Journal of Data Science, Statistics, and Visualisation

MMMMMMM YYYY, Volume VV, Issue II. *doi: XX.XXXXX/jdssv.v000.i00*

Casting Multiple Shadows: High-Dimensional Interactive Data Visualisation with Tours and Embeddings

Stuart Lee
Monash University

Ursula Laa
Monash University

Dianne Cook
Monash University

Abstract

Non-linear dimensionality reduction (NLDR) methods such as t-distributed stochastic neighbour embedding (t-SNE) are ubiquitous in the natural sciences. However, the appropriate use of these methods is difficult because of their complexity; analysts must make trade-offs in order to identify structure in the visualisation of an NLDR technique. We present visual diagnostics for the usage of NLDR methods by combining them with a technique called the tour. A tour is a sequence of interpolated linear projections of multivariate data onto a lower dimensional space. The sequence is displayed as a dynamic visualisation, allowing a user to see the shadows the high-dimensional data casts in a lower dimensional view. By linking the tour to an NLDR view, we can preserve global structure and through user interactions like linked brushing observe where the NLDR view may be misleading. We display several case studies from both simulations and single cell transcriptomics, that shows our approach is useful for cluster orientation tasks and for correcting an NLDR embedding. The implementation of our framework is available as an R package called **liminal**.

Keywords: dimensionality reduction, high-dimensional data, interactive graphics, t-SNE, grand tour, R.

1. Introduction

High-dimensional data is increasingly prevalent in the natural sciences and beyond but presents a challenge to the analyst in terms of data cleaning, pre-processing and visualization. Methods to embed data from a high-dimensional space into a low-dimensional one now form a core step of the data analysis workflow where they are used to ascertain hidden structure and de-noise data for downstream analysis.

Choosing an appropriate embedding presents a challenge to the analyst. How does an analyst know whether the embedding has captured the underlying topology and geometry of the high-dimensional space? The answer depends on the analyst's workflow. Brehmer et al. (2014) characterized two main workflow steps that an analyst performs when using embedding techniques: dimension reduction and cluster orientation. The first relates to dimension reduction achieved by using an embedding method, here an analyst wants to characterize and map meaning onto the embedded form, for example identifying batch effects from a high throughput sequencing experiment, or identifying a gradient or trajectory along the embedded form like changes in cell development or species abundance (Nguyen and Holmes 2019). The second relates to using embeddings as part of a clustering workflow. Here analysts are interested in identifying and naming clusters and verifying them by either applying known labels or coloring by variables that are a-priori known to distinguish clusters like the expression of a marker gene to identify a cell type. Once clusters are identified they are used for further analysis to identify what features in the data make them distinguishable. Both of these workflow steps rely on the embedding being representative of the original high-dimensional dataset, and becomes much more difficult when there is no underlying ground truth.

As part of a visualization workflow, it is important to consider the perception and interpretation of embedding methods as well. Sedlmair et al. (2013) showed that scatter plots were mostly sufficient for detecting class separation, however, they also noted that often multiple embeddings were required. For the task of cluster identification, Lewis et al. (2012) showed experimentally that novice users of non-linear embedding techniques were more likely to consider clusters of points on a scatter plot to be the result of a spurious embedding compared to advanced users who were aware of the inner workings of the embedding algorithm.

There is no one-size fits all; finding an appropriate embedding for a given dataset is a difficult and a somewhat poorly defined problem. For non-linear methods, there are a lot of parameters to explore that can have an effect on the resulting visualization and interpretation. While there has been much work on the algorithmic details of embedding methods; there are relatively few tools designed to assist users to interact with these techniques: when is an embedding sufficient for the task at hand? Several interactive interfaces have been proposed for evaluating or using embedding techniques. Buja et al. (2008) used tours to guide analysts during the optimization of multidimensional scaling methods by extending their interactive visualization software called **XGobi** and **GGobi** into a new tool called **GGvis** (Swayne et al. 1998, 2003; Swayne and Buja 2004). Their interface allows the analyst to dynamically modify and check whether an MDS configuration has preserved the locality and closeness of points between the configuration and the original data. Ovchinnikova and Anders (2020) created the **Sleepwalk** interface for checking non-linear embeddings in single cell RNA-seq data. It provides a click and highlight visualization for coloring points in an embedding according to an estimated pairwise distance in the original high-dimensional space. Similarly, the **TensorFlow** embedding projector is a web interface to running some non-linear embedding methods live in the browser and it provides interactions to color points, and select nearest neighbors (Smilkov et al. 2016). Finally, the work by Pezzotti et al. (2017) provides a user guided and modified form of the t-SNE algorithm, that allows users to modify optimization parameters in real-time.

A complementary approach for visualizing structure in high-dimensional data is the tour. A tour is a sequence of projections of a high-dimensional dataset onto a low-dimensional basis matrix, and is represented as an animated visualization (Asimov 1985; Buja and Asimov 1986). Given the dynamic nature of the tour, user interaction is important for controlling and exploring the visualization; the tour has been used previously by Wickham et al. (2015) for exploring statistical model fits and by Buja et al. (1996) for exploring the space of factorial experimental designs.

The approach used in this paper is to augment the results of an NLDR method with the tour with ourR package called **liminal**. Interfaces for evaluating embeddings require interaction but should also be able to be incorporated into an analysts workflow. Here we implement a more pragmatic workflow by incorporating interactive graphics and tours with embeddings that allows users to see a global overview of their high-dimensional data, allowing them to adjust an NLDR view and assist them with cluster orientation tasks.

The rest of the paper is organized as follows. The next section provides background on dimension reduction methods, including an overview of the tour. Then we describe the visual design of **liminal**, followed by implementation details. Next we provide case studies that show how our interface assists in using embedding methods. Finally, we describe the insights gained by using **liminal** and plans for extensions to the software. Throughout the paper when we refer to high-dimensional data, we mean it in a broad sense not specifically referring to the case of small N large p data.

2. Overview of Dimension Reduction

To begin, we suppose the data is in the form of a rectangular matrix of real numbers, $X = [\mathbf{x_1}, \dots, \mathbf{x_n}]$, where n is the number of observations in p dimensions. The purpose of any dimension reduction (DR) algorithm is to find a low-dimensional representation of the data, $Y = [\mathbf{y_1}, \dots, \mathbf{y_n}]$, such that Y is an $n \times d$ matrix where $d \ll p$. The hope of the analyst is that the DR procedure to produce Y will remove noise in the original dataset while retaining any latent structure.

DR methods can be classified into two broad classes: linear and non-linear methods. Linear methods perform a linear transformation of the data, that is, Y is a linear transformation of X. One example is principal components analysis (PCA) which performs an eigen-decomposition of the estimated sample covariance matrix. The eigenvalues are sorted in decreasing order and represent the variance explained by each component (eigenvector). A common approach to deciding on the number of principal components to retain is to plot the proportion of variance explained by each component and choose a cut-off. When working with linear transformations, we often need more than two dimensions to capture the latent structure. In this case we can use tour methods (Asimov 1985; Buja and Asimov 1986) to show interpolated sequences of projections, providing the viewer with intuition about structure in more than two dimensions, as described below.

For non-linear methods, Y is generated via a pre-processed form of the input X such as the k-nearest neighbors graph or via a kernel transformation. Multidimensional scaling (MDS) is a class of DR methods that aims to construct an embedding Y such that the pair-wise distances (inner products) in Y approximate the pair-wise distances (inner products) in X (Torgerson 1952; Kruskal 1964a). There are many variants of MDS, such as non-metric scaling which amounts to replacing distances with ranks instead (Kruskal 1964b). A related technique is Isomap which uses a k-nearest neighbor graph to estimate the pair-wise geodesic distance of points in X then uses classical MDS to construct Y (Silva and Tenenbaum 2003). Other approaches are based on diffusion processes such as diffusion maps (Coifman et al. 2005). A recent example of this approach is the PHATE algorithm (Moon et al. 2019).

A general difficulty with using non-linear DR methods for exploratory data analysis is selecting and tuning appropriate parameters. For concreteness here, we focus on t-distributed stochastic neighbor embedding (t-SNE), and we will examine its underpinning in some detail below (van der Maaten and Hinton 2008). Similar considerations hold for related methods, for example UMAP (McInnes et al. 2018).

2.1. Introduction to t-SNE

The t-SNE algorithm estimates the pair-wise similarity of points in a high dimensional space based on their (Euclidean) distances using a Gaussian distribution. A configuration in the low dimensional embedding space is then estimated by modelling similarities using a t-distribution with 1 degree of freedom (van der Maaten and Hinton 2008). There are several subtleties of the algorithm that are revealed by stepping through its machinery.

To begin, t-SNE transforms pair-wise distances between $\mathbf{x_i}$ and $\mathbf{x_j}$ to similarities using a Gaussian kernel:

$$p_{i|j} = \frac{\exp(-\|\mathbf{x_i} - \mathbf{x_j}\|^2/2\sigma_i^2)}{\sum_{k\neq i}\exp(-\|\mathbf{x_j} - \mathbf{x_k}\|^2/2\sigma_i^2)}.$$

The conditional probabilities are then normalized and symmetrized to form a joint probability distribution via averaging:

$$p_{ij} = \frac{p_{i|j} + p_{j|i}}{2n}.$$

The variance parameter of the Gaussian kernel is controlled by the analyst using a fixed value of perplexity for all observations:

$$\text{perplexity}_i = \exp(-\log(2)\sum_{i\neq j} p_{j|i}\log_2(p_{j|i})).$$

As the perplexity increases, σ_i^2 increases, until its bounded above by the number of observations , $n-1$, in the data, corresponding to $\sigma_i^2 \to \infty$. This essentially turns t-SNE into a nearest neighbors algorithm, $p_{i|j}$ will be close to zero for all observations that are not in the $\mathcal{O}(\text{perplexity}_i)$ neighborhood graph of the ith observation (van der Maaten 2014).

Next, in the target low-dimensional space, Y, pair-wise distances between $\mathbf{y_i}$ and $\mathbf{y_j}$ are modeled as a symmetric probability distribution using a t-distribution with one degree of freedom (Cauchy kernel):

$$q_{ij} = \frac{w_{ij}}{Z} \text{where } w_{ij} = \frac{1}{1 + \|\mathbf{y_i} - \mathbf{y_j}\|^2} \text{ and } Z = \sum_{k \neq l} w_{kl}.$$

The resulting embedding Y is the one that minimizes the Kullback-Leibler divergence between the probability distributions formed via similarities of observations in X, $\mathcal{P}$ and similarities of observations in Y, $\mathcal{Q}$:

$$\mathcal{L}(\mathcal{P}, \mathcal{Q}) = \sum_{i \neq j} p_{ij} \log \frac{p_{ij}}{q_{ij}}.$$

Re-writing the loss function in terms of attractive (right) and repulsive (left) forces we obtain:

$$\mathcal{L}(\mathcal{P}, \mathcal{Q}) = -\sum_{i \neq j} p_{ij} \log w_{ij} + \log \sum_{i \neq j} w_{ij}.$$

Minimizing the loss function corresponds to large attractive forces, that is, the pair-wise distances in Y are small when there are non-zero p_{ij}, i.e., x_i and x_j are close together. The repulsive force should also be small, that is, overall the pair-wise distances in Y should be large regardless of the magnitude of the corresponding distances in X. As a result, clusters that are separate in X will be placed far from each other in Y. This minimization is done via stochastic gradient decent, and introduces a number of hyperparameters, for example, the number of iterations, the learning rate, and early exaggeration, a multiplier of the attractive force used at the beginning of the optimization.

Taken together, these details reveal the sheer number of decisions that an analyst must make. How does one choose the perplexity and the parameters that control the optimization of the loss function? It is a known problem that t-SNE can have trouble recovering topology and that configurations can be highly dependent on how the algorithm is initialized and parameterized (Wattenberg et al. 2016; Kobak and Berens 2019; Melville 2020). If the goal is cluster orientation a recent theoretical contribution by Linderman and Steinerberger (2019) proved that t-SNE can recover spherical and well separated cluster shapes, and proposed new approaches for tuning the optimization parameters. However, the cluster sizes and their relative orientation from a t-SNE view can be misleading perceptually, due to the algorithms emphasis on locality.

Another recent method, UMAP, has seen a large rise in popularity (at least in single cell transcriptomics) (McInnes et al. 2018). It is a method that is related to LargeVis (Tang et al. 2016), and like t-SNE acts on the k-nearest neighbor graph. Its main differences are that it uses a different cost function (cross entropy) which is optimized using stochastic gradient descent and defines a different kernel for similarities in the low dimensional space. Due to its computational speed it is possible to generate UMAP embeddings in more than three dimensions. It appears to suffer from the same perceptual issues as t-SNE, however, it supposedly preserves global structure better than t-SNE (Coenen and Pearce 2019).

2.2. Tours explore the subspace of low dimensional projections

The tour is a visualization technique that is grounded in mathematical theory, and allows the viewer to ascertain the shape and global structure of a dataset via inspection of the subspace generated by the set of low-dimensional projections (Asimov 1985; Buja and Asimov 1986).

As with other DR techniques, the tour assumes we have a real data matrix X consisting of n observations in p dimensions. First, the tour generates a sequence of $p \times d$ orthonormal projection matrices (bases) A_t, where d is typically 1 or 2. For each pair of orthonormal bases A_t and A_{t+1} that are generated, the geodesic path between them is interpolated to form intermediate frames, giving the sense of continuous movement from one basis to another. The tour is then the continuous visualization of the projected data $Y_t = XA_t$, that is the projection of X onto A_t as the tour path is interpolated between successive bases.

A *grand tour* corresponds to choosing new orthonormal bases at random; allowing a user to ascertain structure via exploring the subspace of d-dimensional projections. In practice, we first form our data into a sphere via principal components to reduce dimensionality of X prior to running the tour. Instead of picking projections at random, a *guided tour* can be used to generate a sequence of ‘interesting’ projections as quantified by an index function (Cook et al. 1995). While our software,**liminal** is able to visualize guided tours, our focus in the case studies uses the grand tour to see global structure in the data.

3. Visual Design

Tours provide a supportive visualization to NLDR graphics, and can be easily incorporated into an analysts workflow with our software package, **liminal**. Our interface allows analysts to quickly compare views from embedding methods and see how an embedding method preserves or alters the geometry of their data. Using multiple concatenated and linked views with the tour enhances interaction techniques, and allows analysts to perform cluster orientation tasks via linked highlighting and brushing (McDonald 1982; Becker and Cleveland 1987). This approach allows our interface to achieve the three principles for interactive high-dimensional data visualization outlined by Buja et al. (1996): finding gestalt (identifying patterns in visual forms), posing queries, and making comparisons.

3.1. Finding Gestalt: focus and context

To understand the data structure, we look for Gestalt features such as (non-)linearities, clusters or outliers. A tour display is preferred for this task, since it accurately captures the geometry, while NLDR methods typically introduce distortions.

To investigate latent structure and the shape of a high-dimensional dataset in **liminal**, a tour can be run without the use of an external embedding. It is often useful to first run principal components on the input as an initial dimension reduction step, and then tour a subset of those components instead, i.e., by selecting them from a scree plot. The default tour layout is a scatter plot with an axis layout displaying the magnitude and direction of each basis vector. Since the tour is dynamic, it is useful to be able to pause and highlight a particular view. In addition to pause, play and reset buttons, brushing will pause the tour path, allowing users to identify 'interesting' projections. The domain of the axis scales from running a tour is called the half range, and is computed by rescaling the input data onto d-dimensional unit cube. We bind the half range to a mouse wheel event, allowing a user to pan and zoom on the tour view dynamically. This is useful for peeling back dense clumps of points to reveal structure.

3.2. Posing Queries: multiple views, many contexts

The initial visualization gives an overview of the data structure, and naturally leads to queries that investigate observed features with the aim to further characterize them. This is an essential aspect of our framework, where we use a tour to better characterize features observed in a NLDR display.

We have combined the tour view in a side by side layout with a scatter plot view as has been done in previous tour interfaces **XGobi** and **DataViewer** (Buja et al. 1986; Swayne et al. 1998). These views are linked; analysts can brush regions or highlight collections of points in either view. Linked highlighting can be performed when points have been previously labelled according to some discrete structure, i.e., cluster labels are available. This is achieved via the analyst clicking on groups in the legend, which causes unselected groupings to have their points become less opaque. Consequently, simple linked highlighting can alleviate a known downfall of methods such as UMAP or t-SNE; that is, distances between clusters are misleading. By highlighting corresponding clusters in the tour view, the analyst can see the relationship between clusters, and therefore obtain a more accurate representation of the topology of their data.

Simple linked brushing is achieved via mouse-click and drag movements. By default, when brushing occurs in the tour view, the current projection is paused and corresponding points in the embedding view are highlighted. Likewise, when brushing occurs in the embedding view, corresponding points in the tour view are highlighted. In this case, an analyst can use brushing for manually identifying clusters and verifying cluster locations and shapes; brushing in the embedding view gives analysts a sense of the shape and proximity of cluster in high-dimensional space.

3.3. Making comparisons: revising embeddings

Combining multiple views of a single data set allows the analyst to make meaningful comparisons. In **liminal** the embedding and tour views are arranged side-by-side for direct cross-checks.

As mentioned previously, when using any DR method, we are assuming the embedding is representative of the high-dimensional dataset it was computed from. Defining what it means for embeddings to be ‘representative‘ or ’faithful’ to high-dimensional data is ill-posed and depends on the underlying task an analyst is trying to achieve. At the very minimum, we are interested in distortions and diffusions of the high-dimensional data. Distortions occur when points that are near each other in the embedding view are far from each other in the original dataset. This implies that the embedding is not continuous. Diffusions occur when points are far from each other in the embedding view are near in the original data. Whether, points are near, or far is reliant on the distance metric used; distortions and diffusions can be thought of as the preservation of distances or the nearest neighbors graphs between the high-dimensional space and the embedding space. As distances can be noisy in high-dimensions, ranks can be used instead as has been proposed by Lee and Verleysen (2009). Identifying distortions and diffusions allows an analyst to investigate the quality of their embedding and revise them iteratively.

These checks are done visually using our side-by-side tour and embedding views. In the simplest case, a local continuity check can be assessed via one to one linked brushing from the embedding to the tour view. Similarly, diffusions are identified from linked brushing on the tour view, highlighting points in the embedding view.

4. Software Infrastructure

We have implemented the above design as an open source R package called **liminal** (Lee and Cook 2020). The package allows analysts to construct concatenated visualizations, drawn with the **Vega-Lite** grammar of interactive graphics via the **vegawidget** package (Satyanarayan et al. 2017; Lyttle and Vega/Vega-Lite Developers 2020). It provides an interface for constructing linked and stand alone interfaces for manipulating tour paths via the **shiny** and **tourr** packages (Chang et al. 2020; Wickham et al. 2011).

4.1. Tours as a streaming data problem

The process of generating successive bases and interpolating between them to construct intermediary frames, means the tour is a dynamic visualization technique. Generally, the user would set $d = 2$ and the tour is visualized as an animated scatter plot. This process of constructing bases and intermediate frames and visualizing the resulting projection is akin to making a “flip book” animation. Like with a flip book, an interface to the tour requires the ability to interact and modify it in real time. The user interface generated in **liminal** allows a user to play, pause, and reset the tour animation, panning and zooming to modify the scales of the plot to provide context and click events to highlight groups of points if a labeling variable has been placed on the legend.

These interactions are enabled by treating the basis generation as a reactive stream. Instead of realizing the entire sequence, which limits the animation to have a discrete number of frames, new bases and their intermediate frames are generated dynamically via pushing the current projection to the visualization interface. The interface listens to events like pressing a button or mouse-dragging and reacts by pausing the stream.

This process allows the user to manipulate the tour in real time rather than having to fix the number of bases ahead of time. Additionally, once the user has identified an interesting projection or is done with the tour, the interface will return the current basis (as a matrix) for use downstream.

4.2. Linking and highlighting views via interactions

The embedding and tour views are linked together via rectangular brushes; when a brush is active, points will be highlighted in the adjacent view. Because the tour is dynamic, brush events that become active will pause the animation, so that a user can interrogate the current view. By default, brushing on the embedding view will produce a one-to-one linking with the tour view. For interpreting specific combinations of clusters, the multiple guides on the legend can be selected in order to see their relative orientations. The interface is constructed as a **shiny** gadget specifically designed for interactive data analysis. Selections such as brushing regions and the current tour path are returned after the user clicks done on the interface and become available for further investigation.

5. Case Studies

The next section steps through case studies of our approach using simulations and an application to single cell RNA-seq data.

The first three case studies use simulations where the cluster structure and geometry of the underlying data is known. We start with a simple example where we generated spherical clusters that are embedded well by t-SNE. Then we move onto more complex examples where the tour provides insight, such as clusters that have substructure and where there is more complex geometry in the data.

In the final case study, we apply our approach to clustering the mouse retina data from Macosko et al. (2015), and apply the tour to the process of verifying marker genes that separate clusters.

We *strongly* recommend viewing the linked videos for each case study while reading. Links to the videos are available in table 1 and in the figures for each case study. The videos presented show the visual appearance of the **liminal** interface, and how we can interact with the tour via the controls previously described. If you are unable to view the videos, the figures in each case study consist of screenshots that summarize what is learned from combining the tour and an embedding view.

5.1. Case Study 1: Exploring spherical Gaussian clusters

To begin, we look at simulated datasets that reproduce known facts about the t-SNE algorithm. Our first data set consists of five spherical 5-d Gaussian clusters embedded in 10-d space, each cluster has the same covariance matrix. We then computed a t-SNE layout with default settings using the **Rtsne** package (Krijthe 2015), and set up the **liminal** linked interface with grand tour on the 10-d observations.

From the video linked in Figure 1, we learn that t-SNE has correctly split out each cluster and laid them out in a star like formation. This agrees with the tour view,

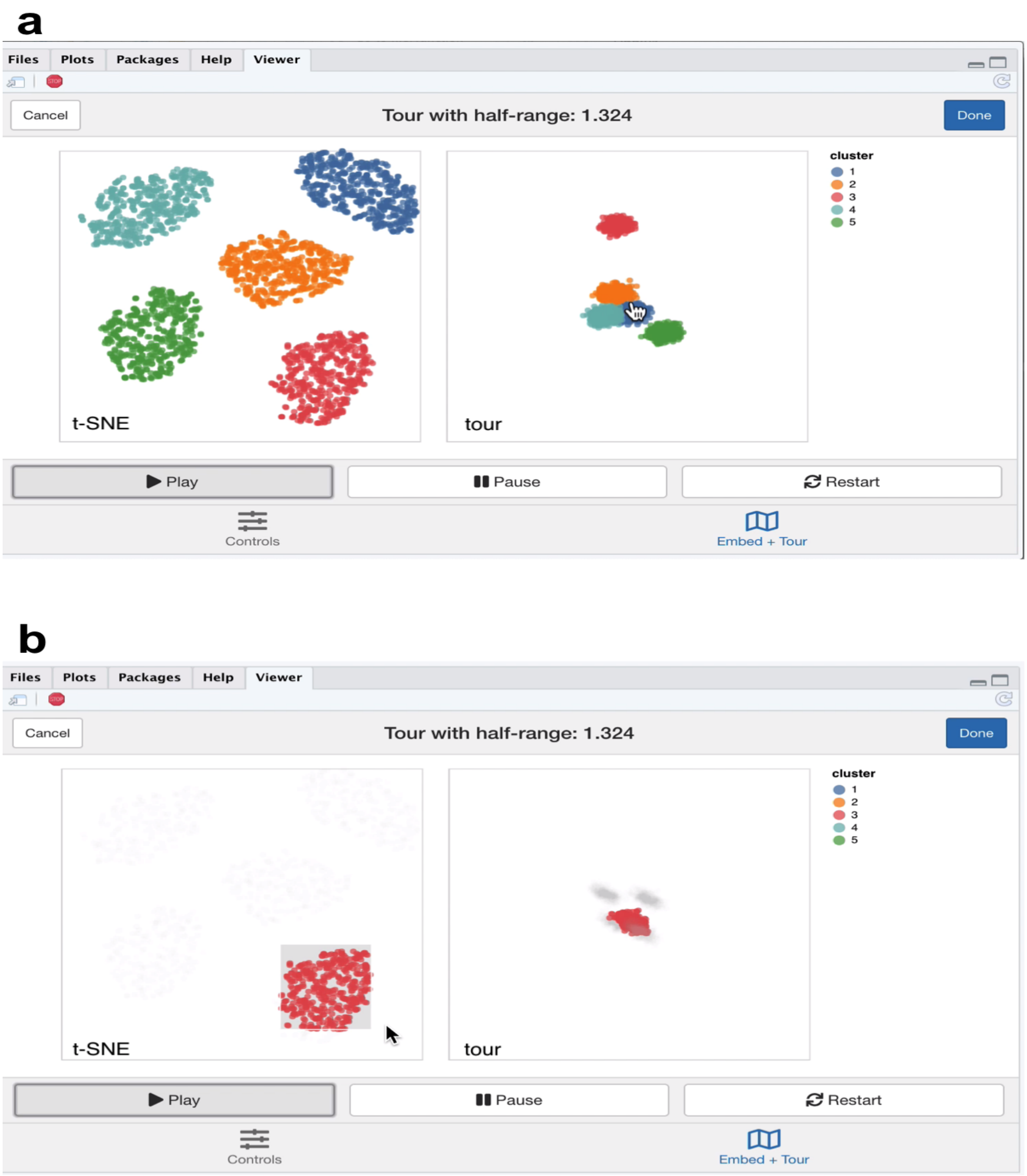

Figure 1: Screenshots of the **liminal** interface applied to well clustered data, a video of the tour animation is available at `https://player.vimeo.com/video/439635921`.

where once we start the animation, the five clusters begin to appear but generally points are more concentrated in the projection view compared to the t-SNE layout (Figure 1a). This can be seen via brushing the t-SNE view (Figure 1b).

5.2. Case Study 2: Exploring spherical Gaussian clusters with hierarchical structure

Next, we view Gaussian clusters from the *Multi Challenge Dataset*, a benchmark simulation data set for clustering tasks (Rauber 2009). This dataset has two Gaussian clusters with equal covariance embedded in 10-d, and a third cluster with hierarchical structure. This cluster has two 3-d clusters embedded in 10-d, where the second cluster is subdivided into three smaller clusters, that are each equidistant from each other and have the same covariance structure. From the video linked in Figure 2, we see that t-SNE has correctly identified the sub-clusters. However, their relative locations to each other is distorted, with the orange and blue groups being far from each other in the tour view (Figure 2a). We see in this case that it is difficult to see the sub-clusters in the tour view, however, once we zoom and highlight they become more apparent (Figure 2b). When we brush the sub-clusters in the t-SNE, their relative placement is again exaggerated, with the tour showing that they are indeed much closer than the impression the t-SNE view gives.

5.3. Case Study 3: Exploring data with piecewise linear structure

Next, we explore some simulated noisy tree structured data (Figure 3). Our interest here is how t-SNE visualizations break the topology of the data, and then seeing if we can resolve this by tweaking the default parameters with reference to the global view of the data set. This simulation aims to mimic branching trajectories of cell differentiation; if there were only mature cells, we would just see the tips of the branches which have a hierarchical pattern of clustering.

First, we apply principal components and restrict the results down to the first twelve principal components (which makes up approximately 70% of the variance explained in the data), to use with the grand tour.

Moreover, we run t-SNE using the default arguments on the complete data (this keeps the first 50 PCs, sets the perplexity to equal 30 and performs random initialization). We then create a linked tour with t-SNE layout with **liminal** as shown in Figure 4.

From the linked video, we see that the t-SNE view has been unable to recapitulate the topology of the tree - the backbone (blue) branch has been split into three fragments (Figure 4a). We can see this immediately via the linked highlighting over both plots. If we click on the legend for the zero branch, the blue colored points on each view are highlighted and the remaining points are made transparent. From here it becomes apparent from the tour view that the blue branch forms the backbone of the tree and is connected to all other branches. From the video, it is easy to see that cluster sizes formed via t-SNE can be misleading; from the tour view there is a lot of noise along the branches, while this does not appear to be the case for the t-SNE result (Figure 4b).

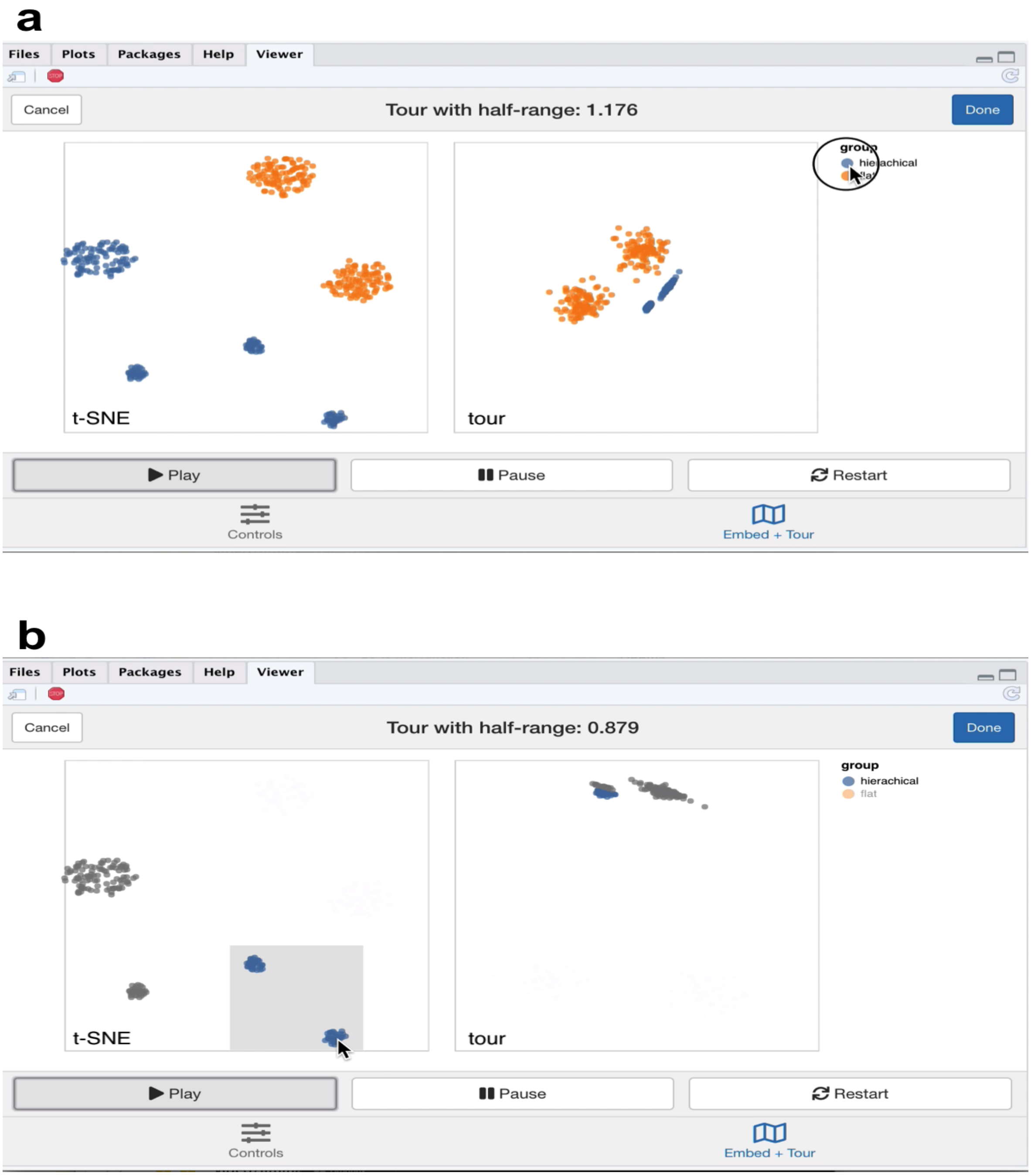

Figure 2: Screenshots of the **liminal** interface applied to sub-clustered data, a video of the tour animation is available at https://player.vimeo.com/video/439635905.

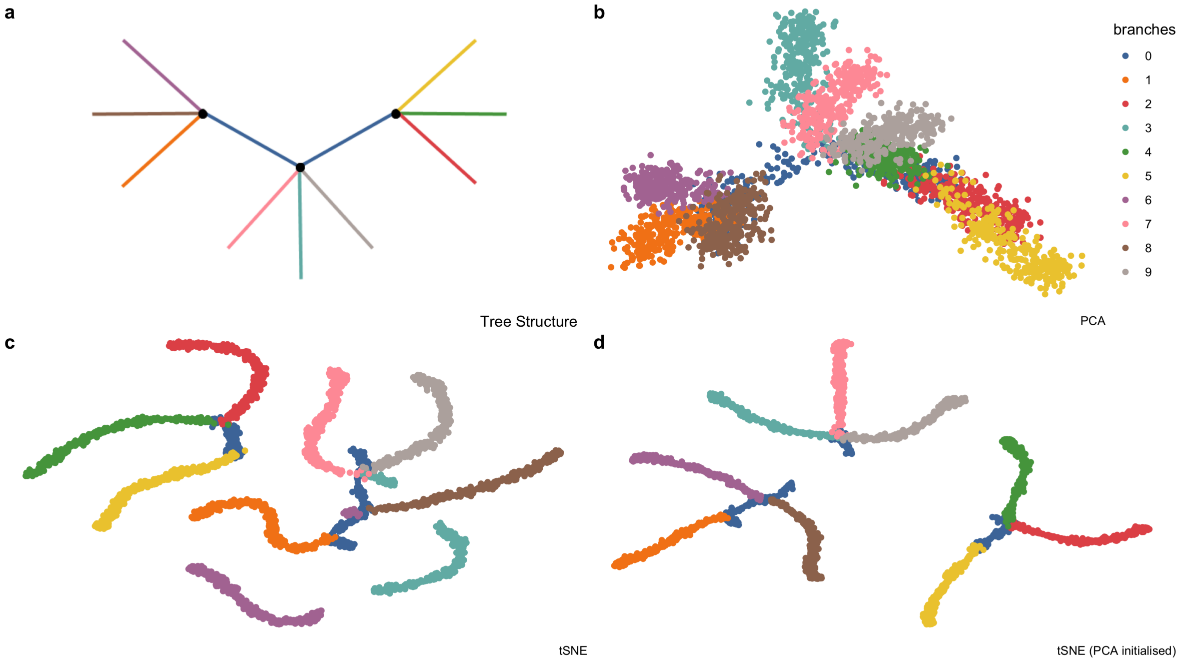

Figure 3: Example high-dimensional tree shaped data, $n = 3000$ and $p = 100$. *(a)* The true data lies on a 2-d tree consisting of ten branches. This data is available in the **phateR** package and is simulated via diffusion-limited aggregation (a random walk along the branches of the tree) with Gaussian noise added (Moon et al. 2019). *(b)* The first two principal components, which form the initial projection for the tour, note that the backbone of the tree is obscured by this view. *(c)* The default t-SNE view breaks the global structure of the tree. *(d)* Altering t-SNE using the first two principal components as the starting coordinates for the embedding, results in clustering the tree at its branching points.

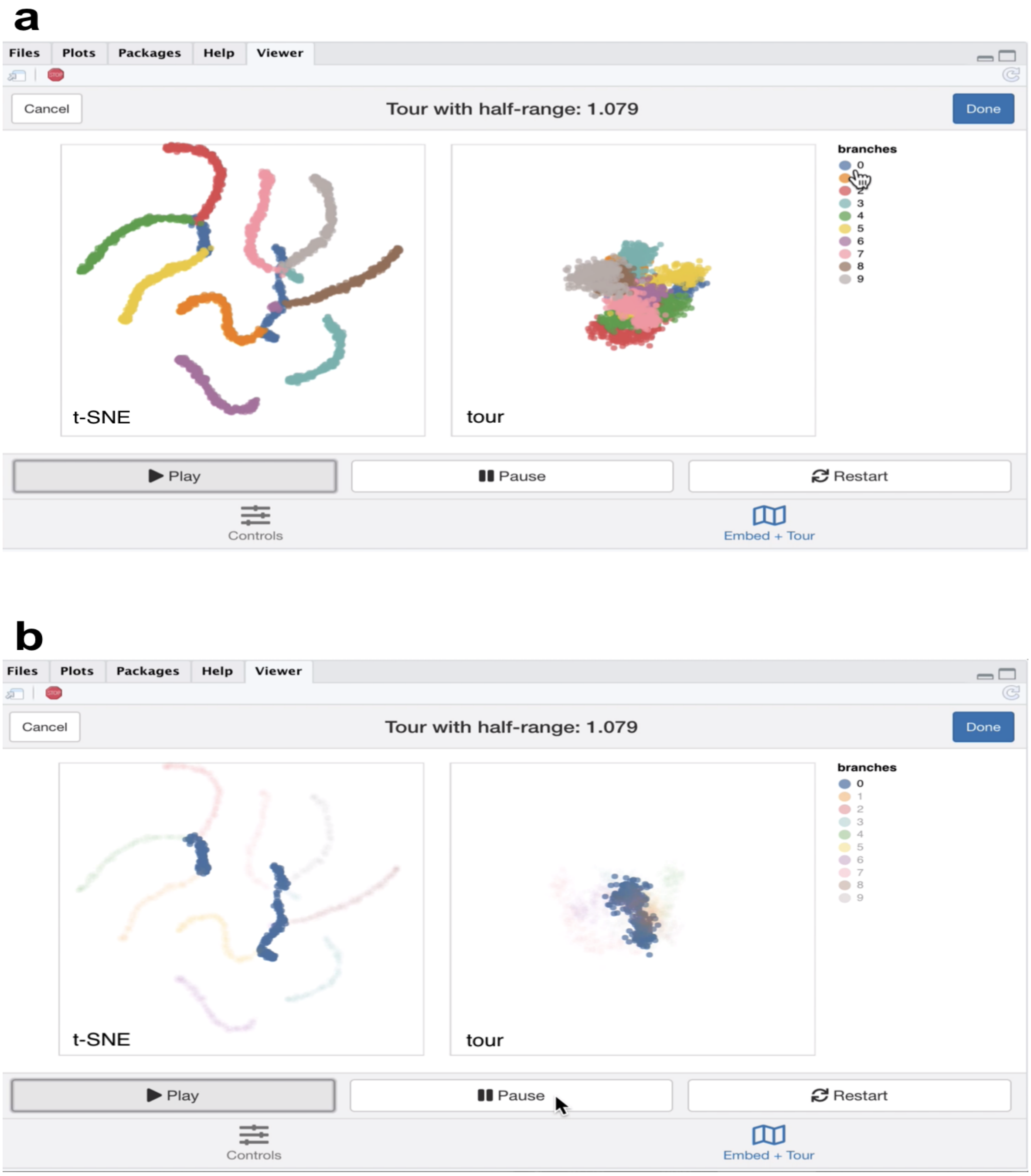

Figure 4: Screenshots of the **liminal** interface applied to tree structured data, a video of the tour animation is available at `https://player.vimeo.com/video/439635892`.

From the first view, we modify the inputs to the t-SNE view, to try and produce a better trade-off between local structure and retain the topology of the data. We keep every parameter the same except that we initialize Y with the first two PCs (scaled to have standard deviation 1e-4) instead of the default random initialization and increase the perplexity from 30 to 100. We then combine these results with our tour view as displayed in the linked video in the caption of Figure 5.

The video linked in Figure 5 shows that this selection of parameters results in the tips of the branches (the three black dots in Figure 3a) being split into three clusters representing the terminal branches of the tree. However, there are perceptual issues following the placement of the three groupings on the t-SNE view that become apparent via simple linked brushing. If we brush the tips of the yellow and brown branches (which appear to be close to each other on the t-SNE view), we immediately see the placement is distorted in the t-SNE view, and in the tour view these tips are at opposite ends of the tree (Figure 5b). Although, this is a known issue of the t-SNE algorithm, we can easily identify it via simple interactivity without knowing the inner workings of the method.

5.4. Case Study 4: Clustering single cell RNA-seq data

A common analysis task in single cell studies is performing clustering to identify groupings of cells with similar expression profiles. Analysts in this area generally use non-linear DR methods for verification and identification of clusters and developmental trajectories (i.e., case study 1). For clustering workflows, the primary task is to verify the existence of clusters and then begin to identify the clusters as cell types using the expression of “known” marker genes. Here a ‘faithful’ embedding should ideally preserve the topology of the data; cells that correspond to a cell type should lie in the same neighborhood in high-dimensional space. In this case study, we use our linked brushing approaches to look at neighborhood preservation and look at marker genes through the lens of the tour. The data we have selected for this case study has features similar to those found in case studies 2 and 3.

First, we downloaded the raw mouse retinal single cell RNA-seq data from Macosko et al. (2015) using the **scRNAseq** Bioconductor package (Risso and Cole 2019). We have followed a standard workflow for pre-processing and normalizing this data (described by Amezquita et al. (2020)). We performed quality control using the **scater** package by removing cells with a high proportion of mitochondrial gene expression (as this indicates poor sample quality), and low numbers of genes detected. We then log-transformed and normalized the expression values and finally selected highly variable genes (HVGs) using **scran** (McCarthy et al. 2017; Lun et al. 2016). The top ten percent of HVGs were used to subset the normalized expression matrix and compute PCA using the first 25 components. Using the PCs we built a shared nearest neighbors graph (with $k = 10$) and used Louvain clustering to generate clusters (Blondel et al. 2008).

To check and verify the clustering we construct a **liminal** view. We tour the first five PCs (approximately 20% of the variance in expression), alongside the t-SNE view which was computed from all 25 PCs. We have selected only the first five PCs because there is a large drop in the percentage of variance explained after the fifth component, with

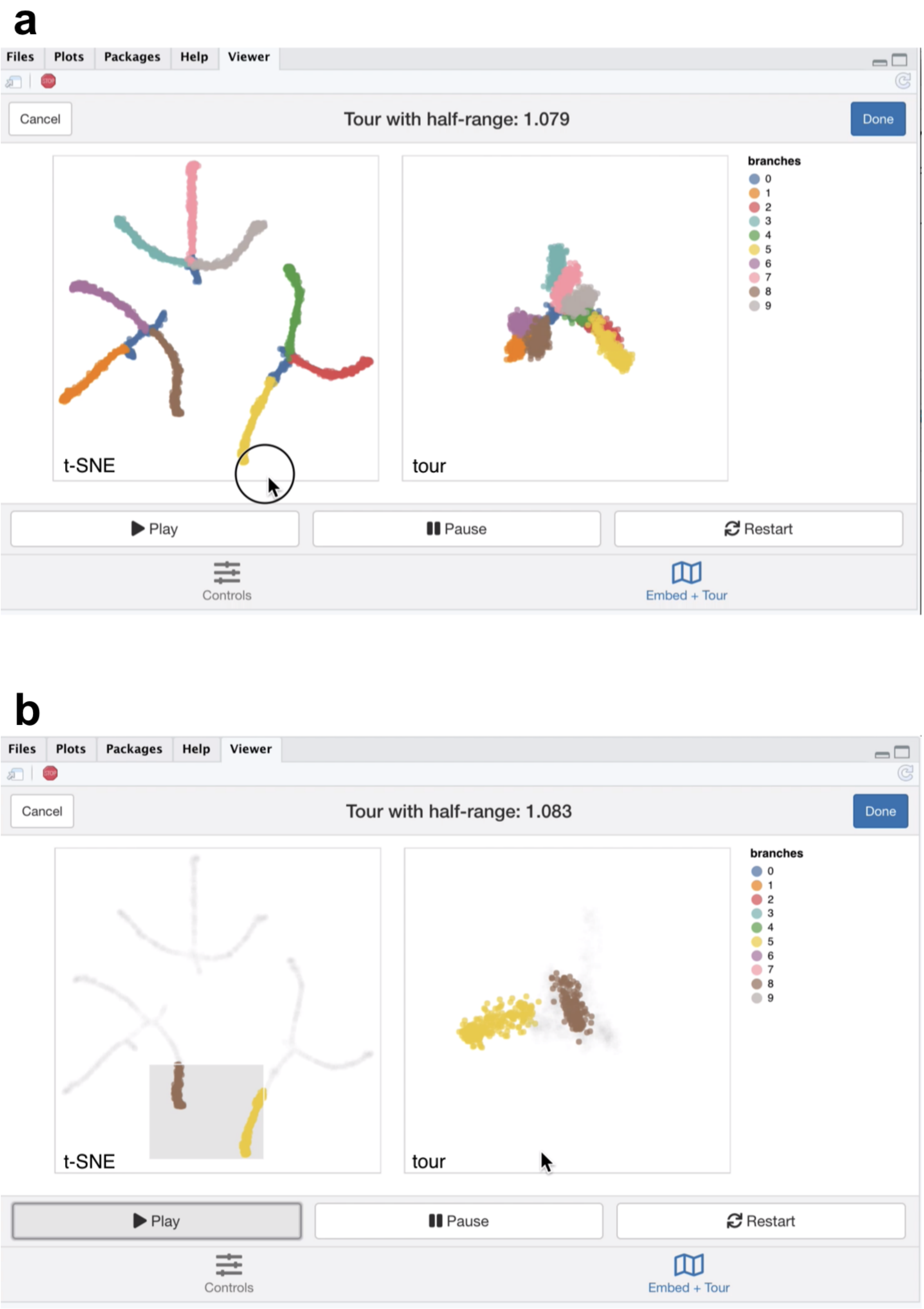

Figure 5: Screenshots of the **liminal** interface applied to tree structured data, a video of the tour animation is available at `https://player.vimeo.com/video/439635863`.

each component after that contributing less than one percent of variance. Consequently, increasing the number of PCs to tour would increase the dimensionality and volume of the subspace we are touring but without adding any additional signal to the view.

Due to latency of the **liminal** interface we do a weighted sample of the rows based on cluster membership, leaving us with approximately 10 per cent of the original data size - 4,590 cells. Although this is not ideal, it still allows us to get a sense of the shape of the clusters as seen from the linked video in Figure 6. If one was interested in performing more in-depth cluster analysis, we recommend an iterative approach of removing large clusters and then re-running the **liminal** view as a way finding more granular cluster structure. One could perform this approach manually via the **liminal** interface by returning the regions identified by brushing on the tour or embedding view.

From the video linked in Figure 6, we learn that the embedding has mostly captured the clusters relative location to each other to their location in high-dimensional space, with a notable exception of points in cluster 3 and 10 as shown with linked brushing (Figure 6a). As expected, t-SNE mitigates the crowding problem that is an issue for tour in this case, where points in clusters 2, 4, 6, and 11 are clumped together in tour view, but are blown up in the embedding view (Figure 6b). The tour appears to form a tetrahedron-like shape, with points lying on the surface and along the vertices of the tetrahedron in 5-d PCA space - a phenomena that has also been observed in Korem et al. (2015) (Figure 6c). Brushing on the tour view, reveals points in cluster 9 that are diffuse in the embedding view, points in cluster 9 are relatively far away and spread apart from other clusters in the tour view, but has points placed in cluster 3 and 9 in the embedding (Figure 6d).

Next, we identify marker genes for clusters using one sided Welch t-tests with a minimum log fold change of one as recommended by Amezquita et al. (2020), which uses the testing framework from McCarthy and Smyth (2009). We select the top 10 marker genes that are expressed in relatively larger quantities (upregulated) in cluster 2, which was one of the clumped clusters when we toured on principal components. Here, the tour becomes an alternative to a standard heatmap view for assessing shared markers; the basis generation (shown as the biplot on the left view) reflects the relative weighting of each gene. We run the tour directly on the log normalized expression values using the same subset as before.

From the video linked in Figure 7, we see that the expression of the marker genes, appear to separate the previously clumped clusters 2, 4, 6, and 11 from the other clusters, indicating that these genes are expressed in all four clusters (Figure 7a). After zooming, we can see a trajectory forming along the clusters, while the axis view shows that magnitude of expression in the marker genes is similar across these separated clusters which is consistent with the results of marker gene analysis (Figure 7b).

6. Discussion

We have shown that the use of tours as a tool for interacting with high-dimensional data provides an additional insight for interrogating views generated from embeddings. The interface we have designed in the **liminal** package, allows a user to gain a deeper understanding of an embedding algorithm, and rectifies perceptual issues associated

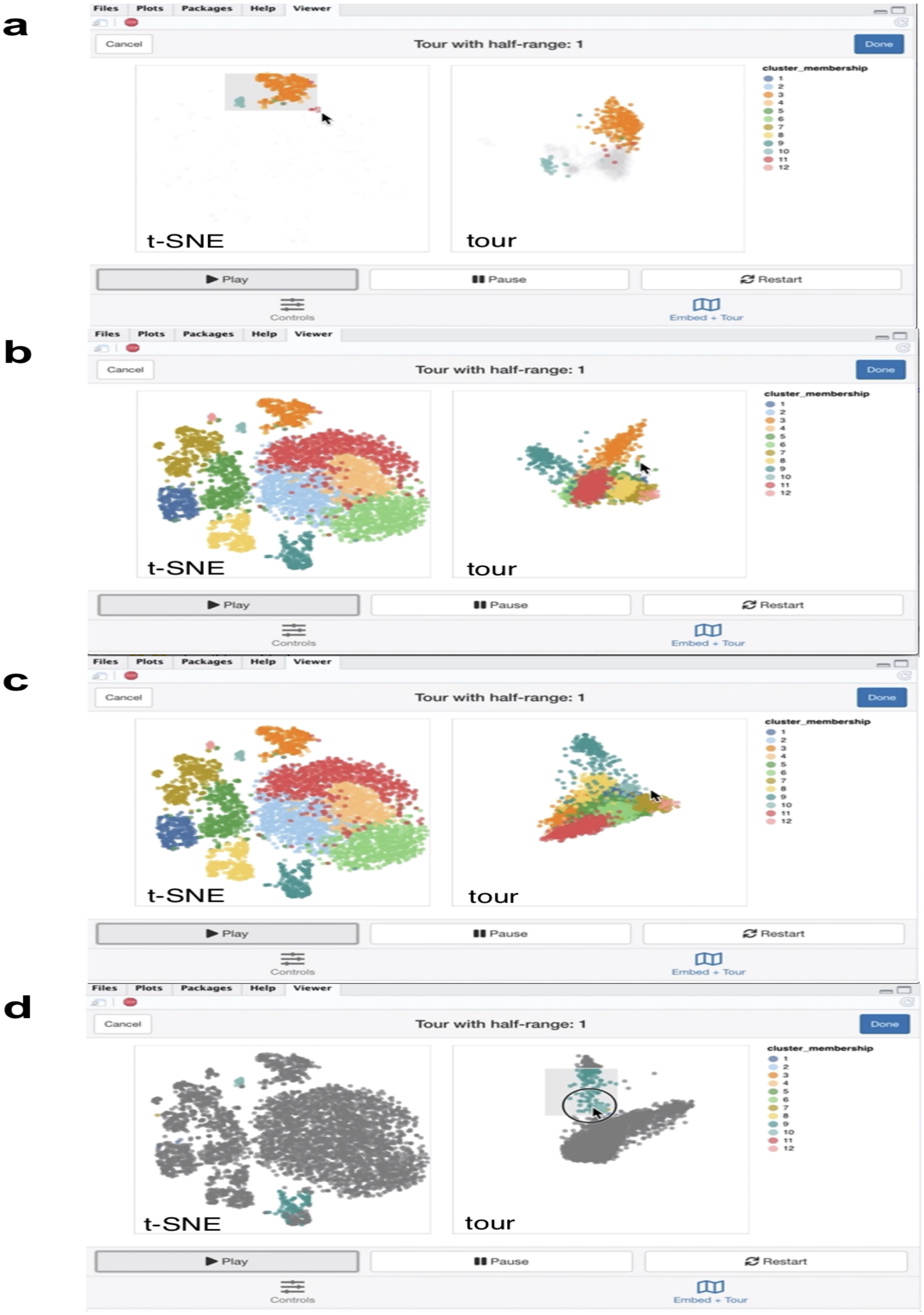

Figure 6: Screenshots of the **liminal** interface applied to single cell data, a video of the tour animation is available at `https://player.vimeo.com/video/439635812`.

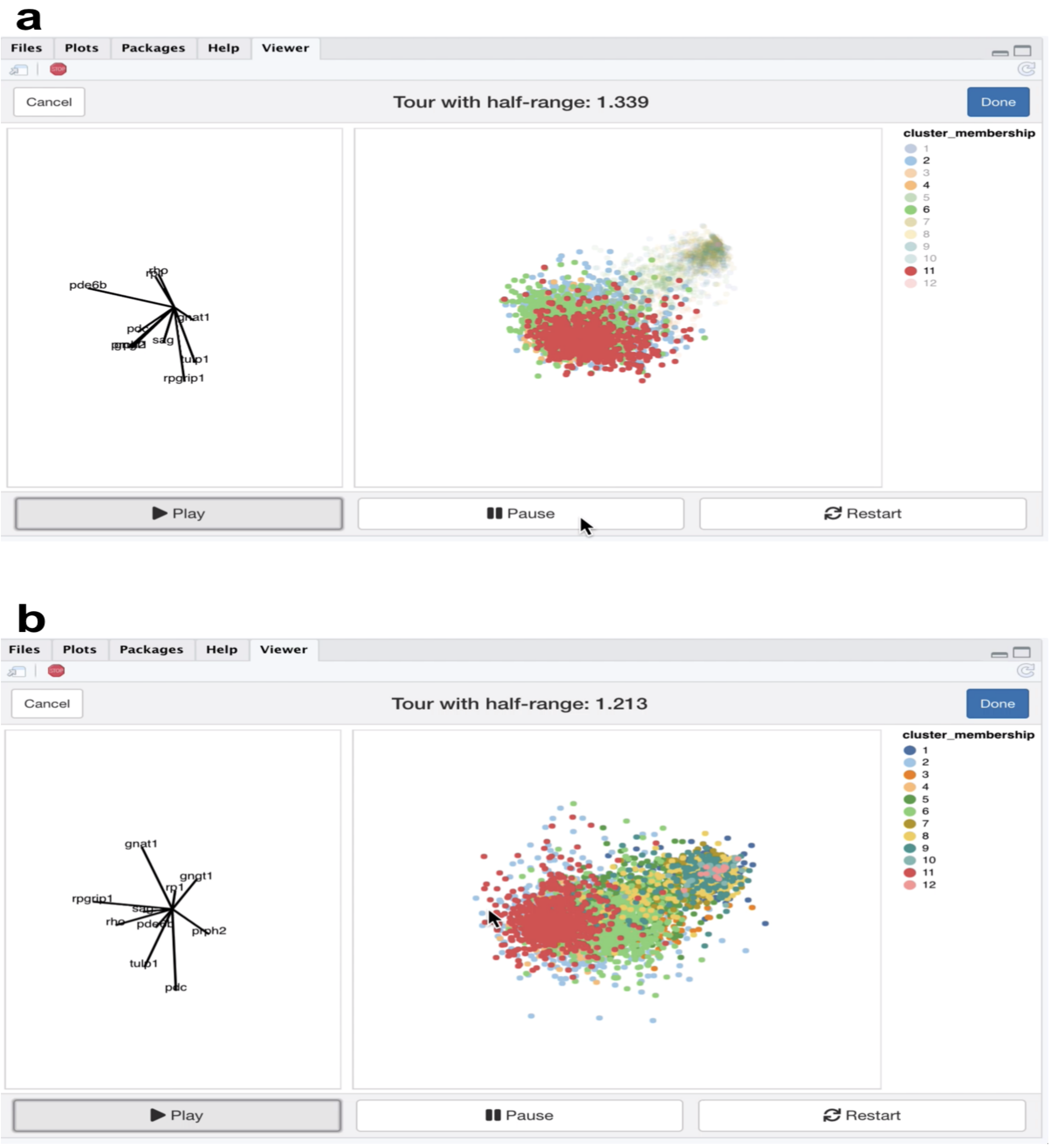

Figure 7: Screenshots of the **liminal** tour applied to a marker gene set, a video of the tour animation is available at `https://player.vimeo.com/video/439635843`.

with NLDR methods via linked interactions with the tour. As we have shown in the simulation case studies, the t-SNE method can produce misleading embeddings which can be detected through the use linked brushing and highlighting. In the case when the data has a piecewise linear geometry, like the tree simulation, the tour preserves the shape of the data which can be obscured by the embedding method.

Our framework can also be useful in practice, as displayed in the fourth case study. The tour when combined with t-SNE allowed us to identify clusters, while giving us an idea of their orientation to each other. Moreover, we could visually inspect the separation of clusters using a tour on marker gene sets. We see our approach as being valuable to the single cell analyst who wants to make their embeddings more interpretable.

We have shown in the case studies, that one to one linked brushing can be used to identify distortions in the embedding, however, we would like extend this to one to many linked brushing, which would allow us to directly interrogate neighborhood preservation. This form of brushing acts directly on a k-nearest neighbors (k-nn) graph computed from a reference dataset: when a user brushes over a region in the embedding, all the points that match the graphs edges are selected on the corresponding tour view. The reference data set for computing nearest neighbors (for example, a distance matrix, or the complete data matrix) can be independent of the tour or embedding views. In place of highlighting, one could use opacity or binned color scales to encode distances or ranks instead of the neighboring points. We have begun implementing this brush in **liminal**, using the **FNN** or **RcppAnnoy** packages for fast neighborhood estimation on the server side, however, there are still technicalities that need be resolved (Beygelzimer et al. 2019; Eddelbuettel 2020). Brush composition, such as 'and', 'or', or 'not' brushes, could be used to further investigate mismatches between the k-nn graphs estimated from both the embedding and tour views.

There are some limitations in using the **liminal** interface for larger datasets. First, t-SNE avoids the crowding problem; points are separated into distinct regions on the display canvas. For the tour, points are concentrated in the centre of the projection and become difficult to see. We have recently proposed a simple non-linear transformation for the tour called a sage tour that aims to fix this problem (Laa et al. 2020b). Second, as n increases both the embedding view and tour view become harder to read due to over-plotting, while the interactivity and animation become slower as there is more data passing from the server to the client. For the tasks we have looked at in this paper, where shape and density are important to the analyst, we think that better displays and sub-sampling strategies are more useful than being able to look at every single point on the canvas. We showed in our single cell clustering case study that doing a weighted sample based on cluster membership still allowed us to get a sense of relative cluster orientation, however, there are alternative sampling approaches that could be applied, like selecting points close to the cluster centers. Alternative displays via statistical transformations could also mitigate the need to show all of the data. Recent work by Laa et al. (2020a) is a promising area for further investigation, as well as work from topological statistics (Rieck 2017; Genovese et al. 2017).

Acknowledgements

The authors gratefully acknowledge the support of the Australian Research Council. We would like to thank Dr David Frazier and Dr Paul Harrison for their feedback on this work. The screenshots and images were compiled using the **cowplot** and **ggplot2** packages (Wilke 2019; Wickham 2016).

Supplementary Materials

Code, data, and video for reproducing this paper are available at `https://github.com/sa-lee/paper-liminal`. Direct links to videos for viewing online are available in Table 1.

Table 1: Case Study Videos

Case Study	Example	URL
1	gaussian	https://player.vimeo.com/video/439635921
2	hierarchical	https://player.vimeo.com/video/439635905
3	trees-01	https://player.vimeo.com/video/439635892
3	trees-02	https://player.vimeo.com/video/439635863
4	mouse-01	https://player.vimeo.com/video/439635812
4	mouse-02	https://player.vimeo.com/video/439635843

References

Amezquita, R. A., Lun, A. T. L., Becht, E., Carey, V. J., Carpp, L. N., Geistlinger, L., Marini, F., Rue-Albrecht, K., Risso, D., Soneson, C., Waldron, L., Pagès, H., Smith, M. L., Huber, W., Morgan, M., Gottardo, R., and Hicks, S. C. (2020). Orchestrating single-cell analysis with bioconductor. *Nat. Methods*, 17(2):137–145.

Asimov, D. (1985). The grand tour: A tool for viewing multidimensional data. *SIAM J. Sci. and Stat. Comput.*, 6(1):128–143.

Becker, R. A. and Cleveland, W. S. (1987). Brushing scatterplots. *Technometrics*, 29(2):127–142.

Beygelzimer, A., Kakadet, S., Langford, J., Arya, S., Mount, D., and Li, S. (2019). *FNN: Fast Nearest Neighbor Search Algorithms and Applications*. R package version 1.1.3.

Blondel, V. D., Guillaume, J.-L., Lambiotte, R., and Lefebvre, E. (2008). Fast unfolding of communities in large networks. *J. Stat. Mech.*, 2008(10):P10008.

Brehmer, M., Sedlmair, M., Ingram, S., and Munzner, T. (2014). Visualizing dimensionally-reduced data: Interviews with analysts and a characterization of task sequences. In *Proceedings of the Fifth Workshop on Beyond Time and Errors: Novel Evaluation Methods for Visualization*, pages 1–8. dl.acm.org.

Buja, A. and Asimov, D. (1986). Grand tour methods: an outline. In *Proceedings of the Seventeenth Symposium on the interface of computer sciences and statistics on Computer science and statistics*, pages 63–67, USA. Elsevier North-Holland, Inc.

Buja, A., Cook, D., and Swayne, D. F. (1996). Interactive High-Dimensional data visualization. *J. Comput. Graph. Stat.*, 5(1):78–99.

Buja, A., Hurley, C., and McDonald, J. (1986). A data viewer for multivariate data. In *Computing Science and Statistics: Proceedings of the 18th Symposium on the Interface*, pages 171–174. American Statistical Association.

Buja, A., Swayne, D. F., Littman, M. L., Dean, N., Hofmann, H., and Chen, L. (2008). Data visualization with multidimensional scaling. *J. Comput. Graph. Stat.*, 17(2):444–472.

Chang, W., Cheng, J., Allaire, J. J., Xie, Y., and McPherson, J. (2020). shiny: Web application framework for R.

Coenen, A. and Pearce, A. (2019). Understanding UMAP. https://pair-code.github.io/understanding-umap/. Accessed: 2020-4-17.

Coifman, R. R., Lafon, S., Lee, A. B., Maggioni, M., Nadler, B., Warner, F., and Zucker, S. W. (2005). Geometric diffusions as a tool for harmonic analysis and structure definition of data: diffusion maps. *Proc. Natl. Acad. Sci. U. S. A.*, 102(21):7426–7431.

Cook, D., Buja, A., Cabrera, J., and Hurley, C. (1995). Grand tour and projection pursuit. *J. Comput. Graph. Stat.*, 4(3):155–172.

Eddelbuettel, D. (2020). *RcppAnnoy: 'Rcpp' Bindings for 'Annoy', a Library for Approximate Nearest Neighbors.* R package version 0.0.16.

Genovese, C., Perone-Pacifico, M., Verdinelli, I., and Wasserman, L. (2017). Finding singular features. *J. Comput. Graph. Stat.*, 26(3):598–609.

Kobak, D. and Berens, P. (2019). The art of using t-SNE for single-cell transcriptomics. *Nat. Commun.*, 10(1):5416.

Korem, Y., Szekely, P., Hart, Y., Sheftel, H., Hausser, J., Mayo, A., Rothenberg, M. E., Kalisky, T., and Alon, U. (2015). Geometry of the gene expression space of individual cells. *PLoS Comput. Biol.*, 11(7):e1004224.

Krijthe, J. H. (2015). *Rtsne: T-Distributed Stochastic Neighbor Embedding using Barnes-Hut Implementation.* R package version 0.15.

Kruskal, J. B. (1964a). Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrika*, 29(1):1–27.

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

Laa, U., Cook, D., Buja, A., and Valencia, G. (2020a). Hole or grain? a section pursuit index for finding hidden structure in multiple dimensions.

Laa, U., Cook, D., and Lee, S. (2020b). Burning sage: Reversing the curse of dimensionality in the visualization of high-dimensional data.

Lee, J. A. and Verleysen, M. (2009). Quality assessment of dimensionality reduction: Rank-based criteria. *Neurocomputing*, 72(7):1431–1443.

Lee, S. and Cook, D. (2020). *liminal: Multivariate Data Visualization With Tours and Embeddings*. R package version 0.0.5.9999.

Lewis, J., Van der Maaten, L., and de Sa, V. (2012). A behavioral investigation of dimensionality reduction. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 34.

Linderman, G. C. and Steinerberger, S. (2019). Clustering with t-SNE, provably. *SIAM Journal on Mathematics of Data Science*, 1(2):313–332.

Lun, A. T. L., McCarthy, D. J., and Marioni, J. C. (2016). A step-by-step workflow for low-level analysis of single-cell rna-seq data with bioconductor. *F1000Res.*, 5:2122.

Lyttle, I. and Vega/Vega-Lite Developers (2020). vegawidget: 'htmlwidget' for 'vega' and 'Vega-Lite'.

Macosko, E. Z., Basu, A., Satija, R., Nemesh, J., Shekhar, K., Goldman, M., Tirosh, I., Bialas, A. R., Kamitaki, N., Martersteck, E. M., Trombetta, J. J., Weitz, D. A., Sanes, J. R., Shalek, A. K., Regev, A., and McCarroll, S. A. (2015). Highly parallel genome-wide expression profiling of individual cells using nanoliter droplets. *Cell*, 161(5):1202–1214.

McCarthy, D. J., Campbell, K. R., Lun, A. T. L., and Willis, Q. F. (2017). Scater: pre-processing, quality control, normalisation and visualisation of single-cell RNA-seq data in R. *Bioinformatics*, 33:1179–1186.

McCarthy, D. J. and Smyth, G. K. (2009). Testing significance relative to a fold-change threshold is a TREAT. *Bioinformatics*, 25(6):765–771.

McDonald, J. A. (1982). *Interactive graphics for data analysis*. PhD thesis, Stanford University.

McInnes, L., Healy, J., and Melville, J. (2018). UMAP: Uniform manifold approximation and projection for dimension reduction.

Melville, J. (2020). t-SNE initialization options. `https://jlmelville.github.io/smallvis/init.html`. Accessed: 2020-3-23.

Moon, K. R., van Dijk, D., Wang, Z., Gigante, S., Burkhardt, D. B., Chen, W. S., Yim, K., van den Elzen, A., Hirn, M. J., Coifman, R. R., Ivanova, N. B., Wolf, G., and Krishnaswamy, S. (2019). Visualizing structure and transitions for biological data exploration.

Nguyen, L. H. and Holmes, S. (2019). Ten quick tips for effective dimensionality reduction. *PLoS Comput. Biol.*, 15(6):e1006907.

Ovchinnikova, S. and Anders, S. (2020). Exploring dimension-reduced embeddings with sleepwalk. *Genome Res.*, 30(5):749–756.

Pezzotti, N., Lelieveldt, B. P. F., Van Der Maaten, L., Hollt, T., Eisemann, E., and Vilanova, A. (2017). Approximated and user steerable tSNE for progressive visual analytics. *IEEE Trans. Vis. Comput. Graph.*, 23(7):1739–1752.

Rauber, A. (2009). Multi-challenge data set. `http://ifs.tuwien.ac.at/dm/dataSets.html`. Accessed: 2020-09-17.

Rieck, B. (2017). *Persistent Homology in Multivariate Data Visualization*. PhD thesis, Ruprecht-Karls-Universität Heidelberg.

Risso, D. and Cole, M. (2019). *scRNAseq: Collection of Public Single-Cell RNA-Seq Datasets*. R package version 2.0.2.

Satyanarayan, A., Moritz, D., Wongsuphasawat, K., and Heer, J. (2017). Vega-Lite: A grammar of interactive graphics. *IEEE Trans. Vis. Comput. Graph.*, 23(1):341–350.

Sedlmair, M., Munzner, T., and Tory, M. (2013). Empirical guidance on scatterplot and dimension reduction technique choices. *IEEE Trans. Vis. Comput. Graph.*, 19(12):2634–2643.

Silva, V. D. and Tenenbaum, J. B. (2003). Global versus local methods in nonlinear dimensionality reduction. *Adv. Neural Inf. Process. Syst.*

Smilkov, D., Thorat, N., Nicholson, C., Reif, E., Viégas, F. B., and Wattenberg, M. (2016). Embedding projector: Interactive visualization and interpretation of embeddings.

Swayne, D. F. and Buja, A. (2004). Exploratory visual analysis of graphs in GGOBI. In *COMPSTAT 2004 — Proceedings in Computational Statistics*, pages 477–488. Physica-Verlag HD.

Swayne, D. F., Cook, D., and Buja, A. (1998). XGobi: Interactive dynamic data visualization in the X window system. *J. Comput. Graph. Stat.*, 7(1):113–130.

Swayne, D. F., Lang, D. T., Buja, A., and Cook, D. (2003). GGobi: Evolving from XGobi into an extensible framework for interactive data visualization. *Comput. Stat. Data Anal.*, 43(4):423–444.

Tang, J., Liu, J., Zhang, M., and Mei, Q. (2016). Visualizing large-scale and high-dimensional data.

Torgerson, W. S. (1952). Multidimensional scaling: I. theory and method. *Psychometrika*, 17(4):401–419.

van der Maaten, L. (2014). Accelerating t-SNE using tree-based algorithms. *J. Mach. Learn. Res.*

van der Maaten, L. and Hinton, G. (2008). Visualizing data using t-SNE. *J. Mach. Learn. Res.*, 9(Nov):2579–2605.

Wattenberg, M., Viégas, F., and Johnson, I. (2016). How to use t-SNE effectively. *Distill*, 1(10).

Wickham, H. (2016). *ggplot2: Elegant Graphics for Data Analysis*. Use R! Springer International Publishing.

Wickham, H., Cook, D., and Hofmann, H. (2015). Visualizing statistical models: Removing the blindfold. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, 8(4):203–225.

Wickham, H., Cook, D., Hofmann, H., and Buja, A. (2011). tourr: An R package for exploring multivariate data with projections. *Journal of Statistical Software, Articles*, 40(2):1–18.

Wilke, C. O. (2019). *cowplot: Streamlined Plot Theme and Plot Annotations for 'ggplot2'*. R package version 1.0.0.

Affiliation:

Stuart Lee
Monash University
Department of Econometrics and Business Statistics,
Monash University
E-mail: stuart.andrew.lee@gmail.com

Ursula Laa
Monash University
Department of Econometrics and Business Statistics,
Monash University

Dianne Cook
Monash University
Department of Econometrics and Business Statistics,
Monash University

Journal of Data Science, Statistics, and Visualisation https://jdssv.org/
published by the International Association for Statistical Computing
http://iasc-isi.org/
MMMMMMM YYYY, Volume VV, Issue II *Submitted:* yyyy-mm-dd
doi:XX.XXXXX/jdssv.v000.i00 *Accepted:* yyyy-mm-dd