Ordination 2

Lecture 10.3 PCoA

Lauren Sullivan

Module: Multivariate Models

Readings

Required for class:

- NA

Optional:

- Legendre, P. (2018) 1.3. Principle Coordinate Analysis

Multivariate Analysis

There are several ways to look at multivariate patterns from a matrix of $\mathbf{Y}$'s.

1. Linear models: MANOVA/regression to test patterns
2. **Ordination: PCA, nMDS, etc to visualize patterns**
3. Permutation tests: PERMANOVA to test patterns

Principal Coordinate Analysis (PCoA)

PCA works well for data where it's appropriate to maintain Euclidean distance among objects.

- However, what do you do when have data that requires a different distance matrix, or where PCA is not a good model (e.g.: many 0's, presence/absence data)?

Principle coordinate analysis decomposes distance matrices (or dissimilarity matrices) such that distance among objects is preserved for **any** distance measure.

- Sometimes PCoA is called metric multidimensional scaling (MDS) because it preserves relationships among objects.
- This is opposed to nMDS that does not preserve distance, but attempts to preserve *relationships* among objects.

Understanding PCoA

Step 1: Start with a distance matrix ($\mathbf{D}$).

Step 2: Transform elements in $\mathbf{D}$ to: $\mathbf{A} = -\frac{1}{2}\mathbf{D}^2$

Step 3: Double-center the matrix.

- Subtract row and column means from each element, and add grand mean. This positions the origin at the centroid of the scatter.

Step 4: Eigen-decomposition of double-centered matrix.

- Eigen vectors are *coordinates* for the ordination plot. They do not describe aspects of the Y variables, only distances among objects/sites/samples (similar to nMDS).

Sparrow Data

Let's look at this sparrow data again.

PCoA

Let's look again at our $\mathbf{Y}$ from the sparrow data but with a Manhattan distance matrix ("city block" distance).

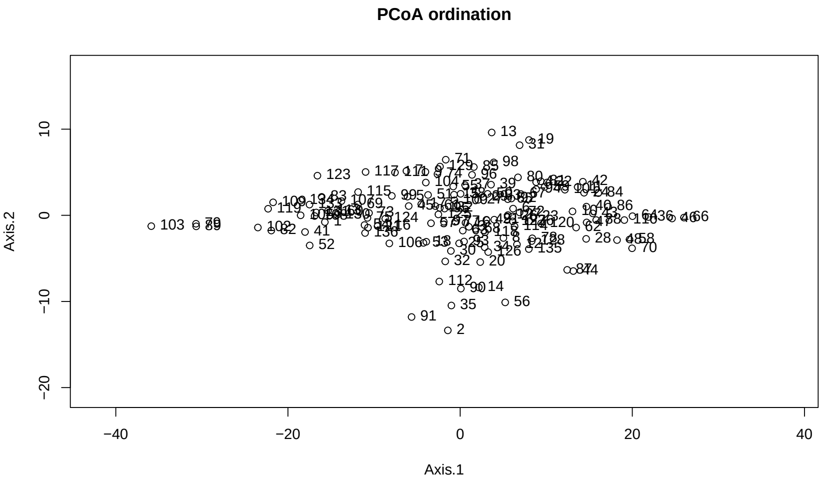
- Use the `pcoa()` function in the `ape` library.

```
sparrow.d <- vegdist(sparrow[, -c(1:3)], "manhattan")
sparrow.pcoa <- pcoa(sparrow.d)
sparrow.pcoa$values[1:10,1:5]
```

```
##    Eigenvalues Relative_eig Rel_corr_eig Broken_stick Cum_corr_eig
## 1   19003.4633   0.78981648  0.120018387   0.04088643    0.1200184
## 2    1940.1974   0.08063793  0.017947368   0.03342375    0.1379658
## 3    1552.0694   0.06450666  0.015625619   0.02969240    0.1535914
## 4    1055.5012   0.04386844  0.012655190   0.02720484    0.1662466
## 5     863.9511   0.03590729  0.011509353   0.02533917    0.1777559
## 6     742.0758   0.03084194  0.010780305   0.02384663    0.1885362
## 7     643.8522   0.02675960  0.010192740   0.02260285    0.1987290
## 8     420.1261   0.01746116  0.008854429   0.02153675    0.2075834
## 9     388.4098   0.01614298  0.008664705   0.02060392    0.2162481
## 10    367.7006   0.01528227  0.008540825   0.01977473    0.2247889
```

Plotting

```r
biplot(sparrow.pcoa, plot.axes = c(1,2))
```



Note: Apparently you can plot the data on these ordination plots similar to nMDS with an env.fit, but I couldn't quite figure out how to get it to work.