Algorithmic Approaches to Aid Species’ Delimitation in

Multidimensional Morphospace: Online Appendix 1

Thomas H.G. Ezard, Paul N. Pearson and Andy Purvis

This appendix contains:

1. a comparison of different threshold-stopping criteria for use in dimension reduction;
2. a simulation study to illustrate the importance of dimension reduction in cases such as this;
3. self-contained R code to follow the framework as we implemented it here;
4. References.

1 Comparing Different Threshold Criteria

The problem of where the threshold between useful information and irrelevant noise in principal components has attracted extended attention, with rules being proposed and amended for well over half a century [1-6]. Retaining too few axes risks neglecting an influential one whereas retaining too many factors can deliver attention to relatively unimportant components: either can generate bias, although the former is more serious [1]. Each retained variable should provide a significant improvement in explanatory power, whilst each discarded variable should only cause incremental improvement. The performance of multiple methods has been rigorously investigated elsewhere [2, 3], concluding that no
single measure consistently outperforms all others, but that consistency among
measures is greatest when variables are highly correlated [3].

In our samples, the tendency for Kaiser-Guttman to retain more
components than other criteria is borne out, while the similarity in performance
of the broken-stick and random average under parallel analysis is also expected
when variables are highly correlated [3]. Correcting parallel analysis using
random averages alters the recommended number of components in this
instance because a number of the traits employed are highly correlated [3].

<table>
<thead>
<tr>
<th>Method</th>
<th>Middle Eocene</th>
<th>Upper Eocene</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kaiser-Guttman [4, 5]</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>Broken Stick [6]</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>Parallel Analysis [7]</td>
<td>5</td>
<td>3</td>
</tr>
<tr>
<td>Random Average under Parallel Analysis [3, 7]</td>
<td>2</td>
<td>2</td>
</tr>
</tbody>
</table>

2 The Importance of Dimension Reduction

We use simulation studies to illustrate the importance of dimension-
reduction in situations such as the one presented here, and follow the
suggestion made in [3] that simulation studies should approximate the
situation at hand. That is one of numerous weakly correlated traits, but also several highly correlated ones. We restrict ourselves to one example of these situations, using three traits for ease of exposition.

We simulate three traits for two groups of individuals using gamma distributions (a typical example is given overleaf). Trait 3 is however merely the first trait with noise sampled from a uniform distribution, i.e. these two traits are highly correlated with one another:

<table>
<thead>
<tr>
<th>Trait 1</th>
<th>Trait 2</th>
<th>Trait 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.466</td>
<td>0.849</td>
<td>0.396</td>
</tr>
</tbody>
</table>

We deliberately choose overlapping distributions to illustrate the power of the model-based clustering approach [8, 9] to delimit groups; the second cluster is restricted to lower values of trait 1 but can still extend throughout the approximate range of trait 2. A randomly selected simulation from the 1000 replicates used is:
From 1000 replicates of the simulation, the median number of clusters when no dimension reduction was implemented was 3 (mean 3.061), whereas the median with dimension reduction was 2 (mean 2.414). In every case, there was support for dimension reduction using random average with parallel analysis [7] [3] and broken stick [6] methods: 1 component was retained in around 10% of cases with 2 components being retained in the remainder.

The Gamma distribution was chosen as it can be tweaked to approximate normal or skewed distributions by changing the size and/or shape parameters. The reason for the large skew in these distributions is that parameter values were chosen to approximate sections of populations diverging away from the majority (see Figure 1 in the main manuscript) and therefore extreme or unusual individuals were more likely. In the overwhelming majority of cases (>90%), the additional clusters have very low numbers of individuals.

An advantage of robust approaches is that they de-emphasise extreme values: medians and median absolute deviations are less affected by long-tailed or asymmetric distributions than means and standard deviations [10], hence obscuring potentially critical differences and restricting the ability to delimit species. The notion of being ‘well-separated’ applies to distinct clusters as well as extreme data points, meaning that identification of genuine outliers can be problematic. In these simulations, these individuals around the margins of the distributions may be genuine, statistical outliers or may reflect under-sampled or hidden groups.
3 Self-Contained R Code

# you may need to install these packages first
library(pcaPP)# [11]
library(mclust)# [8]
library(mvoutlier)# [11, 12]
library(vegan)

# generate some random data
t1 <- rgamma(150, 1.5, .1)
t2 <- rnorm(150, 10, 2)
tt <- matrix(c(t1, t2), ncol=2, byrow=FALSE)

# fit the robust PCA using the GRID algorithm
m1 <- PCAgrid(tt, k=11, scale=mad, center=median, method="qn")
nk <- max(which(m1$sdev^2 > bstick(m1)))
nk  # this is the number of components to retain using broken stick

pp <- predict(m1)[,1:nk]
  # take the PCA predictions, i.e. rotated values, and cluster
m2 <- Mclust(pp)
m2  # this is the modelled structure

grps <- as.numeric(m2$classification)
out01 <- sign2(as.data.frame(pp), qcrit=.975)
out <- out01$wfinal01
  # this line creates a column of 0s and 1s whether an individual
  # ..is a significant outlier at the 5% level of significance.

plot(tt, col=grps, pch=grps)
  # plot, with colour determined by model classification
4 References


The parallel analysis and random averages under parallel analysis criteria were implemented by adapting the paran package by Alexis Dinno. See doyenne.com/Software/index.shtml#paran and http://cran.r-project.org/web/packages/paran/index.html.

We do this by altering line 14 of the paran function to read:

mpp <- PCAgrid(xx, scale=mad, center=median
eigenvalues <- mpp$sdev^2

The previous line read:
eigen(cor(x), only.values = TRUE, EISPACK = FALSE)[[1]]