TreeScaper Workshop Activity 1: Non-Linear Dimensionality Reduction

Jeremy Ash

January 5, 2017

Introduction

Phylogenetic trees are now routinely inferred from enormous genome-scale data sets, revealing extensive variation in phylogenetic signal both within and between individual genes. This variation may result from a wide range of biological phenomena, such as recombination, horizontal gene transfer, or hybridization. It may also indicate stochastic and/or systematic error. However, current approaches for summarizing the variation in a tree set typically condense it into point estimates, such as consensus trees, resulting in extensive loss of information.

We have written TreeScaper to provide a set of visual and quantitative tools for exploring and characterizing the full complement of phylogenetic information contained in a tree set. These tools can be broadly categorized into three types:

1. Utilities for calculating basic information about topologies and bipartitions.
2. Visualization of treespace in 2- or 3-dimensional space through non-linear dimensionality reduction (NLDR)
3. Detection and delineation of distinct ‘communities’ of trees (discussed in part 2).

Utilities

Much of TreeScaper’s functionality requires calculating distances between trees, transforming distances into affinities, translating trees into their component bipartitions, and summarizing how these bipartitions are distributed across trees (i.e., their variances and covariances). However, this information can also be useful in its own right. Therefore, TreeScaper provides a set of built-in utilities to calculate a range of useful tree- and bipartition-related summaries. Once calculated, these values may be used for other tasks in TreeScaper or may be written to file for use in other applications.

NLDR

One way to visually explore tree sets is to plot a 2- or 3-dimensional representation of treespace using non-linear dimensionality reduction (NLDR). This approach was first suggested for the visualization of phylogenetic trees by Amenta and Klingner (2002) and Hillis et al. (2005), and recently extended by Wilgenbusch et al. (2016). The general idea behind NLDR is to find a lower dimensional representation of the relationships among trees that best preserves the true distances between them. TreeScaper implements several different methods for calculating tree-to-tree distances [e.g., Robinson-Foulds (RF) distances, matching distances, and subtree-prune-regraft (SPR) distances], several stress functions to assess how the original tree-to-tree distances should be optimally represented in lower dimensional space [e.g., Normalized stress, Kruskal-1 stress, Sammon’s nonlinear mapping (NLM) stress, and Curvilinear Components Analysis (CCA) stress], and several optimization algorithms for finding the best low-dimensional representation given a chosen stress function [e.g., Gauss-Seidel-Newton, stochastic gradient descent, and simulated annealing].

Practical: Visualizing a Squamate mtDNA Tree Set

First, to get an idea of what tree space looks like for this tree set, we will visualize it in 2D and 3D using non-linear dimensionality reduction (NLDR) projection methods. In these projections, each point represents a tree in the tree set, and the distance between two points is an approximation of the actual tree-to-tree distance.

The Data

Download the github repository. In the top right corner of the github page there is a button that reads *clone or download*. Click this and download the repository. The data subdirectory will contain the tree set files we will work with during the tutorials. The tree set we will be working with is called squamates.nex.

These data consist of 1,300 trees sampled from individual Bayesian analyses of 13 mitochondrial protein-coding genes in squamates (Castoe et al. 2009). Each tree was sampled from the posterior distribution of one gene (ie. one hundred trees were sampled from each of the 13 posterior distributions).

Open the tree set file in a text editor of your choice. The file is in standard NEXUS format with a tree block and a translate command. TreeScaper will not have problems parsing a NEXUS file with a TREES block, but a translate command *IS* necessary.

Guiding Research Questions

- Are there topological differences between the phylogenies inferred from the 13 mtDNA genes?
- Do some NLDR projections better preserve the topological distances in the original treespace than others?
- Is 3D necessary to visualize this tree space?

Formatting Conventions

Throughout this tutorial, we format the text differently when referring to different TreeScaper components for clarity. *Tabs and dropdown menus* are italicized, **buttons** are bolded, “options” are in quotes.

Output

At any point, you can use the data menu in the bottom left pane to see what data structures you have saved in memory. Select any data structure and double click it to output or delete that data. Deleting data structures will allow you to control TreeScaper’s memory footprint. When outputting data structures, the name of the output file will be printed to the log.

Load in the Data

Starting off in *Trees and Bipartition Computation*, **Load Tree Data** should already be selected. **Browse** and select the tree file. Select “weighted trees”, but do not select “rooted”. Click **Load all Trees**. This will read from file a set of unrooted trees with branch lengths. You should see a message in the log that reads:

```
Successfully read 1300 trees
```

You should also see a weighted treeset data structure in the data menu.

Computing Tree-to-Tree Distance Matrix

From the main dropdown menu, select “Load Distance/Coordinate Data or Compute Tree-to-Tree Distances”. In this case, we will calculate the tree-to-tree distances from scratch, but you could also load distances from file if they were calculated previously. If you plan on conducting many analyses with the same large tree set, storing and loading the tree distances will save considerable time.

From the *Method* dropdown select “Weighted Robinson Foulds”, then click **Distance** to compute a matrix of pairwise distances between trees in your tree set. When the weighted Robinson Foulds (RF) distance is computed, bipartitions are weighted by their internal branch length.

You should see a dialogue box appear that reads:

```
Warning: There is no bipartition matrix in the memory! Do you want to compute it?
```

Click *OK*. After a few moments the distance matrix should be computed. The frequency of each bipartition in the tree set will be output to the log. Disregard this - we will discuss this output later. For now, all you need to know is that the bipartitions occuring in each tree topology need to be identified in order for TreeScaper to compute the weighted RF distance. After the bipartition frequency ouput, two messages should appear:

```
Successfully computed bipartition matrix
```

```
Successfully computed Weighted Robinson Foulds distance
```

You should also see a bipartition matrix and weighted RF distance in the data menu.

We computed the weighted RF distance since it is fast to compute and the metric is intuitive. We have analyzed this dataset with the other distance metrics and arrived at the same conclusions we will come to in this tutorial. Note that the unrooted SPR distance is computationally intensive and for moderate to large tree sets.

Non-Linear Dimensionality Reduction: 2D

In the topmost tab menu, now select **NLDR and Dimension Estimation**. In the main dropdown menu under this tab, “Non-linear Dimension Reduction”" should already be selected. Make sure that the appropriate distance matrix is selected (“Weighted RF-distance”) in the *Distance* dropdown, with “CCA” and “Stochastic” selected in *Method* and *Algorithm* dropdowns, respectively. The *Dimension* selected should be 2. Make sure that “Random” is selected in the *Initial Projection* dropdown.

Next, click the **Plot Parameters** button. A new dialogue box should appear.

Note, you may see an error message warning that the indices have not been set correctly. This is due to the fact the TreeScaper default indexing does not work with this data set. Disregard this as we will be setting these indices correctly soon.

Points should be selected in the dialogue’s dropdown menu. Set the “Number of Clusters” to 13, then click **OK**. 13 clusters should be present in the “Cluster Index” section. Check the “Step Size” box. Set the step size to 100. The 13 clusters should now include all 1300 trees in the tree set (with indices starting at 0). This assigns trees into clusters according to the mitochondrial gene they belong to. The NLDR visualization will now color points by mitchondrial gene.

Under “Other parameters” change “point size” to 4, then click **Apply**, followed by **Close**.

Now, click **Run NLDR**. Once the NLDR method is finished running, click **Plot Result**. A window with the plot of tree space should appear. A number of files should also be output to the directory from which you are running TreeScaper.

Each time an NLDR analysis is run in TreeScaper, a series of output files are automatically created. These files have names indicating the name of the tree set, the tree-to-tree distance, the NLDR method, and the optimization algorithm. If NLDR is run multiple times and any of these parameters are changed, a new set of files will be created. If NLDR is run multiple times with the same parameters, only the final run will be saved. The log will tell the directory in which the files are written and the file prefix in case you have forgotten:

```
Start program: /home/jrash/Documents/Castoe_etal_analyses/squamates_Match_2D_CCA_STOCHASTIC
```

Here are descriptions of the output files (*s represent the filename, dimension, NLDR method and optimization algorithm, respectively):

- *_*D_*_COR_*.out: This file contains the NLDR coordinates. Each row corresponds to a tree depicted as a point in low-dimensional space. These coordinates can be used to generate figures of the NLDR projections in other plotting software.
- *_*D_*_DIS_*.out: The Euclidean distance matrix of the NLDR coordinates.
- *_*D_*_STR_*.out: The value of stress function after optimization.
- *_*D_*_CON_*.out: The continuity. The first column is the number of neighbors considered. The second column is the continuities.
- *_*D_*_TRU_*.out: The trustworthiness. The first column is the number of neighbors considered. The second column is the trustworthiness.

Your plot should look similar to Figure 1. Scroll up and down to zoom in and out.

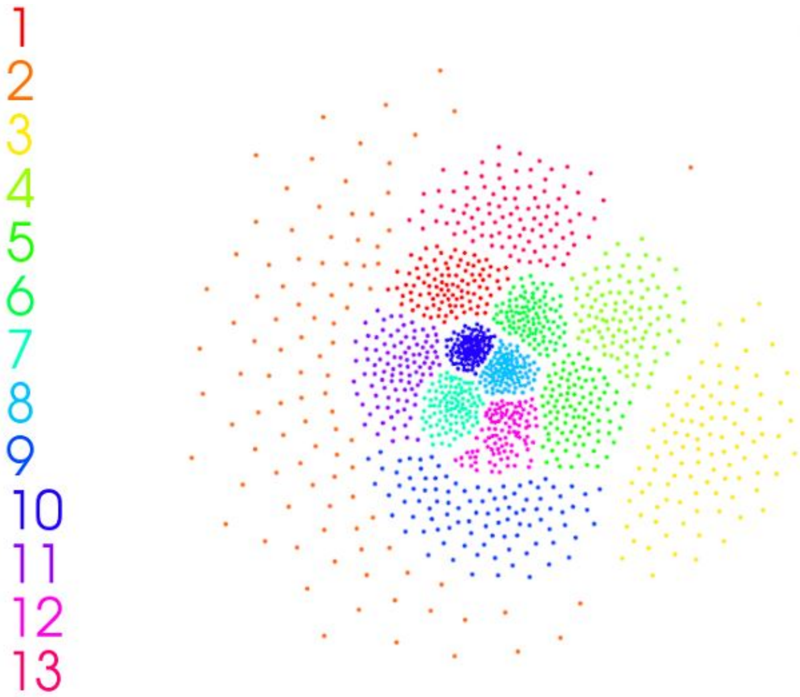


Figure 1: Squamate mitochondrial gene trees projected into 2D using Curvilinear Components Analysis. Points are colored by gene.

Note that like phylogenetic tree searching, NLDR optimization is non-convex. This means that the optimization alogorithm may become trapped in local optima. We advise running the NLDR projection multiple times from different starting points (using the *Initial Projection*→“Random” option). See if the results change considerably. If they do, use the projection with smallest associated stress (what the cost function is minimizing - a measure of the distortion of the tree-to-tree distances in the NLDR visualization). This value is provided in the log output:

```
stress value : 147.565
```

In this example, running the NLDR multiple times from different starting points doesn’t result in any noticeable differences in the visualization.

The 2D projection suggests that distinct tree topologies have been inferred from each of the mitochondrial genes. If you are using a Mac, you can interact with this plot to investigate these topological differences. Hold down Shift+R, select a small number of points (1-3) in one cluster of trees then click the **Plot trees** button to examine each of these trees individually. Repeat this for the another cluster of trees. Note the differences in topology.

Intrinsic Dimensionality Estimation

To examine whether a projection in 3D will better represent the true tree-to-tree distances, we will perform intrinsic dimensionality estimation. For the sake of concision, we will only use the Maximum Likelihood estimator. We have also implemented the Correlation and the Nearest Neighbor estimators in TreeScaper.

See the Appendix for an in-depth explanation of the Maximum Likelihood estimator of intrinsic dimensionality. In short, the location of each data point is assumed to be idependently identically distributed according to a probability distribution that has a parameter for the intrinsic dimensionality of the data. This parameter is estimated using maximum likelihood.

In the main dropdown menu, select **Dimension Estimation**. Select **Maximum Likelihood** in the *Estimator* dropdown. Set the **# of points** to 20. This will set the maximum number of nearest neighbors to consider for the estimation of the intrinsic dimension. Click **Run and obtain analysis result**.

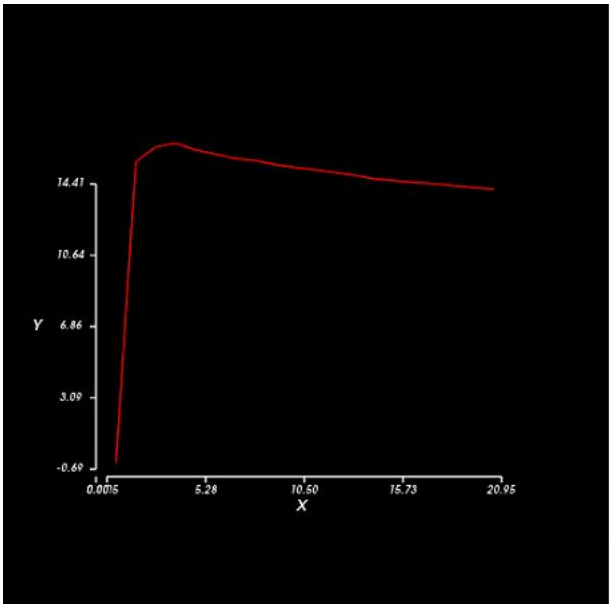


Figure 2: Dimensionality estimation of the Squamate mtDNA tree set using the Maximum Likelihood estimator

The maximum likelihood estimate of the intrinsic dimensionality is affected by the number of k-nearest neighbors considered for each data point. The common approach to this problem is to estimate the dimensionality over a range of values of k and then average the dimension estimates (**???**). Levina and Bickel recommend plotting k versus the intrinsic dimensionality estimate and finding a range of small values of k for which the plot is reasonably flat (**???**). Click **Plot analysis result**. You should see the plot in Figure 2. On the x-axis is the value of k and on the y axis is the estimate of intrinisic dimensionality. Every number from 1 to the maximal number of k we set previously was used to compute the intrinsic dimensionality.

We choose to take the range of k from 10 to 20. The dimensionality estimated should be 14.23. This implies that a projection on a 15 dimensional manifold would result in an accurate representation of the data. We conclude that we should visualize the tree set in 3 dimensions to obtain a more accurate representation.

Non-Linear Dimensionality Reduction: 3D

We proceed to visualize the tree set in 3 dimensions. In the topmost tab menu, now select **NLDR and Dimension Estimation**. Change *Dimension* to 3. Keep the other options set the way they were during the 2D visualization. Click **Run NLDR** and **Plot Result** once the projection has been computed. The resulting figure should look like Figure 3.

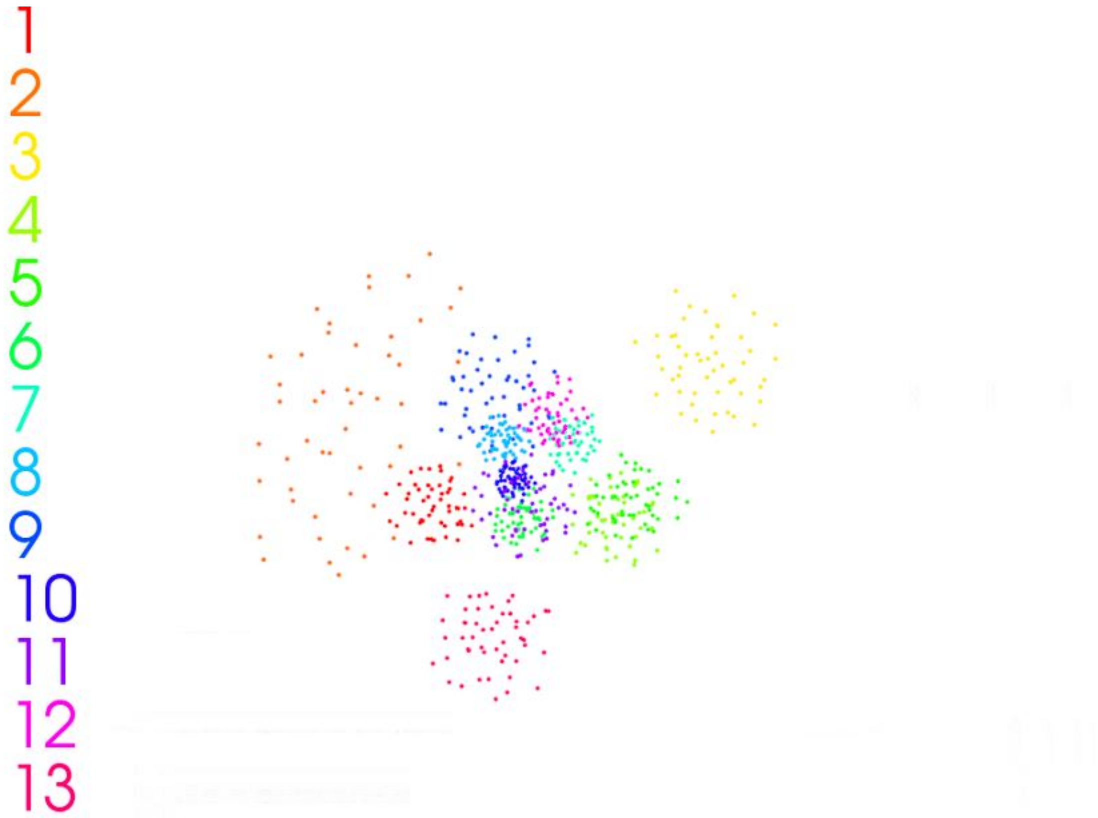


Figure 3: One rotation of the squamate mitochondrial gene trees projected into 3D using Curvilinear Components Analysis. Points are colored by gene.

Click and drag to rotate the 3D plot. Scroll up and down to zoom in and out. It appears that the trees from separate genes are still forming distinct clusters.

Convex Hulls

It is difficult to determine if the clusters are indeed distinct using individual points for each tree. Convex hulls are particularly useful for seeing the relative location of each cluster. A convex hull can be thought of as a convex polygon that encloses every point in a cluster and has smallest possible volume. A convex polygon is a polygon that is solid (ie. it has no 'holes').

Click **Plot Parameters**. You may see an error saying that the tree indices were not set correctly. This is a bug we are currently fixing. Make sure that there are still 13 clusters and 100 trees in each cluster. You may need to reset the tree indices as we did for the 2D projection. Select "Convex Hull" in the dropdown.

Convex hulls are sensitive to large outliers because a few large outliers will greatly increase the convex hull volume. We will eliminate points that are large outliers. Set "outliers stop condition" to .01. A point will be considered an outlier and removed from a set of points if the variance of the distance among all points decreases by .01 when the point is excluded from the variance calculation. Decreasing the threshold value will eleminate points more aggressively. Apply the changes and close the dialogue box.

Now click **Plot Result**. Your plot should now resemble Figure 4.

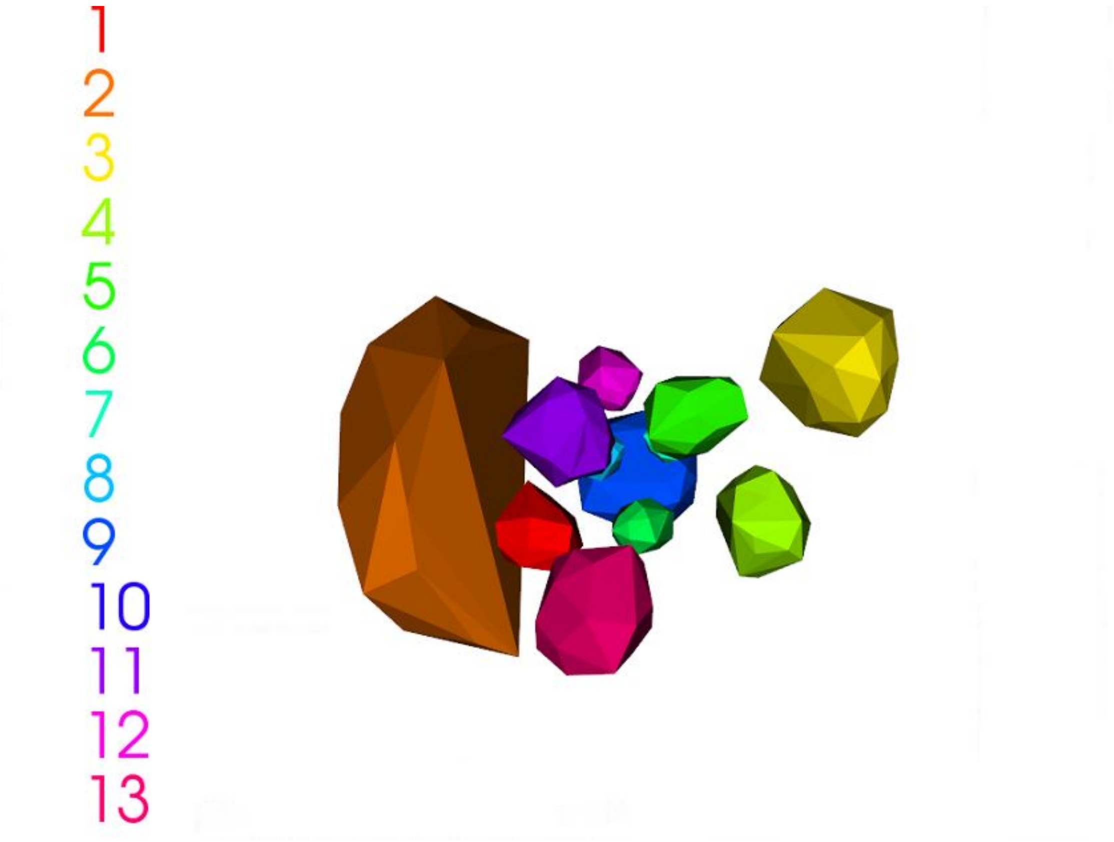


Figure 4: One rotation of the squamate mitochondrial gene trees projected into 3D using Curvilinear Components Analysis. Each set of gene trees are represented as a convex hull.

You can see that several clusters of genes have considerable separation in this 3D projection. This suggests that the phylogenies inferred from these mitochondrial genes have different topologies. We will attempt to more objectively determine whether the trees from different genes form distinct clusters in the second tutorial. We will also demonstrate how to determine what topological features differ between clustes of trees.

Measures of Visualization Quality

We have implemented two measures of the NLDR visualization quality in TreeScaper. When multiple NLDR methods are used to visualize a tree set, these measures can be used to determine their relative performance. We will compare the visualization of our tree set using the CCA, NLM, and Normalized stress NLDR methods. We will only use the stochastic gradient descent optimization algorithm as we have found this optimization algorithm to often have the best performance.

Both continuity and trustworthiness range from 0 to 1.

Continuity is a measure of the extent to which the k nearest neighbors of each tree in the original space are close by in the display. This measure is computed for a sequence of values of k in the output file labeled: ‘squamates_RF_3D_CCA_CON_STOCHASTIC.out’.

Trustworthiness is a measure of the extent to which the k nearest neighbors of each tree in the display are close by in the original space. This measure is computed for a sequence of values of k in the output file labeled: ‘squamates_RF_3D_CCA_TRU_STOCHASTIC.out’.

Perform NLDR using two other methods “NLM” (Sammon’s Non-Linear Mapping) and “Normalized” (Normalized stress). In R, one can easily plot the measures of visualization quality for each NLDR method and assess their relative performance. We start with trustworthiness.

For this code to work, you will need to set the working directory to the directory containing the output files for this analysis (the directory containing the tree set input into TreeScaper).

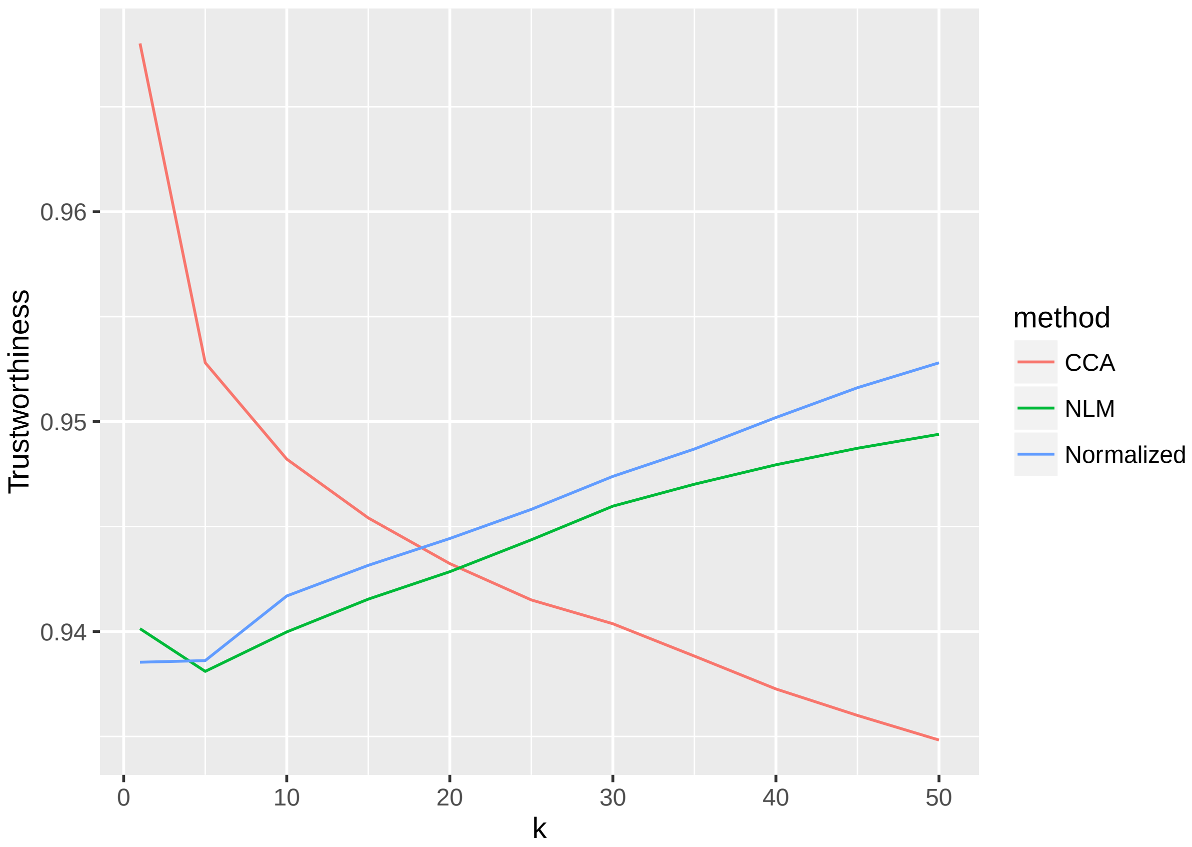
```r
#you will need to install ggplot2 with: install.packages("ggplot2")
library(ggplot2)

cca.tru <- data.frame(read.table("squamates_RF_3D_CCA_TRU_STOCHASTIC.out"), method = "CCA")
nlm.tru <- data.frame(read.table("squamates_RF_3D_SAMMON_TRU_STOCHASTIC.out"), method = "NLM")
norm.tru <- data.frame(read.table("squamates_RF_3D_NORMALIZED_TRU_STOCHASTIC.out"), method = "Normalize

all.tru <- rbind(cca.tru, nlm.tru, norm.tru)

colnames(all.tru) <- c("k", "Trustworthiness", "method")

ggplot(all.tru, aes(x = k, y = Trustworthiness, colour = method)) + geom_line()
```



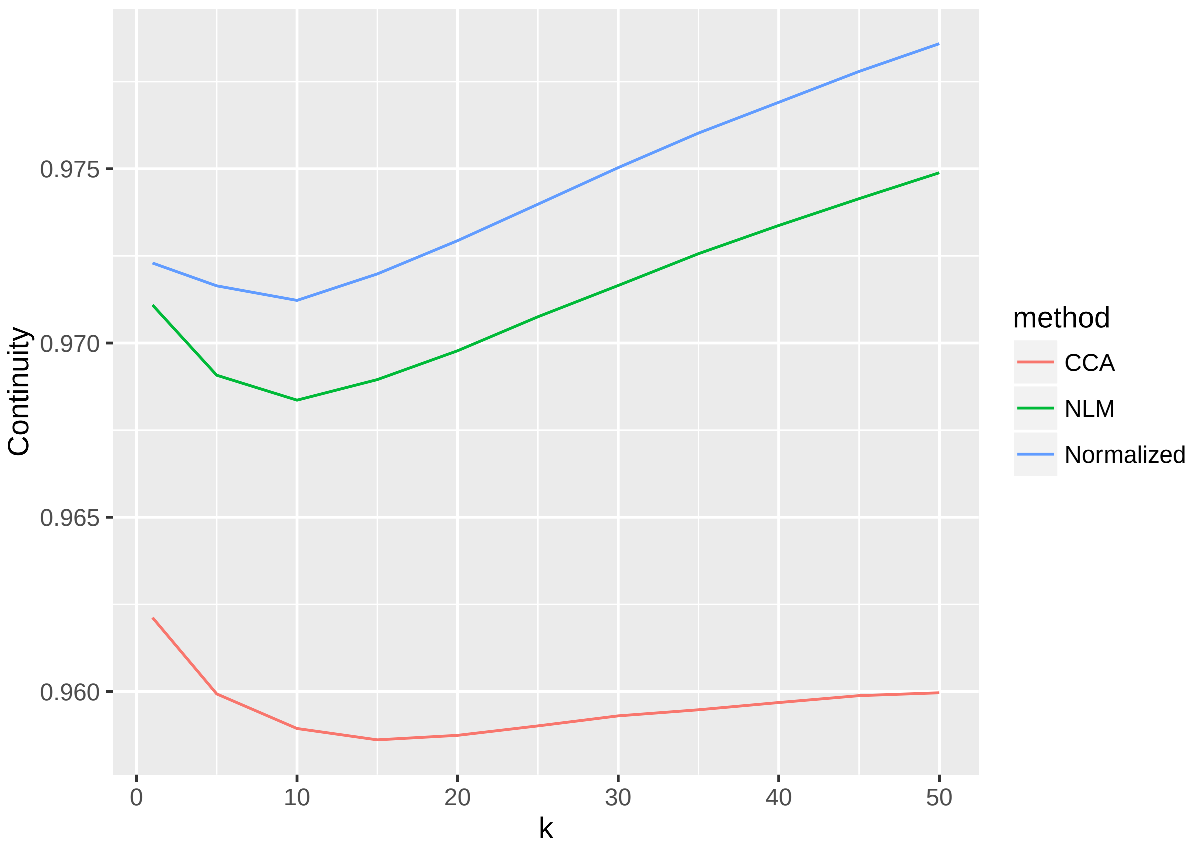
Normalized stress appears to perform better than the other NLDR methods across the majority of the values of k considered. The trustworthiness value is consistently close to one, implying that points that cluster together in the original space also cluster together in the display.

Next we consider continuity.

```
cca.con <- data.frame(read.table("squamates_RF_3D_CCA_CON_STOCHASTIC.out"), method = "CCA")
nlm.con <- data.frame(read.table("squamates_RF_3D_SAMMON_CON_STOCHASTIC.out"), method = "NLM")
norm.con <- data.frame(read.table("squamates_RF_3D_NORMALIZED_CON_STOCHASTIC.out"), method = "Normalize

all.con <- rbind(cca.con, nlm.con, norm.con)
colnames(all.con) <- c("k", "Continuity", "method")

ggplot(all.con, aes(x = k, y = Continuity, colour = method)) + geom_line()
```



Normalized stress outperforms the other NLDR methods for all the values of k considered. The continuity value is consistently close to one, implying that points that cluster together in the display also cluster together in the original space.

Since both continuity and trustworthiness are close to one for Normalized stress, we are confident that points that are forming distinct clusters in the original space are also forming distinct clusters in the display.

The plots demonstrate that NLDR with Normalized stress mostly outperforms other NLDR methods. We conclude that though NLDR with the CCA method seems to be performing well, it would be preferable to visualize this data set with Normalized stress. Visualizing the tree set with this method does not change our conclusions (Figure 5), but it does show better separation for the mitchondrial gene clusters.

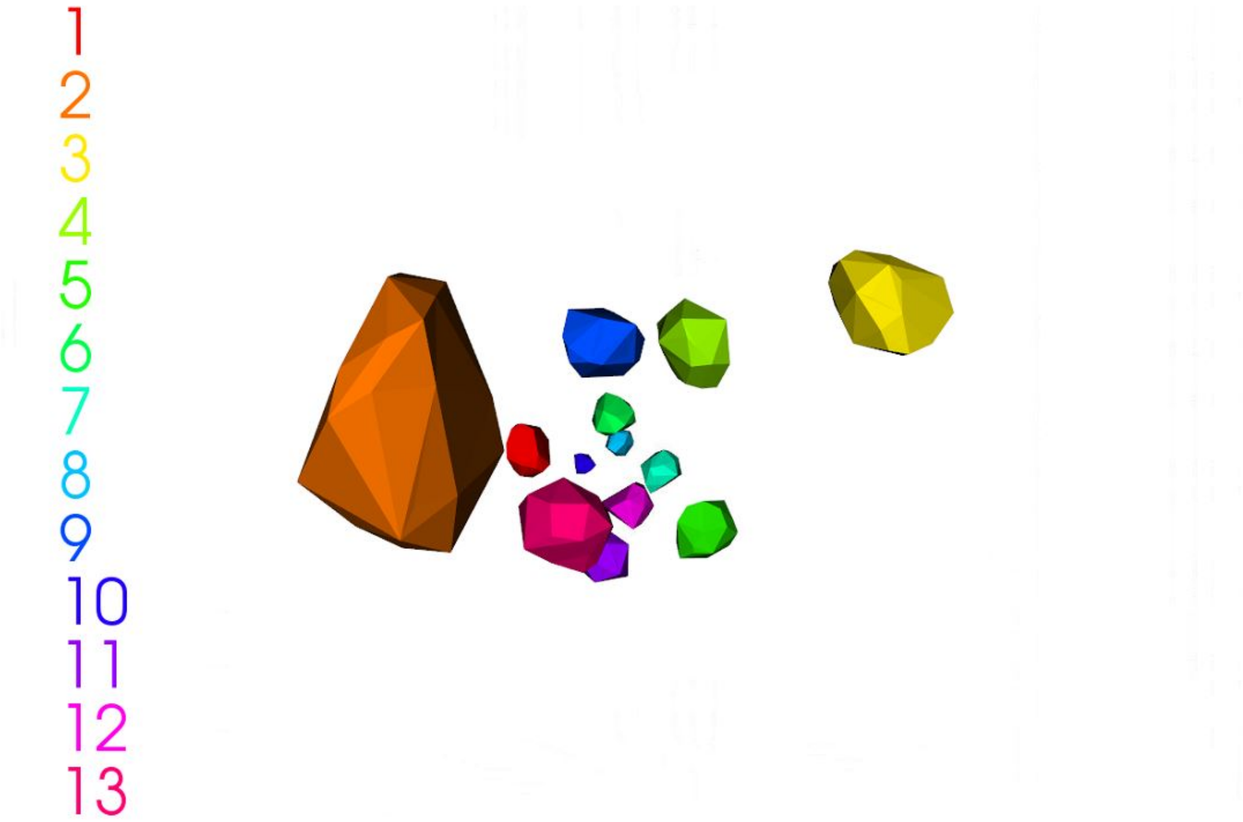


Figure 5: One rotation of the squamate mitochondrial gene trees projected into 3D using Normalized stress. Each set of gene trees are represented as a convex hull.