spinifex: An R Package for Creating a Manual Tour of Low-dimensional Projections of Multivariate Data

by Nicholas Spyrison, Dianne Cook

Abstract Dynamic low-dimensional linear projections of multivariate data collectively known as **tours** provide an important tool for exploring multivariate data and models. The R package **tourr** provides functions for several types of tours: grand, guided, little, local and frozen. Each of these can be viewed dynamically, or saved into a data object for animation. This paper describes a new package, **spinifex**, to provide a manual tour, that allows the coefficient for a single variable to be controlled. The variable is rotated fully into the projection with a coefficient of 1 or -1, or completely out of the projection with a coefficient of 0. The resulting sequence of projections can be displayed using animation, with functions from either the **plotly** and **gganimate** packages. By varying the coefficient of a single variable, it is possible to explore the sensitivity of structure in the projection to that variable. It is particularly useful when used with a projection pursuit guided tour to simplify and understand the solution. The use of the manual tour is illustrated using a problem from particle physics.

Introduction

Exploring multivariate spaces is a challenging task, increasingly so as dimensionality increases. Traditionally, static low-dimensional projections are used to display multivariate data in two dimensions, such as principal component analysis, linear discriminant spaces or projection pursuit. These are useful for finding relationships between multiple variables, but they are limited because they show only a glimpse of the high-dimensional space. An alternative approach is to use a tour of dynamic linear projections, to look at many different low-dimensional projections. Tours can be considered to extend the dimensionality of visualization, which is important as data and models exist in more than 3D. The package **tourr** (Wickham et al. 2011) provides a platform for generating tours. It has many types of tours available, and many types of displays possible. A user can make a grand, guided, little, local or frozen tour, and display the resulting projected data as a scatterplot, density plot, histogram, or even as Chernoff faces if the projection dimension is more than 3.

This work adds a manual tour to the collection. The manual tour was described in Cook and Buja (1997) and allows a user to control the projection coefficients of a selected variable in a 2D projection. The manipulation of these coefficients allows the analyst to explore their sensitivity to the structure within the projection. As manual tours operate on only one variable at a time, they are particularly useful once a feature of interest has been identified.

One way to identify "interesting" features is with the use of a guided tour (Cook et al. 1995). Guided tours select a very specific path, which approaches a projection that optimizes an objective-function. The optimization is conducted in a manner similar to simulated annealing (Kirkpatrick, Gelatt, and Vecchi 1983). The direct optimization of a function allows guided tours to rapidly identify interesting projection features given the relatively large parameter-space. After a projection of interest is identified an analyst can then use the "finer brush" of the manual tour, by controlling the contributions of individual variables to explore the sensitivity they have on the structure visible in the projection.

The paper is organized as follows. Section 2.2 describes the algorithm used to perform a radial manual tour as implemented in the package **spinifex**. Section 2.2.4 explains how to generate an animation of the manual tour using the animation frameworks offered by **plotly** (Sievert 2018) and **gganimate** (Pedersen and Robinson 2019). Package functionality and code usage following the order applied in the algorithm follows in section 2.3. Section 2.4 illustrates how this can be used for sensitivity analysis applied to multivariate data collected on high energy physics experiments (Wang et al. 2018). Section 2.4.3 summarizes this paper and discusses potential future directions.

Algorithm

The algorithm to conduct a manual tour interactively, by recording mouse/cursor motion, is described in detail in Cook and Buja (1997). Movement can be in any direction and magnitude, but it can also be constrained in several ways:

- *radial*: fix the direction of contribution, and allow the magnitude to change.
- *angular*: fix the magnitude, and allow the angle or direction of the contribution to vary.
- *horizontal*, *vertical*: allow rotation only around the horizontal or vertical axis of the current 2D projection.

The algorithm described here produces a **radial** tour as an *animation sequence*. It takes the current contribution of the chosen variable, and using rotation brings this variable fully into the projection, completely removes it, and then returns to the original position. This is an overview of the steps:

1. Provided with a 2D projection, choose a variable to manipulate. This is called the "manip var".
2. Create a 3D manipulation space, where the manip variable has the full contribution.
3. Generate a rotation sequence which increases the norm of the coefficient to 1 and zeros it.

More detail is provided in the following sections. Function names are provided, but code structure and usage is discussed in section 2.3.

Notation

The notation used to describe the algorithm for a 2D radial manual tour is as follows:

- $\mathbf{X}$, the data, an $n \times p$ numeric matrix to be projected to 2D.
- $\mathbf{B} = (B_1, B_2)$, any 2D orthonormal projection basis, $p \times 2$ matrix, describing the projection from p to two dimensions. We will call this the "original projection" because it is the starting point for the manual tour.
- k, is the index of the variable to manipulate, called the "manip var".
- $|\mathbf{B}_k|$, the norm of the manip var, k
- $\mathbf{e}$, a 1D coordinate basis vector of length p with the k-th element set to one.
- θ, the angle of in-projection-plane rotation, for example, on the reference axes; c_θ, s_θ are the cosine and sine.
- ϕ, the angle of out-of-projection-plane rotation, into the manipulation space; c_ϕ, s_ϕ are the cosine and sine.
- $\mathbf{U}$ is the axis of rotation for ϕ.

The algorithm operates entirely on projection bases and utilizes the data only later when making a display.

Steps

Step 0) Set up

The flea data, originally from Lubischew (1962), made available in **tourr** is used to illustrate the algorithm. The data contains 74 observations across 6 variables, which are physical measurements of the flea beetles. Each observation belongs to one of three species.

An initial 2D projection basis set must be provided. One way to identify a projection containing interesting features is to apply a guided tour. In a guided tour, the projection sequence is selected by optimizing an index function via hill-climbing on the projection space. In this case, the holes index is selected and applied to the standardized flea data. The holes index is maximized when the projected observations are furthest from the center. Figure 1 shows a locally optimized projection of the data. The left panel displays the reference axes of the projection basis, a visual indication of the magnitude and direction each variable contributes to the projection. The right panel shows the data as projected through the basis set described by the reference axes (left). Data points are colored and given shape according to the species.

Call `view_basis()` on a basis to produce a **ggplot2** (Wickham 2016) graphic similar to Figure 1. The projected data is processed for display using matrix multiplication $\mathbf{X}_{[n,\ p]}\ *\ \mathbf{B}_{[p,\ d]}\ =\ \mathbf{P}_{[n,\ d]}$.

Step 1) Choose manip variable

In Figure 1 the contribution of the variables tars1 and aede2 mostly contrast the contribution of the other four variables. These two variables combined contribute in the direction of the projection where the purple cluster is separated from the other two clusters. The variable aede2 is selected as the manip var, the variable to be controlled in the tour. The question that will be explored is how important this variable is to the separation of the clusters.
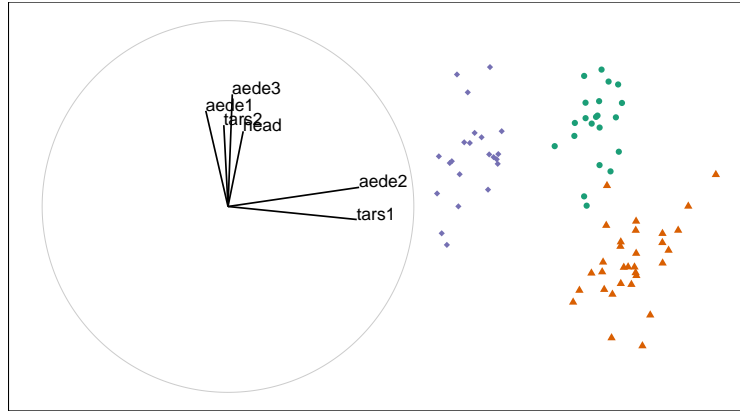
Figure 1: Initial 2D projection: representation of the basis (left) and resulting data projection (right) of standardized flea data. The color and shape of data points are mapped to beetle species. The basis was identified using a projection pursuit guided tour, with the holes index. The contribution of the variables aede2 and tars1 approximately contrasts the other variables. The visible structure in the projection are the three clusters corresponding to the three species. Produced with the function `view_basis()`.

Step 2) Create the 3D manip space

Initialize the coordinate basis vector as a zero vector **e** of length p, and set the k-th element to 1. In the example data, aede2 is the fifth variable in the data, so $k = 5$, set $e_5 = 1$. Use a Gram-Schmidt process to orthonormalize the coordinate basis vector on the original 2D projection to describe a 3D manipulation space, **M**.

$$e_k \leftarrow 1$$
$$\mathbf{e}^* \leftarrow \mathbf{e} - \langle \mathbf{e}, \mathbf{B}_1 \rangle \mathbf{B}_1 - \langle \mathbf{e}, \mathbf{B}_2 \rangle \mathbf{B}_2$$
$$\mathbf{M}_{[p,\, 3]} = (\mathbf{B}_1, \mathbf{B}_2, \mathbf{e}^*)$$

The manip space provides a 3D projection from p-dimensional space, where the coefficient of the manip var can range completely between [-1, 1]. 3D rotation can be used to rotate the mani variable completely into or completely out of a 2D projection. Figure 2 illustrates this 3D manipulation space with the manip var highlighted. This representation is produced by calling the `view_manip_space()` function. Functionally, the manipulation space is more of a mathematical construct that needn't be visualized but this diagram provides intuition for the technique.

Step 3) Generate rotation

Imagine holding the manip var, the red axis in Figure 2, one end fixed to the origin and moving it like a lever. This is effectively the process of the radial manual to rotate the manip variable into and out of the 2D projection (Figure 3). As the variable is controlled, the manipulation space rotates about the origin, the projection onto the horizontal projection plane correspondingly moves. This is what happens in a manual tour. Generating a sequence of values for the horizontal and vertical, angles produces a path for the rotation of the manipulation space. This defines the (orthonormally-constrained) rotation on the basis coefficients.

For a radial tour fix the (horizontal) angle within the projection plane, θ, and define a sequence for the (vertical) angle coming out of the projection plane, ϕ, bringing the initial XY contributions of the manip var to a maximum and then to zero before returning to the initial position. Alternatively, for an oblique manual tour, capture the user manipulation directly on the XY of the projection plane.

Rotations are defined by the axes they pivot on, θ then is the rotation around the Z axis. While ϕ will be the rotation around, **U**, a new axis on the XY plane, orthogonal to ϕ. Given these axes, the SO(3) rotation matrix can be derived by the following.
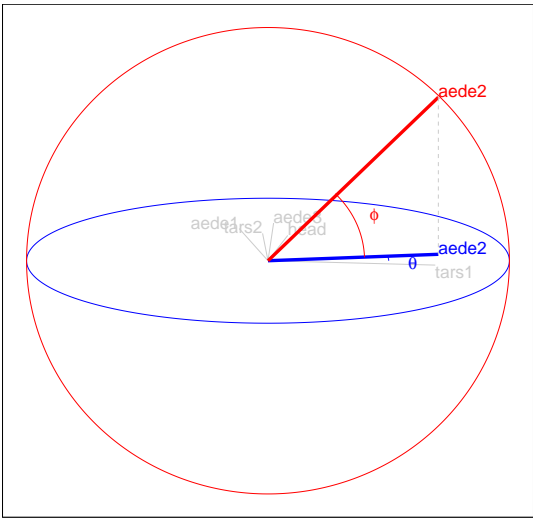
Figure 2: Illustration of a 3D manipulation space, constructed to change the contribution of the variable aede2 in the example data. The red circle indicates a unit sphere. The 2D projection is represented by the blue circle, with the projection coefficients represented by grey lines and text. The manip var axis, in red, has length 1 touching the sphere, extends the projection to a third dimension. The shadow of this axis (blue) is its contribution in the original 2D projection. Illustrated with the `view_manip_space()` function.
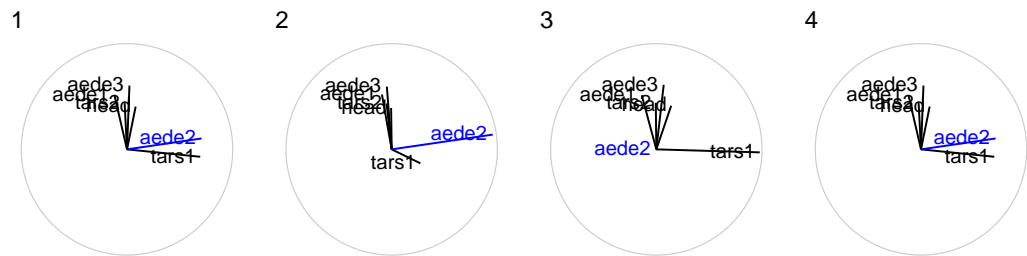
Figure 3: Snapshots of a radial manual tour manipulating aede2: (1) original projection, (2) full contribution, (3) zero contribution, (4) back to original.

$$\mathbf{U} = (u_x, u_y, u_z) = (s_\theta, -c_\theta, 0)$$

$$= \begin{bmatrix} 0 & -u_z & u_y \\ u_z & 0 & -u_x \\ -u_y & u_x & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & -c_\theta \\ 0 & 0 & -s_\theta \\ c_\theta & s_\theta & 0 \end{bmatrix}$$

Apply Rodrigues' rotation formula\protect\T1\textbraceleft@rodrigues_lois_1840}:

$$\mathbf{R}_{[3, 3]} = \mathbf{I}_3 + s_\phi \mathbf{U} + (1 - c_\phi)\mathbf{U}^2$$

$$= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} + \begin{bmatrix} 0 & 0 & c_\theta s_\phi \\ 0 & 0 & s_\theta s_\phi \\ -c_\theta s_\phi & -s_\theta s_\phi & 0 \end{bmatrix} + \begin{bmatrix} -c_\theta(1 - c_\phi) & s_\theta^2(1 - c_\phi) & 0 \\ -c_\theta s_\theta(1 - c_\phi) & -s_\theta^2(1 - c_\phi) & 0 \\ 0 & 0 & c_\phi - 1 \end{bmatrix}$$

$$= \begin{bmatrix} c_\theta^2 c_\phi + s_\theta^2 & -c_\theta s_\theta(1 - c_\phi) & -c_\theta s_\phi \\ -c_\theta s_\theta(1 - c_\phi) & s_\theta^2 c_\phi + c_\theta^2 & -s_\theta s_\phi \\ c_\theta s_\phi & s_\theta s_\phi & c_\phi \end{bmatrix}$$

Defining the axis of rotation for ϕ and the application of Rogrigues' rotation formula results in the rotation matrix, the rotation group SO(3). Continue with the rotation of the manipulation space, post-multiply the space by the above rotation matrix producing **RM**, the rotated manip space. This is basis values, the linear combination of variables, for one frame of the animation, repeat this process over a sequence of values for ϕ to produce an animation.

Creating the animation

The steps outlined above can be used to create any arbitrary rotation in the manip space. To use these for sensitivity analysis, the radial rotation is built into an animation where the manip var is rotated fully into the projection, completely out, and then back to the start. This involves changing ϕ, the angle relative to its initial value.

When ϕ is 0, the manip var is completely in the projection, and when ϕ is $\pi/2$, $\mathbf{X}_k$ is not contributing at all.

Sequence of ϕ_i

1. Set initial value to be ϕ_1.
2. Set an angle value (Δ_ϕ) that sets the step size to make in rotating the manip var into and out of the projection. (Note: Using angle value to control this is consistent with the tour algorithm, as implemented in the **tourr**).
3. Step towards 0, where the manip var has a completely in the projection.
4. Step towards $\pi/2$ where the manip variable has no contribution.
5. Step back to ϕ_1.

In each of steps 3-5, a small step may be made to bring it exactly to the end position.

Checks that need to be made

The value of ϕ is relative to its initial value. The transformation ϕ_i - ϕ_1 is useful to think about ϕ relative to the basis plane. Additionally, the value of ϕ may be out of phase by a factor of pi. If the manip variable doesn't move as expected these are the first places to check.

Figure 4 illustrates a sequence with 15 projected bases, showing the reference axes on the top half with the corresponding projected data points below. Changes in the basis coefficients of the manip var correspond to the distance separating the purple cluster and the remaining sample, aede2 is crucial in distinguishing the variation of species. Tours are typically viewed as an animation. The animation of this tour can be viewed online at https://nspyrison.netlify.com/thesis/flea_manualtour_mvar5/. The page may take a moment to load. Animations can be produced using the function play_manual_tour().
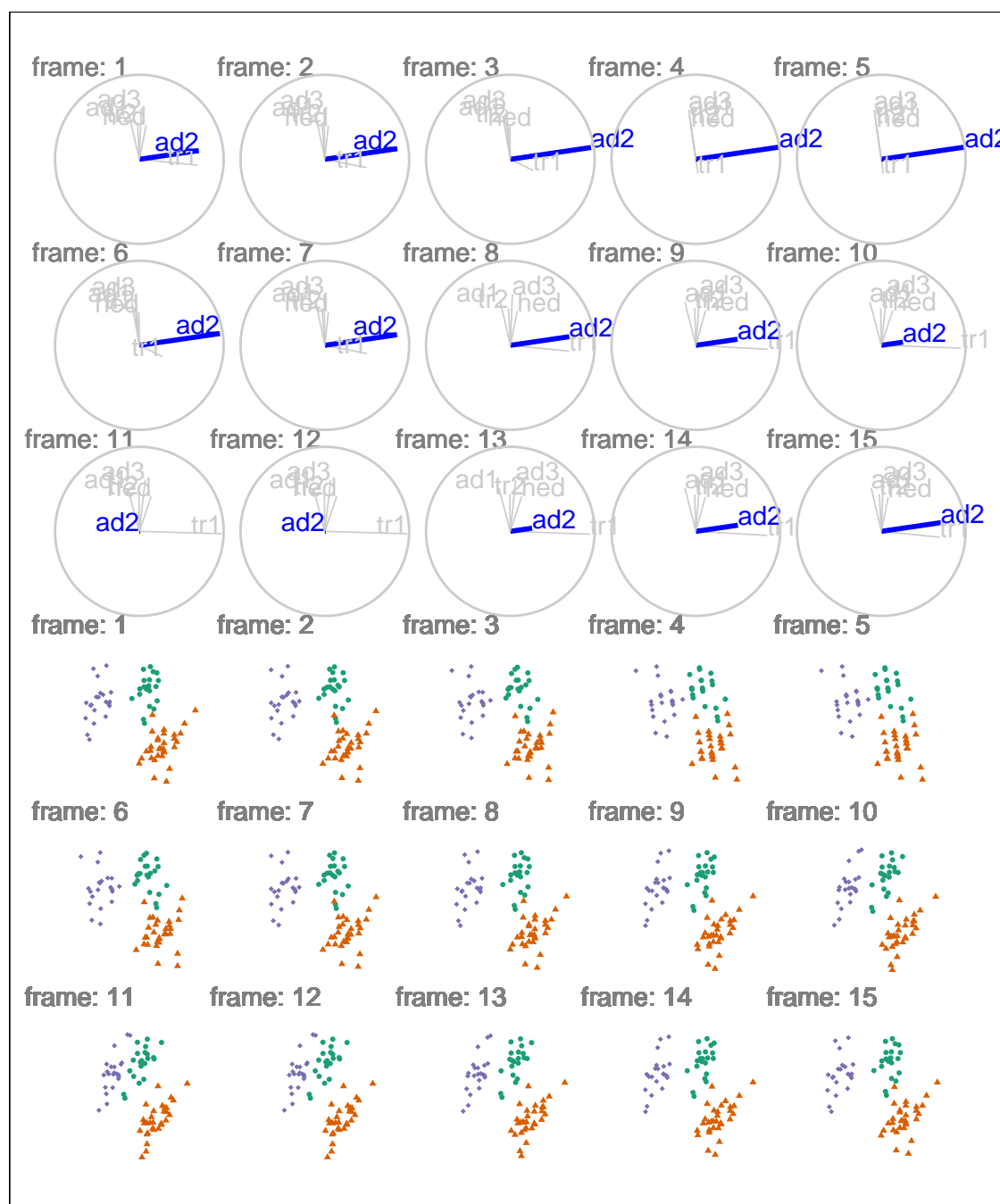
Figure 4: Radial manual tour manipulating aede2 of standardized flea data. The contributions increase from its initial contribution to a full contribution to the projection before decreasing to zero and then returning to its initial value. Manipulating the coefficeients of aede2 changes the distance separating the purple cluster. This shows that aede2 is important for distinguishing this species from the ramaining data. Animated versions made with the `play_manual_tour()` function and can be viewed at https://nspyrison.netlify.com/thesis/flea_manualtour_mvar5/.

Table 1: Primary functions

Class	Function	Description
tour preparation	view_basis	displays the reference frame of the basis
tour preparation	view_manip_space	displays the manipulation space
tour animation	play_tour_path	animates given tour path
tour animation	play_manual_tour	animates the manual tour algorithm
utility	render_gganimate	uses gganimate to render the animation
utility	render_plotly	uses plotly to render the animation

Data in projection-space

In an appeal to performance, the operations of a tour are performed on the bases while the larger datasets are not operated on until the display of data points are needed. After the bases are brought into the projection-space, however, it is helpful to observe them with projected data in the same space. Pre-multiply the data by basis frame bringing the data into the projection space.

$$\mathbf{P}_{[n,\,3]} = \mathbf{X}_{[n,\,p]} \ * \ \mathbf{RM}_{[p,\,3]}$$

For a 2D scatterplot, plot the first two variables within each frame. Show each frame in sequence to form an animation. The remaining manipulation space variable can be linked to point aesthetic (such as size or color) to produce depth cues used in conjunction with the *XY* scatterplot.

Package structure and functionality

This section discusses the functions and code usage of the package before delving into a domain-specific application of the manual tour.

Installation

```
# remotes::install_github("nspyrison/spinifex") # Development version
install.package("spinifex")
library("spinifex")

# Also see the vignette:
vignette("spinifex_vignette")
```

Primary package functions

The primary functions that will aid users in performing their own manual tours are broken into the following groupings: tour preparation, tour animation, and utility.

In the following section describes and demonstrates the usage of the above functions following the order in the algorithm. The utility functions are called by the tour animation functions, passed to the argument render_type. Usage is demonstrated below and in the example documentation.

Algorithm code

We'll start by initializing values including a standardized data set (numeric columns only), a starting basis, a categorical variable for point aesthetics (optional), and a manip var. To get bearings on the projection, start by observing the reference axes of the basis with view_basis() producing Figure 1.

```
f_data  <- tourr::rescale(flea[, 1:6])                 ## standardize data
f_path  <- save_history(f_data, guided_tour(holes()))  ## produce guided tour
f_basis <- matrix(f_path[,, max(dim(f_path)[3])], ncol=2) ## end of guided tour
f_cat   <- factor(flea$species)                        ## categorical var

view_basis(basis = f_basis,
```

```
                data = f_data,
                labels = colnames(f_data))
```

After becoming familiar with this space, select a manip var, the variable to change the contributions of. Use `view_manip_space()` to view the new space with a dimension orthogonal to the projection plane where the manip var has a full contribution. This illustrates how the manip var is manipulated with the addition of the manipulation space as shown in Figure 2.

```
f_mvar  <- 5  ## manip var number

view_manip_space(basis = f_basis,
                 manip_var = f_mvar,
                 labels = colnames(f_data))
```

Now we are ready to perform a manual tour on the selected variable. Use `play_manual_tour()` to perform the algorithm as discussed above, in section 2.2. This is the animated equivalent of Figure 3.

```
angle_speed <- .26

play_manual_tour(data = f_data,
                 basis = f_basis,
                 manip_var = f_mvar,
                 angle = angle_speed,
                 col = f_cat,
                 pch = f_cat)
```

This concludes the content of the algorithm section, however, lets cover animating paths generated in **tourr**. Animate the previously generated guided tour path via `play_tour_path()`. Utility functions can also be passed as arguments into either of the tour animation function to change the resulting graphics object, set `render_type = render_gganimate` to view the animation as a GIF.

```
play_tour_path(tour_path = f_path,
               data = f_data,
               render_type = render_gganimate,
               angle = angle_speed
)
```

Rendering and sharing

The **tourr** package utilizes **base** graphics for the display of tours. **spinifex** allows tours to be rendered in **plotly** as an HTML5 object or **gganimate** as GIF or MP4 files. Sharing of animations is not trivial, especially static formats like print and PDF. Even with dynamic display capturing the correct resolution and aspect ratio can be challenging, while many formats quickly bloat file sizes limiting sharing options. Keep in mind hosting options and exporting functions offered in **plotly**, **gganimate** and **tourr**.

Storage

Storing each data point for every frame of the animation is redundant. Just as operations are performed on the bases, so too should tour paths be stored as bases and a single instance of the data. Consider a radial manual tour, we can store the salient features in 3 bases, where ϕ is at its starting, minimum, and maximum values. The frames in between can be interpolated by supplying angular speed. With the use of the `tourr::save_history()` function, the target bases can be saved. From there geodesic interpolation can be used to populate the intermittent frames. This type of interpolation should not be used on manual tours, which have already been initialized into a 3D manipulation space where direct linear interpolation is appropriate.

Application

In a recent paper, Wang et al. (2018), the authors aggregate and visualize the sensitivity of hadronic experiments to nucleon structure. The authors introduce a new tool, PDFSense, to aid in the visualization of parton distribution functions (PDF). The parameter-space of these experiments lies in 56 dimensions, $\delta \in \mathbb{R}^{56}$, and are visualized as 3D subspaces of the 10 first principal components in linear (PCA) and non-linear (t-SNE) embeddings.

Using the same data, another study, Cook, Laa, and Valencia (2018), applied grand tours (Asimov 1985) to the same subspaces. Grand tours are dynamic linear projections of high dimensional spaces where basis sets are selected at random and animated with geodesic interpolation of the intermediate frames. Because of the change in the basis, or orientation to the subspace, tours are able to better resolve the distribution shape of clusters, intra-cluster detail, and better outlier detection than the use of PDFSense & TFEP (TensorFlow embedded projections) or traditional static embeddings. Before applying manual tours the structure of the data is discussed.

The data has a hierarchical structure with top-level clusters; DIS, VBP, and jet. Each cluster is a particular class of experiments, each with many experimental datasets which, in turn, have many observations. In consideration of data density, we conduct manual tours on subsets of the DIS and jet clusters. This explores the sensitivity of the structure to each of the variables in turn and we present the subjectively best and worst variable to manipulate for identifying dimensionality of the clusters and describing the span of the clusters.

Jet cluster

The jet cluster resides in a smaller dimensionality than the full set of experiments with four principal components explaining 95% of the variation in the cluster (Cook, Laa, and Valencia 2018). The data within this 4D embedding is subset down to ATLAS7old and ATLAS7new to focus in on two groups with a reasonable number of observations that occupy different parts of the subspace. Radial manual tours controlling contributions from PC4 and PC3 are shown in Figure 5 and Figure 6 respectively. These variables are selected to contrast the information conveyed by different manip variables. Links to dynamic HTML5 animations controlling each of the four variables are also provided.

When manipulating PC4, there is a clear difference in the parameter space spanned by the experiment types ATLAS7new and ATLAS7old. Specifically, the variation of ATLAS7new (green) becomes more singular. The experiments are probing different parameter space and PC4 is important to demonstrate this. Yet, when PC3 is manipulated there is no clear indication that the different experiments probe different parameter space. Performing a radial manual tour on PC4 is more insightful than for PC3. Radial manual tours manipulating each of the principal components in the jet cluster can be viewed by following the links: PC1, PC2, PC3, and PC4.

DIS cluster

A different space is used to explore the DIS cluster; specifically the first six principal components, which explains 48% of the variation contained within the aggregated data (Cook, Laa, and Valencia 2018). Radial manual tours are performed on PC6 and PC2 in Figure 7 and Figure 8 respectively.

The selection of the manip variable is important, as the manipulation spaces convey substantially different information. The manual tour of PC6 offers information about the dimensionality, shape, and orientations of the different experiment classes. PC6 is particularly important to describe the variations of DIS HERA1+2 and charm SIDIS observations, Whereas manipulating the contributions of PC2 only shows a subset of the dimensionality and shape information. Manipulating the contributions of PC6 turned out to be much more insightful than PC2. This result might seem counter-intuitive at first as PC2 should explain much more of the variation in the data. However, features and structures in the data regularly reside in smaller dimensionality which can be overlooked when optimizing on full sample statistics. DIS cluster manual tours manipulating each of the principal components can be viewed from the links: PC1, PC2, PC3, PC4, PC5, and PC6.

Discussion

Tours, which are a dynamic linear projection of numeric multivariate data, play an important role in data visualization; they extend the dimensionality of visuals while data- and parameter-spaces become ever larger. This research has modified the algorithm producing manual tours which and has made this functionality available in package **spinifex**. The package adds to **tourr**, extending the graphics offerings that can be used to display tours.

Radial manual tours were applied to a dataset across different experiments of hadronic collisions. The importance of selecting the correct manip var, as demonstrated by comparing tours of varying amounts of structural information. Manual tours, by giving the full contribution of the manipulation space to the manip var, enable analysts to explore the sensitivity of the structure with the selected variable. This information can be used by domain experts to identify which variables are probing which parameter spaces and how sensitive those variables are to structural features. This insight can be indispensable for variable inclusion/exclusions in a model, or in high-level decisions, such as budgeting time and funding.
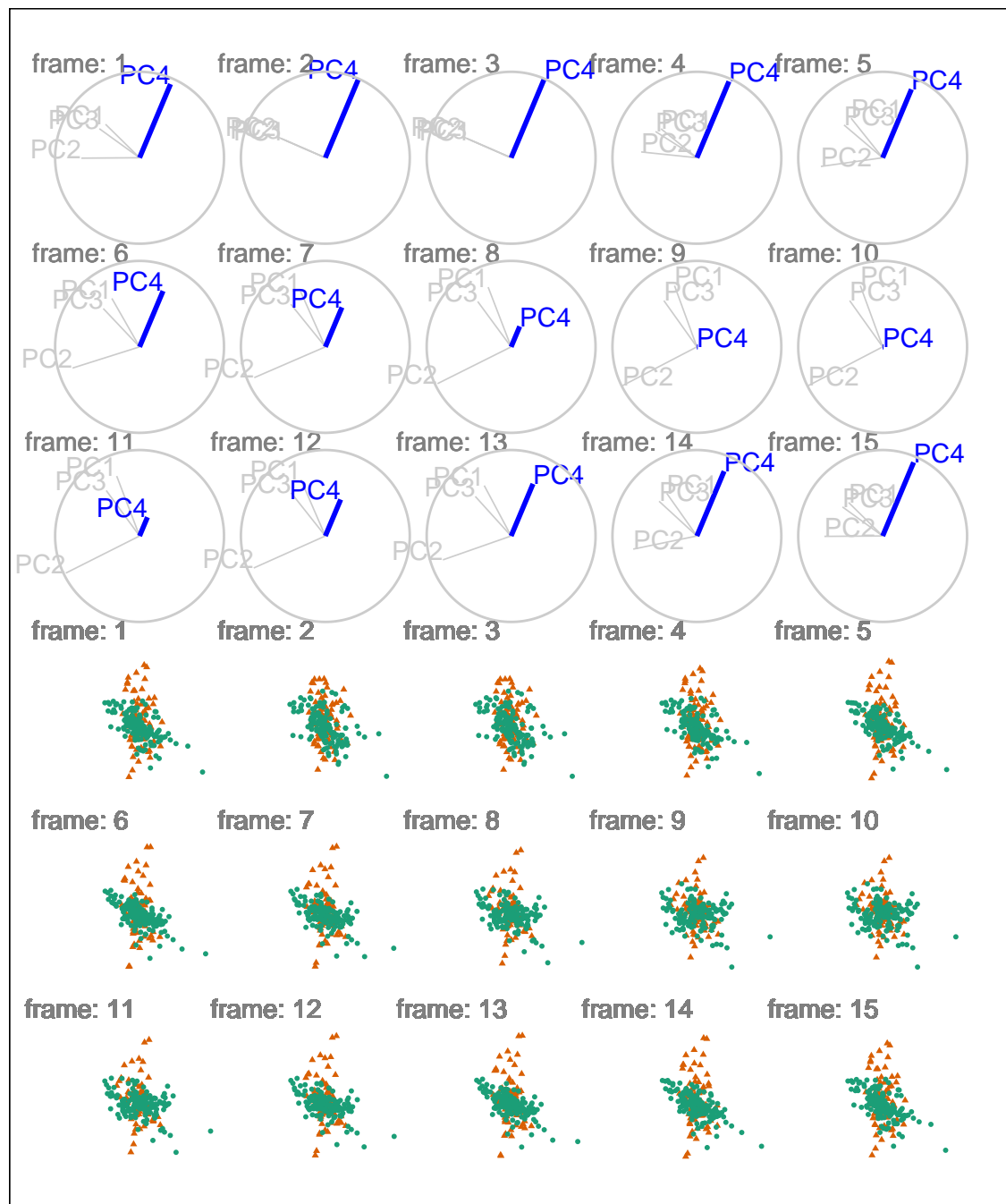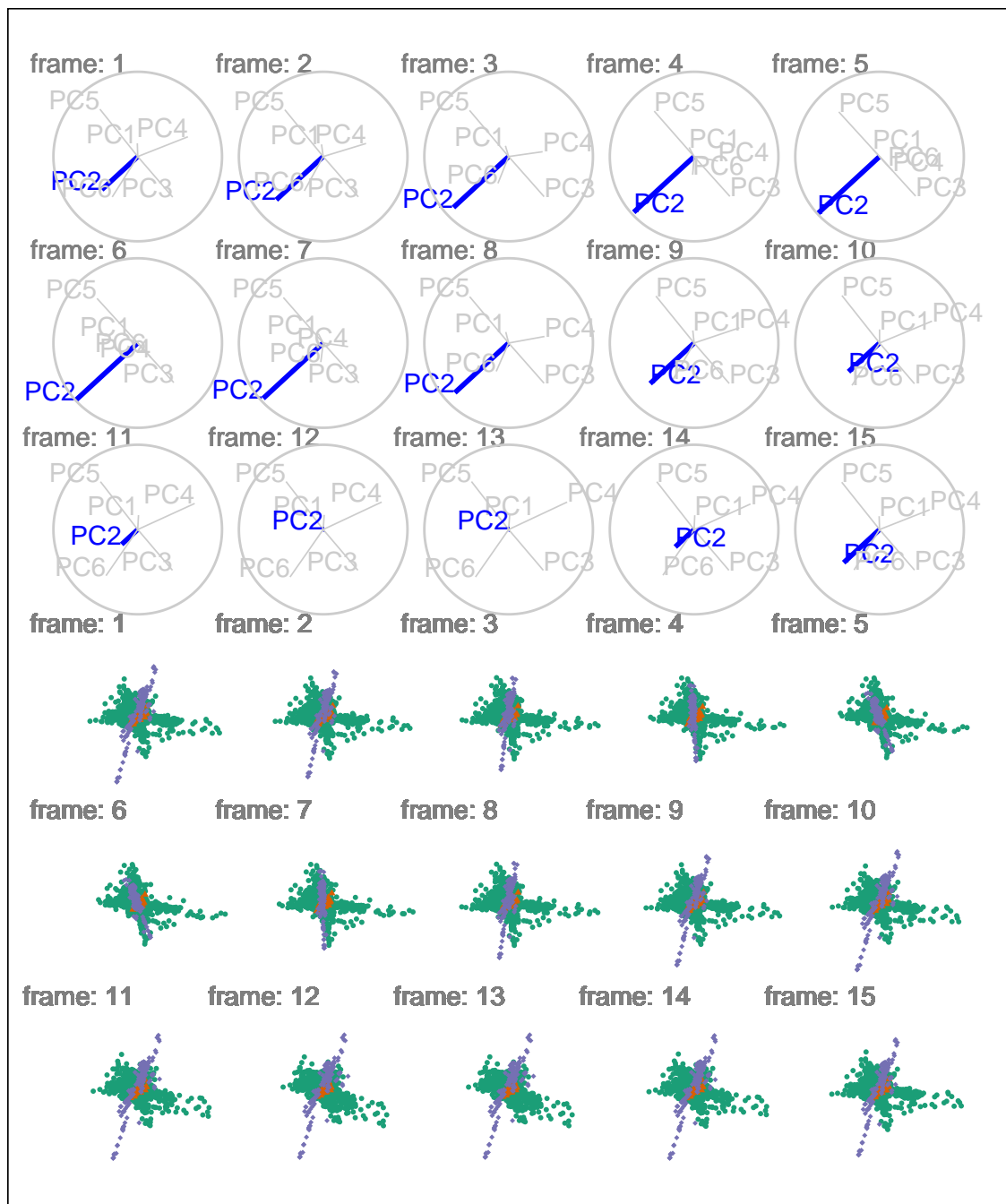
Figure 5: A radial manual tour of PC4 within the jet cluster. Colored by experiment type: AT-LAS7new in green and ATLAS7old in orange. When PC4 fully/negligibly contributes to the projection ATLAS7new (green) spans the same space as the orange points. During the intermediate frames, the ATLAS7new is compressed in the direction radial to PC4. The differece in distribution shape demonstrates the experiments probe different phase-space, which has a linear mapping back to the original variables for interpretation and further exploration. An HTML5 version can be viewed at https://nspyrison.netlify.com/thesis/jetcluster_manualtour_pc4/.
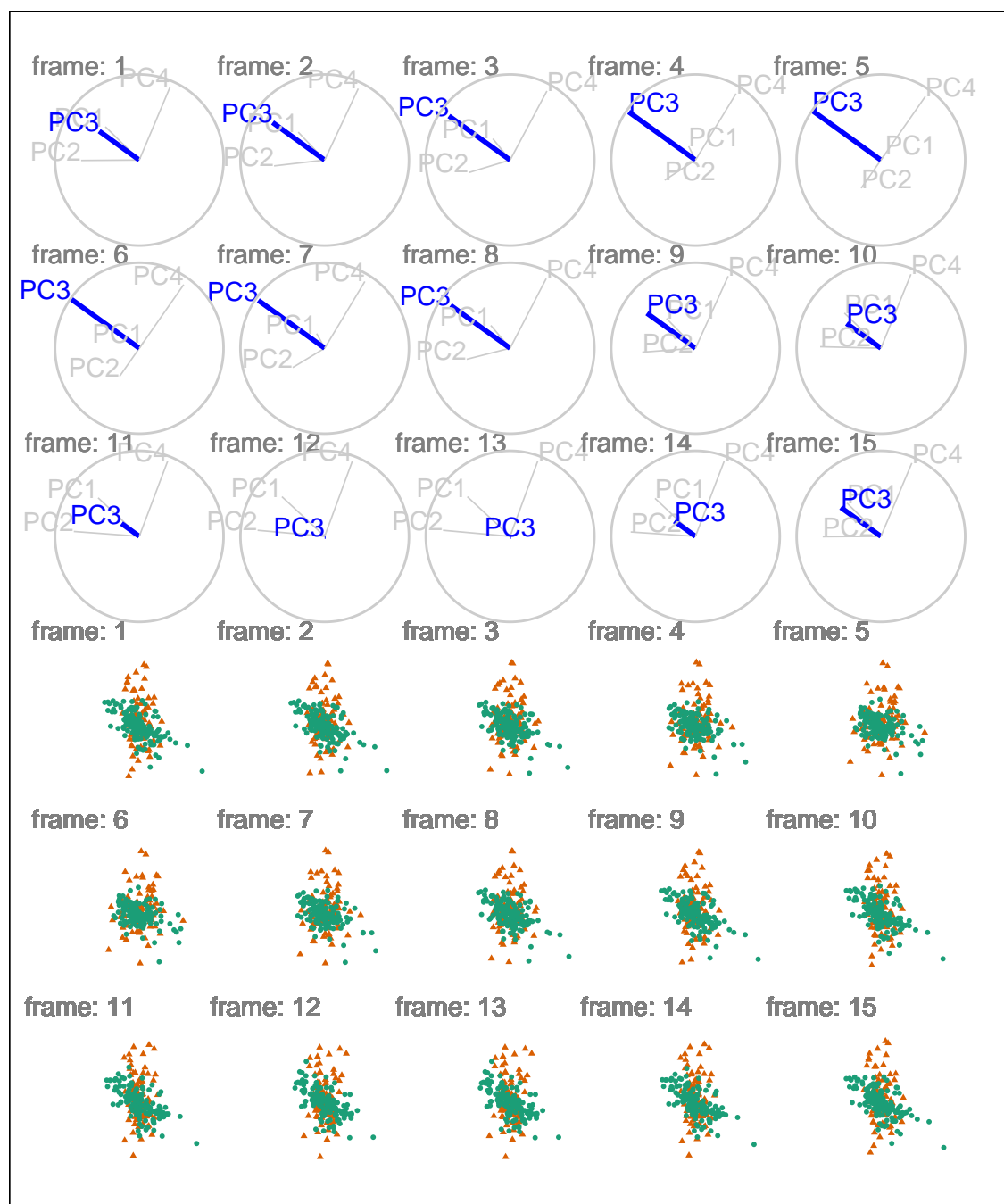
Figure 6: A radial manual tour of PC4 within the jet cluster. Colored by experiment type: ATLAS7new in green and ATLAS7old in orange. Data from ATLAS7new (green) spans mostly the same space as ALTLAS7old (orange) with no evident difference in cluster structure across varying contributions of PC3. An HTML5 version can be viewed at https://nspyrison.netlify.com/thesis/jetcluster_manualtour_pc3/.
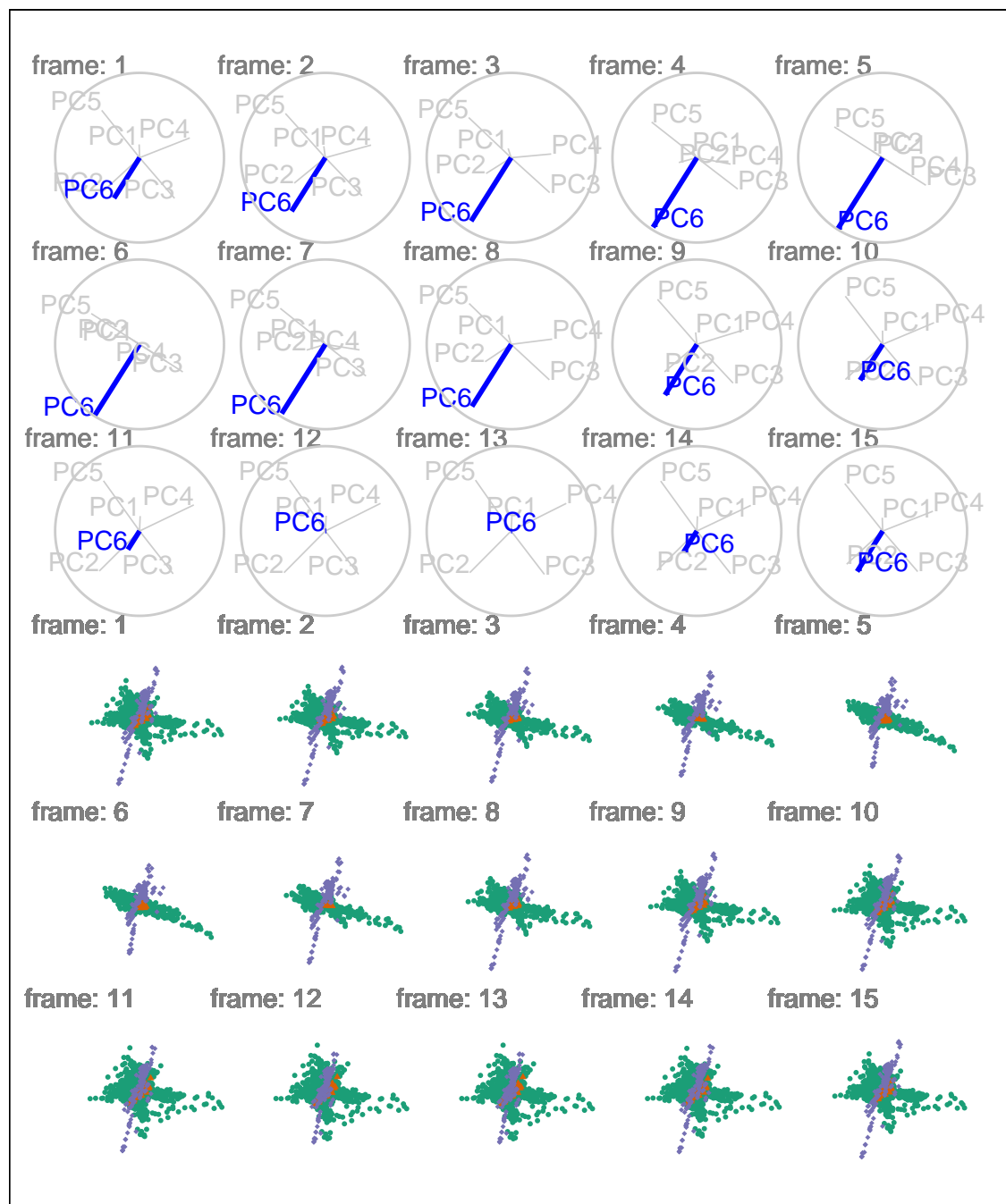
Figure 7: A radial manual tour manipulating the contribution of PC6 within the DIS cluster. Points are colored by experiment type: DIS HERA1+2 in green, dimuon SIDIS in purple, and charm SIDIS in orange. The cluster DIS HERA1+2 (green) is distributed in a cross-shaped plane, charm SIDIS (orange) occupies the center space of this cross. As the contribution of PC6 becomes whole the distributions of DIS HERA1+2 (green) and charm SIDIS (orange) become singular but offset by a small angle. Less evident is the linear dimuon SIDIS (purple) observations approaching the line of view for intermediate values of PC6. An HTML5 version can be viewed at https://nspyrison.netlify.com/thesis/discluster_manualtour_pc6/.

Figure 8: A radial manual tour manipulating the contribution of PC2 within the DIS cluster. Points are colored by experiment type: DIS HERA1+2 in green, dimuon SIDIS in purple, and charm SIDIS in orange. The plane of cross distributed DIS HERA data (green) and a nearly orthogonal jet of dimuon SIDIS (purple) is present. This jet does extend more in the plane of view when the contribution of PC2 is full, giving insight to its orientation. However, less information about the shape of DIS HERA (green) and charm SIDIS (orange) is available compared to PC6 as the manip var. An HTML5 version can be viewed at https://nspyrison.netlify.com/thesis/discluster_manualtour_pc2/.

Future research on the algorithm would include extending it for use in 3D projections. The addition of another dimension theoretically allows for improved perception. In some frameworks, such as the game engine **unity**, this would allow for the exploration in immersive virtual reality or mixed reality, which may further allow for a better perception of structure and aid in higher-dimensional function visualization. Functions with many parameters suffer from the same dimensionality problem as data while their possible values lie on a surface of values rather than discrete points. Occulation, or the closer surface blocking further surfaces, will likely be an issue that may be alleviated by the use of wire mesh, changing opacity, or looking at sections of the projections known as prosections (Furnas and Buja 1994).

The **tourr** package provides many other geometric displays with the `tourr::display_*()` family. These geometric options could be integrated into the **ggplot2** framework for display on **plotly** and **gganimate**. Additionally, the **animation** package Xie et al. (2018) could be implemented for another graphics framework. However, **animation** builds from **base** graphics while **spinifex** utilizes **ggplot2** graphics, a significant paradigm shift, which may have a low return on investment.

The Givens rotations and Householder reflections as outlined in Buja et al. (2005) could also be added. Currently, Gram-Schmidt is the only form of frame interpolation used. In a Givens rotation, the x and y components (for example $\theta = 0$, $pi/2$) of the in-plane rotation are calculated separately and would be applied sequentially to produce the radial rotation. Householder reflections define reflection axes to project points on to the axes to generate rotations.

Having script-only interaction with tours causes a significant barrier to entry. To a lesser extent, **plotly** offers some static interactions with the contained object, such as tooltips, brushing, and linking without communicating back to the R console. The development of a dynamic graphical user interface, perhaps with the use of a **shiny** (Chang et al. 2018) application, would mitigate the barrier to entry, allow for more rapid analysis, and offer an approachable demo tool. The user could easily switch between variables to control, adjust the interpolation step angle, or flag/save specific frame basis sets.

Acknowledgments

Bibliography

Asimov, Daniel. 1985. "The Grand Tour: A Tool for Viewing Multidimensional Data." *SIAM Journal on Scientific and Statistical Computing* 6 (1): 128–43. https://doi.org/https://doi.org/10.1137/0906011.

Buja, Andreas, Dianne Cook, Daniel Asimov, and Catherine Hurley. 2005. "Computational Methods for High-Dimensional Rotations in Data Visualization." In *Handbook of Statistics*, 24:391–413. Elsevier. https://doi.org/10.1016/S0169-7161(04)24014-7.

Chang, Winston, Joe Cheng, J. J. Allaire, Yihui Xie, and Jonathan McPherson. 2018. *Shiny: Web Application Framework for R*. https://CRAN.R-project.org/package=shiny.

Cook, Dianne, and Andreas Buja. 1997. "Manual Controls for High-Dimensional Data Projections." *Journal of Computational and Graphical Statistics* 6 (4): 464–80. https://doi.org/10.2307/1390747.

Cook, Dianne, Andreas Buja, Javier Cabrera, and Catherine Hurley. 1995. "Grand Tour and Projection Pursuit." *Journal of Computational and Graphical Statistics* 4 (3): 155. https://doi.org/10.2307/1390844.

Cook, Dianne, Ursula Laa, and German Valencia. 2018. "Dynamical Projections for the Visualization of PDFSense Data." *Eur. Phys. J. C* 78 (9): 742. https://doi.org/10.1140/epjc/s10052-018-6205-2.

Furnas, George W., and Andreas Buja. 1994. "Prosection Views: Dimensional Inference Through Sections and Projections." *Journal of Computational and Graphical Statistics* 3 (4): 323–53. https://doi.org/10.2307/1390897.

Kirkpatrick, Scott, C. Daniel Gelatt, and Mario P. Vecchi. 1983. "Optimization by Simulated Annealing." *Science* 220 (4598): 671–80. https://doi.org/10.1126/science.220.4598.671.

Lubischew, Alexander A. 1962. "On the Use of Discriminant Functions in Taxonomy." *Biometrics*, 455–77. https://doi.org/10.2307/2527894.

Pedersen, Thomas Lin, and David Robinson. 2019. *Gganimate: A Grammar of Animated Graphics*. http://github.com/thomasp85/gganimate.

R Core Team. 2018. *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing. https://www.R-project.org/.

Sievert, Carson. 2018. *Plotly for R*. https://plotly-book.cpsievert.me.

Wang, Bo-Ting, T. J. Hobbs, Sean Doyle, Jun Gao, Tie-Jiun Hou, Pavel M. Nadolsky, and Fredrick I. Olness. 2018. "Mapping the Sensitivity of Hadronic Experiments to Nucleon Structure." *Physical Review D* 98 (9): 094030. https://doi.org/10.1103/PhysRevD.98.094030.

Wickham, Hadley. 2016. *Ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York. http://ggplot2.org.

Wickham, Hadley, Dianne Cook, Heike Hofmann, and Andreas Buja. 2011. "**Tourr** : An *R* Package for Exploring Multivariate Data with Projections." *Journal of Statistical Software* 40 (2). https://doi.org/10.18637/jss.v040.i02.

Xie, Yihui. 2014. "Knitr: A Comprehensive Tool for Reproducible Research in R." In *Implementing Reproducible Computational Research*, edited by Victoria Stodden, Friedrich Leisch, and Roger D. Peng. Chapman; Hall/CRC. http://www.crcpress.com/product/isbn/9781466561595.

Xie, Yihui, J. J. Allaire, and Garrett Grolemund. 2018. *R Markdown: The Definitive Guide*. Boca Raton, Florida: Chapman; Hall/CRC. https://bookdown.org/yihui/rmarkdown.

Xie, Yihui, Christian Mueller, Lijia Yu, and Weicheng Zhu. 2018. *Animation: A Gallery of Animations in Statistics and Utilities to Create Animations*. https://yihui.name/animation.

Nicholas Spyrison
Monash University
Faculty of Information Technology
Nicholas.Spyrison@monash.edu

Dianne Cook
Monash University
Department of Econometrics and Business Statistics
dicook@monash.edu