Appendix to:

Legendre, P. & Gauthier, O. 2014. Statistical methods for temporal and space-time analysis of community composition data. *Proceedings of the Royal Society B* **281**, 20132728.

Appendix S2

Temporal eigenfunction methods – Practicals in R

This Appendix gives the calculation details in the R statistical language (R Core Team 2013) for the case study developed in section 8 of the paper.

Table of Contents

0. REQUIRED R PACKAGES

The following packages, available on CRAN, will be used: 'ade4', 'adespatial','mvpart', 'vegan'.

1. THE DATA USED IN THESE EXERCISES

The dataset used in these applications are taken from the Maryland Data Sets of the
Chesapeake Bay Benthic Monitoring Program (http://www.baybenthos.versar.com/data.htm),
a part of the *Chesapeake Bay Program* (http://www.chesapeakebay.net/). You will find
detailed information about the sampling protocol on the web page. The whole dataset is made
available online in numerous .txt files, one per group of variables and per year.

We compiled and formatted these files in an .Rdata file for immediate use in R. The 'reshape'
R package (Wickham 2007) was most useful to accomplish this task.
The file is called <ChesapeakeBay.Maryland.RData>.
Double-click on the RData file, or drag it onto the R icon or in the R console. Else, you can
type `load("ChesapeakeBay.Maryland.RData")` if your R console is set to the working
directory corresponding to the folder containing the file. Check this by typing getwd().

```
ls()
```

1.1. DESCRIPTION OF THE DATA FILES

Please refer to the *Maryland Dataset Data Dictionary* found on
(http://www.baybenthos.versar.com/DOCS/DataDictionaryMD.pdf) for an in-depth description
of the environmental variables and sampling protocols.

fauna (702x205) – Abundances of 205 benthic macrofaunal taxa in alphabetic order. This
includes all animals retained on a 0.5 mm sieve. Nearly all (n = 203) are invertebrates, but two
chordates (*Molgula manhattensis* and *Branchiostoma caribaeum*) are also encountered in the
retained samples.

sampling (702x6)
STATION, SAMPLE_DATE, SAMP_TYPE, GMETHOD, YEAR, SEASON
STATION – A factor, ID tags 1 to 204 corresponding to 27 sites, each with 26 data rows.
SAMPLE_DATE – Sampling date, from 1996-05-06 to 2008-10-01.
SAMP_TYPE – A factor, FIXED or RANDOM sampling sites. Only the FIXED sites are
included in our RData file; see http://www.baybenthos.versar.com/data.htm for details.
GMETHOD – A factor, four gear types for sampling the benthic macrofauna.
Either "BC-PH" ("Post-Hole digger", 250 cm^2 surface area, n = 156), "BC-WC" (Wildco
box corer, 225 cm^2 surface area, n = 468), "PP" (Petite Ponar, 250 cm^2 surface area,
n = 26), or "VV-YM" (Van-Veen modified Young Grab, 440 cm^2 surface area, n = 52).
YEAR – Sampling years, 1996 to 2008.
SEASON – Season, a factor: Fall (n = 351) or Spring (n = 351).

```
summary(sampling, maxsum=27)
```

sediment (702x5)
MOIST, SAND, SILTCLAY, TC, TN
MOIST – Sediment moisture content in percent.
SAND – Sand content in percent by mass.
SILTCLAY – Silt-clay content in percent by mass.
TC – Total carbon content in percent.
TN – Total nitrogen content in percent.

waterquality (702x5)
CONDUCT, DO, PH, SALINITY, WTEMP
CONDUCT – Conductivity in µmho/cm, US equivalent to µS/cm.

```
#       DO – Dissolved oxygen in ppm, US equivalent to mg/L.
#       PH – pH of water sample.
#       SALINITY – In practical salinity units (PSU), equivalent to parts per thousand (‰).
#       WTEMP – Water temperature in Celsius (ºC).

# xy (27x2)
# LATITUDE and LONGITUDE in decimal degrees for each of the 27 sampling sites.
# The original ID tags of the 27 sites are found in vector rownames(xy).

# Please note the following decisions that were made in order to produce the data tables used in
# the exercises that follow.

# 1) The Chesapeake Bay Benthic Monitoring Program includes both FIXED and RANDOM
# sites. FIXED sites were sampled every year whereas RANDOM sites changed from year to
# year. We only included the FIXED sites in our data tables.

# 2) While the monitoring program started in 1995 and is ongoing, we decided to restrict our
# analyses to calendar years for which both a Spring (May) and Fall (late August to early
# October) sampling were conducted. The dataset thus covers 13 years and 26 sampling
# campaigns, spawning from Spring of 1996 to Fall of 2008.

# 3) Among the environmental variables available about the sediment, we removed Total
# Inorganic Carbon (TIC) and Total Organic Carbon (TOC) from the sediment file because no
# data were available for 1996.

# 4) Although the Data Dictionary states that SALINITY was measured in Practical Salinity
# Units (PSU), data files for 1997 report SALINITY in Parts Per Thousand (PPT). We took this
# to be a data entry error and merged the data accordingly.

# 5) Dissolved oxygen in the water column was available both in Parts Per Million (DO) and as
# percent saturation (DO_PSAT). We elected to use only DO due to the fairly large number (16)
# of missing values for variable DO_PSAT.

# 6) For one STATION/SAMPLE_DATE combination (Station 74 on 05/30/2000), the sum of
# SAND and SILTCLAY granulometric fractions was greater then 100%. We rescaled these
# values for their sum to be 100%.

# 7) A total of 8 measurements were missing in the environmental data tables: 5 for water quality
# (all 5 measurements for site 68 on 05/17/2000) and 3 for sediment (MOIST for site 22
# on 09/10/2007, and TC and TN for site 26 on 05/10/1999). In each case, we estimated the
# empty cell using the mean value of the variable at the same site during the same season,
# computed over the year interval (1996 to 2008) considered here.

# 8) Within the Chesapeake Bay Benthic Monitoring Program, three replicate faunal samples
# were scheduled on each sampling occasion. In the fauna data frame, all available samples from
# a given sampling occasion were summed. However, for some rare sampling occasions, only 2,
# or even only 1, sample was available. This is not a big concern here as all analyses will be
# conducted on Hellinger-transformed faunal abundances, where each faunal data vector is first
# transformed into relative abundances, then the values are square-rooted. A total of 8 samples,
# from an expected total of 2106, were missing due to various field or laboratory mishaps. These
# are: samples number 2 and 3 for site 79, sample number 1 for site 68, and sample number 1 for
# site 23 in the Spring of 1998; sample number 1 for site 1 in the Spring of 1999; sample number
# 3 for site 26 in the Fall of 1999; sample number 3 for site 79 in the Spring of 2001; and,
# sample number 3 for site 79 in the Fall of 2008.

# 1.2. PLOT A ROUGH MAP OF THE SITES IN CHESEAPEAKE BAY

plot(xy[,c(2,1)], xlab="Longitude W", ylab="Latitude N", asp=1)
text(xy[,c(2,1)], labels=rownames(xy), pos=4)
```

```
# 1.3. PLOT A NICER MAP OF THE SITES WITH CHESAPEAKE BAY BACKGROUND
#       using RgoogleMaps (Appendix A3, Figure A3.2)

# install.packages("RgoogleMaps", dependencies=TRUE)
require(RgoogleMaps)

# Get the map background from the Google server.
# You will need to be connected to the Internet to fetch the map background from a
# Google server.  Choices for the type of background are:
# maptype = c("roadmap", "mobile", "satellite", "terrain", "hybrid",
#                   "mapmaker-roadmap", "mapmaker-hybrid")
MapBackground(lat=xy$LATITUDE, lon=xy$LONGITUDE, destfile= "bckg", maptype =
"terrain")

# Load the map to an R object
my.map <- ReadMapTile(destfile="bckg")

# Add points at the site coordinates
PlotOnStaticMap(my.map, lat=xy$LATITUDE, lon=xy$LONGITUDE, cex=1.2, col="red",
pch=19)

# Add site labels at positions offset from the site points
TextOnStaticMap(my.map, xy$LATITUDE+0.0005*abs(xy$LATITUDE),
xy$LONGITUDE+0.0005*abs(xy$LONGITUDE), labels=rownames(xy), add=T, col="black")

# Overlapping labels may be touched-up using a graphics editor such as Inkscape (free).

# Close the graphic window before proceeding, otherwise the next plots will be cropped.
# Either close the window manually, or type: dev.off() in the console.

# ========
```

2. LOAD THE NECESSARY PACKAGES, PREPARE THE DATA MATRICES

2.1. LOAD THE R PACKAGES NECESSARY FOR ANALYSIS

```
library(ade4)
library(adespatial)
library(vegan)
```

2.2. LOAD USEFUL FUNCTIONS

```
# Load file "R_functions_for_Practicals.txt", which contains three R functions written for
# these Practicals, using the "Source R Code..." menu for Windows clients
# or the "Source File..." menu for MacOS X clients.
```

2.3. IS THERE A LINEAR TREND IN TEMPERATURE AT ANY OF THE SITES?

```
# For each site, regress water temperature data against sampling dates.
# Function temperature.trend() is in file "R_functions_for_Practicals.txt" loaded in section 2.2.

    (res <- temperature.trend(waterquality, rownames(xy), sampling))

# Are any of the regression coefficients significant?

# Because temperature is an important environmental variable in determining the structure of
# invertebrate communities, the presence of a temporal trend in temperature (over the sampling
# years) may suggest the presence of an induced temporal trend in the community data.

# The presence of such a trend in the faunal data can be checked by computing an RDA of the
# community data over time. This will be done in section 3.2 of these Practicals for the site 40
# data prior to MEM modelling.

# Environmental variables other than temperature may be driving community structure in
# ecological time series. Drift may also generate trends in time series (Appendix A1).

# ========
```

3. TIME SERIES ANALYSIS OF A SINGLE SITE: THE SITE 40 DATA

3.1. SELECT SITE 40 AMONG THE 27 SAMPLING SITES

```
# Site 40 is located in the upper (brackish) course of the Potomac River, which constitutes the
# border between Maryland and Virginia in the USA. Mean salinity at site 40 over the 13 years
# period was 2.3 PSU.

curr.site <- 40
sampling.40 <- sampling[sampling$STATION == curr.site, ]
sediment.40 <- sediment[sampling$STATION == curr.site, ]
waterquality.40 <- waterquality[sampling$STATION == curr.site, ]
fauna.40 <- fauna[sampling$STATION == curr.site, ]
fauna.40 <- fauna.40[ , colSums(fauna.40)!=0]       # (26 sampling units x 36 taxa)

# Quick analysis: identify the temporal structure of the sampling design
dates.40 <- sampling.40$SAMPLE_DATE
plot(dates.40, rep(1,26), yaxt="n", ylab="")
```

3.2. TEMPORAL EIGENFUNCTION ANALYSIS OF COMMUNITY DATA FROM SITE 40

```
# Hellinger transformation of the faunal data prior to analysis by RDA
# Transformation: square root of the relative abundances by rows
# See reference to Legendre & Gallagher (2001) in ?decostand
fauna.hel.40 <- decostand(fauna.40, method="hellinger")

# Is there a linear temporal trend in the response data?
fauna.trend <- rda(fauna.hel.40, sampling.40$SAMPLE_DATE)
anova(fauna.trend, step=10000, perm.max=10000)            # See note[1]
RsquareAdj(fauna.trend)
# Examine the R-square and the p-value to decide if there is a significant temporal trend.
# Detrend the response data (i.e. compute residuals) only if the trend is [highly] significant.
# The MEM and AEM methods are equally suitable to analyze the temporal structure of the data.
```

 # 3.2.1. dbMEM analysis

```
# Construct the dbMEM eigenfunctions. Generate all dbMEM eigenfunctions
# As used here, function dist() computes the number of days between sampling occasions
time.mem.40 <- dbmem(dist(dates.40), MEM.autocor="all")
summary(time.mem.40)
attributes(time.mem.40)
dim(time.mem.40)

# Which dbMEM model positive temporal correlation?
attr(time.mem.40,"value")
moran.40 <- moran.randtest(time.mem.40, nrepet = 99)
plot(moran.40$obs, attr(time.mem.40, "values"), xlab = "Moran's I", ylab = "Eigenvalues")
plot(moran.40$obs, xlab="MEM rank", ylab="Moran's I")
abline(h=-1/(nrow(sampling.40) - 1), col="red")
```

[1] There are many functions called anova() in R. They do not compute an Analyse Of Variance. Instead, they compute different tests of significance and produce anova-like tables. R automatically selects the anova function that corresponds to the class of the object. The class associated with object "fauna.trend" is found with class(fauna.trend) [result: "rda" "cca"]. For this object, function anova.cca(), which performs a permutation test of the RDA statistic, was called by R and used for the analysis.

```
# Compute the redundancy analysis (RDA) of the fauna by the dbMEM eigenvectors modelling
# positive temporal correlation
time.mem.40.pos <- time.mem.40[,attr(time.mem.40,"value")>0]
fauna.mem.40.pos <- rda(fauna.hel.40 ~ ., time.mem.40.pos)
anova(fauna.mem.40.pos, step=10000, perm.max=10000)
RsquareAdj(fauna.mem.40.pos)
anova(fauna.mem.40.pos, by="axis")# Examine the R-square, the adjusted R-square and the p-value.
# The model based upon the 12 MEM modelling positive temporal correlation is significant.
# It produces two significant canonical axes.

# Compute the redundancy analysis (RDA) of the fauna by the dbMEM eigenvectors modelling
# negative temporal correlation
time.mem.40.neg <- time.mem.40[,attr(time.mem.40,"value")<0]
fauna.mem.40.neg <- rda(fauna.hel.40 ~ ., time.mem.40.neg)
anova(fauna.mem.40.neg, step=10000, perm.max=10000)
RsquareAdj(fauna.mem.40.neg)
anova(fauna.mem.40.neg, by="axis")# Examine the R-square, the adjusted R-square and the p-value
# The model containing the 13 MEM modelling negative temporal correlation is not globally
# significant. It produces a significant canonical axis, however, and that axis is worth looking at.

# The canonical axes (MEM models) are all orthogonal to one another.
# Property of orthogonal vectors: their cross-product is 0. Example:
tmp <- as.matrix(summary(fauna.mem.40.pos)$constraints)
round(t(tmp) %*% tmp, 4)

# Plot the RDA axes values of the three canonical models (Appendix A3, Figure A3.3)
par(mfrow=c(3,1))
# The positive ones
for(i in 1:2) {
      plot(dates.40, scores(fauna.mem.40.pos, display="lc", choice=i), type="b", pch=19, main =
paste("RDA axis", i, ", positive temporal correlation model"), xlab="Date", ylab="RDA axis")
      }
# The single significant negative axis
      plot(dates.40, scores(fauna.mem.40.neg, display="lc", choice=1), type="b", pch=19, main =
paste("RDA axis", 1, ", negative temporal correlation model"), xlab="Date", ylab="RDA axis",
col="red", col.main="red")
par(mfrow=c(1,1))

# --------

# Select the dbMEM that are useful for modelling
sel.mem.40 <- forward.sel(fauna.hel.40, time.mem.40, nperm=9999, alpha=0.10)
sel.mem.40
# Do not include variables with p-values that are much larger than 0.05

# Three models: all selected dbMEM, then those modelling positive and negative correlation
mem.select <- sort(sel.mem.40$order[sel.mem.40$pval<=0.05])
# This selection is the same as: mem.select <- c(2,3,5,6,8,11,21,25)
mem.select.pos <- c(2,3,5,6,8,11)
mem.select.neg <- c(21,25)

# Plot the selected dbMEM to see what they look like[2] (Appendix A3, Figure A3.4)
par(mfrow=c(4,2))
```

[2] In eigenvectors (which include eigenfunctions) and ordination axes computed on different computers or different software, all signs may be reversed, producing graphs that are mirror images of graphs produced on other software or computers. Such inversions have no effect on the interpretation of ordination diagrams or eigenfunctions.

```
# The positive ones
for(i in 1:6) {
        # plot(dates.40, time.mem.40$vectors[,mem.select.pos[i]], type="b", pch=19, main =
paste("Positive", mem.select.pos[i]), xlab="Date", ylab="dbMEM")
          plot(dates.40, time.mem.40[,mem.select.pos[i]], type="b", pch=19, main =
paste("Positive", mem.select.pos[i]), xlab="Date", ylab="dbMEM")
}
# The negative ones
for(i in 1:2) {
# plot(dates.40, time.mem.40$vectors[,mem.select.neg[i]], type="b", pch=19, main =
paste("Negative", mem.select.neg[i]), xlab="Date", ylab="dbMEM", col="red", col.main="red")
}
plot(dates.40, time.mem.40[,mem.select.neg[i]], type="b", pch=19, main = paste("Negative",
mem.select.neg[i]), xlab="Date", ylab="dbMEM", col="red", col.main="red")
}
par(mfrow=c(1,1))

# Analysis based on the selected dbMEM eigenvectors only
# Compute RDA of the fauna by the selected dbMEM in each group (positive, negative)
fauna.mem.40.pos.sel <- rda(fauna.hel.40 ~ .,as.data.frame(time.mem.40[,mem.select.pos]))
anova(fauna.mem.40.pos.sel, by="axis")

fauna.mem.40.neg.sel <- rda(fauna.hel.40 ~ ., as.data.frame(time.mem.40[,mem.select.neg]))
anova(fauna.mem.40.neg.sel, by="axis")

# Plot the RDA axes of the significant canonical dbMEM models produced by the selected MEM
# (Appendix A3, Figure A3.5)
par(mfrow=c(3,1))
# The positive ones
for(i in 1:2) {
      plot(dates.40, scores(fauna.mem.40.pos.sel, display= "lc", choices=i), type="b", pch=19,
main = paste("RDA axis", i, ", positive temporal correlation model"), xlab="Date", ylab="RDA
axis")
      }
# The single significant negative axis
      plot(dates.40, scores(fauna.mem.40.neg.sel, display= "lc", choices=1), type="b", pch=19,
main = paste("RDA axis", 1, ", negative temporal correlation model"), xlab="Date", ylab="RDA
axis", col="red", col.main="red")
par(mfrow=c(1,1))

# --------

# Variation along the significant MEM model axes can be interpreted by stepwise selection in
regression, using the available environmental variables assembled in file envir.40.

# Among the explanatory variables, we added annual Principal Component (PC)-based indices
# of the North Atlantic Oscillation (NAO), obtained from the National Center for Atmospheric
# Research Climate Data Guide (https://climatedataguide.ucar.edu/)

# Values of the North Atlantic Oscillation (NAO), yearly data from 1996 to 2008
NAO <- c(-1.07, -1.07, 0.03, 0.03, -0.29, -0.29, 0.55, 0.55, 0.63, 0.63, -0.57, -0.57, 0.35, 0.35,
0.32, 0.32, 0.10, 0.10, -0.76, -0.76, -0.26, -0.26, 0.54, 0.54, 0.08, 0.08)

# Analyse MEM model 1 representing positive temporal correlation by stepwise regression
# (variable target.1)
envir.40 <- cbind(sediment.40, waterquality.40, sampling.40$SEASON, NAO)
target.1 <- summary(fauna.mem.40.pos.sel)$constraints[,1]
m0 <- lm(target.1 ~ 1, data= envir.40)
mtot <-
```

```
lm(target.1 ~ CONDUCT + DO + PH + SALINITY + WTEMP + MOIST + SAND +
SILTCLAY + TC + TN + NAO, sampling.40$SEASON, data= envir.40)
res.step.1 <- step(m0, scope=formula(mtot), direction="both", trace=-1)
summary(res.step.1)

# NAO and TN are selected and significant at the 0.05 level.
# Repeat the regression using only these two variables:
res.model1 <- lm(target.1 ~ NAO + TN, data= envir.40)
summary(res.model1)

# Analyse MEM model 2 representing positive temporal correlation by stepwise regression
# (variable target.2)
target.2 <- summary(fauna.mem.40.pos.sel)$constraints[,2]
m0 <- lm(target.2 ~ 1, data= envir.40)
mtot <-
lm(target.2 ~ CONDUCT + DO + PH + SALINITY + WTEMP + MOIST + SAND +
SILTCLAY + TC + TN + NAO + sampling.40$SEASON, data= envir.40)
res.step.2 <- step(m0, scope=formula(mtot), direction="both", trace=-1)
summary(res.step.2)   # TN is the only explanatory variable selected, but it is not significant.

# Analyse the single axis representing negative temporal correlation by stepwise regression
# (variable target.3)
target.3 <- summary(fauna.mem.40.neg.sel)$constraints[,1]
m0 <- lm(target.3 ~ 1, data= envir.40)
mtot <-
lm(target.3 ~ CONDUCT + DO + PH + SALINITY + WTEMP + MOIST + SAND +
SILTCLAY + TC + TN + NAO + sampling.40$SEASON, data= envir.40)
res.step.3 <- step(m0, scope=formula(mtot), direction="both", trace=-1)
summary(res.step.3)

# Sampling season is selected and significant at the 0.05 level.
# Repeat the regression using only that variable:
model.4 <- lm(target.3 ~ SEASON, data = sampling.40)
```

 # 3.2.2. AEM analysis

```
# Construct the AEM eigenfunctions. Generate all AEM eigenfunctions.

# Construct a vector of weigths for the edges, each representing the easiness of exchange
# between adjacent dates (nodes).
# The 'max.d' value used here to scale the distances through weighting function 1
# (?weight.time) is the smallest distance for which no significant autocorrelation is found in the
# multivariate Mantel correlogram (correlog.40, section 3.4 of this document; Fig. A3.8).
# if you have the AEM library installed, you can use the following line to compute the weights
# weights <- weight.time(dates.40, alpha=2, max.d=522)
# Otherwise, use this quick fix or use equal weights ('w=NULL')
weights <- c(0.9280692, 0.8041683, 0.9370202, 0.8007626, 0.9539643, 0.7669441,  0.9435710,
0.7632301, 0.9435710, 0.8108916, 0.9379780, 0.8041683,  0.9331153, 0.7779099, 0.9661778,
0.7284831, 0.9523055, 0.7650908,  0.9435710, 0.7903693, 0.9555937, 0.7850736, 0.9290931,
0.8041683,  0.9290931)

# Construct the AEMs themselves
aem.40.out <- aem.time(26, w=weights, moran=TRUE, plot.moran=TRUE)
aem.40.out$Moran
# How many AEM have positive Moran's I > E(I) and model positive temporal correlation?
# Current version of the function not computing Moran's I, even if asked to, will be fixed.
#  => The AEMs c(1:11, 14) model positive temporal correlation; the other c(12, 13, 15:25)
model negative temporal correlation
```

```
# Compute the redundancy analysis (RDA) of the fauna by the matrix of AEM eigenvectors
# modelling positive temporal correlation
fauna.aem.40.pos <- rda(fauna.hel.40, aem.40.out$aem[,  c(1:11, 14)])
anova(fauna.aem.40.pos)
RsquareAdj(fauna.aem.40.pos)
# Examine the R², the R²adj and the p-value.
# You may want to recompute the AEM eigenfunctions under the assumption that the dates are
# equidistant, using aem.time with option 'w=NULL', and compare the R² coefficients.

# Compute the redundancy analysis (RDA) of the fauna by the matrix of AEM eigenvectors
# modelling negative temporal correlation
fauna.aem.40.neg <- rda(fauna.hel.40, aem.40.out$aem[,  c(12, 13, 15:25)])
anova(fauna.aem.40.neg)
RsquareAdj(fauna.aem.40.neg)
# Examine the R², the R²adj and the p-value.

# Select the AEM that are useful for modelling
sel.aem.40 <- forward.sel(fauna.hel.40, aem.40.out$aem, nperm=9999, alpha=0.10)
sel.aem.40
# Do not include selected variables with p-values that are much larger than 0.05
# Compare to the results of forward selection of AEM assuming equidistant observations.

# Three models: all AEM selected, then those modelling positive and negative correlation
aem.select <- sort(sel.aem.40$order[sel.aem.40$pval<=0.08])
# This selection is the same as: aem.select <- c(1,2,3,6,10,15,21,24,25)
aem.select.pos <- c(1,2,3,6,10)
aem.select.neg <- c(15,21,24,25)

# Plot the selected AEM to see what they look like (Appendix A3, Figure A3.6)
par(mfrow=c(3,3))
# The positive ones
for(i in 1:5) {
      plot(dates.40, aem.40.out$aem[,aem.select.pos[i]], type="b", pch=19, main =
paste("Positive", aem.select.pos[i]), xlab="Date", ylab="AEM")
      }
# The negative ones
for(i in 1:4) {
      plot(dates.40, aem.40.out$aem[,aem.select.neg[i]], type="b", pch=19, main =
paste("Negative", aem.select.neg[i]), xlab="Date", ylab="AEM", col="red", col.main="red")
      }
par(mfrow=c(1,1))

# Compute RDA of the fauna by the selected AEM in each group (positive, negative), p <= 0.08
fauna.aem.40.pos.5 <- rda(fauna.hel.40~ ., as.data.frame(aem.40.out$aem[,aem.select.pos]))
anova(fauna.aem.40.pos.5, by="axis")
RsquareAdj(fauna.aem.40.pos.5)
fauna.aem.40.neg.4 <- rda(fauna.hel.40~ ., as.data.frame(aem.40.out$aem[,aem.select.neg]))
anova(fauna.aem.40.neg.4, by="axis")
RsquareAdj(fauna.aem.40.neg.4)

# Plot the RDA axes of the significant AEM models produced by the selected AEM
# (Appendix A3, Figure A3.7)
par(mfrow=c(3,1))
# The positive ones
for(i in 1:2) {
      plot(dates.40, scores(fauna.aem.40.pos.5, display="lc", choices=i), type="b", pch=19, main
= paste("RDA axis", i, ", positive temporal correlation model"), xlab="Seasons within years
1996-2008", ylab="RDA axis")
      }
# The single significant negative axis
```

```
plot(dates.40, scores(fauna.aem.40.neg.4, display="lc", choices=1), type="b", pch=19, main
= paste("RDA axis", 1, ", negative temporal correlation model"), xlab="Seasons within years
1996-2008", ylab="RDA axis", col="red", col.main="red")
par(mfrow=c(1,1))

# Compare the MEM and AEM models. If one looks like a mirror image of the other, see
# footnote 2 a few pages back. If it is necessary to produce a mirror-image plot for the
# comparison, add a minus sign in the plotting script in front of the model to be drawn.

# --------

# Compare the MEM and AEM eigenfunctions through RV coefficients

# RV coefficient between MEM and AEM modeling positive temporal correlation
RV.pos <- RV.rtest(as.data.frame(time.mem.40.pos), as.data.frame(aem.40.out$aem[,  c(1:11,
14)]))

RV.pos

# RV coefficient between MEM and AEM modeling negative temporal correlation
RV.neg <- RV.rtest(as.data.frame(time.mem.40.neg), as.data.frame(aem.40.out$aem[,  c(12, 13,
15:25)]))
RV.neg
```

 # 3.2.3. Scalogram of the dbMEM eigenfunctions

```
# Compute semipartial R² for each dbMEM modelling positive temporal correlation, separately
# Function R2.by.variable() is found in file "R_functions_for_Practicals.txt" loaded in sect. 2.2.

R2.out <- R2.by.variable(fauna.hel.40, time.mem.40[,1:12], scale.Y=FALSE)
R2.out
# The output table lists the semipartial R² computed separately for each dbMEM, the F-statistics
# and the p-values of the semipartial test. The semipartial R² will be used in the scalogram. The
# test displayed in the table is very conservative because it tests the contribution of each MEM
# eigenfunction above and beyond all other MEM variables in the analysis.

# In the scalogram, the squares represent the semipartial R² coefficients associated with each
# MEM eigenfunction. We chose to highlight (black squares) the dbMEM that significantly
# contribute sequentially to the explanation of the faunal response data, as in Legendre &
# Legendre (2012, Fig. 14.5). These significant dbMEM are found in the output list of the
# forward.sel() function. The scalogram produced here only concerns the dbMEM variables
# modelling positive temporal correlation.

# Plot the scalogram (Appendix A3, Figure A3.8)
# Open squares: pch=22; filled black or white squares: pch=15.
plot(1:12, R2.out[,1], type="o", pch=22, cex=1.0, xlim=c(1,12), ylim=c(0,0.25), xlab="dbMEM
1-12", ylab="R-square", main="Scalogram of positively correlated dbMEM, Chesapeake site
40")
points(1:12, R2.out[,1], pch=15, col="white", cex=0.8)
points(mem.select.pos, R2.out[mem.select.pos,1], pch=15, cex=1.0)
```

3.3. VARIATION PARTITIONING

```
#       involving environmental variables and dbMEM eigenfunctions (Figure 5 of the paper)

# Selection of the site 40 sediment variables
(res.sel1 <- forward.sel(fauna.hel.40, sediment.40, nperm=9999, alpha=0.10))
# You will receive an error message because no sediment variable was selected
```

```
# Selection of water quality variables
(res.sel2 <- forward.sel(fauna.hel.40, waterquality.40, nperm=9999, alpha=0.10))
# A single variable was selected: SALINITY (variable 4)

# Variation partitioning: waterquality, positive MEM, negative MEM
res.part <- varpart(fauna.hel.40, waterquality.40[,4], time.mem.40[,mem.select.pos],
time.mem.40[,mem.select.neg])
res.part
plot(res.part, digits=2)

# Example of partial test: test partial effect of waterquality while controlling for MEM
mod1 <- rda(fauna.hel.40, waterquality.40[,4], time.mem.40[,mem.select])
anova(mod1, step=1000, perm.max=1000)
RsquareAdj(mod1)
# Then, compare the adjusted R-square ($adj.r.squared) to line [a] of the varpart result.
# This illustrates the calculations done by function varpart().

# Example of partial test: test MEM.pos while controlling for (waterquality and MEM.neg)
mod2 <- rda(fauna.hel.40, time.mem.40[,mem.select.pos], cbind(waterquality.40[,4],
time.mem.40[,mem.select.neg]))
anova(mod2, step=1000, perm.max=1000)
RsquareAdj(mod2)

# Example of partial test: test MEM.neg while controlling for (waterquality and MEM.neg)
mod3 <- rda(fauna.hel.40, time.mem.40[,mem.select.neg], cbind(waterquality.40[,4],
time.mem.40[,mem.select.pos]))
anova(mod3, step=1000, perm.max=1000)
RsquareAdj(mod3)


# 3.4. MULTIVARIATE CORRELOGRAM
#       See “Numerical ecology with R” (2011) and “Numerical ecology” (2012) for details.

# Multivariate Mantel correlogram  (Appendix A3, Figure A3.9)
# See "Numerical Ecology" (2012), Section 13.1.6
# See "Numerical Ecology with R" (2011), p. 235
correlog.40 <- mantel.correlog(dist(fauna.hel.40), XY=dates.40, n.class=26)
plot(correlog.40)


# 3.5. TIME-CONSTRAINED CLUSTERING: MULTIVARIATE REGRESSION TREE
#       See “Numerical ecology with R” (2011) and “Numerical ecology” (2012) for details.

# Use this method to identify one or several breakpoints in a data series
# See "Numerical Ecology" (2012), Section 12.6.4
# See "Numerical Ecology with R" (2011), Section 4.11.5, for details about graph interpretation
library(mvpart)

# Generate a "time.seq" variable containing the integers 1 to 26
time.seq = as.data.frame(1:26)          # Generate a "time.seq" variable containing integers 1 to 26
colnames(time.seq) = "time.seq"
part.res <- mvpart(data.matrix(fauna.hel.40) ~ time.seq, data=time.seq, xv="pick", xvmult=100)

# Click on the red dot of the graph to obtain the tree.
# Two groups represent the solution with the smallest CVRE (“best” in that sense).
# Find the observations in which group:
part.res$where

# ========
```

4. TWO-WAY MANOVA BY PARTIAL RDA

This section shows how to carry out a two-way multivariate ANOVA (called MANOVA) for community composition data. The two factors of interest are year and season.

4.1. TWO-WAY TEMPORAL MANOVA OF FIVE SITES

```
# Test whether the year and season factors can explain a significant fraction of the multivariate
# dispersion within a group of geographically close sites. Sites 22, 23, 201, 202 and 203 are
# considered replicates in this analysis. Distances between these sites ranges from 2.8 to 15.9 km
# (mean of 7.9 km); they are thus far enough from one another that the faunal data should not
# be pseudoreplicated.

# Select a subset of five sites for multivariate MANOVA
curr.siteS <- c(201, 202, 203, 22, 23)

# These five sites are clustered in the northern portion of Chesapeake Bay near the city of
# Baltimore, Maryland. Run the code to see the map, which is not shown in Appendix A3
plot(xy[,c(2,1)], xlab="Longitude W", ylab="Latitude N", asp=1)
points(xy[rownames(xy) %in% curr.siteS,c(2,1)], pch=19, col= "red")
text(xy[,c(2,1)], labels=rownames(xy), pos=4)

# Extract sampling and fauna data for the sites in set curr.siteS
sampling.manova1 <- sampling[sampling$STATION %in% curr.siteS, ]
fauna.manova1 <- fauna[sampling$STATION %in% curr.siteS, ]

# Remove the "empty" taxa
fauna.manova1 <- fauna.manova1[ , colSums(fauna.manova1)!=0]   # (130x70)

# Remove unused levels from sampling data
# Function drop.levels() is found in file "R_functions_for_Practicals.txt" loaded in section 2.2.
sampling.manova1 <- drop.levels(sampling.manova1)

# What do we have on hand?
dim(fauna.manova1)
dim(sampling.manova1)
summary(sampling.manova1)

# Create a factor for YEAR by transforming vector sampling.manova1$YEAR
year.fac <- as.factor(sampling.manova1$YEAR)
year.fac

# Create a factor for SEASON
season.fac <- sampling.manova1$SEASON
season.fac

# Make sure that the factors are balanced, i.e. same number of observations in each cell
table(season.fac, year.fac)

# Create Helmert contrasts for the factors and their interaction.
# Explanation – In two-way anova, Helmert contrasts are used to code for the main factors A and
# B. Variables representing the interaction are generated by computing the products of all
# Helmert variables coding for A by all variables coding for B. As a result, the set of variables
# coding for the interaction is orthogonal to the set of variables coding for A and for B. The
# fractions of variation explained by A, B and the interaction are thus linearly independent of
# one another.

year.season.helm <- model.matrix(~ season.fac * year.fac,
    contrasts=list(season.fac="contr.helmert", year.fac="contr.helmert"))
# year.season.helm   # If you want to look at the Helmert contrasts
```

```
# Property 1 of Helmert contrasts: all variables should sum to 0
apply(year.season.helm[, 2:ncol(year.season.helm)], 2, sum)

# Property 2: the cross products (scalar products) of the Helmert contrasts should all be 0,
# showing that they are orthogonal to one another
res <- t(year.season.helm[,-1]) %*% year.season.helm[,-1]
head(res)   # Check that the non-diagonal terms of matrix "res" are all 0.

# Transform the species abundance data using the Hellinger transformation
fauna.hel.manova1 <- decostand(fauna.manova1, method="hellinger")

# --------

# Before proceeding with MANOVA, we must test whether there is homogeneity of the
# multivariate within-group covariance matrices

# Compute the Hellinger distance matrix from the transformed data
fauna.hel.manova1.D1 <- dist(fauna.hel.manova1)

# We cross the season & year factors to create the groups
year.season.fac <- as.factor(paste(year.fac, season.fac, sep="."))
year.season.fac

# Test of homogeneity of the multivariate within-group covariance matrices
fauna.hel.manova1.MHV <- betadisper(fauna.hel.manova1.D1, year.season.fac)
permutest(fauna.hel.manova1.MHV)

# If p < 0.05, the test rejects H0: the multivariate within-group covariance matrices are
# homogeneous. If H0 is not rejected, we can proceed with the analysis of variance.

# --------

# Multivariate analysis of variance

# Step 1. Check if there is a significant interaction between sampling YEAR and SEASON.
# We use the interaction terms as explanatory variables, the factors themselves as covariables.
# Do not use column 1 (Intercept). Look at colnames(year.season.helm) to find out which
# columns represent the different terms (factors and interactions). It is important to choose the
# correct columns as explanatory variables and covariables in the three analyses that follow.
season.year.rda1 <- rda(fauna.hel.manova1, year.season.helm[, 15:26], year.season.helm[, 2:14])
anova(season.year.rda1, step=1000, perm.max=1000, model="direct")
RsquareAdj(season.year.rda1)

# Is the interaction significant? If it is, MANOVAs of the YEAR factor should be computed for
# each season separately, and conversely.

# Step 2. Can factor SEASON explain a significant portion of the multivariate dispersion?
# Sampling YEAR and interaction are used as covariables.
season.rda1 <- rda(fauna.hel.manova1, year.season.helm[, 2], year.season.helm[, 3:26])
anova(season.rda1, step=1000, perm.max=1000, strata=year.fac, model="direct")
RsquareAdj(season.rda1)      # Measure of effect size

# Step 3. Can factor sampling YEAR explain a significant portion of the multivariate dispersion?
# SEASON and interaction are used as covariables.
year.rda1 <- rda(fauna.hel.manova1, year.season.helm[, 3:14], year.season.helm[, c(2, 15:26)])
anova(year.rda1, step=1000, perm.max=1000, strata=season.fac, model="direct")
RsquareAdj(year.rda1)          # Measure of effect size

# Which factor explains the largest fraction of the multivariate faunal dispersion?
```

```
# ========

# 4.2. SPACE-TIME STUDY: VARIABILITY AMONG YEARS AND REGIONS, ONE SEASON AT A TIME

# Sites 43, 44 and 47 are in the Potomac River estuary in the south-west of Chesapeake Bay,
# sites 201, 202 and 203 are in the inlet near Baltimore in the north. The 6 sites form two groups
# with three replicates each. We will test whether the year and site.group factors can explain the
# multivariate dispersion between two groups of geographically distant sites during each season.

# Select a subset of sites for multivariate MANOVA
curr.siteS <- c(43, 44, 47, 201, 202, 203)

# Plot sites on a simple map; run the code to see the map
plot(xy[,c(2,1)], xlab="Longitude W", ylab="Latitude N", asp=1)
points(xy[rownames(xy) %in% curr.siteS,c(2,1)], pch=19, col= "red")
text(xy[,c(2,1)], labels=rownames(xy), pos=4)

# Extract sampling and fauna for curr.siteS
fauna.6sites <- fauna[sampling$STATION %in% curr.siteS, ]
sampling.6sites <- sampling[sampling$STATION %in% curr.siteS, ]


# 4.2.1. MANOVA of the spring surveys
#          sites 43, 44, 47, 201, 202 and 203

# Select the spring sampling units.
fauna.manova2 <- fauna.6sites[sampling.6sites$SEASON == "Spring", ]
sampling.manova2 <- sampling.6sites[sampling.6sites$SEASON == "Spring", ]

# Remove the "empty" taxa
fauna.manova2 <- fauna.manova2[ , colSums(fauna.manova2)!=0]

# Remove unused levels from sampling data
# Function drop.levels() is found in file "R_functions_for_Practicals.txt" loaded in section 2.2.
sampling.manova2 <- drop.levels(sampling.manova2)

# What do we have on hand?
dim(fauna.manova2)
dim(sampling.manova2)
summary(sampling.manova2)

# Create a factor for YEAR (identical to year.fac generated in 4.1)
year.fac <- as.factor(sampling.manova2$YEAR)
year.fac

# Create a factor describing the two groups of sites.
site.vec <- as.character(sampling.manova2$STATION)
site.vec[site.vec %in% c("43","44","47")] <- "43.44.47"
site.vec[site.vec %in% c("201","202","203")] <- "201.202.203"
site.group.fac <- as.factor(site.vec)
site.group.fac

# Make sure that the factors are balanced, i.e. same number of observations in each cell
table(site.group.fac, year.fac)

# Helmert contrasts for the factors and interaction
year.group.helm <- model.matrix(~ site.group.fac * year.fac,
    contrasts=list(site.group.fac="contr.helmert", year.fac="contr.helmert"))
# year.group.helm   # If you want to look at the Helmert contrasts
```

```
# Property 1 of Helmert contrasts: all variables should sum to 0
apply(year.group.helm[, 2:ncol(year.group.helm)], 2, sum)

# Property 2: the cross products (scalar products) of the Helmert contrasts should all be 0,
# showing that they are orthogonal to one another
res <- t(year.group.helm[,-1]) %*% year.group.helm[,-1]
head(res)   # Check that the non-diagonal terms of matrix "res" are all 0.

# Transform the species abundance data using the Hellinger transformation
fauna.hel.manova2 <- decostand(fauna.manova2, method="hellinger")

# --------

# Test homogeneity of the multivariate within-group covariance matrices

# Compute the Hellinger distance matrix from the transformed data
fauna.hel.manova2.D1 <- dist(fauna.hel.manova2)

# We cross the year and site.group factors to create the groups
year.site.group.fac <- as.factor(paste(year.fac, site.group.fac, sep="."))

# Test of homogeneity of the multivariate within-group covariance matrices
fauna.hel.manova2.MHV <- betadisper(fauna.hel.manova2.D1, year.site.group.fac)
permutest(fauna.hel.manova2.MHV)

# --------

# Multivariate analysis of variance of the spring surveys

# Step 1. Check if there is a significant interaction between the site groups and YEAR.
# We use the interaction terms as explanatory variables, the factors themselves as covariables.
# Do not use column 1 (Intercept). Look at colnames(year.group.helm) to find out which
# columns represent the different terms (factors and interactions). It is important to choose the
# correct columns as explanatory variables and covariables in the three analyses that follow.
site.year.rda2 <- rda(fauna.hel.manova2, year.group.helm[, 15:26], year.group.helm[, 2:14])
anova(site.year.rda2, step=1000, perm.max=1000, model="direct")
RsquareAdj(site.year.rda2)
# Is the interaction significant? If it is, MANOVAs of the YEAR factor should be computed for
# each site group separately, and conversely.

# Step 2. Can factor site.group.fac explain a significant portion of the multivariate dispersion?
# Sampling YEAR and interaction are used as covariables.
site.rda2 <- rda(fauna.hel.manova2, year.group.helm[, 2], year.group.helm[, 3:26])
anova(site.rda2, step=1000, perm.max=1000, strata=year.fac, model="direct")
RsquareAdj(site.rda2)

# Step 3. Can factor sampling YEAR explain a significant portion of the multivariate dispersion?
# Factor site.group.fac and interaction are used as covariables.
year.rda2 <- rda(fauna.hel.manova2, year.group.helm[, 3:14], year.group.helm[, c(2, 15:26)])
anova(year.rda2, step=1000, perm.max=1000, strata=site.group.fac, model="direct")
RsquareAdj(year.rda2)


# 4.2.2. MANOVA of the fall surveys
#        sites 43, 44, 47, 201, 202 and 203

# Select the fall sampling units.
fauna.manova3 <- fauna.6sites[sampling.6sites$SEASON == "Fall", ]
sampling.manova3 <- sampling.6sites[sampling.6sites$SEASON == "Fall", ]
```

```
# Remove the "empty" taxa
fauna.manova3 <- fauna.manova3[ , colSums(fauna.manova3)!=0]

# Remove unused levels from sampling data
# Function drop.levels() is found in file “R_functions_for_Practicals.txt” loaded in section 2.2.
sampling.manova3 <- drop.levels(sampling.manova3)

# What do we have on hand?
dim(fauna.manova3)
dim(sampling.manova3)
summary(sampling.manova3)

# Create a factor for YEAR (identical to year.fac generated in 4.1)
year.fac <- as.factor(sampling.manova3$YEAR)
year.fac

# Create a factor describing the two groups of sites (identical to site.group.fac generated in 4.2.1)
site.vec <- as.character(sampling.manova3$STATION)
site.vec[site.vec %in% c("43","44","47")] <- "43.44.47"
site.vec[site.vec %in% c("201","202","203")] <- "201.202.203"
site.group.fac <- as.factor(site.vec)
site.group.fac

# Make sure that the factors are balanced, i.e. same number of observations in each cell
table(site.group.fac, year.fac)

# Helmert contrasts for the factors and their interaction (identical to year.group.helm in 4.2.1)
year.group.helm <- model.matrix(~ site.group.fac * year.fac,
    contrasts=list(site.group.fac="contr.helmert", year.fac="contr.helmert"))
# year.group.helm   # If you want to look at the Helmert contrasts

# Property 1 of Helmert contrasts: all variables should sum to 0
apply(year.group.helm[, 2:ncol(year.group.helm)], 2, sum)

# Property 2: the cross products (scalar products) of the Helmert contrasts should all be 0,
# showing that they are orthogonal to one another
res <- t(year.group.helm[,-1]) %*% year.group.helm[,-1]
head(res)   # Check that the non-diagonal terms of matrix “res” are all 0.

# Transform the species abundance data using the Hellinger transformation
fauna.hel.manova3 <- decostand(fauna.manova3, method="hellinger")

# --------

# Test homogeneity of the multivariate within-group covariance matrices

# Compute the Hellinger distance matrix from the transformed data
fauna.hel.manova3.D1 <- dist(fauna.hel.manova3)

# We cross the year and site.group factors to create the groups
year.site.group.fac <- as.factor(paste(year.fac, site.group.fac, sep="."))

# Test of homogeneity of the multivariate within-group covariance matrices
fauna.hel.manova3.MHV <- betadisper(fauna.hel.manova3.D1, year.site.group.fac)
permutest(fauna.hel.manova3.MHV)

# --------

# Multivariate analysis of variance of the fall surveys
```

```
# Step 1. Check if there is a significant interaction between the site groups and YEAR.
# We use the interaction terms as explanatory variables, the factors themselves as covariables.
# Do not use column 1 (Intercept). Look at colnames(year.group.helm) to find out which
# columns represent the different terms (factors and interactions). It is important to choose the
# correct columns as explanatory variables and covariables in the three analyses that follow.
site.year.rda3 <- rda(fauna.hel.manova3, year.group.helm[, 15:26], year.group.helm[, 2:14])
anova(site.year.rda3, step=1000, perm.max=1000, model="direct")
RsquareAdj(site.year.rda3)
# Is the interaction significant? If it is, MANOVAs of the YEAR factor should be computed for
# each site group separately, and conversely.

# Step 2. Can factor site.group.fac explain a significant portion of the multivariate dispersion?
# Sampling YEAR and interaction are used as covariables.
site.group.rda3 <- rda(fauna.hel.manova3, year.group.helm[, 2], year.group.helm[, 3:26])
anova(site.group.rda3, step=1000, perm.max=1000, strata=year.fac, model="direct")
RsquareAdj(site.group.rda3)

# Step 3. Can factor sampling YEAR explain a significant portion of the multivariate dispersion?
# Factor site.group.fac and interaction are used as covariables.
year.rda3 <- rda(fauna.hel.manova3, year.group.helm[, 3:14], year.group.helm[, c(2, 15:26)])
anova(year.rda3, step=1000, perm.max=1000, strata=site.group.fac, model="direct")
RsquareAdj(year.rda3)

# Which factor explains the largest fraction of the multivariate faunal dispersion?

# ========
```

5. SPACE-TIME ANALYSIS: LOCAL CONTRIBUTIONS TO BETA DIVERSITY (LCBD)

```
# Local Contributions to Beta Diversity (LCBD indices) are comparative indicators of the
# ecological uniqueness of the sampling units. The LCBD values indicate how much each
# observation contributes to beta diversity compared to a site with average species composition,
# which would have a LCBD value of 0. Sampling units may have large LCBD values for
# different reasons, as explained in the paper.

# Download function beta.div() from Appendix S4 of the Legendre & De Cáceres (2013) paper
# at http://onlinelibrary.wiley.com/doi/10.1111/ele.12141/suppinfo (Supporting information).
# The function itself is in file ele12141-sup-0005-AppendixS4.R, whereas the documentation
# is found in file ele12141-sup-0004-AppendixS4.pdf.


# 5.1. PREPARE A DATA FILE

# Create a simple factor for seasons for the 27 sites, 13 years and 2 seasons, length = 702
season27 <- sampling$SEASON

# 5.2. COMPUTE LCDB INDICES OVER THE 27 SITES, SPRING AND FALL SEPARATELY

beta.out.27.spring <- beta.div(fauna[season27=="Spring",], method="hellinger", nperm=499)
beta.out.27.fall <- beta.div(fauna[season27=="Fall",], method="hellinger", nperm=499)
# Warning – It is better to use 999 permutations. Permutation tests take a bit of time.
# The LCBD indices are available in the $LCBD vector, the permutational p-values in $p.LCBD
signif.27.spring <- which(beta.out.27.spring$p.LCBD <= 0.05)
signif.27.fall <- which(beta.out.27.fall$p.LCBD <= 0.05)

# Copy the LCBD computed for each season to matrices (27 sites x 13 years)
LCBD.27.spring <- matrix(beta.out.27.spring$LCBD, 27, 13, byrow=TRUE)
LCBD.27.fall <- matrix(beta.out.27.fall$LCBD, 27, 13, byrow=TRUE)
rownames(LCBD.27.spring) <- rownames(LCBD.27.fall) <- rownames(xy)
colnames(LCBD.27.spring) <- colnames(LCBD.27.fall) <- 1996:2008

# Compute LCBD values per site, summed over the years, for each season separately
LCBD.27.per.site.spring <- apply(LCBD.27.spring, 1, sum)
LCBD.27.per.site.fall <- apply(LCBD.27.fall, 1, sum)

# Compute LCBD values per year, summed over the sites, for each season separately
LCBD.27.per.year.spring <- apply(LCBD.27.spring, 2, sum)
LCBD.27.per.year.fall <- apply(LCBD.27.fall, 2, sum)

# --------

# Maps of the LCBD values per site, summed over the years, for each season separately.
# Note – In this plot and onwards, the square roots of the LCBD values are used in the argument
# value of parameter "cex" in order to plot bubbles whose areas (rather than radii) are
# proportional to the LCBD values.

# Map of LCBD values per site, summed over the years, spring; Appendix A3, Figure A3.10
plot(xy[,c(2,1)], asp=1, type="n", xlab="Latitude", ylab="Longitude", main="LCBD indices,
Chesapeake sites, spring, all years", xlim=c(-77.3,-75.7))
points(xy[,c(2,1)],pch=21, col="white", bg="steelblue2", cex=15*sqrt(LCBD.27.per.site.spring))
text(xy[,c(2,1)], labels=rownames(xy), pos=4)

# Map of LCBD values per site, summed over the years, fall; run the code to see the map
plot(xy[,c(2,1)], asp=1, type="n", xlab="Latitude", ylab="Longitude", main=" LCBD indices,
Chesapeake sites, fall, all years ", xlim=c(-77.3,-75.7))
points(xy[,c(2,1)], pch=21, col="white", bg="steelblue2", cex=15*sqrt(LCBD.27.per.site.fall))
text(xy[,c(2,1)], labels=rownames(xy), pos=4)
```

```
# Time maps of LCBD per year, summed over sites, for each season; App. A3, Figure A3.11
par(mfrow=c(2,1))
plot(1996:2008, rep(0, 13), type="p", xlab="Sampling years", ylab="", main="LCBD indices
along years, Chesapeake, spring", ylim=c(-1,1), pch=21, col="white", bg="steelblue2",
cex=20*sqrt(LCBD.27.per.year.spring))
plot(1996:2008, rep(0, 13), type="p", xlab="Sampling years", ylab="", main="LCBD indices
along years, Chesapeake, fall", ylim=c(-1,1), pch=21, col="white", bg="steelblue2",
cex=20*sqrt(LCBD.27.per.year.fall))
par(mfrow=c(1,1))

# --------

# Plot space-time maps of LCBD, spring and fall data; Appendix A3, Figure A3.12
par(mfrow=c(1,2))
seq.X.27 <- rep(1996:2008, 27)
seq.Y.27 <- rep(1:27, each=13)

plot(seq.X.27, seq.Y.27, asp=1, type="n", ylab="Sites", xlab="Years", main="Space-time map,
LCBD, spring", ylim=c(1,27), xlim=c(1996,2008), yaxt="n", cex.axis=0.8)
points(seq.X.27, seq.Y.27, pch=21, col="white", bg="steelblue2",
cex=30*sqrt(beta.out.27.spring$LCBD))
points(seq.X.27[signif.27.spring], seq.Y.27[signif.27.spring], pch=21, col="black",
bg="steelblue2", cex=30*sqrt(beta.out.27.spring$LCBD[signif.27.spring]))
axis(side=2, 1:27, labels=rownames(xy), las=1, cex.axis=0.8)

plot(seq.X.27, seq.Y.27, asp=1, type="n", ylab="Sites", xlab="Years", main="Space-time map,
LCBD, fall", ylim=c(1,27), xlim=c(1996,2008), yaxt="n", cex.axis=0.8)
points(seq.X.27, seq.Y.27, pch=21, col="white", bg="steelblue2",
cex=30*sqrt(beta.out.27.fall$LCBD))
points(seq.X.27[signif.27.fall], seq.Y.27[signif.27.fall], pch=21, col="black", bg="steelblue2",
cex=30*sqrt(beta.out.27.fall$LCBD[signif.27.fall]))
axis(side=2, 1:27, labels=rownames(xy), las=1, cex.axis=0.8)
par(mfrow=c(1,1))


# 5.3. REPEAT THE LCBD ANALYSES ON 25 BRACKISH SITES, SPRING AND FALL SEPARATELY
#       after excluding sites #36 and #79, located in freshwater


# 5.3.1. Prepare the data files

freshwater <- which(sampling$STATION %in% c(36,79))
fauna.25 <- fauna[-freshwater,]                                  # 650 x 205
sampling.25 <- sampling[-freshwater,]                            # 650 x 6
waterquality.25 <- waterquality[-freshwater,]                    # 650 x 7
sediment.25 <- sediment[-freshwater,]                            # 650 x 7
xy.25 <- xy[-c(12,27),]                                          # 27 x 2

# Create a simple factor for seasons for the 25 sites, 13 years and 2 seasons, length = 650
season25 <- sampling.25$SEASON

# Assemble separate sampling data frames for the spring and fall data
temp = as.matrix(sampling.25)
sampling.25.spring <- as.data.frame(temp[temp[,6]=="Spring", ])   # 325 x 6
sampling.25.fall <- as.data.frame(temp[temp[,6]=="Fall", ])       # 325 x 6


# 5.3.2. Compute LCDB indices

beta.out.25 <- beta.div(fauna.25, method="hellinger", nperm=0)          ### Used in section 5.3.3
beta.out.25.spring <- beta.div(fauna.25[season25=="Spring",], method="hellinger", nperm=499)
```

```
beta.out.25.fall <- beta.div(fauna.25[season25=="Fall",], method="hellinger", nperm=499)
# Warning – It is better to use 999 permutations. Permutation tests take a bit of time.
# The LCBD indices are available in the $LCBD vector, the permutational p-values in $p.LCBD
signif.25.spring <- which(beta.out.25.spring$p.LCBD <= 0.05)
signif.25.fall <- which(beta.out.25.fall$p.LCBD <= 0.05)

# --------

# Plot space-time maps of LCBD, spring and fall data; Appendix A3, Figure A3.13
# Significant LCBD values at the 0.05 level are plotted with a black rim
par(mfrow=c(1,2))
seq.X.25 <- rep(1996:2008, 25)
seq.Y.25 <- rep(1:25, each=13)

plot(seq.X.25, seq.Y.25, asp=1, type="n", ylab="Sites", xlab="Years", main="Space-time map,
LCBD, spring", ylim=c(1,25), xlim=c(1996,2008), yaxt="n", cex.axis=0.8)
points(seq.X.25, seq.Y.25, pch=21, col="white", bg="steelblue2",
cex=30*sqrt(beta.out.25.spring$LCBD))
points(seq.X.25[signif.25.spring], seq.Y.25[signif.25.spring], pch=21, col="black",
bg="steelblue2", cex=30*sqrt(beta.out.25.spring$LCBD[signif.25.spring]))   # Significant
LCBD values, spring
axis(side=2, 1:25, labels=rownames(xy.25), las=1, cex.axis=0.8)

plot(seq.X.25, seq.Y.25, asp=1, type="n", ylab="Sites", xlab="Years", main="Space-time map,
LCBD, fall", ylim=c(1,25), xlim=c(1996,2008), yaxt="n", cex.axis=0.8)
points(seq.X.25, seq.Y.25, pch=21, col="white", bg="steelblue2",
cex=30*sqrt(beta.out.25.fall$LCBD))
points(seq.X.25[signif.25.fall], seq.Y.25[signif.25.fall], pch=21, col="black", bg="steelblue2",
cex=30*sqrt(beta.out.25.fall$LCBD[signif.25.fall]))   # Significant LCBD values, fall
axis(side=2, 1:25, labels=rownames(xy.25), las=1, cex.axis=0.8)
par(mfrow=c(1,1))

# --------

# Analysis of variance of the LCBD indices, 25 sites, factors site and year

# Two-way anova for LCBD computed for 325 spring data only
tmp.25.1 <- sampling.25[sampling.25$SEASON=="Spring",]
tmp.25.1$YEAR <- as.factor(tmp.25.1$YEAR)
aov.LCBD.25.spring.out <- aov(beta.out.25.spring$LCBD ~ STATION+YEAR, data= tmp.25.1)
summary(aov.LCBD.25.spring.out)

# Two-way anova for LCBD computed for 325 fall data only
tmp.25.2 <- sampling.25[sampling.25$SEASON=="Fall",]
tmp.25.2$YEAR <- as.factor(tmp.25.2$YEAR)
aov.LCBD.25.fall.out <- aov(beta.out.25.fall$LCBD ~ STATION+YEAR, data= tmp.25.2)
summary(aov.LCBD.25.fall.out)
```

 # 5.3.3. Compare spring and fall LCBD: paired t*-test*

```
# The data in vector beta.out.25$LCBD subjected to the t-test were computed at the beginning
# of section 5.3.2 for the spring and fall faunal data together
(t.test.spring.fall.res <- t.test(beta.out.25$LCBD ~ rep(c(1,2), 325), paired=TRUE))

# LCBD values in a study always sum to 1
(sum.spring <- sum(beta.out.25$LCBD[seq(1, 650, 2)]))
(sum.fall <- sum(beta.out.25$LCBD[seq(2, 650, 2)]))
sum.spring + sum.fall
```

```
# 5.4. ENVIRONMENTAL VARIABLES EXPLAINING LCBD VARIATION ACROSS YEARS, 25 SITES

# Values of North Atlantic Oscillation (NAO), yearly data, from 1996 to 2008. See section 3.2.1.
NAO <- c(-1.07, -1.07, 0.03, 0.03, -0.29, -0.29, 0.55, 0.55, 0.63, 0.63, -0.57, -0.57, 0.35, 0.35,
0.32, 0.32, 0.10, 0.10, -0.76, -0.76, -0.26, -0.26, 0.54, 0.54, 0.08, 0.08)
NAO <- rep(NAO, 25)

site.names.25 <- rep(rownames(xy)[-c(12,27)], each=26)

# Recode factor site using Helmert contrast variables
sites.helmert <- model.matrix(~as.factor(site.names.25), contrasts="contr.helmert")[,-1]
colnames(sites.helmert) <- rownames(xy)[-c(1,12,27)]       # Site 1 has no Helmert variable
explanatory.25bis <- cbind(waterquality.25, sediment.25, NAO, sites.helmert,
sampling.25$SEASON)
```

 # 5.4.1. Spring only

```
# Vector listing only the spring obervations
spring.obs <- which(sampling.25$SEASON == "Spring")

# Selection of explanatory variables will be done using partial regression controlling for factor
# site, which is likely to account for a large fraction of the variation in the data. Partial linear
# regression is linear regression computed after residualizing the response and explanatory
# variables onto the variables or factors one wants to control for, here factor site.
# See Legendre & Legendre (2012, section 10.3.5).

# Compute the two regression models using lm() and obtain residuals
# 1. Residualize the LCBD data on factor site
LCBD.spring.resid <- resid(lm(beta.out.25.spring$LCBD ~ ., explanatory.25bis[spring.obs,
12:35]))

# 2. Residualize the explanatory data on factor site
envir.spring.resid <- resid(lm(as.matrix(explanatory.25bis[spring.obs, 1:11]) ~ .,
explanatory.25bis[spring.obs, 12:35]))

# Selection of environmental variables using function forward.sel() of package 'packfor'
(sel.spring.res <- forward.sel(LCBD.spring.resid, envir.spring.resid))

# For partitioning, the response data will be the original (not residualized) LCBD values.
# For explanatory, use CONDUCT (var. 1), DO (var. 2), SALINITY (var. 4) and NAO (var. 11)
# and the Helmert contrasts representing factor site.
(part.25.spring.res <- varpart(beta.out.25.spring$LCBD, explanatory.25bis[spring.obs,
c(1,2,4,11)], explanatory.25bis[spring.obs, 12:35]))

# Test of the unique contribution of the 4 environmental variables, controlling for site
test.4var <- rda(beta.out.25.spring$LCBD, explanatory.25bis[spring.obs, c(1,2,4,11)],
explanatory.25bis[spring.obs, 12:35])
anova(test.4var)

# Test of the unique contribution of factor site, controlling for the 4 environmental
test.site <- rda(beta.out.25.spring$LCBD, explanatory.25bis[spring.obs, 12:35],
explanatory.25bis[spring.obs, c(1,2,4,11)])
anova(test.site)
```

 # 5.4.2. Fall only

```
# Vector listing only the fall obervations
fall.obs <- which(sampling.25$SEASON == "Fall")
```

```
# Compute the two regression models using lm() and obtain residuals
# 1. Residualize the LCBD data on factor site
LCBD.fall.resid <- resid(lm(beta.out.25.fall$LCBD ~ ., explanatory.25bis[fall.obs, 12:35]))

# 2. Residualize the explanatory data on factor site
envir.fall.resid <- resid(lm(as.matrix(explanatory.25bis[fall.obs, 1:11]) ~ .,
explanatory.25bis[fall.obs, 12:35]))

# Selection of environmental variables using function forward.sel() of package 'packfor'
 (sel.fall.res <- forward.sel(LCBD.fall.resid, envir.fall.resid, alpha = 0.08))

# --------

# For partitioning, the response data will be the original (not residualized) LCBD values.
# For explanatory, use CONDUCT (var. 1), PH (var 3), SALINITY (var. 4) and NAO (var. 11)
# and the Helmert contrasts representing factor site.
(part.25.fall.res <- varpart(beta.out.25.fall$LCBD, explanatory.25bis[fall.obs, c(1,3,4,11)],
explanatory.25bis[fall.obs, 12:35]))

# Test of the unique contribution of the 4 environmental variables, controlling for site
test.4var <- rda(beta.out.25.fall$LCBD, explanatory.25bis[fall.obs, c(1,3,4,11)],
explanatory.25bis[fall.obs, 12:35])
anova(test.4var)

# Test of the unique contribution of factor site, controlling for the 4 environmental
test.site <- rda(beta.out.25.fall$LCBD, explanatory.25bis[fall.obs, 12:35],
explanatory.25bis[fall.obs, c(1,3,4,11)])
anova(test.site)


# 5.5. CHANGES IN SPECIES COMPOSITION RELATED TO CHANGES IN LCBD AMONG YEARS

# This analysis will examine among-year variation at site 40, which was also used in section 3.1.
# This site has 36 taxa and exhibits strong variability among years, in both the spring and fall.

# Extract sampling and fauna data for site 40, spring and fall separately
sampling.40 <- sampling[sampling$STATION==40, ]
fauna.40 <- fauna[sampling$STATION==40, ]
fauna.40.spring <- fauna.40[sampling.40$SEASON=="Spring",]
fauna.40.fall <- fauna.40[sampling.40$SEASON=="Fall",]

# Remove the absent taxa and apply a Hellinger transformation
fauna.40.spring <- fauna.40.spring[ , colSums(fauna.40.spring)!=0]
dim(fauna.40.spring)     # 13 rows x 19 taxa
fauna.40.spring.hel <- decostand(fauna.40.spring, method="hellinger")
rownames(fauna.40.spring.hel) <- 1996:2008

fauna.40.fall <- fauna.40.fall[ , colSums(fauna.40.fall)!=0]
dim(fauna.40.fall)     # 13 rows x 19 taxa
fauna.40.fall.hel <- decostand(fauna.40.fall, method="hellinger")
rownames(fauna.40.fall.hel) <- 1996:2008

# Compute the LCBD values of site 40 for spring and fall separately
beta.out.spring.40 <- beta.div(fauna.40.spring, method="hellinger", nperm=999)
LCBD.spring.40 <- beta.out.spring.40$LCBD
LCBD.spring.40
beta.out.spring.40$p.LCBD

beta.out.fall.40 <- beta.div(fauna.40.fall, method="hellinger", nperm=999)
LCBD.fall.40 <- beta.out.fall.40$LCBD
```

```
LCBD.fall.40
beta.out.fall.40$p.LCBD

# Which LCDB values are significant at the alpha=0.05 level ?
signif.40.spring <- which(beta.out.spring.40$p.LCBD <= 0.05)
signif.40.fall <- which(beta.out.fall.40$p.LCBD <= 0.05)

# Plot LCBD values along the years at site 40, spring and fall; Appendix A3, Figure A3.14
# Significant LCBD values at the 0.05 level are plotted with a black rim
par(mfrow=c(2,1))

plot(1996:2008, rep(0, 13), type="p", xlab="Sampling years", ylab="", main="LCBD indices
along years, Site 40, spring", ylim=c(-1,1), pch=21, col="white", bg="steelblue2",
cex=20*sqrt(LCBD.spring.40))
points((1996:2008)[signif.40.spring], rep(0, 13)[signif.40.spring], pch=21, col="black",
bg="steelblue2", cex=20*sqrt(beta.out.spring.40$LCBD[signif.40.spring]))

plot(1996:2008, rep(0, 13), type="p", xlab="Sampling years", ylab="", main="LCBD indices
along years, Site 40, fall", ylim=c(-1,1), pch=21, col="white", bg="steelblue2",
cex=20*sqrt(LCBD.fall.40))
points((1996:2008)[signif.40.fall], rep(0, 13)[signif.40.fall], pch=21, col="black",
bg="steelblue2", cex=20*sqrt(beta.out.fall.40$LCBD[signif.40.fall]))
par(mfrow=c(1,1))

# Compute correlations between taxon abundances and LCDB values. Only display the taxa for
# which the absolute values of the correlations are greater than or equal to 0.5 (arbitrarily chosen
# value). Do not perform tests of significance: the LCBD values and taxon abundances are not
# independent of each other.

# Spring
cor.tax.LCBD.spring.40 <- cor(fauna.40.spring.hel, LCBD.spring.40)
as.matrix(cor.tax.LCBD.spring.40[abs(cor.tax.LCBD.spring.40)>=0.5,])

# Fall
cor.tax.LCBD.fall.40 <- cor(fauna.40.fall.hel, LCBD.fall.40)
as.matrix(cor.tax.LCBD.fall.40[abs(cor.tax.LCBD.fall.40)>=0.5,])


# 5.6. REPEAT THE GRAPHICAL SPACE-TIME ANALYSIS FOR TAXONOMIC RICHNESS, 25 SITES

# Compute species richness (written in a vector)
rich <- apply(decostand(fauna.25, method="pa"), 1, sum)

# Put the richness data in a matrix with rows = sites and columns = sampling times
rich.mat <- matrix(rich, 25, 26, byrow=TRUE)
rownames(rich.mat) <- rownames(xy.25)
rich.spring.mat <- rich.mat[, seq(from=1, to=25, by=2)]
rich.fall.mat <- rich.mat[ , seq(from=2, to=26, by=2)]
colnames(rich.spring.mat) <- 1996:2008
colnames(rich.fall.mat) <- 1996:2008

rich.per.year.spring <- apply(rich.spring.mat, 2, mean)
rich.per.year.fall <- apply(rich.fall.mat, 2, mean)

# --------
```

```
# Plot the taxonomic richness values along the years
par(mfrow=c(2,1))

plot(1996:2008, rep(0, 13), type="p", xlab="Sampling years", ylab="", main="Richness along
years, Chesapeake, spring", ylim=c(-1,1), pch=21, col="white", bg="steelblue2",
cex=sqrt(rich.per.year.spring))

plot(1996:2008, rep(0, 13), type="p", xlab="Sampling years", ylab="", main=" Richness along
years, Chesapeake, fall", ylim=c(-1,1), pch=21, col="white", bg="steelblue2",
cex=sqrt(rich.per.year.fall))
par(mfrow=c(1,1))

# --------

# Plot space-time maps of richness, spring and fall data; Appendix A3, Figure A3.15

rich.spring <- rich[seq(from=1, to=649, by=2)]
rich.fall <- rich[seq(from=2, to=650, by=2)]

# Alternative way: transform the rich.spring.mat and rich.fall.mat matrices (18 lines above) into
# vectors. Take into account the fact that the sites for each year form a column in these matrices
# rich.spring <- as.vector(t(rich.spring.mat))
# rich.fall <- as.vector(t(rich.fall.mat))

par(mfrow=c(1,2))
seq.X.25 <- rep(1996:2008, 25)
seq.Y.25 <- rep(1:25, each=13)

plot(seq.X.25, seq.Y.25, asp=1, type="n", ylab="Sites", xlab="Years", main="Space-time map,
Richness, spring", ylim=c(1,25), xlim=c(1996,2008), yaxt="n", cex.axis=0.8)
points(seq.X.25, seq.Y.25, pch=21, col="white", bg="steelblue2", cex=0.5*sqrt(rich.spring))
axis(side=2, 1:25, labels=rownames(xy.25), las=1, cex.axis=0.8)

plot(seq.X.25, seq.Y.25, asp=1, type="n", ylab="Sites", xlab="Years", main="Space-time map,
Richness, fall", ylim=c(1,25), xlim=c(1996,2008), yaxt="n", cex.axis=0.8)
points(seq.X.25, seq.Y.25, pch=21, col="white", bg="steelblue2", cex=0.5*sqrt(rich.fall))
axis(side=2, 1:25, labels=rownames(xy.25), las=1, cex.axis=0.8)
par(mfrow=c(1,1))

# --------

# Is there a significant difference in mean between the spring and fall richness data?

t.test.res <- t.test(rich.spring, rich.fall, paired=TRUE)

# --------

# Relationship between LCBD and richness

# Spring only
cor.test(rich.spring, beta.out.25.spring$LCBD)

# Fall only
cor.test(rich.fall, beta.out.25.fall$LCBD)

# Note – The correlations between richness and LCDB could be recomputed based upon rarefied
# estimates. On the one hand, use vegan's rarefy() function to obtain rarefied richness estimates
# for a standard sampling effort. On the other hand, use function beta.div() to compute LCBD
# based upon one of Chao et al. (2006) indices, which account for unseen species in the survey
# data. Compute correlations between the resulting richness and LCBD indices for the spring and
```

```
# fall surveys separately.

# ========
```

REFERENCES NOT FOUND IN THE MAIN PAPER

Chao, A., Chazdon, R. L., Colwell, R. K. & Shen, T. J. 2006 Abundance-based similarity indices and their estimation when there are unseen species in samples. *Biometrics* **62**, 361–371.

R Core Team. 2013 *R: A language and environment for statistical computing*. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, URL http://www.R-project.org/.

Wickham, H. 2007 Reshaping data with the reshape package. *Journal of Statistical Software* **21**, 1–20.