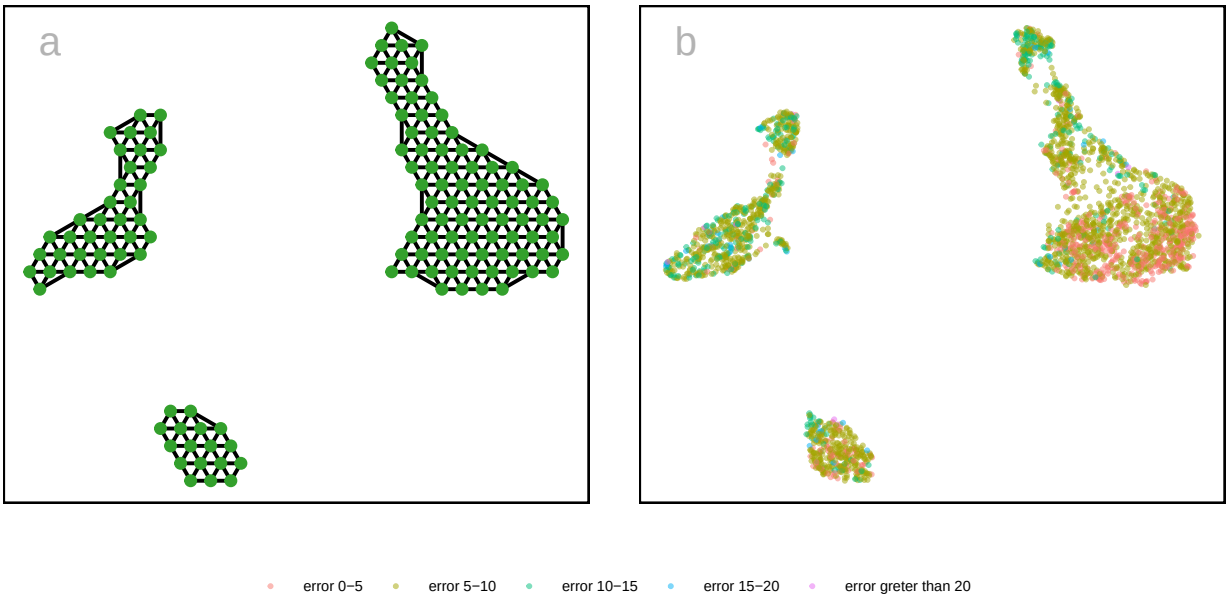


Figure 15: (a) Model generated in 2-D with UMAP, and (b) p-D model error in 2-D. The 2-D model shows three well-separated distant clusters. The p-D model errors are distributed along clusters.

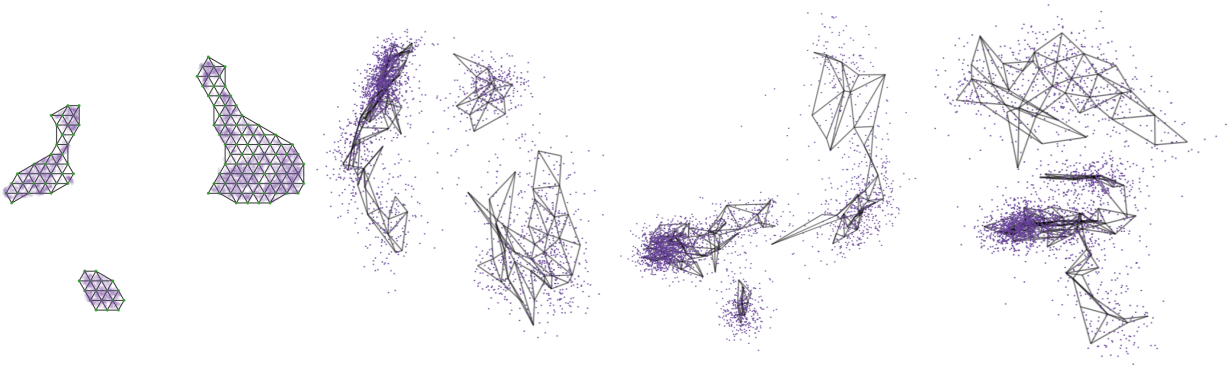


Figure 16: Model in 2-D, on the UMAP layout, and three views of the fit in projections from 9-D, for the PBMC3k data $((s_1,\ s_2) = (-0.050,\ -0.041)$, $b = 870\ (30,\ 29)$, $m = 135$, benchmark value to remove large edges is 0.099). (The **langevitour** software is used to view the data with a tour, and the full video is available at https://youtu.be/VqqWuE0Jj6A).

In order to find a reasonable NLDR representation for the PBMC3k dataset, the absolute error for different numbers of non-empty bins using various NLDR techniques and different (hyper-)parameter settings (Figure 17) were calculated. After analyzing the results, it was found that tSNE with default (hyper-)parameter setting (perplexity: 30) achieved the lowest error when there were 136 number of non-empty bins. Therefore, tSNE with a perplexity value set to 30, the default parameter setting, is considered as a reasonable representation for the PBMC3k dataset.

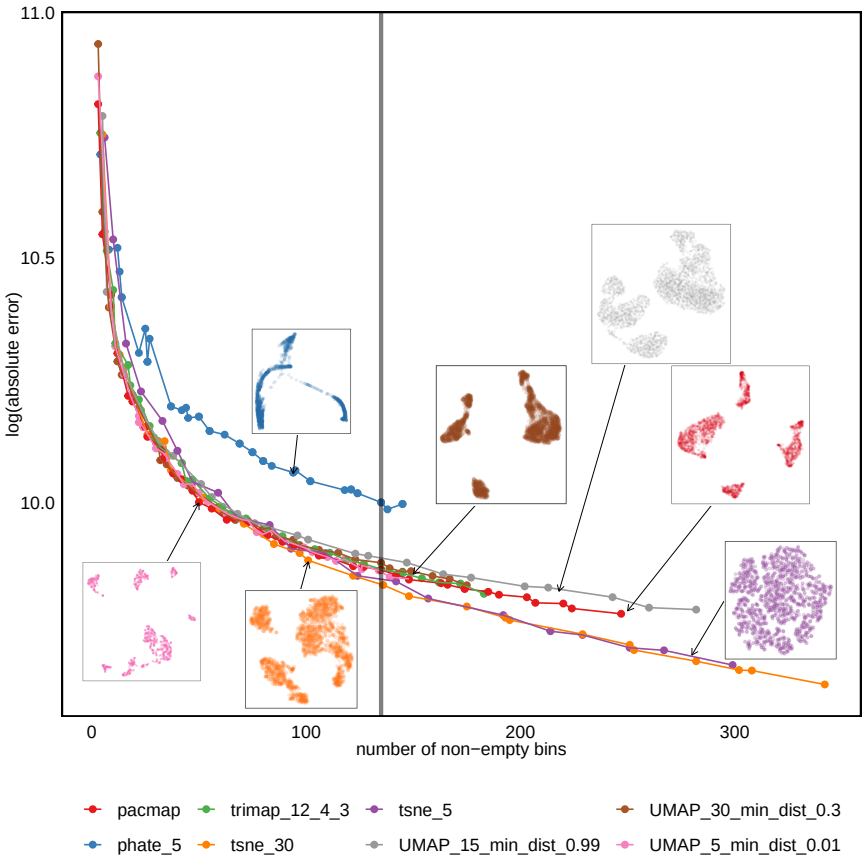


Figure 17: Absolute error from UMAP and tSNE applied to training PBMC3k dataset with diffferent parameter choices. What is the best parameter choice to create the model? The residual plot have a steep slope at the beginning, indicating that a smaller number of non-empty bins causes a larger amount of error. Then, the slope gradually declines or level off, indicating that a higher number of non-empty bins generates a smaller error. Using the elbow method, it was observed that when the number of non-empty bins is set to 136, the lowest error occurred with the parameters perplexity: 30.

As shown in Figure 18, there are three well-separated clusters, although they are located close to each other. Additionally, non-linear structures can also be observed within the clusters (Figure 18). This demonstrates that tSNE accurately captures the data structure of the PBMC3k dataset, which UMAP did not achieve.

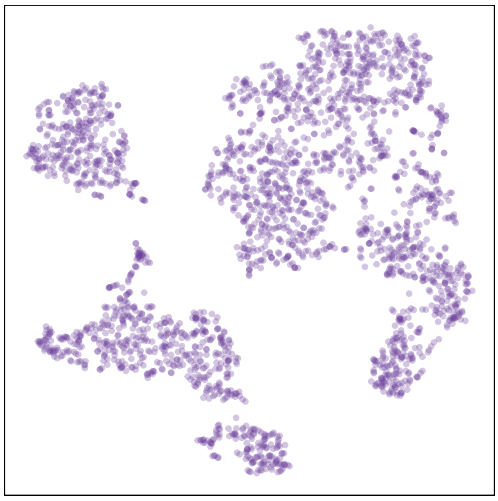


Figure 18: The 2-D layout from tSNE applied for the PBMC3k dataset. The default (hyper-)parameter setting is a perplexity 30. We use our model-in-the-data-space to assess whether this is an accurate representation of structure present in the high-dimensional data, or if it is misleading.

We then fit the model for tSNE, and visualize the resultant model in the p-D data space. The model shows a quirk, as shown in Figure 19. All three clusters are connected by an edge except the small and large clusters. Because the clusters are so close in 2-D, they attempt to maintain the structure in p-D as well. This is evident that tSNE with perplexity 30 provides a reasonable representation of PBMC3k data.

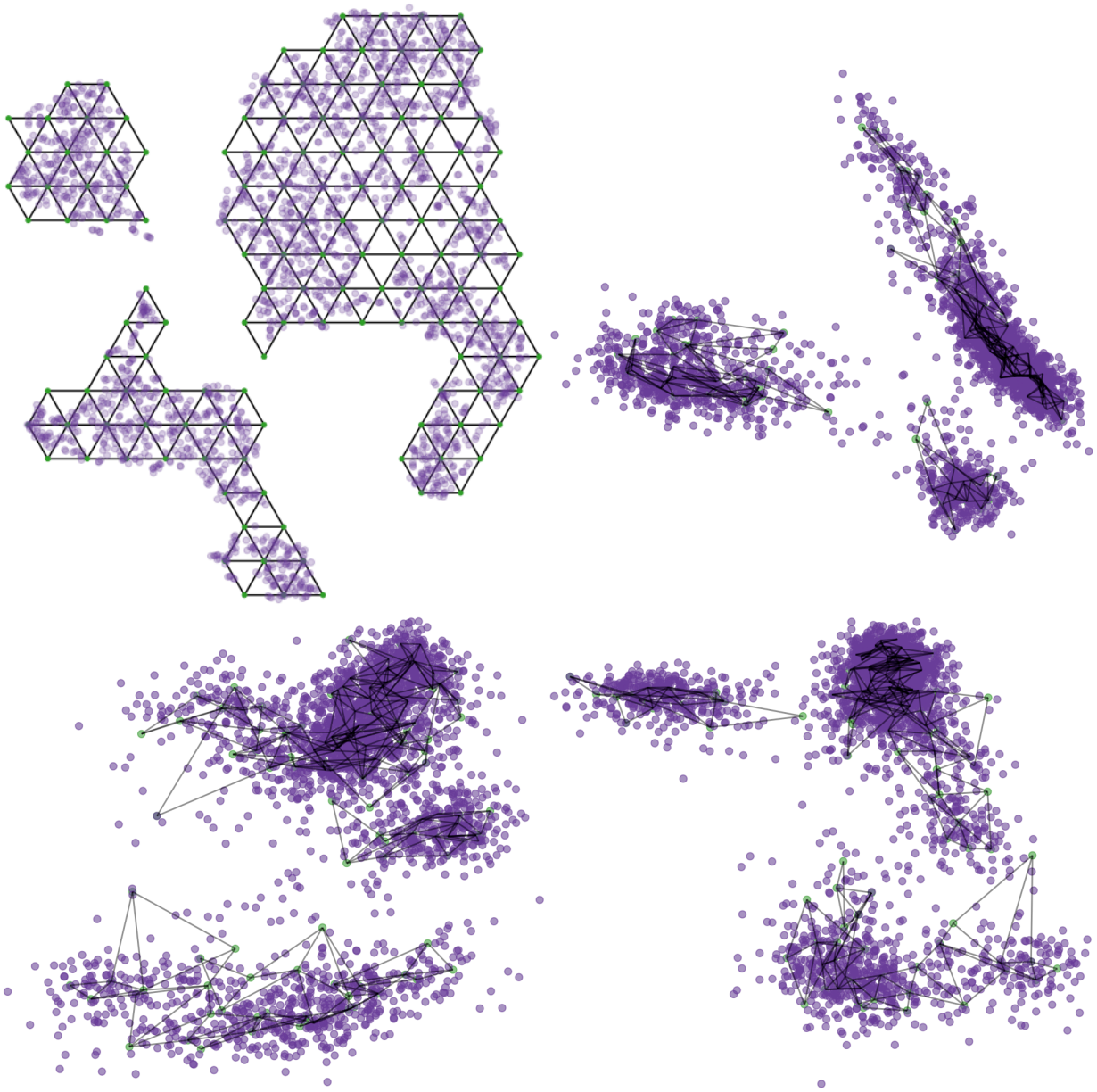


Figure 19: Model in 2-D, on the tSNE layout, and three views of the fit in projections from 9-D, for the PBMC3k data ($(s_1,\ s_2) = (-0.050,\ -0.058)$, $b = 300$ $(15,\ 20)$, $m = 136$, benchmark value to remove large edges is 0.133). (The **langevitour** software is used to view the data with a tour, and the full video is available at https://youtu.be/5Y1hE4i7N2k).

6.2 Hand-written digits

The MNIST dataset consists of grayscale images of handwritten digits (LeCun & Cortes 2010). Wang et al. (2021) used this dataset to demonstrate how PaCMAP preserves non-linear structure in p-D. To evaluate whether PaCMAP provides a reasonable representation of the data, the 2-D embedding of the handwritten digit 1 was selected. As shown in Figure 20, the angle of the digit 1 images varies along the 2-D structure.

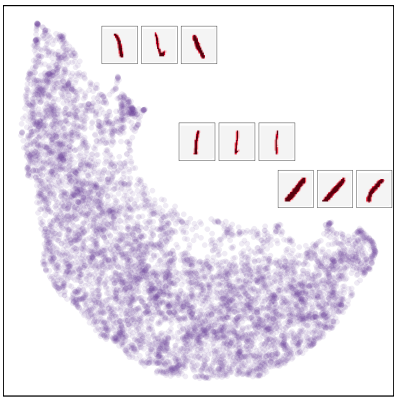


Figure 20: 2-D layout from PaCMAP applied for the digit 1 of the MNIST dataset. The (hyper-)parameter settings, beyond the defaults, are 10 nearest neighbors, ratio of the number of mid-near pairs to the number of neighbors equal to 0.5, and the ratio of the number of further pairs to the number of neighbors is 2. We use our model-in-the-data-space to assess whether this is an accurate representation of structure present in the high-dimensional data, or if it is misleading. The angle of the digit 1 varies along this structure. Images at the top-left of the 2-D layout show the digit 1 angled more to the left, while images at the bottom-right show the digit 1 angled more to the right.

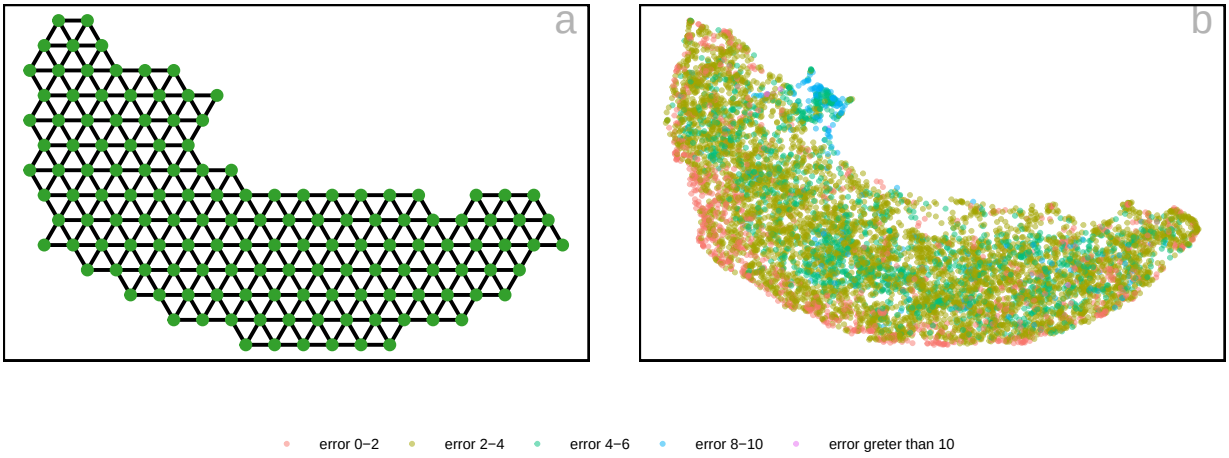
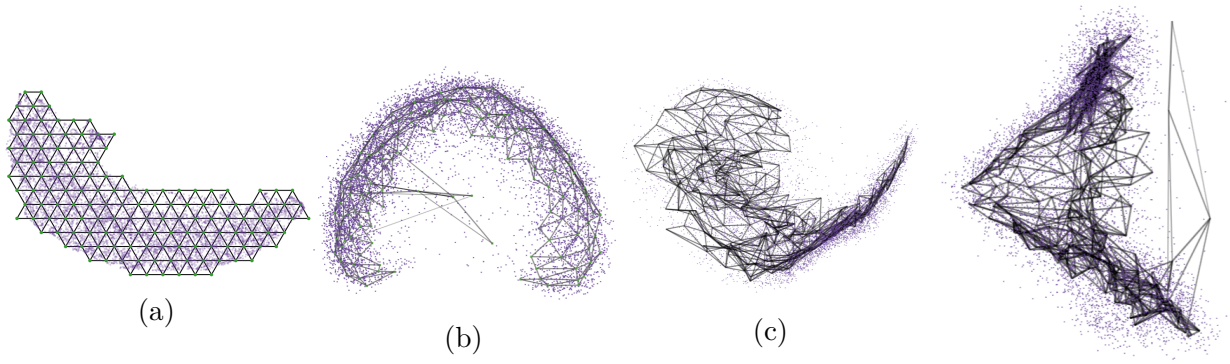


Figure 21: (a) Model generated in 2-D, and (b) p-D model error in 2-D. The 2-D model shows a non-linear continuous structure. Most low p-D model errors are distributed along the lower edge of the 2-D structure, while most high p-D model errors are concentrated along the upper edge.

According to Figure 22b, the non-linear continuous structure observed in the 2-D representation of PaCMAP (Figure 20) is also visible when visualizing the model overlaid on the data space. This indicates that PaCMAP accurately captures the structure of the p-D data. Additionally, the model shows a twisted pattern within the non-linear structure in p-D space (Figure 22c), which is an additional pattern not visible in the 2-D representation (Figure 20). Furthermore, as shown in Figure 22d, some long edges exist in the p-D space that are not recognized as long edges in the 2-D representation. However, PaCMAP is

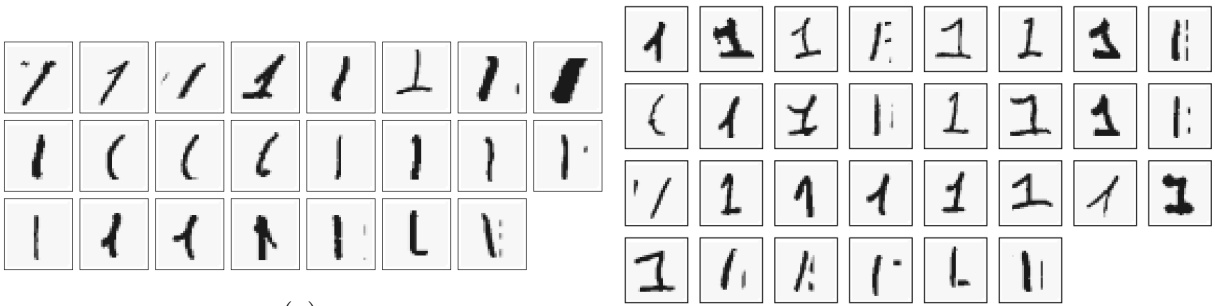
a reasonable 2-D representation of MNIST digit 1 data, because PaCMAP preserves the non-linear structure present in the p-D data.



(a) (b) (c) (d)

Figure 22: Model in 2-D, on the PaCMAP layout, and three views of the fit in projections from 10-D, for the digit 1 of MNIST data $((s_1,\ s_2) = (-0.100,\ -0.059)$, $b = 374\ (22,\ 17)$, $m = 140$, benchmark value to remove large edges is 0.094). (The **langevitour** software is used to view the data with a tour, and the full video is available at https://youtu.be/zcg_GXBmqjA).

There are certain data points that exhibit high error rates due to their deviation from the usual p-D data structure, which makes them anomalies (Figure 21 b). These anomalies can be classified into two types: those that are anomalies within the non-linear structure and those that lie outside of it. The images associated with high model error points within the non-linear structure display different patterns of the digit 1, as shown in Figure 23a. However, when comparing these images to the ones found outside of the non-linear structure, it becomes evident that the latter display different patterns of the digit 1 (Figure 23b).



(a) (b)

Figure 23: Some images of handwritten digit 1 which occur high model error (a) within the non-linear structure, and (b) outside the non-linear structure. The images shows different patterns of digit 1.

7 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 a reasonable 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 S-curve 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 **pbmc** 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.

8 Supplementary Materials

Code, and data for reproducing this paper are available at https://github.com/JayaniLakshika/paper-nldr-vis-algorithm.

References

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.
URL: *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.
URL: *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* .
URL: *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.
URL: *https://doi.org/10.1007/BF01025868*

Gebhardt, A., Bivand, R. & Sinclair, D. (2024), *interp: Interpolation Methods*. R package version 1.1-6.
URL: *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.
URL: *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.
URL: *https://royalsocietypublishing.org/doi/abs/10.1098/rsta.2009.0159*

Jolliffe, I. (2011), *Principal Component Analysis*, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 1094–1096.
URL: *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.
URL: *https://doi.org/10.1007/BF02289343*

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.
URL: *https://doi.org/10.1080/10618600.2021.1963264*

LeCun, Y. & Cortes, C. (2010), 'MNIST handwritten digit database'.
URL: *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.
URL: *https://doi.org/10.1007/BF00977785*

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

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.

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

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.
URL: *http://jmlr.org/papers/v22/20-1061.html*

Wickham, H., Cook, D. & Hofmann, H. (2015), 'Visualizing statistical models: Removing

the blindfold', *Statistical Analysis and Data Mining: The ASA Data Science Journal* **8**(4), 203–225.
URL: *https://onlinelibrary.wiley.com/doi/abs/10.1002/sam.11271*

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