Introduction to R for the Geoscience: Stratigraphic & Palaeo Data

Gavin Simpson

February , 2017

1 Rates of Change

In this part of the practical you will compute some rate of change estimates for a short core sequence from the Round Loch of Glenhead covering the last 140 years. There aren't any canned functions for computing rates of change in R so you'll perform all the neccessary steps yourself — don't worry, they aren't too onerous and you'll learn more about R coding as you go along.

Start by loading the rioja package and the dataset needed

```
> library("rioja")                          # install.packages("rioja")
> ## load data and extract
> data(RLGH)
> spp <- RLGH$spec
> age <- RLGH$depth$Age
```

1.1 Ordination-based rates of change

The first method you'll use is the ordination based rate of change, for which we need to smooth and interpolate the species data to a common time interval. This can be done using the `interp.dataset()` function

```
> ## interpolate new dataset to every 5 years
> ## using a smoothing splint
> x.new <- seq(0, max(age), by=5)
> sp.interp <- interp.dataset(y=spp, x=age, xout=x.new,
+                            method = "sspline")
> rownames(sp.interp) <- x.new
```

Next you need to ordinate the data. Here we use DCA to follow the published version, but you could try other ordination techniques should you wish.

```
> ## ordinate
> library("vegan")
> ord <- decorana(sp.interp)
> ord

Call:
decorana(veg = sp.interp)

Detrended correspondence analysis with 26 segments.
Rescaling of axes with 4 iterations.

                  DCA1    DCA2     DCA3     DCA4
Eigenvalues     0.1312 0.02112 0.007314 0.007082
Decorana values 0.1316 0.01130 0.005090 0.003085
Axis lengths    1.1056 0.53068 0.427959 0.374999
```

```
> ## plot of points showing ages
> plot(ord, display = "sites", type = "text")
```

How many axes do you want to retain?

Finally, to compute the rate of change we need to extract the axis scores we need and compute the distance between observations in ordination space. This has to be done by hand for the most part.

```
> ## get scores
> scrs <- scores(ord, display = "sites", choices = 1:2)
> roc <- as.matrix(dist(scrs))
> roc <- roc[row(roc) == col(roc) + 1] ## extract off-diagonal
> roc
```

```
 [1] 0.145791683 0.038388466 0.056476075 0.096765764 0.141675748 0.089316724
 [7] 0.148886370 0.065164981 0.123568620 0.180866597 0.135311608 0.072936429
[13] 0.009012417 0.054150418 0.058022625 0.065137156 0.081900194 0.093647074
[19] 0.102022525 0.106469426 0.102384652 0.095557955 0.078070381 0.054041243
[25] 0.024817872 0.004879166 0.010482557 0.017234419
```

Plot the rates of change for the core sequence

```
> ## plot roc against age
> plot(roc ~ head(x.new, -1), type = "l", ylab = "Rate of Change", xlab = "Age")
> points(roc ~ head(x.new, -1), type = "h")
```

Identify the periods with greatest rates of change. You'll compare these with those identified using the dissimilarity-based approach in the next section.

1.2 Dissimilarity-based rates of change

Now you'll calculate rates of change for the same samples using the raw dissimilarity method. In this method the data don't need to be interpolate, instead we work with the actual species data themselves. Start by computing the squared chord distances between all samples and extract the off-diagonal elements (the pairwise dissimilarities)

```
> roc2 <- as.matrix(paldist(spp/100, dist.method="sq.chord"))
> roc2 <- roc2[row(roc2) == col(roc2) + 1] ## extract off-diagonal
```

Next scale these by the number of years between each sample. We can compute the sample intervals in years using the `diff()` function

```
> intervals <- diff(age)
> roc2 <- roc2 / intervals
```

Plot the rates of change for the core sequence

```
> ## plot roc2 against age
> plot(roc2 ~ head(age, -1), type = "l", ylab = "Rate of Change", xlab = "Age")
> points(roc2 ~ head(age, -1), type = "h")
```

Identify the periods with greatest rates of change. How do these compare with the rates extract using the ordination-based approach? Can you think why there might be differences? Which method would you trust most? Why?

To aid you in your comparison, you can plot both rates of change on the same device

```
> ## compare plots
> layout(matrix(1:2, nrow = 2))
> ## plot roc against new age
> xlim <- range(age, x.new)
> plot(roc ~ head(x.new, -1), type = "l", ylab = "Rate of Change", xlab = "Age",
+      main = "Ordination-based", xlim = xlim)
```

```
> points(roc ~ head(x.new, -1), type = "h")
> ## plot roc2 against age
> plot(roc2 ~ head(age, -1), type = "l", ylab = "Rate of Change", xlab = "Age",
+      main = "Dissimilarity-based", xlim = xlim)
> points(roc2 ~ head(age, -1), type = "h")
> layout(1)
```

2 Chronological clustering

Now we'll look at a coupld of chronological clustering methods that are available in R.

2.1 CONISS

First we'll apply CONISS to the Abernethy Forest pollen record. We load the data set, convert the observations to proportions (from percentages) and apply a square root transformation. Ths means that when we compute the euclidean distances between samples using `dist()` we are computing hellinger distances instead. Finally we use the `chclust()` function to perform the CONISS clustering

```
> library("analogue")
> data(abernethy)
> aber <- abernethy[, seq_len(ncol(abernethy) - 2L)]
> diss <- dist(tran(aber, method = "pcent2prop"))
> clust <- chclust(diss)
```

We can plot the CONISS results and the variances "explained" by varying the number of zones.

```
> layout(matrix(1:2, ncol = 2))
> op <- par(mar = c(5,4,1,2) + 0.1)
> plot(clust, labels = abernethy$Age, hang=-1, horiz=TRUE, x.rev = TRUE, cex = 0.5)
> bstick(clust, 10)
> par(op)
> layout(1)
```

Using the screeplot method or Bennett's broken stick approach, suggest how many zones you would consider splitting the Abernethy Forest pollen record into?

2.2 Binary splitting

Unfortunately, the **mvpart** package that I used previously to do binary or recursive partitioning has been removed from CRAN as it contains problems that were never addressed by its author/maintainer. A new way of fitting almost the same model has become available in the **partykit** package. Rather than use cross-validation and cost-complexity pruning to choose the number of binary splits, it uses conditional inference trees. I'm not exactly sure how to describe this, but it is essentially using statistical methods to decide if it should make a new split at all. This is in contrast to standard tree models where they are grown very large and we use cross validation to prune the branches back.

In this section we'll take a quick look at doing clustering via binary splitting using the **partykit** package. First we load the package (you'll need to install it first) and then we remove some of the rare species (you should just do this casually — I'm doing it here to make fitting the tree easier). We also need to clean up (remove spaces and hyphens) the taxon names so we can create the formula needed to fit the model

```
> library("partykit")                    # install.packages("partykit")
> aber2 <- chooseTaxa(aber, n.occ = 2, max.abun = 5)
> names(aber2) <- gsub(" ", "", names(aber2))
> names(aber2) <- gsub("-", "", names(aber2))
```

Now we can create the formula which describes the conditional tree we want to fit. All the species we didn't remove will end up on the left hand side of the formula, and `Age` will be the sole predictor. We add on the `Age` variable to the reduced set of species so all the variables are in one object.

```
> frm <- formula(paste0(paste(names(aber2), collapse = " + "), " ~ Age"))
> aber2 <- cbind(aber2, Age = abernethy$Age)
```

Now we can fit the conditional inference tree and plot the results

```
> sptree <- ctree(frm, data = aber2, teststat = "max", minsplit = 2)
> plot(sptree, terminal_panel = node_barplot, tp_args = list(rot = 45, just = c("right")))
```

The leaves of the tree are illustrated by barplots showing the abundances of each species. These barplots give you some idea of the species composition in each zone, but the code isn't flexible enough yet to allow you to mke nicer plots or show taxon names clearly (or I haven't worked out how to do it!) Note that here we just used the raw percentage abundances to fit the tree, you might get a different tree if you transformed or standardised the species data.

How many zones are suggested by the tree? Do these zones match up with the ones CONISS identified?

3 Principal Curves

In this section of the practical you'll explore fitting a principal curve to the Abernethy Forest pollen data set. The functions you need are available in the analogue package. Load it, and ignore the warnings for now.

```
> library("analogue")
> data(abernethy)
```

A simple summary of the data can be produced using the `Stratiplot()` function

```
> (plt <- Stratiplot(Age ~ . - Depth, data =
+                    chooseTaxa(abernethy, max.abun = 15, n.occ = 10),
+                    type = c("h","g","l"), sort = "wa"))
```

Next, remove two variables from the data set (the samples depths and ages) and then proceed to fit a PCA and a CA, as well as the principal curve to the data

```
> abernethy2 <- abernethy[, -(37:38)]
> aber.pca <- rda(abernethy2)
> aber.ca <- cca(abernethy2)
> aber.pc2 <- prcurve(abernethy2, method = "ca", trace = TRUE, plotit = TRUE,
+                     vary = TRUE, penalty = 1.4)

   Determining initial DFs for each variable...

  |
  |                                                                      |   0%
  |
  |==                                                                    |   3%
  |
  |====                                                                  |   6%
  |
  |======                                                                |   8%
  |
  |========                                                              |  11%
  |
  |==========                                                            |  14%
  |
```

```
|============                                                     |  17%
|
|==============                                                   |  19%
|
|================                                                 |  22%
|
|==================                                               |  25%
|
|====================                                             |  28%
|
|======================                                           |  31%
|
|========================                                         |  33%
|
|==========================                                       |  36%
|
|============================                                     |  39%
|
|==============================                                   |  42%
|
|================================                                 |  44%
|
|==================================                               |  47%
|
|====================================                             |  50%
|
|======================================                           |  53%
|
|========================================                         |  56%
|
|==========================================                       |  58%
|
|============================================                     |  61%
|
|==============================================                   |  64%
|
|================================================                 |  67%
|
|==================================================               |  69%
|
|====================================================             |  72%
|
|======================================================           |  75%
|
|========================================================         |  78%
|
|==========================================================       |  81%
|
|============================================================     |  83%
|
|==============================================================   |  86%
|
|================================================================ |  89%
|
|==================================================================|  92%
|
```

```
  |======================================================================    |  94%
  |
  |======================================================================  |  97%
  |
  |======================================================================| 100%


Fitting Principal Curve:

Initial curve: d.sq: 103233.450
Iteration   1: d.sq: 4283.431
Iteration   2: d.sq: 4312.298
Iteration   3: d.sq: 4340.691
Iteration   4: d.sq: 4355.388
Iteration   5: d.sq: 4366.497
Iteration   6: d.sq: 4369.944

PC Converged in 6 iterations.
```

The arguments to `prcurve()` tell it to start from a CA solution, show it working by printing to the console and plotting the curve as it converges, taxa are allowed differeing complexity smoothers and the penalty per degree of freedom is increased by 40% to help avoid over-fitting.

Look at the fitted object. How much of the variance in the dataset is explained by the principal curve?

```
> aber.pc2

        Principal Curve Fitting

Call: prcurve(X = abernethy2, method = "ca", vary = TRUE, trace = TRUE,
plotit = TRUE, penalty = 1.4)

Algorithm converged after 6 iterations

           SumSq Proportion
Total     103234      1.000
Explained  98864      0.958
Residual    4370      0.042

Fitted curve uses 218.3391 degrees of freedom.
```

Compare that with the variance explained by the PCA and CA

```
> varExpl(aber.pca)

      PC1
0.4649883

> varExpl(aber.ca)

      CA1
0.3098955
```

How well has the principal curve done compared to the more traditional techniques?

Plot the principal curve

```
> ## Plot the fitted curve
> plot(aber.pc2)
```

Can you see why the principal curve performs much better than the other two ordination techniques?

A longer chunk of code is required to draw rates of change and compare the various methods

```
> Depth <- abernethy$Depth
> Age <- abernethy$Age
> layout(matrix(c(1,2,2), ncol = 3))
> nseg <- nrow(abernethy2) - 1
> RoC <- unclass(diff(gradientDist(aber.pc2))) / diff(Age)
> plot(y = Age[-length(Age)],
+      x = RoC * 1000, type = "n",
+      ylim = rev(range(Age)),
+      ylab = "Age (Radiocarbon years BP)",
+      xlab  = expression("Rate of Change" ~ (kyr^{-1})),
+      main = "a", cex.main = 1.5)
> segments(x0 = rep(0, nseg), y0 = Age[-length(Age)],
+          x1 = RoC * 1000, y1 = Age[-length(Age)])
> plot(gradientDist(aber.pc2), orderBy = Age,
+      xlim = rev(range(Age)),
+      type = "o", flipAxes = TRUE, xlab = "Age (Radiocarbon years BP)",
+      main = "b",
+      cex = 0.8, pch = 21, col = "black", bg = "black", cex.main = 1.5)
> lines(gradientDist(aber.pca), orderBy = Age,
+       lty = "dashed", flipAxes = TRUE, type = "o",
+       cex = 0.8, pch = 22, col = "black", bg = "black")
> lines(1 - gradientDist(aber.ca), orderBy = Age,
+       lty = "dotdash", flipAxes = TRUE, type = "o",
+       cex = 0.8, pch = 23, col = "black", bg = "black")
> legend("topright", bty = "n", pch = 21:23,
+        legend = c(expression(PCurve), expression(PCA[1]), expression(CA[1])),
+        lty = c("solid","dashed","dotdash"),
+        col = c("black","black","black"),
+        pt.bg = c("black","black","black"), inset = 0.01, cex = 1.4,
+        seg.len = 4)
> layout(1)
```

The individual curves fitted to each taxon can be visualised, again with a long chunk of code. Here, only the most abundant taxa are shown

```
> taxaWant <- chooseTaxa(aber, max.abun = 25, n.occ = 10, value = FALSE)
> layout(matrix(seq_len(sum(taxaWant)), ncol = 3))
> op <- par(mar = c(4,4,3,1) + 0.1)
> plot(sppResponse(aber.pc2), which = taxaWant)
> par(op)
> layout(1)
```

Using the two plots, do you feel the principal curve has done a good job of describing floristic change in the Abernethy core?