

DETRENDED CORRESPONDENCE ANALYSIS (DCA)

Steven M. Holland

Department of Geology, University of Georgia, Athens, GA 30602-2501



May 2008

Introduction

Correspondence analysis (CA) was first introduced to ecology in the early 1970s as reciprocal averaging (RA) and quickly gained in popularity because of its better recovery of one-dimensional simulated gradients, compared to principal components analysis. Correspondence analysis also assumed modal relationships of taxa relative to ecological gradients, as opposed to the linear relationship underlying principal components analysis, and was a more attractive method on these theoretical grounds.

Detrended correspondence analysis (DCA) was developed to overcome the distortions inherent to correspondence analysis ordination, in particular the tendency for one-dimensional gradients to be distorted into an arch on the second ordination axis and for the tendency for samples to be unevenly spaced along the axis 1. DCA overcomes these problems by flattening this arch and rescaling the positions of samples along an axis. In general, DCA ordinations perform better with simulated data than do CA/RA ordinations.

Computation

Correspondence analysis / reciprocal averaging can be calculated with an eigenanalysis approach and with a reciprocal averaging approach. The latter is more intuitive, so the reciprocal averaging approach will be presented here. Beginning as usual with a matrix of n rows of samples and p columns of taxa, each sample is assigned an arbitrarily chosen score (x_i). Scores for each taxon (y_j) are then calculated as a weighted average, where the abundance of a taxon (a_{ij}) is multiplied by the sample score, and these are summed across all samples and divided by

$$y_j = \frac{\sum_{i=1}^n a_{ij} x_i}{a_{+j}}$$

the total abundance for that taxon. These taxon scores are then used to calculate a new set of sample scores following the same procedure in which the abundance of a taxon is multiplied by the taxon score and these are summed across all taxa and divided by the total abundance

$$x_i = \frac{\sum_{j=1}^p a_{ij} y_j}{a_{i+}}$$

within each sample. Sample scores are then centered and standardized such that their mean

$$\sum_{i=1}^n a_i x_i = 0 \text{ and } \sum_{i=1}^n a_i x_i^2 = 1$$

is zero and their variance is one. This procedure of alternately calculating sample and taxon scores is repeated until the scores stabilize. This initial pass produces the CA axis 1 scores for both samples and taxa. The same procedure can be performed to produce higher-order axes that are orthogonal with lower-order axes.

Detrended correspondence analysis begins with a correspondence analysis, but follows it with steps to detrend (hence its name) and rescale axes. Several methods have been used to detrend an axis. In the simplest approach, the axis is first divided into an arbitrary number of equal-length segments. Within each segment, the scores on the next higher-order axis are recentered such that their mean is zero. In effect, if there is arch present, it is flattened onto the lower-order axis. The detrending process is sensitive to the number of segments and the default value is 26, which has empirically produced acceptable results. Others have proposed using polynomial regressions for rescaling and others have used a sliding moving average window, which is what the algorithm in R does. The rescaling of an axis is accomplished by equalizing the weighted variance of taxon scores along the axis segments.

Considerations

Correspondence analysis suffers from two well-known distortions that limit its usefulness for ecological ordinations. First, a one-dimensional ecological gradient is commonly bent into an arch in axis 1 - axis 2 space. In general, higher-order axes contain quadratic distortions of lower-order axes, which hampers their interpretation. Second, scores are typically compressed near the ends along axis 1 relative to their original spacing along the ecological gradient, limiting the ability to use axis 1 scores of a linear proxy of position along an environmental gradient. In addition to these distortions, correspondence analysis has an underlying chi-square distance measure that can overemphasize the importance of rare taxa. In large samples that contain multiple rare taxa, perhaps solely because of the large size of the sample, the chi-square distance can exaggerate how distinctive the sample is.

The detrending and rescaling algorithms in detrended correspondence analysis do not fully eliminate distortions of the underlying data structure and can introduce new distortions in some cases. One common distortion of a two-dimensional gradient can be envisioned by taking a piece of paper and twisting it, such that axis 1 is preserved reasonably undistorted, but the second axis of variation is expressed on DCA axis 2 at one end and on DCA axis 3 at the opposite end. This produces the well-known DCA wedge, consisting of a tapering of sample points in axis 1-2 space and an opposing wedge in axis 1-3 space.

Several ecological studies have simulated two-dimensional gradients and subjected them to a variety of ordination methods to compare their results. Principal components analysis and correspondence analysis typically display severe distortions of a horseshoe and an arch, respectively, so they should be used in ecological data only with considerable caution. De-

trended correspondence analysis and multidimensional scaling commonly perform much better and display minimal distortions in most cases. Because DCA can produce stronger distortions than MDS in some of the simulations, ecologists have embraced MDS as the method of choice. However, similar explorations of such simulations have also revealed cases of strong distortion in MDS. Likewise, some authors have reported that DCA ordinations are more interpretable than those from MDS. Both of these observations suggest that the stronger condemnations of DCA that have appeared in the ecological literature may be overstated.

CA and DCA in R

Correspondence analysis and detrended correspondence analysis can each be run from the `decorana()` command in the `vegan` package.

1) Download and install the `vegan` library, necessary for running the `decorana()` command and the data transformations.

```
> library(vegan)      # necessary for decorana()
```

2) Perform a correspondence analysis / reciprocal averaging on a data set

```
> mydata <- read.table(file="frankfort.txt", header=TRUE,
  row.names=1, sep=",")
```

```
> mydata.ra <- decorana(mydata, ira=1)
# ira=1 specifies RA
```

3) Perform a detrended correspondence analysis on a data set, preceded by transformations to correct for differences in sample size and differences in abundance among taxa.

```
> mydata.t1 <- decostand(mydata, "total")
# percent transformation on samples
# "total" method defaults to rows (samples)
```

```
> mydata.t2 <- decostand(mydata.t1, "max")
# percent of maximum transformation on taxa
# "max" method defaults to columns (taxa)
```

```
> mydata.t2.dca <- decorana(mydata.t2)
# default ira=0 specifies DCA
```

```
> mydata.t2.dca.DW <- decorana(mydata.t2, iweigh=1)
# iweigh=1 turns on down-weighting of rare taxa,
# dampening their
# effects on the ordination.
```

4) View items in the list produced by `decorana`. The most useful elements will be `rproj` and `cproj`.

```
> names(mydata.t2.dca)
```

```

# mydata.t2.dca$rproj: sample scores for the first 4 axes
# mydata.t2.dca$cproj: taxon scores for the first 4 axes
# mydata.t2.dca$evals: eigenvalues for the first 4 axes
# mydata.t2.dca$evals.decorana: legacy eigenvalues -
# do not use
# mydata.t2.dca$origin: origin in DCA; used for plotting
# mydata.t2.dca$v: unknown
# mydata.t2.dca$fraction: abundance fraction where
# downweighting begins
# mydata.t2.dca$adotj: sum of abundances of taxa
# mydata.t2.dca$aidot: sum of abundances within samples
# mydata.t2.dca$iweigh: whether down-weighting of rare
# taxa was turned on, 0 for no, 1 for yes
# mydata.t2.dca$iresc: number of rescaling cycles
# mydata.t2.dca$ira: method invoked - 0 for DCA (default)
# 1 for RA/CA
# mydata.t2.dca$mk: number of segments in rescaling
# mydata.t2.dca$short: length of shortest gradient to be
# rescaled
# mydata.t2.dca$before: values before Hill's piecewise
# transformation, if used;
# by default, transformation is not used.
# mydata.t2.dca$after: values after Hills's piecewise
# transformation
# mydata.t2.dca$call: function call

```

5) View summary of dca results, including function call, eigenvalues, taxon scores, and sample scores.

```
> summary(mydata.t2.dca)
```

6) Extract scores from decorana output.

```

> mydata.t2.dca.taxonscores <- scores(mydata.t2.dca,
  display=c("species"), choices=c(1,2))
# extracts axis 1 & 2 scores for taxa

> mydata.t2.dca.taxonscores <- mydata.t2.dca$cproj[,1:2]
# alternative approach

> mydata.t2.dca.samplescores <- scores(mydata.t2.dca,
  display=c("sites"), choices=1)
# extracts only axis 1 scores for samples

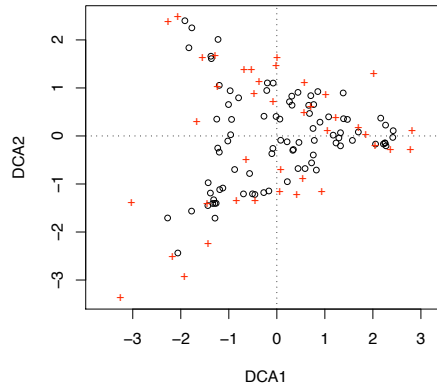
> mydata.t2.dca.taxonscores <- mydata.t2.dca$rproj[,1]
# alternative approach

```

7) Plot scores from decorana.

```
> plot(mydata.t2.dca)
```

```
# Samples are shown as black open circles, taxa are
# displayed with red crosses
```

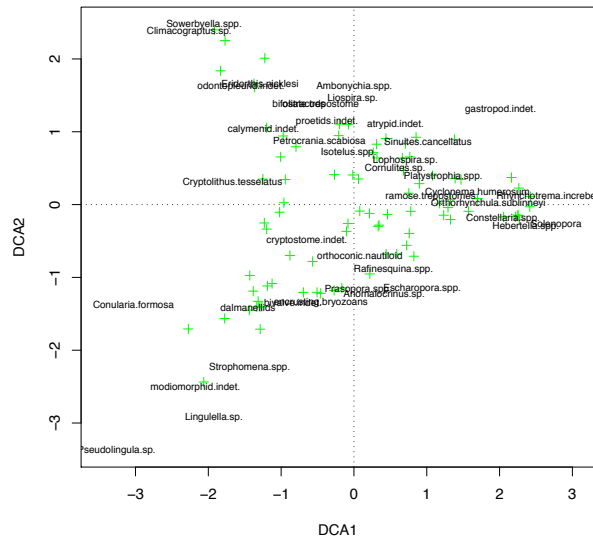


8) Plot scores, but with customization of symbols and labels

```
> plot(mydata.t2.dca, display=c("none"), cols=c(1,2))
# plot axis 1 & 2, but display no points or labels

> points(mydata.t2.dca, display=c("sites"), choices=1:2,
        pch=3, col="green")
# plot samples as green crosses for axis 1 and 2

> text(mydata.t2.dca, display=c("species"), choices=1:2,
       cex=0.7)
# plot labels for taxa for axis 1 and 2, using cex to shrink
# size of labels. Larger plots may also be used to alleviate
# congestion of labels.
```



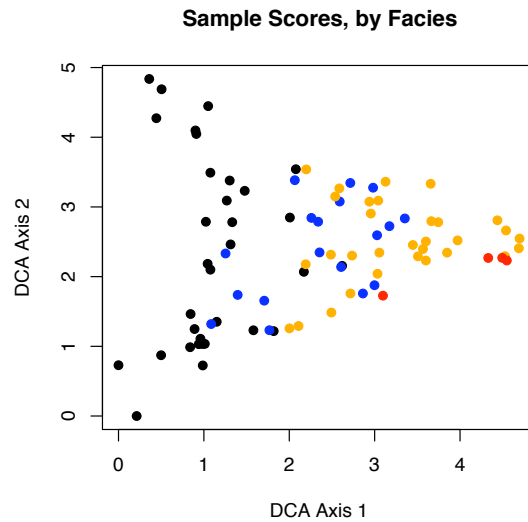
9) Plot sample scores, color-coded by an external variable, such as lithofacies.

```
> myFacies <- read.table(file="frankfacies.txt", header=TRUE,
  row.names=1, sep=",")
# Read in external file with numeric code for each facies.
# The first column of this file has the same locality codes,
# in the same order as the original data file.

> t2.dca.scores <- mydata.t2.dca$rproj[ ,1:2]
# Extract axis 1 & 2 sample scores

> t2.dca.scoresFacies <- data.frame(t2.dca1=t2.dca.scores[,1],
  t2.dca2=t2.dca.scores[,2], facies=myFacies)
> attach(t2.dca.scoresFacies)
# Merge facies codes and samples scores into single data frame

> plot(t2.dca1, t2.dca2, type="n", xlab="DCA Axis 1",
  ylab="DCA Axis 2", main="Sample Scores, by Facies")
> points(t2.dca1[facies=="OF"], t2.dca2[facies=="OF"],
  col="black", pch=16)
> points(t2.dca1[facies=="DS"], t2.dca2[facies=="DS"],
  col="blue", pch=16)
> points(t2.dca1[facies=="SS"], t2.dca2[facies=="SS"],
  col="orange", pch=16)
> points(t2.dca1[facies=="SH"], t2.dca2[facies=="SH"],
  col="red", pch=16)
```

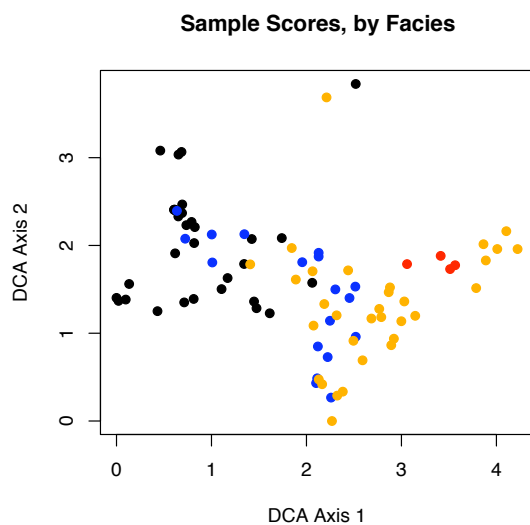


10) Visualize the effects of not running the second transformation, which equalized differences among abundance of taxa.

```

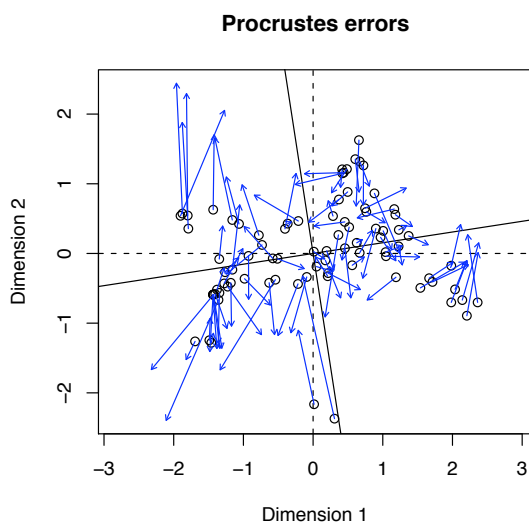
> mydata.t1.dca <- decorana(mydata.t1)
> t1.dca.scores <- mydata.t1.dca$rproj[ ,1:2]
> t1.dca.scoresFacies <- data.frame(t1.dca1=t1.dca.scores[,1],
  t1.dca2=t1.dca.scores[,2], facies=myFacies)
> attach(t1.dca.scoresFacies)
> plot(t1.dca1, t1.dca2, type="n", xlab="DCA Axis 1",
  ylab="DCA Axis 2", main="Sample Scores, by Facies")
> points(t1.dca1[facies=="OF"], t1.dca2[facies=="OF"],
  col="black", pch=16)
> points(t1.dca1[facies=="DS"], t1.dca2[facies=="DS"],
  col="blue", pch=16)
> points(t1.dca1[facies=="SS"], t1.dca2[facies=="SS"],
  col="orange", pch=16)
> points(t1.dca1[facies=="SH"], t1.dca2[facies=="SH"],
  col="red", pch=16)

```

11) Compare results of two ordinations quantitatively, using `procrustes()`. This example will compare DCA with and without the second data transformation. The `procrustes()` function rotates, inverts, translates, and rescales two ordinations to produce the best fit in the positions of objects.

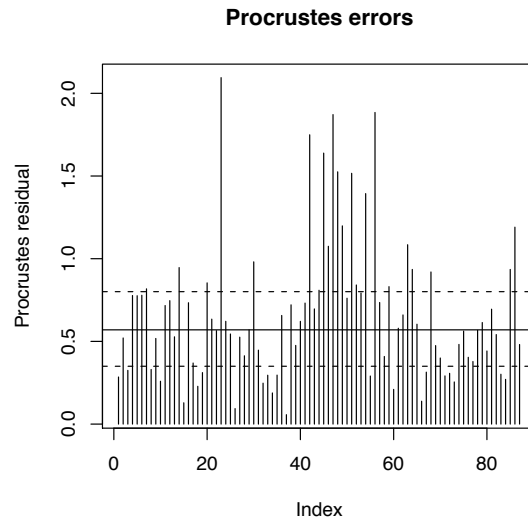
```
> comparison <- procrustes(t2.dca.scores, t1.dca.scores)
> plot(comparison)
# displays plot showing first ordination (tips of blue
# arrow) relative to second ordination (open black circles)
```



```

> plot(comparison, kind=2)
# displays differences in ordination position for each
# sample. Note that some samples (e.g., #23) have greatly
# different ordination positions, indicated by the large
# Procrustes errors

```



References

- Hill, M.O., and H.G. Gauch, Jr., 1980. Detrended correspondence analysis: an improved ordination technique. *Vegetatio* 42: 47-58.
- Jackson, D.A., and K.M. Somers, 1991. Putting things in order: the ups and downs of detrended correspondence analysis. *American Naturalist* 137: 704-712.
- Kenkel, N.C., and L. Orlóci, 1986. Applying metric and nonmetric multidimensional scaling to ecological studies: some new results. *Ecology* 67: 919-928.
- Legendre, P., and L. Legendre, 1998. *Numerical Ecology*. Elsevier: Amsterdam, 853 p.
- McCune, B., and J.B. Grace, 2002. *Analysis of Ecological Communities*. MjM Software Design: Gleneden Beach, Oregon, 300 p.
- Minchin, P.R., 1987. An evaluation of the relative robustness of techniques for ecological ordination. *Vegetatio* 69: 89-107.
- Wartenberg, D., S. Ferson, F.J. Rohlf, 1987. Putting things in order: a critique of detrended correspondence analysis. *American Naturalist* 129: 434-448.