# Analogue Methods in Palaeoecology: Using the analogue Package

**Gavin L. Simpson**
Environmental Change Research Centre — UCL

**Abstract**

Palaeoecology is an important branch of ecology that uses the subfossil remains of organisms preserved in lake, ocean and bog sediments to inform on changes in ecosystems and the environment through time. The **analogue** package contains functions to perform modern analogue technique (MAT) transfer functions, which can be used to predict past changes in the environment, such as climate or lake-water pH from species data. A related technique is that of analogue matching, which is concerned with identifying modern sites that are floristically and faunistically similar to fossil samples. These techniques, and others, are increasingly being used to inform public policy on environmental pollution and conservation practices. These methods and other functionality in **analogue** are illustrated using the Surface Waters Acidification Project diatom:pH training set and diatom counts on samples of a sediment core from the Round Loch of Glenhead, Galloway, Scotland. The paper is aimed at palaeoecologists who are familiar with the techniques described but not with R.

*Keywords*: analogue matching, palaeoecology, modern analogue technique, dissimilarity, R.

## 1. Introduction

Palaeoecology is a small but increasingly important branch of ecology. Sub-fossil remains of a range of organisms are well preserved in a number of media, primarily lake and ocean sediments and peat bogs. Analysis of these remains can show how individual organisms through to whole ecosystems develop and evolve, and how they respond to external environmental pressures, such as climate change and anthropogenic pollution. In recent decades palaeoecology has progressed from a primarily descriptive science to one which today involves a wide range of quantitative analysis. This development has been required as palaeoecology has begun to be used to answer questions in areas relating to public policy on pollution impacts and in conservation biology.

Two important quantitative applications of palaeoecology are palaeoenvironmental reconstructions and approaches to define reference conditions and restoration success.

Quantitative palaeoecology has played a key role in identifying the problem and the causes of major environmental issues that have been at the centre of much public concern over the past 20 years or so, such as acid rain and surface water acidification, eutrophication and anthropogenic climate change. In each of these cases, the onset of change or pollution occurred long before environmental monitoring programs were around to detect any change. A key issue, therefore, is to be able to reconstruct past changes in the environment (e.g. lake water pH or nutrient concentrations, air temperatures, and sea surface temperature and salinity) from the remains of organisms preserved in sediments, so that the extent and timing of the change can be determined. These may in turn suggest particular causative mechanisms.

Acknowledging that many aquatic environments are today degraded as a result of anthropogenic activities major new pieces of legislation have been enacted in Europe (the European Council Water Framework Directive, WFD; European Union 2000) and the USA (Clean Water Act; Barbour *et al.* 2000), which at their heart contain the concept of change over a baseline state, the reference condition. In Europe for example, the WFD requires member states to restore all degraded fresh waters to at least good status by 2015. Good status is defined as very minor change compared to the reference condition. In many cases we simply do not know what the appropriate reference state should be as there are invariably few, if any, reliable records that predate the onset of change.

Palaeoecology can also play a role here; palaeoenvironmental reconstructions can inform us as to the likely hydrochemical conditions in the past for certain key parameters, and the remains of various species groups preserved in lake sediments can tell us about the flora and fauna living in a lake prior to change. However, because only certain species groups preserve well in lake sediments, direct palaeoecological analysis of lake sediments can provide only part of the answer. Analogue matching can then be used to identify lakes that are today most similar to the reference conditions of the target lake, and the missing species information filled in from surveys of those species living in the identified sites (Simpson *et al.* 2005).

## 1.1. Calibration

Palaeoenvironmental reconstruction is a multivariate calibration problem. Calibration methods (known as *transfer functions* in the palaeoecological literature) can be classified into two main types; *classical* and *inverse* methods. In general, the species assemblages, $\mathbf{Y}$, in a training set are assumed to be some function $f$ of the environment at those sites, $\mathbf{X}$, plus an error term. This is commonly written as

$$\mathbf{Y} = f(\mathbf{X}) + \epsilon \tag{1}$$

where $\mathbf{Y}$ is an $n \times m$ matrix of counts on $m$ species and $\mathbf{Y}$ is an $n \times p$ matrix of $p$ environmental variables for $n$ samples or sites.

In the classical approach to calibration, $f$ is estimated from a set of training data via regression of $\mathbf{Y}$ on $\mathbf{X}$. Given a sample of fossil species data, $y_0$, $f$ is inverted to yield an estimate of the environment, $x_0$, that gave rise to the fossil assemblage. In all but the simplest cases, however, the inverse of $f$ does not exist and must be estimated from the data, for example via numerical optimisation techniques.

The inverse approach avoids the problem of inverting $f$ by directly estimating the inverse of $f$, denoted $g$, from the data by regressing $\mathbf{X}$ on $\mathbf{Y}$

$$\mathbf{X} = g(\mathbf{Y}) + \epsilon. \tag{2}$$

Note that we do not believe that the species ($\mathbf{Y}$) influence their environment ($\mathbf{X}$).

Inverse approaches are known to perform slightly better in situations where the fossil samples are from the central part of the distribution of the training set, whereas classical approaches perform slightly better at the extremes of the training set and with a small amount of extrapolation (ter Braak 1995). The modern analogue technique, described below, is an inverse multivariate calibration approach.

## 1.2. The modern analogue technique (MAT)

The quantitative analysis of stratigraphic records from sediment archives is predicated on the concept of Uniformitarianism (Rymer 1978), which is summarised by the phrase *the present is the key to the past*. Through knowledge of the present-day ecology of species, inferences about past environmental conditions can be made via analogy to that same set of conditions existing where those species are found living today. This is known as space-for-time substitution, or more commonly as the modern analogue technique (MAT). In MAT, the environment of samples from a modern set of lakes that are most similar in terms of their species composition to a fossil sample can be used as a direct prediction of the environment that existed at the time the fossil sample was deposited (Jackson and Williams 2004). MAT is a $k$-nearest neighbours ($k$-NN) method.

Defining how similar two samples are to one another is a critical consideration in MAT. Dissimilarity or distance coefficients are used, which measure the floristic or faunistic similarity between a fossil sample and each modern training set sample. One recommended dissimilarity coefficient for use with compositional data is the chord distance as it has good signal to noise properties (Overpeck *et al.* 1985; Gavin *et al.* 2003).

The chord distance between samples $j$ and $k$, $d_{jk}$, is

$$d_{jk} = \sqrt{\sum_{k=1}^{m} \left(x_{ij}^{0.5} - x_{ik}^{0.5}\right)^2} \tag{3}$$

where $x_{ij}$ is the proportion of taxon $i$ in sample $k$. For the chord distance, values for $d_{jk}$ range from 0 to $\sqrt{2}$. Another commonly used measure is the $\chi^2$ distance (Prentice 1980; Birks *et al.* 1990). Often the squared forms of these coefficients have been used for no other reason than computational efficiency.

Despite having some optimal properties for percentage compositional data, Faith *et al.* (1987) have criticised the chord distance as a weak measure of compositional dissimilarity.

A wide range of dissimilarity coefficients have been proposed, several of which have been implemented in the function `distance` (see Section 4.1), including several of the coefficients recommended by Faith *et al.* (1987) as good measures of compositional dissimilarity.

### 1.3. Analogue matching

Analogue matching (Overpeck *et al.* 1985; Flower *et al.* 1997) is a palaeoecological technique used to identify the $k$-closest sites from a modern set of lakes that are biologically most similar to the impacted lake prior to the onset of change. The $k$-closest sites are selected on the basis of how similar they are to the target sample in those organisms that are preserved in lake sediments, and are known as modern analogues. The pre-impact or reference condition flora and fauna for the target lake from groups that do not preserve in lake sediments can then be inferred on the basis of the species found living in the modern analogues today (Simpson *et al.* 2005).

### 1.4. Outline of the paper

Section 2 contains a worked example providing an overview of the **analogue** package for R (R Development Core Team 2007). In Section 3 we look at alternative ways of selecting the number of analogues, $k$, to retain in a MAT model. Section 4 describes the wider functionality contained within **analogue**, including the dissimilarity coefficients available, an overview of the plotting functions provided, and how to produce sample specific error estimates for fossil samples and use an independent test set in MAT transfer functions. The paper concludes with a short description of future plans for the package (Section 5).

## 2. Using analogue

This section contains a worked example of how to use the **analogue** package to fit MAT transfer function models and to perform analogue matching. The **analogue** package first has to be loaded before it can be used:

```
R> library("analogue")

This is analogue 0.5-2
```

The version of **analogue** installed is printed if the package has been successfully loaded.

To illustrate **analogue**, the Surface Waters Acidification Project (SWAP) diatom:pH training set is used (Stevenson *et al.* 1995), along with diatom counts from a sediment core taken from the Round Loch of Glenhead, Galloway, Scotland (Jones *et al.* 1989). The data sets also need to be loaded before they can be used:

```
R> data(swapdiat, swappH, rlgh, package = "analogue")
```

The `swapdiat` data set contains diatom[1] counts on 277 species from 167 lakes. Matching measurements of lake water pH (acidity) are available for each lake in `swappH`. These pH measurements are the average of four quarterly samples.

The sediment core from the Round Loch of Glenhead (RLGH from now on) contains diatom counts on 139 species from 101 levels.

In both datasets the diatom counts are expressed as percentage abundances.

## 2.1. MAT transfer functions

MAT transfer functions are built using the generic function `mat`. The default method for `mat` takes three arguments; `x` — a data frame of diatom counts for the training set, `y` — a numeric vector of observations of the environmental variable of interest, and `method` — the dissimilarity coefficient to use.

The data frame of diatom counts (`x`), must have the same columns (species) as the data frame of counts for the sediment core for which MAT reconstructions are required. To ensure that both data frames have the same set of columns, the `join` function is used to merge the two data sets.

```
R> dat <- join(swapdiat, rlgh, verbose = TRUE)

Summary:

             Rows Cols
Data set 1:   167  277
Data set 2:   101  139
Merged:       268  277
```

The `verbose = TRUE` argument instructs the function to print out summaries of the merged data sets. `dat` is a list containing two data frames. These are the original datasets but now with a common set of columns (species). The defaults for `join` also replace the missing values created when merging the two data sets with zeros. This behaviour can be controlled through the `na.replace` argument.

An alternative to merging the two data sets would be to select only the intersect of the data sets, i.e. select only those columns in common between the two datasets. This is a non-standard approach however, and is not consistent with implementations in other software packages. One potential problem with the merging approach employed by `join` is the additional zero values added to one or both of the training set or fossil samples, which may exacerbate the double-zero problem or have an unduly large effect on the values of the chosen dissimilarity coefficient. As such, care must be taken when forming training sets and fossil samples, as well as in the choice of dissimilarity coefficient.

By convention, dissimilarity coefficients are defined for proportional data. As the data used in this example are percentages we need to convert them to proportions. We extract each of the merged data sets (the components of `dat`) back into the training set and the fossil set, converting the data into proportions as we do so.

```
R> swapdiat <- dat$swapdiat/100
R> rlgh <- dat$rlgh/100
```

The data are now ready for analysis. We will fit a MAT model to the SWAP training set using the squared chord distance (SCD) coefficient:

[1]Diatoms are unicellular algae that possess a frustule (cell wall) composed of a form of silica. Diatoms live wherever there is water and light. Diatom frustules are highly resistant and as such preserve well in lake sediments. Individual diatom species are identified by different ornamentation of the frustule.

```
R> swap.mat <- mat(swapdiat, swappH, method = "SQchord")
```

An overview of the fitted model is produced by printing the stored object:

```
R> swap.mat

        Modern Analogue Technique

Call:
mat(x = swapdiat, y = swappH, method = "SQchord")

Percentiles of the dissimilarities for the training set:

   1%    2%    5%   10%   20%
0.416 0.476 0.574 0.668 0.815

Inferences based on the mean of k-closest analogues:

  k   RMSEP      R2 Avg Bias Max Bias
  1  0.4227  0.7139  -0.0254  -0.3973
  2  0.3741  0.7702  -0.0493  -0.4689
  3  0.3387  0.8088  -0.0379  -0.4034
  4  0.3282  0.8200  -0.0335  -0.4438
  5  0.3136  0.8356  -0.0287  -0.4124
  6  0.3072  0.8444  -0.0386  -0.4152
  7  0.3167  0.8364  -0.0481  -0.4179
  8  0.3065  0.8474  -0.0433  -0.4130
  9  0.3049  0.8495  -0.0436  -0.4111
 10  0.3015  0.8548  -0.0473  -0.4083

Inferences based on the weighted mean of k-closest analogues:

  k   RMSEP      R2 Avg Bias Max Bias
  1  0.4227  0.7139  -0.0254  -0.3973
  2  0.3711  0.7734  -0.0476  -0.4614
  3  0.3375  0.8102  -0.0385  -0.4088
  4  0.3272  0.8213  -0.0346  -0.4433
  5  0.3144  0.8348  -0.0298  -0.4205
  6  0.3077  0.8435  -0.0371  -0.4253
  7  0.3148  0.8377  -0.0451  -0.4250
  8  0.3049  0.8483  -0.0407  -0.4206
  9  0.3035  0.8500  -0.0408  -0.4205
 10  0.3005  0.8546  -0.0442  -0.4180
```

The percentiles of the distribution of SCD values for the training set are displayed, along with model performance statistics for the training data of inferences for pH based on the mean and weighted mean of the $k$ closest analogues. The weights used are the inverse of the dissimilarity, $1/d_{jk}$, for each of the $k$-closest analogues. It should be noted that this may give overly large weights to nearly identical analogues, which may be of concern in species poor oceanic data sets, but not generally in species rich limnological training sets. By default only statistics for $k = 1, \ldots, 10$ closest analogues are shown. The RMSEP values shown are leave-one-out errors; the prediction for each sample in the training set is based on $k$-closest analogues excluding that sample. These values are not strongly biased, unlike the apparent (RMSE) errors from other methods such as the weighted averaging-based techniques. There is not much to choose between models that use

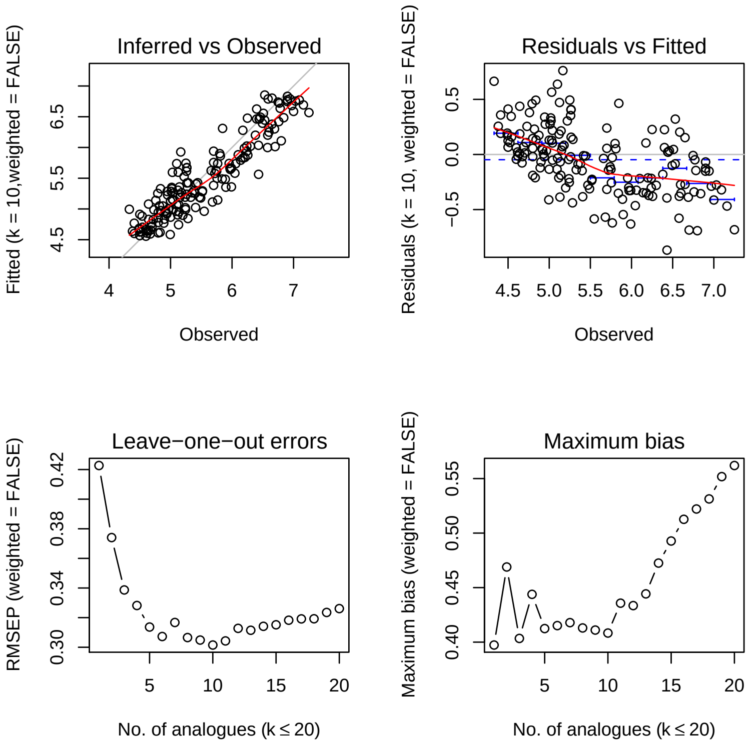


Figure 1: Summary diagram of the results of a MAT model applied to predict lake water pH from the SWAP diatom data set — see text for details.

the mean or weighted mean. For the rest of this example, we restrict ourselves to non-weighted versions of the models.

A more detailed summary of the results may be displayed using the `summary` method:

```
R> summary(swap.mat)
```

Before using this model to reconstruct pH for the RLGH core, the number of analogues, $k$, to use in the reconstructions must be determined. A simple way of choosing $k$ is to select $k$ from the model with lowest RMSEP. In the printed results shown above, the model with the lowest RMSEP was a model with $k = 10$ closest analogues for both the mean and weighted mean indices. We should check this number however, as the displayed lists were restricted to show only the $k = 1, \ldots, 10$ closest analogues. Whenever $k$ is not specified, the functions in **analogue** automatically choose the model with lowest RMSEP. The simplest way to check this is to the use the `getK` extractor function:

```
R> getK(swap.mat)

[1] 10
attr(,"auto")
[1] TRUE
attr(,"weighted")
[1] FALSE
```

This shows that the model with 10 closest analogues has the lowest RMSEP, and that this value was chosen automatically and not set by the user.

mat has a plot method, which provides a plot.lm-like function to graphically summarise the fitted model. By default 4 different plots of the model are produced, so we split the plotting region in four before plotting and subsequently restore the original settings:

```
R> opar <- par(mfrow = c(2, 2))
R> plot(swap.mat)
R> par(opar)
```

The resulting plot is displayed in Figure 1. The upper left panel of Figure 1 shows a plot of the observed versus fitted values, whilst the upper right panel shows a plot of the observed values versus model residuals. The dashed blue line in the residuals plot shows the average bias in the model. In both plots, the solid red line is a LOWESS smoother (span = 2/3).

The labels for the y-axes of both plots show the value of $k$ selected automatically by mat — in this case $k = 10$ analogues. We can confirm this value by looking at the plot of the leave-one-out errors (RMSEP) in the lower left panel of Figure 1. This is a screeplot of the RMSEP values for models with various values of $k$ (by default this is restricted to be $\leq 20$ to avoid clutter). We can see that a model with 10 analogues has lowest RMSEP although there is not a lot of difference in the RMSEP of models with between 6 and 11 analogues. The lower right panel of Figure 1 shows a screeplot, similar to the plot of leave-one-out errors, but which displays the maximum bias in models of various sizes.

This choice of $k$ is generally not strongly biased despite being determined *post hoc* from the training data. However, Telford *et al.* (2004) demonstrate a worst case where this $k$ is badly biased. The use of an independent optimsation set, alongside the usual training and test sets, is recommended to avoid this bias (Telford *et al.* 2004). Section 4.2.2 shows how to use independent test or optimsation sets with **analogue**.

This model can now be used to reconstruct past pH values for the RLGH core. The predict method of mat can be used for reconstructions:

```
R> rlgh.mat <- predict(swap.mat, rlgh, k = 10)
R> rlgh.mat
```

The reconPlot method can be used to plot the reconstructed values as a time series-like plot — the resulting plot is shown in Figure 2:

```
R> reconPlot(rlgh.mat, use.labels = TRUE, ylab = "pH", xlab = "Depth (cm.)")
```

The argument use.labels = TRUE instructs the function to take the names component of the predicted values as the values for the x-axis. Here depth is a surrogate for time.

If we are interested in how reliable our reconstructed values are, a useful descriptor is the minimum dissimilarity between a core sample and the training set samples (minDC). If there are no close modern analogues in the training set for certain fossil samples, we will have less faith in the MAT reconstructions for those fossil samples than for samples that do have close modern analogues. The minDC function can be used to extract the minimum dissimilarity for each fossil sample:

```
R> rlgh.mdc <- minDC(rlgh.mat)
```

Printing the resulting object (rlgh.mdc) doesn't yield very much information. It is easier to display the minDC values in a plot similar to the one produced by reconPlot above:

```
R> plot(rlgh.mdc, use.labels = TRUE, xlab = "Depth (cm.)")
```

The resulting plot is shown in Figure 3. The dotted horizontal lines are the probability quantiles of the distribution of dissimilarity values for the training samples. A useful rule of thumb is that

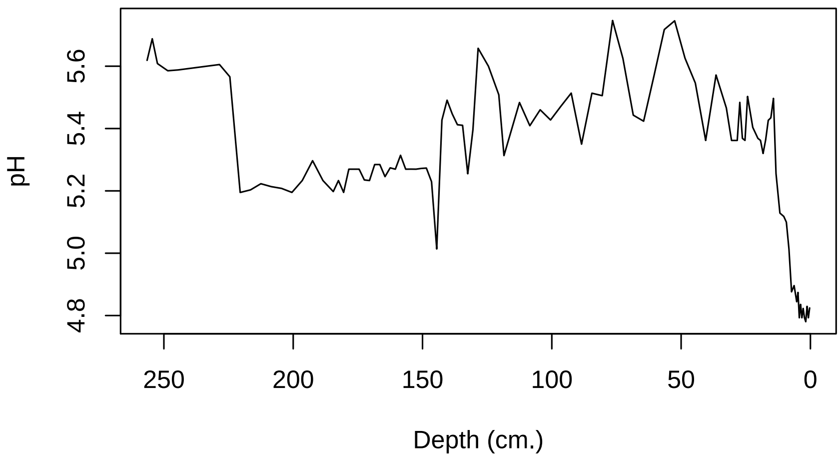


Figure 2: Time series plot of the pH reconstruction for the RLGH core. Depth is a surrogate for time, with 0 being the most recent period represented by the core.

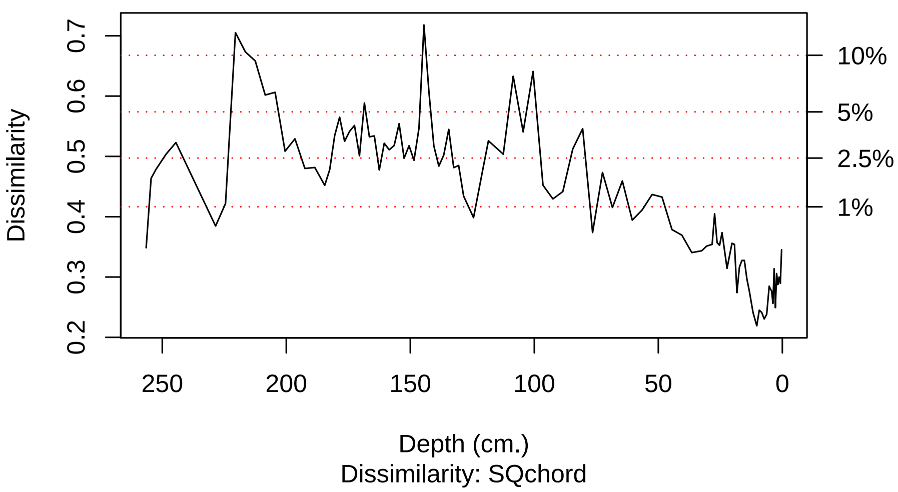


Figure 3: Time series plot of the minimum dissimilarity between each core (fossil) sample and the SWAP training set samples. The dotted, horizontal lines are drawn at various percentiles of the distribution of the pair-wise dissimilarities for the training set samples.

a fossil sample has no close modern analogues where the minDC for the sample is greater than the 5th percentile of the distribution of dissimilarity values for the training samples. As Figure 3 shows, there are several periods of the RLGH core that have no close modern analogues.

## 2.2. Analogue matching

Analogue matching (AM) is a more general version of MAT and the two techniques are used for different purposes. As such, a different set of functions are provided in **analogue** to perform AM. The main function is `analog` and it is used in much the same way as `mat` was earlier, but now both `x` and `y` are data frames of species data.

Returning to the RLGH example, in AM all we are interested in is identifying those samples from the modern training set that are close modern analogues for samples from the RLGH core. In particular, we define the reference condition or period for acidified lakes to be immediately prior to the onset of the industrial revolution, *c.* 1800. We accept that this period is not the "natural" state of the RLGH as many UK surface waters have experienced several thousand years of human impact, but this reference condition is appropriate for assessing recovery from recent acidification resulting from the burning of fossil fuels for energy generation and industrial activities. We use `analog`, this time with the chord distance (CD) measure and select only those samples from the reference period of the RLGH (samples 25–37):

```
R> rlgh.ref <- rlgh[25:37, ]
R> swap.ana <- analog(swapdiat, rlgh.ref, method = "chord")
R> swap.ana

        Analogue matching for fossil samples

Call: analog(x = swapdiat, y = rlgh.ref, method = "chord")
Dissimilarity: chord

Percentiles of the dissimilarities for the training set:

   1%    2%    5%   10%   20% 
0.645 0.690 0.758 0.817 0.903 

        Minimum dissimilarity per sample

Dissimilarity: chord 

020.3 022.3 024.3 025.3 026.3 027.3 028.3 030.5 032.5 036.5 040.5 044.5 048.5 
0.597 0.561 0.611 0.594 0.597 0.636 0.595 0.593 0.586 0.584 0.608 0.615 0.658 
```

In the minimum dissimilarity section of the printed results, the upper row is the core sample label — here these are numbers representing depth down the core. The lower row is the minimum dissimilarity between the fossil sample and a training set sample. A more detailed display of the $k$ best analogues ($k = 10$ by default) is given by the `summary` method.

Having performed the main AM computations, we need to extract information from the resulting object, particularly those samples from the training set that are as close or closer than $c$ to each fossil sample, where $c$ is some critical threshold or cutoff. The `cma` function (*c*lose *m*odern *a*nalogues) does this:

```
R> swap.cma <- cma(swap.ana)
R> swap.cma
```

```
        Close modern analogues of fossil samples

Call: cma(object = swap.ana)

Dissimilarity: chord
Cutoff: 0.705

        Number of analogues per fossil sample:

020.3 022.3 024.3 025.3 026.3 027.3 028.3 030.5 032.5 036.5 040.5 044.5 048.5
   14    13    10     9    10     9    10    11    10    19     9    12     5
```

Notice that we do not need to specify a cutoff, $c$. By default, `cma` uses the 2.5th percentile of the distribution of dissimilarities for the modern training set as the value of $c$ if none is supplied. Argument `"cutoff"` is used if you want to supply a different cutoff value:

```
R> cma(swap.ana, cutoff = 0.5)
```

The close modern analogues can be displayed graphically using the `plot` method for `cma`. This is a wrapper for `stripchart`, and only displays samples that have one or more close modern analogues. Stripcharts are one dimensional scatter plots and are a good alternative to boxplots when sample sizes are small, as they generally are when selecting close modern analogues for fossil samples.

```
R> plot(swap.cma)
```

The stripchart is shown in Figure 4. The y-axis contains the samples of interest, and for each of these a point is drawn along the x-axis for each close modern analogue within the dissimilarity cutoff, $c$, chosen. Recall that the sample labels for the RLGH sediment core are just the depths from the core top, it is, therefore, only coincidental that the y-axis appears numeric and continuous.

One problem with analogue methods is the need to decide what level of dissimilarity between two samples should accept before we consider the two samples as being truly dissimilar. We avoid this problem with MAT by selecting the number of analogues that minimises the RMSEP. We cannot do this in AM, however, as invariably we do not have known environmental data for the fossil samples we are comparing with the training set. Instead we must choose a suitable cutoff for the dissimilarity, as described above.

One solution to this problem is to take a low percentile of the distribution of training set dissimilarities as the cutoff; often the 5th or 10th percentile (Anderson *et al.* 1989). However, if the shape of the distribution of dissimilarities is strongly left skewed, taking the 5th or 10th percentile would lead to the use of an overly large cutoff, and if there is strong right skew, a smaller cutoff will be chosen. Depending on the shape distribution of training set dissimilarities one may decide to choose a lower or higher percentile to guide their choice of cutoff. We can examine the distribution of training set dissimilarities using the `dissim` extractor function and its plot method:

```
R> plot(dissim(swap.ana))
```

The resulting plot is shown in Figure 5. A reference normal is overlaid with the same mean and standard deviation as the observed set of dissimilarities, with the same sample size. The two vertical, dotted lines are drawn at the 5th percentiles of the observed and reference distributions. The actual percentile drawn can be changed using argument `"prob"`. As Figure 5 shows, the observed distribution of dissimilarities for the training set is not too far from a normal distribution, though there is some slight skewness to the left. The 5th percentile would suggest a cutoff of $c \leq 0.758$ in this case.

An alternative solution to the problem of deciding on a suitable cutoff is to use Monte Carlo simulation to determine a dissimilarity threshold that is unlikely to have occurred by chance (Sawada

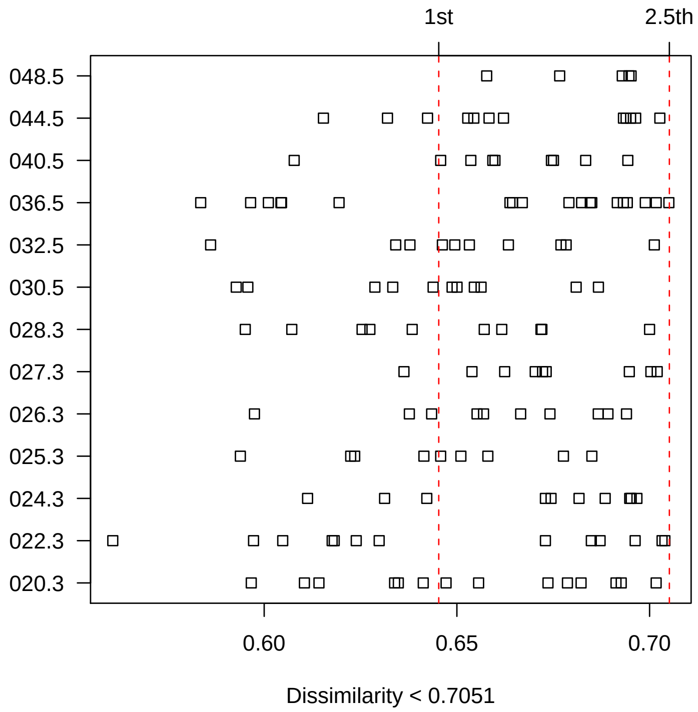


Figure 4: Plot of the number of close modern analogues from the SWAP training set and their dissimilarity to samples from the RLGH core (y-axis).

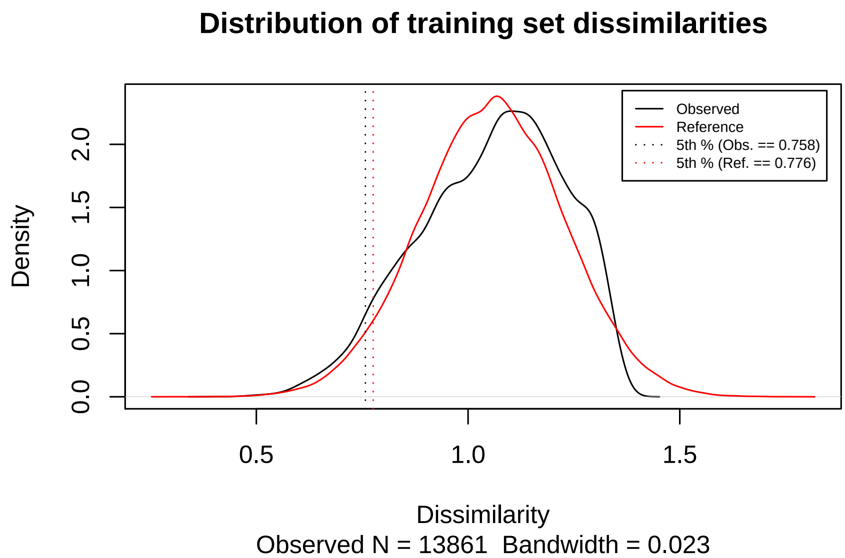


Figure 5: Density plot of the distribution of the pair-wise dissimilarities for the SWAP training set samples and a reference normal distribution.

*et al.* 2004). At random, two samples are drawn from the training set and the dissimilarity between the two samples is recorded. This process is repeated many times to generate a randomisation distribution of dissimilarity values expected by random comparison of samples. A threshold value that occurred one time in a hundred would correspond to a significance level of 0.01. The dissimilarity value that achieves this level of significance can be determined by selecting the 0.01 probability quantile of the randomisation distribution (the 1st percentile).

The `mcarlo` function provides this functionality and methods are available for `"mat"` and `"analog"` objects.

```
R> swap.mc <- mcarlo(swap.ana)
R> swap.mc

        Simulated Dissimilarities

Simulation type : paired
No. simulations : 10000
Coefficient     : chord

Summary of simulated distribution:
    Min 1st Qu.  Median    Mean 3rd Qu.     Max
  0.616   0.847   1.009   0.987   1.132   1.317

Percentiles of simulated distribution:
   1%  2.5%    5%   10%   90%   95% 97.5%   99%
0.633 0.663 0.690 0.734 1.236 1.262 1.286 1.298
```

See Section 3.3 for details on how Receiver Operating Characteristic curves may be used to determine and optimal value for $c$.

# 3. Alternative methods for choosing $k$

A wide range of techniques have been described in the literature for choosing a value of $k$ that gives the best model predictions/reconstructions with the lowest error. Some of these techniques are available in **analogue**.

## 3.1. Bootstrapping

The most objective way of determining an optimal value for $k$ is to use some form of cross-validation (CV). **analogue** currently contains functions to implement bootstrapping (Birks *et al.* 1990). Repeated bootstrap samples are drawn from the training set and a MAT model fitted to the selected samples. These models are then used to predict for the out-of-bag (OOB) samples. A RMSEP measure is then calculated by averaging over the OOB predictions. This procedure is the same as bagging (Breiman 1996), but a different form of RMSEP than the normal definition is used (Birks *et al.* 1990). The $\mathrm{RMSEP}_{\mathrm{boot}}$ of the training set is calculated as:

$$\mathrm{RMSEP}_{\mathrm{boot}} = \sqrt{s_1^2 + s_2^2}, \tag{4}$$

where $s_1$ is the standard deviation of the OOB residuals and $s_2$ is the mean bias or the mean of the OOB residuals.

The `bootstrap` function is used to bootstrap resample the training set from a MAT model. Continuing the RLGH MAT example from earlier, we take 100 bootstrap samples and examine the returned object:

```
R> set.seed(1234)
R> swap.boot <- bootstrap(swap.mat, n.boot = 100)
R> swap.boot

        Bootstrap results for palaeoecological models

Model type: MAT
Weighted mean: FALSE
Number of bootstrap cycles: 100

Leave-one-out and bootstrap-derived error estimates:

           k  RMSEP     S1     S2 r.squared avg.bias max.bias
LOO       10 0.3015      -      -    0.8548 -0.04729  -0.4083
Bootstrap 11 0.3246 0.1168 0.3028    0.9250 -0.05109  -0.4452
```

The bootstrap procedure suggests that $k = 11$ analogues provides the lowest $\text{RMSEP}_{\text{boot}}$. We cannot directly compare the RMSEP values shown, as a different method was used to calculate the two values. The leave-one-out RMSEP is calculated in the normal way:

$$\text{RMSEP}_{\text{loo}} = \sqrt{\frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{n}}, \quad (5)$$

where $i = 1, \ldots, n$ and $n$ is the number of samples, whilst the bootstrap RMSEP is calculated following (4). We can compute a RMSEP that can be compared with the leave-one-out RMSEP as follows:

```
R> RMSEP(swap.boot, type = "standard")

[1] 0.3028484
```

It is felt that the RMSEP of Birks *et al.* (1990) gives a more reliable estimate of the real prediction error than the standard RMSEP definition. Furthermore, the alternate RMSEP formulation is used to produce bootstrap sample-specific errors (see Section 7).

## 3.2. Changing the stored value of $k$

Having used `bootstrap` to select a value for $k$, it would be useful if this value could be stored in the MAT model so that functions that utilise the stored value of $k$ will use the new value automatically. The `getK` function can be used extract the stored value of $k$ from certain objects, whilst `setK` can be used to alter or set the stored value. To illustrate, we extract the bootstrap selected value of $k$ and store this in the `swap.mat` object created earlier:

```
R> getK(swap.boot)

[1] 11
attr(,"auto")
[1] TRUE
attr(,"weighted")
[1] FALSE

R> setK(swap.mat) <- getK(swap.boot)
```

## 3.3. Receiver Operating Characteristic (ROC) curves

Recently Wahl (2004) and Gavin *et al.* (2003) have presented a framework for identifying an optimal critical threshold, $c$, for dissimilarity, that best discriminates between known analogue and non-analogue samples. This framework is based on the use of Receiver Operating Characteristic (ROC) curves but is only applicable where training set samples can be *a priori* assigned to groups or types of samples (e.g. samples classified into vegetation types). A site is an analogue for another site if they belong to the same group, and not an analogue if they come from different groups.

There are two types of error that arise when a cutoff value for the dissimilarity is used: i) *false positive error*, which occurs when two samples that are not analogues are determined to be analogues on the basis of the chosen cutoff; and ii) *false negative error*, when two samples that really are analogues are determined non-analogous. The optimal cutoff value is the one that jointly minimises these two types of error. ROC curve analysis allows us to compare the rates of these two different errors for various cutoff values and to determine the optimal cutoff.

ROC curves are drawn using two measures of performance: i) *sensitivity*, the proportion of true analogues out of all sites said to be analogues on the basis of the cutoff; and ii) *sensitivity*, the proportion of true non-analogues out of all non-analogues. Sensitivity is drawn on the y-axis and $1-$specificity is drawn on x-axis. The point on the ROC curve closest to the top-left corner of the plot corresponds to the cutoff value that jointly minimises the two types of error. The so-called area under the ROC curve (AUC) is a measure of the ability of the community dissimilarity to discriminate between analogue samples and non-analogue ones, and is equivalent to the Mann-Whitney U.

The general idea is that by using the ROC curve for your training set/model a critical threshold $c$ is determined. Instead of choosing the $k$-closest analogues for each fossil sample, you now choose the $m$-closest samples from the training set with a dissimilarity of $\leq c$. The implication being that a variable number of analogues is used for each fossil sample in the reconstruction because only those samples that really are analogues are used. Contrast this with the approach presented earlier, where a fixed number of $k$-closest analogues is used for all fossil samples. In effect, by using a fixed value of $k$, the standard approach is employing a variable threshold $c$ in its predictions.

**analogue** contains functions that implement a modified version of the ROC method of Wahl (2004) and Gavin *et al.* (2003). The major difference is that **analogue** considers all pair-wise comparisons in building the ROC curve, whereas the the methodology proposed by Wahl (2004) and Gavin *et al.* (2003) uses only the $k$-closest analogues.

The `roc` function is used to produce ROC curves from `mat` and `analog` objects. We continue the worked example by calculating a ROC curve for the SWAP training set. As these data do not fall into natural groupings, we first need to cluster the lakes into groups of similar lake types, arbitrarily splitting the training set into 12 groups. Note that we do this only to illustrate the approach. In reality, the groups should have been determined *a priori*, on the basis of a lake-typology (such as in the case of WFD assessments of standing waters) or vegetation types for example, and not via a clustering of the species data in the training set.

```
R> clust <- hclust(as.dist(swap.mat$Dij), method = "ward")
R> grps <- cutree(clust, k = 12)

R> swap.roc <- roc(swap.mat, groups = grps)
R> swap.roc

        ROC curve of dissimilarities

Optimal Dissimilarity = 0.894

AUC = 0.889, p-value: < 2.22e-16
No. within: 1214   No. outside: 12647
```

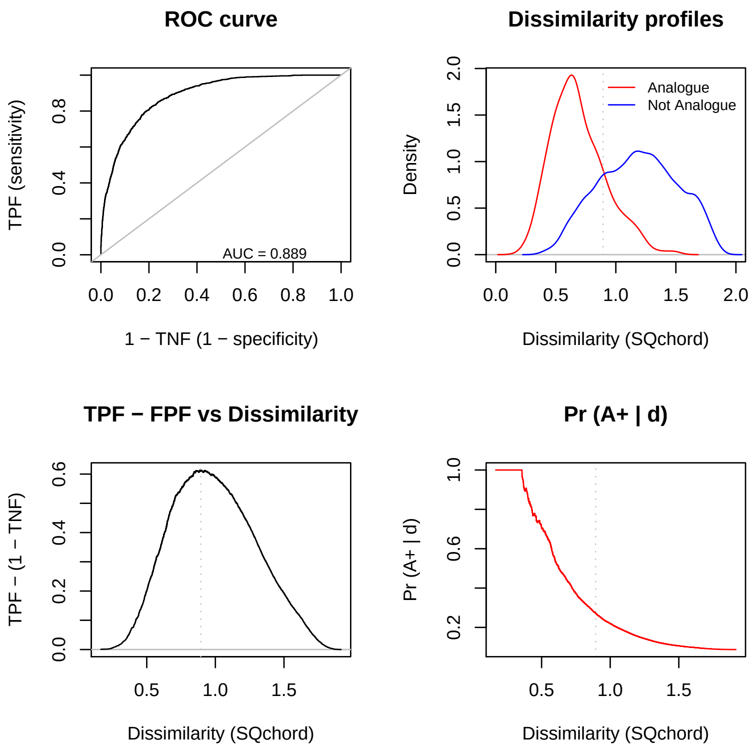


Figure 6: Plot summarising the results of the ROC curve analysis of the SWAP MAT model — see text for details.

The printed results show the optimal dissimilarity $c$, the AUC statistic and its $p$-value. The latter two are determined by the standard R function `wilcox.test`.

The `plot` method for `roc` can display a number of different plots of the ROC results:

```
R> opar <- par(mfrow = c(2, 2))
R> plot(swap.roc)
R> par(opar)
```

The resulting plot is shown in Figure 6. The ROC curve itself is drawn in the upper-left panel. The upper-right panel displays density plots of the distributions of the dissimilarities between analogue and non-analogue samples. The point where the two curves cross is the optimal decision threshold. The vertical, dotted line is the optimal dissimilarity based on the ROC curve. This line may not always pass exactly through the optimal decision threshold as the ROC curve has been evaluated on a finite set of dissimilarities, but it is usually very close.

The lower left panel of Figure 6 is a plot of the difference between the true positive fraction (TPF) and the false positive fraction (FPF) as a function of dissimilarity. The vertical, dotted line is the optimal dissimilarity based on the ROC curve. The lower right panel of Figure 6 is a plot of the posterior probability of two samples being analogues as a function of the dissimilarity, $d$. It is worth noting that the posterior probability of analogue is based on the slope of ROC curve, and that there are various definitions of the slope of a ROC curve in the literature. The slope used in `plot.roc` is different to that used by Gavin *et al.* (2003), who use a measure of the instantaneous rate of change at points on the ROC curve[2], where as in **analogue**, the slope of the ROC curve is TPF/FPF (Henderson 1993).

[2] Gavin *et al.* (2003) used binned data from a histogram of dissimilarities for analogue and no-analogue compar-

The data plotted in the lower right panel of Figure 6 is based on the likelihood ratio of a positive event (LR+), which is calculated as LR(+) = TPF/FPF (Henderson 1993). This likelihood ratio is converted into a posterior odds:

$$O^{+}_{\text{post.}} = \text{LR}(+) \times O^{+}_{\text{pri.}} \tag{6}$$

where $O^{+}_{\text{pri.}}$ is

$$O^{+}_{\text{pri.}} = \frac{\text{Pr}^{+}_{\text{pri.}}}{1 - \text{Pr}^{+}_{\text{pri.}}} \tag{7}$$

and $\text{Pr}^{+}_{\text{pri.}}$ is the prior probability of any two samples being analogous (Brown and Davis 2006). $\text{Pr}^{+}_{\text{pri.}}$ may be set at 0.5 (i.e. a 50% probability of two samples being analogues) or may be determined from the observed probability of two samples being analogue (i.e. in the same group) in the modern training set.

The posterior odds of analogue $O^{+}_{\text{post.}}$ are converted to a posterior probability of analogue via

$$\text{Pr}^{+}_{\text{post.}} = \frac{O^{+}_{\text{post.}}}{1 + O^{+}_{\text{post.}}}. \tag{8}$$

The workhorse function used by `plot.roc` to draw the posterior probability of any two samples being analogues is `bayesF`. The help page for `bayesF` contains additional details.

# 4. Other features of analogue

We briefly describe some of the other features of the **analogue** package.

## 4.1. Dissimilarity coefficients

Analogue provides a wide range of dissimilarity coefficients via the `distance` function. A list of the coefficients provided is shown in Table 1. All the dissimilarity coefficients are coded in pure R code. As such, `distance` will not be as quick as other similar functions available in R, such as `dist`, or `vegdist` in **vegan**, where the computations are done in compiled C code. Where there is overlap with coefficients available explicitly or indirectly (via transformation), in functions `dist` or `vegdist`, these faster functions are used by default, but only if no second argument `y` is supplied.

The existing implementation is sufficiently speedy for most problems that might be encountered with training sets of up to about 200 samples. Beyond this, a faster implementation may be desirable to save compute time. C versions of the dissimilarity coefficients already implemented in `distance` are currently being written and will be made available in a future version of **analogue**.

The implementation in `distance` has one main advantage over other implementations. In many situations we are interested in computing the dissimilarities between training set samples and fossil samples, not the pair-wise dissimilarities between samples in a single data set. With other R functions for computing dissimilarities, such as those mentioned above, this is not possible unless the two data sets are merged and the required dissimilarities subsequently extracted from the resulting object. `distance` was primarily designed to work with two separate data frames of species data and to calculate only the required dissimilarities between the two data frames. Pairwise dissimilarities for a single data frame can be calculated using `distance`, by providing the sole data frame as argument `x` and leaving argument `y` as missing, as the following snippet shows.

isons to calculate the slope of the curve across each bin. It is not clear what advantage binning the data has over the method employed in **analogue** or whether it is even necessary.

*where $R_i$ is the range of proportions for descriptor (variable) $i$.

†where $w_i$ is the weight for descriptor $i$ and $s_{jki}$ is the similarity between samples $j$ and $k$ for descriptor (variable) $i$.

| Distance metric | Method | Formula |
|---|---|---|
| Euclidean distance | `euclidean` | $d_{jk} = \sqrt{\sum_i (x_{ij} - x_{ik})^2}$ |
| Squared Euclidean distance | `SQeuclidean` | $d_{jk} = \sum_i (x_{ij} - x_{ik})^2$ |
| Chord distance | `chord` | $d_{jk} = \sqrt{\sum_i (\sqrt{x_{ij}} - \sqrt{x_{ik}})^2}$ |
| Squared chord distance | `SQchord` | $d_{jk} = \sum_i (\sqrt{x_{ij}} - \sqrt{x_{ik}})^2$ |
| Bray-Curtis dissimilarity | `bray` | $d_{jk} = \frac{\sum_i \lvert x_{ij} - x_{ik} \rvert}{\sum_i (x_{ij} + x_{ik})}$ |
| $\chi^2$ distance | `chi.square` | $d_{jk} = \sqrt{\sum_i \frac{(x_{ij} - x_{ik})^2}{x_{ij} + x_{ik}}}$ |
| Squared $\chi^2$ distance | `SQchi.square` | $d_{jk} = \sqrt{\sum_i (x_{ij} - x_{ik})^2 / (x_{i+}/x_{++})}$ |
| Information statistic | `information` | $d_{jk} = \sum_i (p_{ij} \log(\frac{2x_{ij}}{x_{ij} + x_{ik}}) + x_{ik} \log(\frac{2x_{ik}}{x_{ij} + x_{ik}}))$ |
| $\chi^2$ distance | `chi.distance` | $d_{jk} = \sqrt{\sum_i (x_{ij} - x_{ik})^2 / (x_{i+}/x_{++})}$ |
| Manhattan distance | `manhattan` | $d_{jk} = \sum_i (\lvert x_{ij} - x_{ik} \rvert)$ |
| Kendall's coefficient | `kendall` | $d_{jk} = \sum_i \mathrm{MAX_i} - \mathrm{minimum}(x_{ij}, x_{ik})$ |
| Gower's coefficient* | `gower` | $d_{jk} = \sum_i \frac{\lvert x_{ij} - x_{ik} \rvert}{R_i}$ |
| Alternative Gower's coefficient* | `alt.gower` | $d_{jk} = \sqrt{2 \sum_i \frac{\lvert x_{ij} - x_{ik} \rvert}{R_i}}$ |
| Gower's mixed coefficient† | `mixed` | $d_{jk} = \frac{\sum_{i=1}^p w_i s_{jki}}{\sum_{i=1}^p w_i}$ |

Table 1: List of the dissimilarity coefficients currently available in function `distance`.

```
R> dists1 <- distance(swapdiat, method = "bray")
R> dists2 <- distance(swapdiat, rlgh, method = "bray")
```

Object `dists1` contains the pairwise Bray-Curtis dissimilarities between samples in the SWAP diatom data set, where as `dists2` contains the Bray-Cutis dissimilarity between each sample in `rlgh` and each sample in `swapdiat`. The dissimilarity coefficient used is specified using the `method` argument.

## 4.2. Advanced MAT usage

### *Sample specific error estimates*

Using the bootstrap method described above, it is possible to derive sample specific errors of the reconstructed values for core samples. The sample specific RMSEP is calculated using:

$$\mathrm{RMSEP} = \sqrt{s_{1_{\mathrm{fossil}}}^2 + s_{2_{\mathrm{model}}}^2}, \tag{9}$$

where $s_{1_{\mathrm{fossil}}}$ is the standard deviation of the bootstrap estimates of the environment for an individual fossil sample and $s_{2_{\mathrm{model}}}$ is the average bias (mean of residuals) from the MAT model.

We continue the RLGH example from above and generate sample specific RMSEPs for each of the RLGH core samples using the `predict` method for `mat` and 100 bootstraps:

```
R> set.seed(1234)
R> rlgh.boot <- predict(swap.mat, rlgh, bootstrap = TRUE, n.boot = 100)
R> reconPlot(rlgh.boot, use.labels = TRUE, ylab = "pH", xlab = "Depth (cm.)",
+     display.error = "bars", predictions = "bootstrap")
```

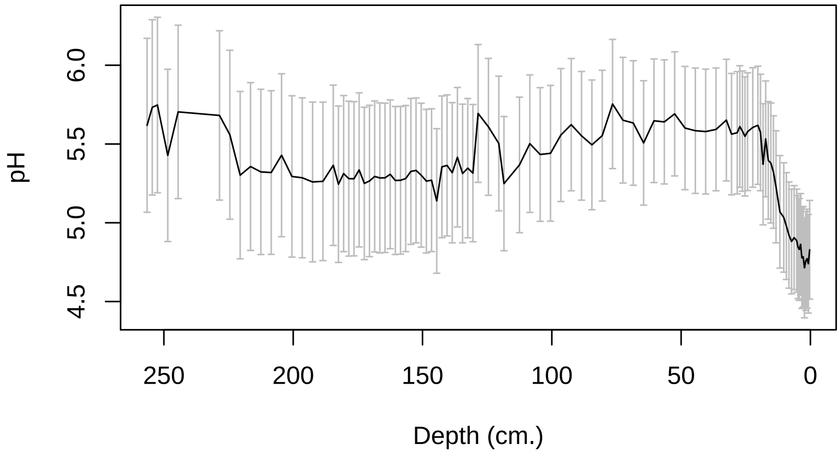


Figure 7: Time series plot of the pH reconstruction for the RLGH core, with bootstrap-derived sample specific errors. Depth is a surrogate for time, with 0 being the most recent period represented by the core.

The bootstrap predictions are plotted with error bars representing the sample specific RMSEP of the estimated value. The resulting plot is shown in Figure 7. The `display.errors` argument controls how the model errors are displayed; available options are `"none"`, `"bars"` or `"lines"`.

### *Using an independent test set*

The `bootstrap` function can also be used to provide a realistic RMSEP using an independent test set. A test set is one where both the predictor and the response variables have been observed, invariably by random splitting of the a full data set into a training and a test set.

We begin by randomly splitting the SWAP data into a training set of 100 samples and a test set of 67 samples:

```
R> set.seed(1234)
R> want <- sample(1:nrow(swapdiat), 67, replace = FALSE)
R> train <- swapdiat[-want, ]
R> train.env <- swappH[-want]
R> test <- swapdiat[want, ]
R> test.env <- swappH[want]
```

Now we draw 100 bootstrap samples from the training set and predict for the test set:

```
R> train.mat <- mat(train, train.env, method = "SQchord")
R> test.boot <- bootstrap(train.mat, newdata = test, newenv = test.env,
+     n.boot = 100)
R> test.boot

        Bootstrap results for palaeoecological models

Model type: MAT
```

```
Weighted mean: FALSE
Number of bootstrap cycles: 100

Leave-one-out and bootstrap-derived error estimates:

             k  RMSEP     S1     S2 r.squared avg.bias max.bias
LOO          9 0.3298      -      -    0.8329 -0.06244  -0.5729
Bootstrap   12 0.3621 0.1320 0.3371    0.9167 -0.07278  -0.6136
Test         7 0.3948      -      -    0.8807  0.04068   0.7456
Test (Boot)  9 0.4335 0.1629 0.4018    0.8871 -0.04978  -0.7130
```

The printed results now show two additional lines for the model and bootstrap summary statistics for the test set. The bootstrap RMSEP for the test set is $\sim 0.07$ pH units higher than the standard bootstrap RMSEP for the training set, suggesting that simply bootstrapping a training set slightly underestimates the real error performance. It should be noted that, ideally, the test set samples should be taken as a random, stratified sample from the full data set, such that the test set samples cover the entire range of the full data set.

### *Using an optimisation set*

Telford *et al.* (2004) demonstrated that choosing $k$ *post hoc* by selecting the $k$ with lowest RMSEP for the training set can be biased, and that in some cases this bias can be quite large. The solution to this problem is to use an optimisation set alongside the usual training and test sets (Telford *et al.* 2004). The model is built on a subset of the training data, just as in the previous section, except that we split the test set into a small optimisation set as well as a test set. The optimisation set is used to select $k$, and is the number of analogues that produces the lowest RMSEP for the optimisation set samples. **analogue** provides both the model-based RMSEP as well as the bootstrap RMSEP for the optimisation test. This value of $k$ is then used to predict for the test set samples to produce an independent assessment of the RMSEP of the predictions.

We illustrate this process, first by selecting out the optimisation set samples from the test set,

```
R> set.seed(9876)
R> want <- sample(nrow(test), 40)
R> opti <- test[-want, ]
R> opti.env <- test.env[-want]
R> test <- test[want, ]
R> test.env <- test.env[want]
```

Using the test set created in the previous section, 40 of these samples are randomly selected for the new test set and the remaining 27 are allocated to the optimisation set. The training set is the same as that generated previously.

Using `train.mat`, we bootstrap the training set to produce predictions for the optimisation set,

```
R> opti.boot <- bootstrap(train.mat, newdata = opti, newenv = opti.env,
+     n.boot = 100)
R> opti.boot

        Bootstrap results for palaeoecological models

Model type: MAT
Weighted mean: FALSE
Number of bootstrap cycles: 100

Leave-one-out and bootstrap-derived error estimates:
```

```
             k  RMSEP     S1     S2 r.squared avg.bias max.bias
LOO          9 0.3298      -      -    0.8329 -0.06244  -0.5729
Bootstrap    9 0.3611 0.1456 0.3305    0.9134 -0.06269  -0.5854
Test         5 0.4321      -      -    0.8625  0.01053   0.5994
Test (Boot) 7 0.4895 0.1892 0.4514    0.8651 -0.05449  -0.6596
```

The number of analogues that gives the lowest RMSEP for the optimisation samples is 5 for the model-based predictions and 7 for the bootstrap-based predictions. We continue by selecting the value of $k$ for the model-based predictions and use this to produce predictions for the test set.

```
R> use.k <- getK(opti.boot, prediction = TRUE, which = "model")
R> test.boot <- bootstrap(train.mat, newdata = test, newenv = test.env,
+      k = use.k, n.boot = 100)
R> test.boot

        Bootstrap results for palaeoecological models

Model type: MAT
Weighted mean: FALSE
Number of bootstrap cycles: 100

Leave-one-out and bootstrap-derived error estimates:

             k  RMSEP     S1     S2 r.squared avg.bias max.bias
LOO          5 0.3415      -      -    0.8010 -0.05200  -0.5725
Bootstrap    5 0.3686 0.1734 0.3253    0.9071 -0.05437  -0.5511
Test         5 0.4111      -      -    0.8483  0.03091   0.8416
Test (Boot) 5 0.4177 0.1876 0.3733    0.8800 -0.03407  -0.7895
```

`getK` is used to select the appropriate $k$ from `opti.boot` and this is passed to `bootstrap` as its argument `k`. The printed results show the model- and bootstrap-based RMSEP in the lines labelled "Test".

## *The curse of dimensionality*

The curse of dimensionality, a term coined by Bellman (1961), describes the problem of defining localness in high dimensions; neighbourhoods with a fixed number of samples become less local as the number of dimensions increases (Hastie and Tibshirani 1990). It is common for the dimensionality of palaeoecological data sets to be high, especially with diverse proxies such as diatoms. In the SWAP and RLGH example presented here, there are 277 dimensions (species) and only 167 sites in the modern training set. However, MAT and AM have been applied routinely in palaeoecology without any prior dimension reduction.

Despite this, MAT and AM appear to defy the curse of dimensionality. This may be, as Härdle (1990) shows, because the relevant dimensionality is not $m$, the number of species, but $p$, the number of environmental variables (ter Braak 1995). ter Braak (1995) also suggests that this defiance of the curse is due to the dissimilarity just summing over dimensions, the species.

A common method of dimension reduction in palaeoecology is to delete rare taxa from the training set. Various definitions of what is rare have been used, but taxa that are found in fewer than a set number of sites/samples or whose maximum abundance is less than some prescribed limit are often deleted. Commonly, taxa are retained if they are present in, say, at least 5 or 10 samples in the training set or are found at at least 2% abundance in one or more sample. Often these two measures are combined. This deletion of rare taxa runs counter to ecology, especially in AM, where these rare taxa may be important indicators of particular environments and as yet our knowledge of the autecology of many of the taxa employed in transfer functions is not sufficiently developed to determine their worth. As such, rare taxa should be deleted with care.

The following snippet illustrates how to subset the merged SWAP and RLGH data set to select only those species present in at least 5 sites and with a maximum abundance of at least 2%. `max.abb` and `n.occ` are the maximum abundances and the number of occurrences for each taxon respectively.

```
R> dat <- join(swapdiat, rlgh, split = FALSE)
R> max.abb <- apply(dat, 2, max)
R> n.occ <- colSums(dat > 0)
R> spp.want <- which(max.abb >= 0.02 & n.occ >= 5)
R> swapdiat2 <- swapdiat[, spp.want]
R> rlgh2 <- rlgh[, spp.want]
```

Note that we generate the maximum abundance and number of occurrences per taxon on the combined SWAP and RLGH data sets, by using `join`, but this time with the argument `split = FALSE` so that `dat` contains the full merged species data set rather than the two data sets split out (see page 3). The process has considerably reduced the number of diatom taxa from 277 to 173. One can now proceed to refit the models described earlier, but this time using `swapdiat2` and `rlgh2`.

## 4.3. Generating plots

Several **analogue** functions produce a range of plots. In this section we take a brief look at some of the more important plot-types.

The two main plots commonly used to illustrate palaeoecological transfer function models are i) a plot of inferred (fitted) model estimates versus observed values, and ii) a plot of residuals versus inferred (fitted) values. These are two of the plot-types that can be produced by the `plot` method for `mat`. We have seen how to use this function already, but here we illustrate how individual figures can be produced using `plot.mat`, firstly the inferred estimates versus observed values plot:

```
R> plot(swap.mat, which = 1)
```

The resulting plot is shown in Figure 8. The grey line is a 1:1 line, and the red line is a LOWESS smoother. Whether the smoother is displayed is controlled by the global option `options("add.smooth")` or suppressed by specifying `panel = points` in the call to `plot`.

The residuals versus observed plot is produced using `which = 2` in the call to `plot`:

```
R> plot(swap.mat, which = 2)
```

The resulting plot is shown in Figure 9. By default, a number of additional features are drawn on this plot. The blue, dashed line is the mean bias in the model (the mean of the residuals). A related statistic is the maximum bias. Maximum bias is calculated by splitting the environmental gradient (the range of the response, `y`) into 10 sections, and calculating the mean of the residuals within these sections. The maximum bias statistic is taken as the maximum of the mean biases of the 10 sections. These sections, and the mean bias for each, are plotted as blue, error bar-like lines displayed in Figure 9. Display of these maximum bias markers is controlled by argument `max.bias` of `plot.mat`. The red line is again a LOWESS smoother.

The main remaining plotting function not already covered is `screeplot`. This function produces the type of screeplots displayed in the lower row of Figure 1. `screeplot` methods for `mat` and `bootstrap` are currently available, and can draw screeplots of the RMSEP, average bias or maximum bias statistics for models of size $k$. The statistic displayed is controlled by the `display` argument, which defaults to RMSEP. The `bootstrap` method draws both the leave-one-out and bootstrap-derived statistics. We illustrate this by plotting RMSEP as a function of $k$ for the `swap.boot` object created earlier — the resulting plot shown in Figure 10:

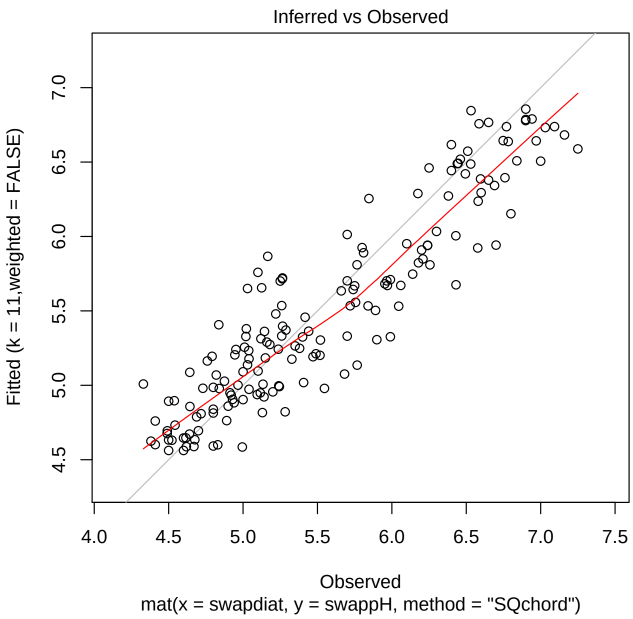


Figure 8: Plot of the MAT-inferred and observed pH values for the SWAP training set — see text for details.

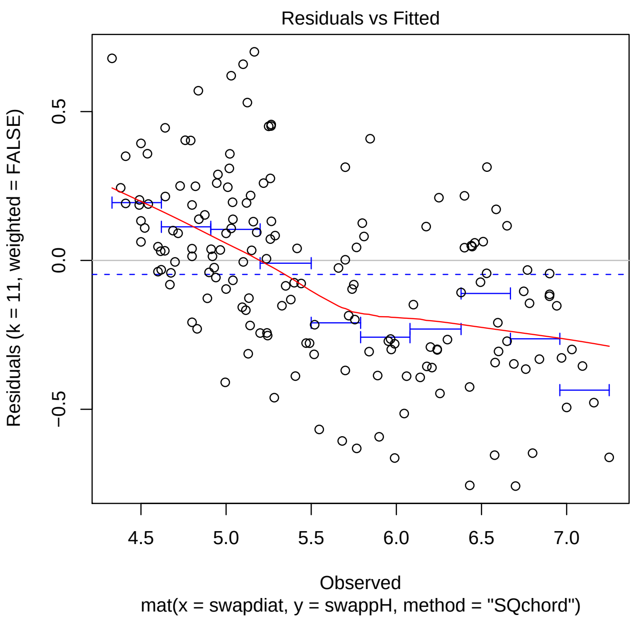


Figure 9: Plot of the MAT model residuals and observed pH values for the swap training set — see text for details.

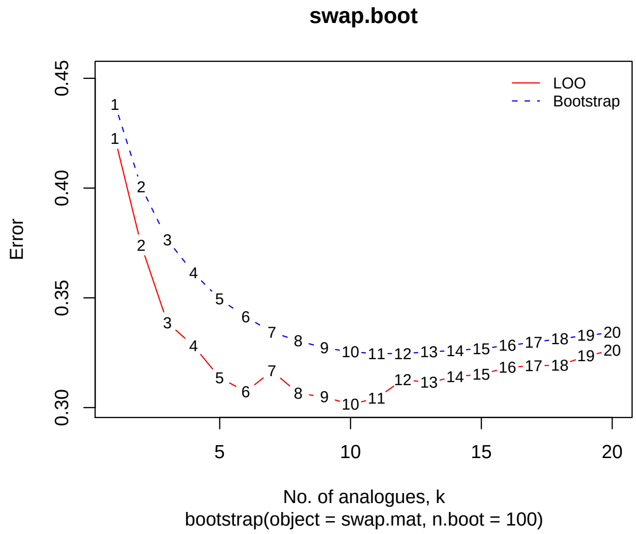


Figure 10: Screeplot of leave-one-out (solid) and boostrap-derived (dashed) RMSEP as a function of $k$ for the SWAP training set.

```
R> screeplot(swap.boot)
```

### 4.4. Generic R functions

Several of the standard R model utility functions have methods for `mat` available in **analogue**. Currently, `fitted` and `resid` methods are provided to extract the fitted values and residuals from a MAT model respectively.

## 5. Final remarks and future development plans

The functionality of R package **analogue** has been demonstrated and explained using the SWAP diatom:pH data set and diatom counts from the RLGH sediment core. The SWAP dataset is a relatively large data set compared to those routinely produced in palaeoecological studies, and as such represents a real-world example of the type of data used in the field.

**analogue** is still in the early stages of planned development. The main functionality for generating MAT transfer functions and reconstructions and for performing AM is already implemented, but several areas of development remain.

As mentioned above, faster C versions of the dissimilarity calculations are planned to speed up the functions for use on large problems. Also, the package code has yet to receive any rigorous optimisation in terms of memory usage or computation time. Once the feature set has stabilised sufficiently, a code review will be performed to identify bottlenecks and to improve the implementation where possible.

It will be noticeable that the functionality is more comprehensive for MAT transfer functions than for analogue matching. This is purely a function of legacy; MAT models have been used in palaeoecology for over 20 years, but analogue matching (in the sense presented in this paper) is a much newer topic and exactly how the results of AM are used in informing conservation policy is an area of ongoing research. As new developments are proposed, they will be added to future versions of **analogue**.

## Acknowledgements

Support, in part, for the development of this package was provided by the European Union Sixth Framework Programme integrated project Euro-limpacs (GOCE-CT-2003-505540). The author wishes to thank Viv Jones for permission to use the RLGH core data and to distribute these data with **analogue**. Two anonymous reviewers and the guest editors provided numerous comments and suggestions that have improved both the manuscript and the package.

## References

Anderson PM, Bartlein PJ, Brubaker LB, Gajewski K, Ritchie JC (1989). "Modern Analogues of Late-Quaternary Pollen Spectra from Western Interior of North America." *Journal of Biogeography*, **16**, 573–596.

Barbour MT, Swietlik WF, Jackson SK, Courtemanch DL, Davies SP, Yoder CO (2000). "Measuring the Attainment of Biological Integrity in the USA: a Critical Element of Ecosystem Integrity." *Hydrobiologia*, **422/433**, 453–464.

Bellman RE (1961). *Adaptive Control Processes.* Princeton University Press.

Birks HJB, Line JM, Juggins S, Stevenson AC, ter Braak CJF (1990). "Diatoms and pH Reconstruction." *Philosophical Transactions of the Royal Society of London Series B-Biological Sciences*, **327**(1240), 263–278.

Breiman L (1996). "Bagging Predictors." *Machine Learning*, **24**(2), 123–140.

Brown C, Davis H (2006). "Receiver Operating Characteristics Curves and Related Decision Measures: a Tutorial." *Chemometrics and Intelligent Laboratory Systems*, **80**, 24–38.

European Union (2000). "Directive 2000/60/EC of the European Parliament and the Council of 23 October 2000 Establishing a Framework for Community Action in the Field of Water Policy." *Official Journal of the European Communities*, **L327**, 1–72.

Faith DP, Minchin PR, Belbin L (1987). "Compositional Dissimilarity as a Robust Measure of Ecological Distance." *Vegetatio*, **69**, 57–68.

Flower RJ, Juggins S, Battarbee RW (1997). "Matching Diatom Aassemblages in Lake Sediment Cores and Modern Surface Sediment Samples: The Implications for Lake Conservation and Restoration with Special Reference to Acidified Systems." *Hydrobiologia*, **344**, 27–40.

Gavin DG, Oswald WW, Wahl ER, Williams JW (2003). "A Statistical Approach to Evaluating Distance Metrics and Analog Assignments for Pollen Records." *Quaternary Research*, **60**, 356–367.

Härdle W (1990). *Applied Nonparametric Regression.* Cambridge University Press.

Hastie T, Tibshirani R (1990). *Generalized Additive Models*, volume 43 of *Monographs on Statistics and Applied Probability.* Chapman and Hall.

Henderson RA (1993). "Assessing Test Accuracy and its Clinical Consequences: a Primer for Receiver Operating Characteristic Curve Analysis." *Annals of Clinical Biochemistry*, **30**, 521–539.

Jackson ST, Williams JW (2004). "Modern Analogues in Quaternary Paleoecology: Here Today, Gone Yesterday, Gone Tomorrow?" *Annual Review of Earth and Planetary Science*, **32**, 495–537.

Jones VJ, Stevenson AC, Battarbee RW (1989). "Acidification of Lakes in Galloway, South West Scotland: a Diatom and Pollen Study of the Post-glacial History of the Round Loch of Glenhead." *Journal of Ecology*, **77**(1), 1–23.

Overpeck JT, Webb T, Prentice IC (1985). "Quantitative Interpretation of Fossil Pollen Spectra—Dissimilarity Coefficients and the Method of Modern Analogs." *Quaternary Research*, **23**(1), 87–108.

Prentice IC (1980). "Multidimensional Scaling as a Research Tool in Quaternary Palynology: a Review of Theory and Methods." *Review of Palaeobotany and Palynology*, **31**, 71–104.

R Development Core Team (2007). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-00-3, URL http://www.R-project.org.

Rymer N (1978). "The Use of Uniformitarianism and Analogy in Palaeoecology, Particularly Pollen Analysis." In D Walker, J Guppy (eds.), "Biology and Quaternary Environments," pp. 245–257. Australian Academy of Science, Canberra.

Sawada M, Viau AE, Vettoretti G, Peltier WR, Gajewski K (2004). "Comparison of North-American Pollen-based Temperature and Global Lake-status with CCCma AGCM2 Output at 6 ka." *Quaternary Science Reviews*, **23**(3–4), 225–244.

Simpson GL, Shilland EM, Winterbottom JM, Keay J (2005). "Defining Reference Conditions for Acidified Waters Using a Modern Analogue Approach." *Environmental Pollution*, **137**, 119–133.

Stevenson AC, Juggins S, Birks HJB, Anderson DS, Anderson NJ, Battarbee RW, Berge F, Davis RB, Flower RJ, Haworth EY, Jones VJ, Kingston JC, Kreiser AM, Line JM, Munro MAR, Renberg I (1995). *The Surface Waters Acidification Project Palaeolimnology Programme: Modern Diatom/Lake-water Chemistry Data-set*. ENSIS Publishing.

Telford R, Andersson C, Birks H, Juggins S (2004). "Biases in the Estimation of Transfer Function Prediction Errors." *Paleoceanography*, **19**, PA4014. doi:10.1029/2004PA001072.

ter Braak CJF (1995). "Non-linear Methods for Multivariate Statistical Calibration and Their Use in Palaeoecology: a Comparison of Inverse ($k$-nearest Neighbours, Partial Least Squares and Weighted Averaging Partial Least Squares) and Classical Approaches." *Chemometrics and Intelligent Laboratory Systems*, **28**, 165–180.

Wahl ER (2004). "A General Framework for Determining Cutoff Values to Select Pollen Analogs with Dissimilarity Metrics in the Modern Analog Technique." *Review of Palaeobotany and Palynology*, **128**, 263–280.