

Ecography

**ECOG-00215**

Guerin, G. R., Biffin, E. and Lowe, A. J. 2013. Spatial modelling of species turnover identifies climate ecotones, climate change tipping points and vulnerable taxonomic groups. – *Ecography* 36: xxx–xxx.

**Supplementary material**

## Appendix 1

Table A1. Major surveys in the study region from which plot data were sourced (titles as recorded in the SA Department for Environment, Water and Natural Resources database *BDBSA*).

Arkaroola (SEG)  
Beetaloo Valley(NCS)  
Bma Mlr General Svy  
Eastern Mt Lofty Ranges (NCS)  
Flinders  
Flinders Ranges  
Flinders Ras Nat Park  
Gammon Ranges (NCS)  
Horsnell Gully C.P.(NPWS 1981)  
Kyeema Cons. Park (NPWS 1981)  
Lower Flinders Ranges  
Mid North Reserves (NCS)  
Midnorth  
Morialta Cons Park (NPWS 1981)  
Mount Lofty Ranges (DEP 1981)  
Mt Brown (NCS)  
Mt Lofty  
Mt Remarkable Nat Park 1982  
Para Wirra Rec. Pk (NPWS 1981)  
Southern Fleurieu (SEG)  
Southern Mt Lofty  
Tothill Ranges Survey (NCS)  
Upper Mid North  
Warraweena

---

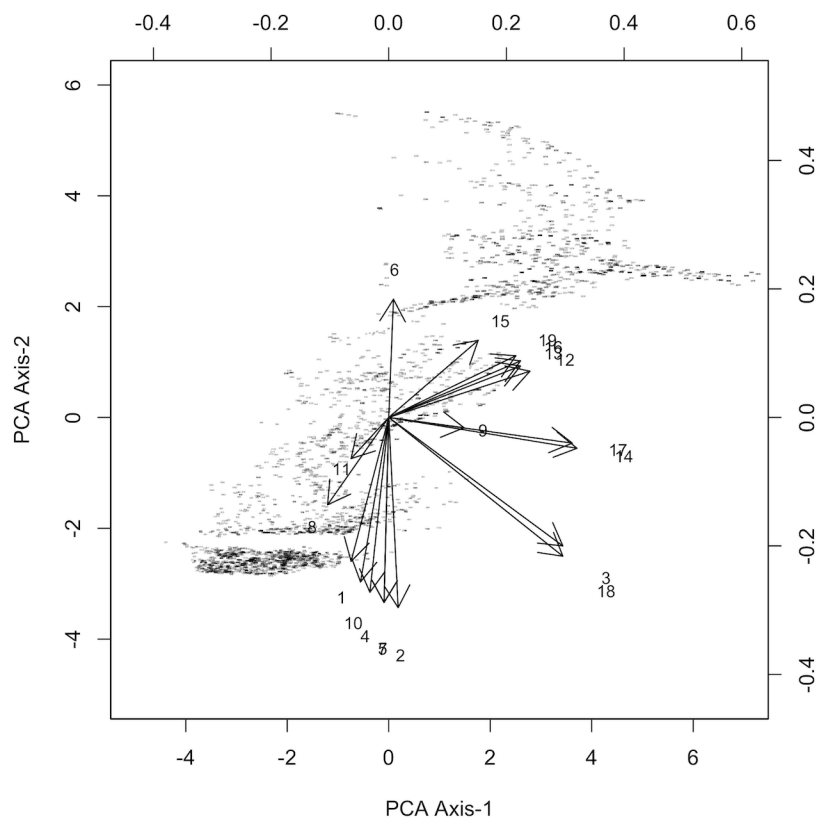


Figure A1. Principal Components Analysis (PCA) of 3,567 survey plots based on 19 bioclimatic variables (Table 1). Bioclimatic variable vectors scaled x1.5.

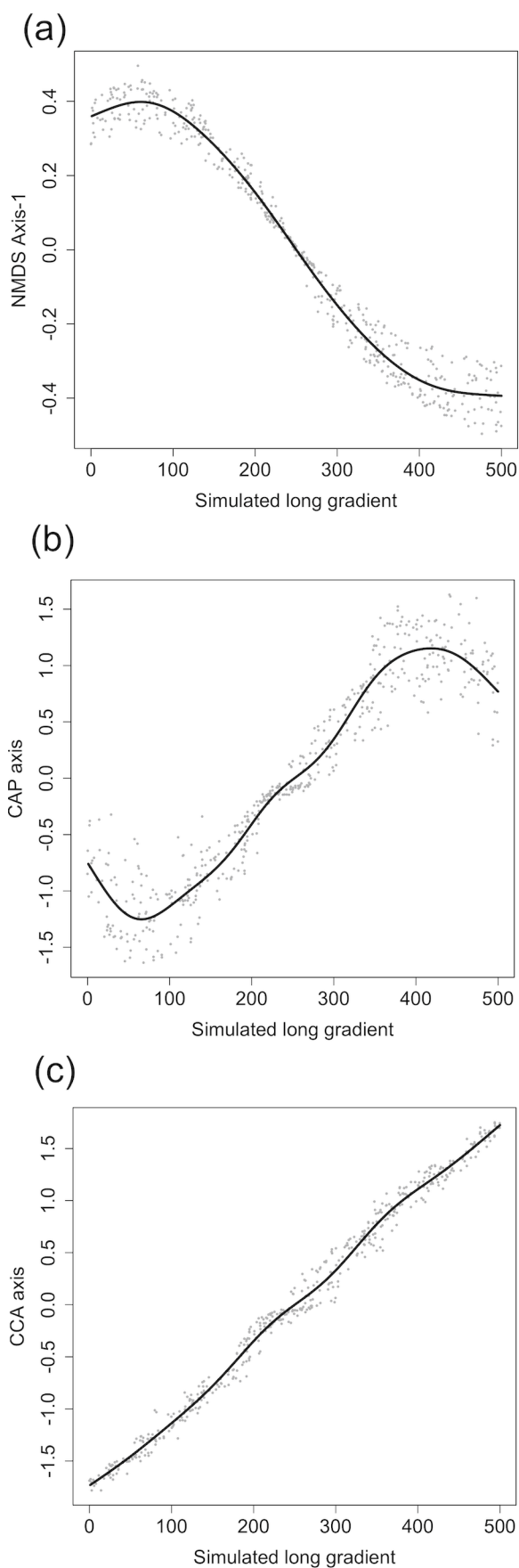


Figure A2. Gradient analysis based on simulated environmental and species occurrence data to trial the robustness of different ordination methods. Data simulate continuous species turnover along orthogonal short and long gradients: (a) using non-metric multidimensional scaling based on Bray-Curtis dissimilarity; (b) using Constrained Analysis of Principle Coordinates based on Bray-Curtis dissimilarity; (c) using WA scores from Canonical Correspondence Analysis. Points: scatterplot of raw data (axis score vs gradient); lines: model. CCA – (c) – is more robust because it recovers a ~linear relationship as expected for continuous turnover scenario. Curves in (a) and (b) do not reflect variation in the rate of species turnover.

Correspondence Analysis also produced a non-linear response. Detrended Correspondence Analysis produced a linear response as for CCA but is less robust for this approach as it removes real arches in data.

## R code demonstrating functions used in presented results

#below are examples of R code and functions used to produce results and figures presented in the main paper. This is not an exhaustive set of code or functions for this analysis (e.g. many data preparation and matrix operation functions not shown) but serves to demonstrate some of the approaches used.  
#there are alternative approaches to achieve similar outcomes and undoubtedly more efficient or elegant code

```
-----  
  
#load packages  
  
require(vegan) # ecological analysis, ordination, correlation plots  
  
require(mgcv) # GAM  
  
require(coenoflex) # simulated ecological data  
  
require(Rcmdr) # correlations  
  
require(segmented) # break.point regression  
  
#require(plotrix) # plot features  
  
-----  
  
#data  
  
Species <- read.table(file="species_v_plots_dataset.csv", header=TRUE, sep="\t",  
row.names=1)  
  
Bray <- vegdist(Species, method='bray')  
  
Bioclim <- read.table(file="scaled_bioclim_v_plots_dataset.csv", header=TRUE,  
sep="\t", row.names=1)  
  
Euclid <- vegdist(Bioclim, method='euclidean')  
  
BioclimRaw <- read.table(file="raw_bioclim_v_plots_dataset.csv", header=TRUE,  
sep="\t", row.names=1)  
  
distance <- as.dist(read.table(file="geographic_distance_symmetrix_matrix.csv",  
header=TRUE, sep=";", row.names=1))  
  
EUCpredictors <- read.table(file="distance-based_predictor_variables.csv",  
header=TRUE, sep="\t")  
  
RAWpredictors <- read.table(file="raw_predictor_variables.csv", header=TRUE,  
sep="\t")  
  
families <- read.table(file="family_nodes_v_plots_dataset.csv", sep="\t",  
header=TRUE, row.names=1)  
  
-----  
  
#REGIONAL DISTANCE ANALYSIS
```

```

rand <- sample(6359961)
Brayord <- as.data.frame(cbind(rand, Bray, distance, Euclid))
Brayorda <- Brayord[order(Brayord[,1]),]
Brayordb <- Brayorda[1:250000, c("Bray", "distance", "Euclid")]
BRAY1 <- Brayordb$Bray
EUCLID1 <- Brayordb$Euclid
DISTANCE1 <- Brayordb$distance

InverseBRAY <- (1-BRAY1) + 1/DISTANCE1
BRAYtransform <- -log(InverseBRAY)
LM_1 <- lm(BRAYtransform ~ EUCLID1 + DISTANCE1)

#

rand1 <- sample(6359961)
Brayord1 <- as.data.frame(cbind(rand1, Bray, distance, Euclid))
Brayord1a <- Brayord1[order(Brayord1[,1]),]
Brayord1b <- Brayord1a[1:250000, c("Bray", "distance", "Euclid")]
BRAY2 <- Brayord1b$Bray
EUCLID2 <- Brayord1b$Euclid
DISTANCE2 <- Brayord1b$distance

InverseBRAY <- (1-BRAY2) + 1/DISTANCE2
BRAYtransform <- -log(InverseBRAY)
LM_2 <- lm(BRAYtransform ~ EUCLID2 + DISTANCE2)

#

rand2 <- sample(6359961)
Brayord2 <- as.data.frame(cbind(rand2, Bray, distance, Euclid))
Brayord2a <- Brayord2[order(Brayord2[,1]),]
Brayord2b <- Brayord2a[1:250000, c("Bray", "distance", "Euclid")]
BRAY3 <- Brayord2b$Bray
EUCLID3 <- Brayord2b$Euclid
DISTANCE3 <- Brayord2b$distance

InverseBRAY <- (1-BRAY3) + 1/DISTANCE3
BRAYtransform <- -log(InverseBRAY)
LM_3 <- lm(BRAYtransform ~ EUCLID3 + DISTANCE3)

#

rand3 <- sample(6359961)
Brayord3 <- as.data.frame(cbind(rand3, Bray, distance, Euclid))
Brayord3a <- Brayord3[order(Brayord3[,1]),]
Brayord3b <- Brayord3a[1:250000, c("Bray", "distance", "Euclid")]
BRAY4 <- Brayord3b$Bray
EUCLID4 <- Brayord3b$Euclid
DISTANCE4 <- Brayord3b$distance

InverseBRAY <- (1-BRAY4) + 1/DISTANCE4
BRAYtransform <- -log(InverseBRAY)
LM_4 <- lm(BRAYtransform ~ EUCLID4 + DISTANCE4)

#

rand4 <- sample(6359961)
Brayord4 <- as.data.frame(cbind(rand4, Bray, distance, Euclid))
Brayord4a <- Brayord4[order(Brayord3[,1]),]
Brayord4b <- Brayord4a[1:250000, c("Bray", "distance", "Euclid")]

```

```

BRAY5 <- Brayord4b$Bray
EUCLID5 <- Brayord4b$Euclid
DISTANCE5 <- Brayord4b$distance

InverseBRAY <- (1-BRAY5) + 1/DISTANCE5
BRAYtransform <- -log(InverseBRAY)
LM_5 <- lm(BRAYtransform ~ EUCLID5 + DISTANCE5)

#determine average slope for bioclimatic distance and make predictions
del1 <- coef(LM_1)
del2 <- coef(LM_2)
del3 <- coef(LM_3)
del4 <- coef(LM_4)
del5 <- coef(LM_5)
euc_av <- mean(del1[2], del2[2], del3[2], del4[2], del5[2])

predict <- (1-exp(-(euc_av*EUCpredictors$EUCLID)))

#calculate future climate scenarios
datamat <-
matrix(c(0.9719450799,0.8542543551,0.749380441,0.9121458222,0.738648426,1.024442
8826,0.9849043476,0.9518834243,0.749380441,0.9121458222,0.738648426,1.0244428826
,1.0108228831,1.1471415626,0.6819362013,0.8300526982,0.6721700677,1.0244428826,1
.0011034323,1.0739197607,0.7418866366,0.903024364,0.7312619417,1.0244428826,1.09
50581234,1.7817305121,0.5245663087,0.6385020755,0.5170538982,1.0244428826), nrow
= 5, ncol=6, byrow=TRUE)
dimnames(datamat) <- list(c("base", "2030low", "2030high", "2070low",
"2070high"), c("bio5", "bio6", "bio12", "bio14", "bio18", "bio2"))

#split rows
base <- datamat[1,]
thlow <- datamat[2,]
thhigh <- datamat[3,]
selow <- datamat[4,]
sehigh <- datamat[5,]
#combine rows for each calc
a <- rbind(base, thlow)
b <- rbind(base, thhigh)
c <- rbind(base, selow)
d <- rbind(base, sehigh)
#calc distances for each scenario
dista <- vegdist(a, method="euclidean")
distb <- vegdist(b, method="euclidean")
distc <- vegdist(c, method="euclidean")
dstd <- vegdist(d, method="euclidean")

#PLOT
plot(predict~EUCpredictors$EUCLID, type='l', lwd=2, xlab="Euclidean bioclimatic
distance", ylab="Predicted Bray-Curtis dissimilarity", cex.lab=1.5,
cex.axis=1.5, ylim=c(0,1), xlim=c(0,1.2), bty="n", xaxs="i", yaxs="i")

ablineclip(v=dista, lty=2, lwd=2, y1=0, y2=0.2536621)
ablineclip(v=distb, lty=2, lwd=2, y1=0, y2=0.6145845)
ablineclip(v=distc, lty=2, lwd=2, y1=0, y2=0.48294)
ablineclip(v=dstd, lty=2, lwd=2, y1=0, y2=0.9523548)

ablineclip(h=LIST[1], lty=2, lwd=2, x1=0, x2=0.09848542)
ablineclip(h=LIST[2], lty=2, lwd=2, x1=0, x2=0.3209389)
ablineclip(h=LIST[3], lty=2, lwd=2, x1=0, x2=0.2220293)

```

```

ablineclip(h=LIST[4], lty=2, lwd=2, x1=0, x2= 1.024644)

text(0.07, 0.06, "2030 L", cex=1.5, srt=90, font=2)
text(0.2, 0.06, "2070 L", cex=1.5, srt=90, font=2)
text(0.3, 0.06, "2030 H", cex=1.5, srt=90, font=2)
text(1.0, 0.06, "2070 H", cex=1.5, srt=90, font=2)

#extract predicted values for scenarios (as plotted)
LIST <- list((1-exp(-(euc_av*dista[1]))), (1-exp(-(euc_av*distb[1]))), (1-exp(-
(euc_av*distc[1]))), (1-exp(-(euc_av*distd[1]))))
print(LIST)

-----

#ORDINATION and GRADIENT ANALYSIS

#generate simulated community and environmental gradient data

example <- coenoflex(numgrd=2, numplt=500,numspc=400,grdtyp=c('e','e'),
grdlen=c(500,200),width=c(100,100),variab=c(20,20),grdprd=c(0,0),
alphad=c(1,1), pdist='r', sdist='r', skew=3.0, aacorr=0.0, cmpasy=3.0,
cmpphy=0.0, maxtot=100, noise=20, slack=0.2, autlin='irm(1,2)')

new <- as.data.frame(example$veg)

trick <- as.data.frame(example$site)

#NMDS indirect gradient analysis
bray3 <- vegdist(new, method="bray")
metaNMDS_BRAY <- metaMDS(bray3, k=2, trymax=10, no.share=0.1, trace=2, pc=TRUE)
SCORES6 <- as.data.frame(scores(metaNMDS_BRAY))
plot(SCORES6$NMDS1~trick$V1)
plot(SCORES6$NMDS1~trick$V2)
plot(SCORES6$NMDS2~trick$V1)
plot(SCORES6$NMDS2~trick$V2)

V1 <- trick$V1

GAMnmds <- gam(SCORES6$NMDS1~s(V1))
summary(GAMnmds)
plot(GAMnmds)

plot(SCORES6$NMDS1 ~ trick$V1, cex=0.3, pch=19, xlab="Simulated long gradient",
ylab="NMDS Axis-1", cex.lab=1.5, cex.axis=1.5, main="", col="darkgray")

SIMpredictors <- matrix(1:500)
SIMpredictors <- as.data.frame(SIMpredictors)
colnames(SIMpredictors) <- "V1"

PREDICT9 <- predict.gam(GAMnmds, newdata=SIMpredictors)

```

```

points(SIMpredictors$V1, PREDICT9, col="black", type="l", lwd=3)

#CCA: direct gradient analysis/constrained ordination
#first analyse the longer gradient
CCAtest1 <- cca(new~trick$V1)
CCAtest1
permutest(CCAtest1, permutations=1000)
plot(CCAtest1)
SCORES7 <- scores(CCAtest1)
SCORES7 <- as.data.frame(SCORES7$sites)
plot(SCORES7$CCA1~trick$V1)
plot(SCORES7$CCA1~trick$V2)
V1 <- trick$V1
GAMcca <- gam(SCORES7$CCA1~s(V1))
plot(GAMcca)
summary(GAMcca)

plot(SCORES7$CCA1 ~ trick$V1, cex=0.3, pch=19, xlab="Simulated long gradient",
ylab="CCA axis", cex.lab=1.5, cex.axis=1.5, main="", col="darkgray")

PREDICT9 <- predict.gam(GAMcca, newdata=SIMpredictors)

points(SIMpredictors$V1, PREDICT9, col="black", type="l", lwd=3)

#shorter gradient test (with long one not analysed)
CCA2 <- cca(new~trick$V2)
CCA2
permutest(CCA2, permutations=1000)
plot(CCA2)
SCORES2 <- scores(CCA2)
SCORES2 <- as.data.frame(SCORES2$sites)
plot(SCORES2$CCA1~trick$V1)
plot(SCORES2$CCA1~trick$V2)
V2 <- trick$V2
GAM <- gam(SCORES2$CCA1~s(V2))
summary(GAM)
plot(GAM)

#capscale: Constrained Analysis of Principle Coordinates (CAP)
CAP1 <- capscale(bray3~trick$V1)
CAP1
permutest(CAP1, permutations=1000)
plot(CAP1)
SCORES5 <- scores(CAP1)
SCORES5 <- as.data.frame(SCORES5$sites)
plot(SCORES5$CAP1~trick$V1)
plot(SCORES5$CAP1~trick$V2)
GAMcap <- gam(SCORES5$CAP1~s(V1))
summary(GAMcap)
plot(GAMcap)

```

```

plot(SCORES5$CAP1 ~ trick$V1, cex=0.3, pch=19, xlab="Simulated long gradient",
ylab="CAP axis", cex.lab=1.5, cex.axis=1.5, main="", col="darkgray")

PREDICT8 <- predict.gam(GAMcap, newdata=SIMPredictors)

points(SIMPredictors$V1, PREDICT8, col="black", type="l", lwd=3)

#ordination with real data

#generate spatial autocovariates via principal coordinates of neighbour matrices
(PCNM) (vegan):

PCNM <- pcnm(distance)
SC2 <- scores(PCNM)
SC2 <- as.data.frame(SC2)

PCNM1 <- SC2$PCNM1
PCNM2 <- SC2$PCNM2
PCNM3 <- SC2$PCNM3
PCNM4 <- SC2$PCNM4
PCNM5 <- SC2$PCNM5
PCNM6 <- SC2$PCNM6
PCNM7 <- SC2$PCNM7
PCNM8 <- SC2$PCNM8
PCNM9 <- SC2$PCNM9
PCNM10 <- SC2$PCNM10

CCA1 <- cca(Species~BioclimRaw$bio1)
CCA1
permutest(CCA1, permutations=1000)
SCORES <- scores(CCA1)
SCORES <- as.data.frame(SCORES$sites)

CCA2 <- cca(Species~BioclimRaw$bio12)
CCA2
permutest(CCA2, permutations=1000)
SCORES2 <- scores(CCA2)
SCORES2 <- as.data.frame(SCORES2$sites)

CCA3 <- cca(Species~BioclimRaw$bio5)
CCA3
permutest(CCA3, permutations=1000)
SCORES3 <- scores(CCA3)
SCORES3 <- as.data.frame(SCORES3$sites)

CCA4 <- cca(Species~BioclimRaw$bio14)
CCA4
permutest(CCA4, permutations=1000)
SCORES4 <- scores(CCA4)
SCORES4 <- as.data.frame(SCORES4$sites)

bio1 <- BioclimRaw$bio1
bio12 <- BioclimRaw$bio12
bio5 <- BioclimRaw$bio5
bio14 <- BioclimRaw$bio14

```

```

##GAM with first 10 spatial eigenvectors - non-significant terms removed:
GAM1 <- gam(SCORES$CCA1~s(bio1, k=8) + PCNM1 + PCNM2 + PCNM3 + PCNM4 + PCNM5 +
PCNM6 + PCNM7 + PCNM8 + PCNM9 + PCNM10)
summary(GAM1)

GAM2 <- gam(SCORES2$CCA1~s(bio12, k=8) + PCNM1 + PCNM2 + PCNM3 + PCNM8 + PCNM10)
summary(GAM2)

GAM3 <- gam(SCORES3$CCA1~s(bio5, k=8) + PCNM1 + PCNM2 + PCNM3 + PCNM4 + PCNM7 +
PCNM8 + PCNM9 + PCNM10)
summary(GAM3)

GAM4 <- gam(SCORES4$CCA1~s(bio14, k=8) + PCNM1 + PCNM2 + PCNM3 + PCNM4 + PCNM5 +
PCNM6 + PCNM7 + PCNM8 + PCNM9)
summary(GAM4)

#PLOTS

par(mfrow=c(2,2),mar=c(4,4,2,1))

##BIO1

fv1 <- predict(GAM1, type="terms")
prsd1 <- residuals(GAM1, type="working")+fv1[,11] # column 11 is the smooth term
in this case
gamline1 <- cbind(bio1, fv1[,11])
gamline1 <- gamline1[order(gamline1[, 1]), ] # so in order for line

plot(prsd1 ~ bio1, cex=0.3, pch=19, xlab=expression(paste("Mean temperature (",
degree, "C)")), ylab="s(mean temperature)", cex.lab=1.2, cex.axis=1.3,
col="darkgray", bty="n", xaxs="i", yaxs="i")

rect(15.2, -3, 17, 3, density = 75, angle = 45, col = "lightgray", border =
FALSE, lty = 1, lwd = 0.5)

points(bio1, prsd1, col="darkgray", cex=0.3, pch=19)

points(gamline1, col="black", type="l", lwd=3)

mtext("(a)", side=3, adj=0, cex=1.5)

arrows(13, 1.2, y1=1.2, x1=16.8, length=0.1, angle=15)
arrows(13, 0.7, y1=0.7, x1=13.9, length=0.1, angle=15)
arrows(13, 0.95, y1=0.95, x1=14.2, length=0.1, angle=15)
arrows(13, 0.45, y1=0.45, x1=13.4, length=0.1, angle=15)

text(13.4, 1.1, "2070 high", cex=0.9)
text(13.4, 0.85, "2030 high", cex=0.9)
text(13.4, 0.6, "2070 low", cex=0.9)
text(13.4, 0.35, "2030 low", cex=0.9)

##BIO12

fv2 <- predict(GAM2, type="terms")
prsd2 <- residuals(GAM2, type="working")+fv2[,6] # column 11 is the smooth term
in this case
gamline2 <- cbind(bio12, fv2[,6])
gamline2 <- gamline2[order(gamline2[, 1]), ] # so in order for line

```

```

plot(prsd2 ~ bio12, cex=0.3, pch=19, xlab="Mean annual rainfall (mm)",
ylab="s(mean annual rainfall)", cex.lab=1.2, cex.axis=1.3, col="darkgray",
bty="n", xaxs="i", yaxs="i")

rect(400, -3, 600, 3, density = 75, angle = 45, col = "lightgray", border =
FALSE, lty = 1, lwd = 0.5)

points(bio12, prsd2, col="darkgray", cex=0.3, pch=19)

points(gamline2, col="black", type="l", lwd=3)

mtext("(b)", side=3, adj=0, cex=1.5)

arrows(1000, -1, y1=-1, x1=700, length=0.1, angle=15)
arrows(1000, -0.5, y1=-0.5, x1=990, length=0.1, angle=15)
arrows(1000, -0.75, y1=-0.75, x1=910, length=0.1, angle=15)

text(960, -0.93, "2070 high", cex=0.9)
text(960, -0.68, "2030 high", cex=0.9)
text(960, -0.43, "2070 low", cex=0.9)

##BIO5

fv3 <- predict(GAM3, type="terms")
prsd3 <- residuals(GAM3, type="working")+fv3[,9] # column 11 is the smooth term
in this case
gamline3 <- cbind(bio5, fv3[,9])
gamline3 <- gamline3[order(gamline3[, 1]), ] # so in order for line

plot(prsd3 ~ bio5, cex=0.3, pch=19, xlab=expression(paste("Mean max. temp. hot.
month (" , degree, "C)")), ylab="s(max. temp. hottest)", cex.lab=1.2,
cex.axis=1.3, col="darkgray", bty="n", xaxs="i", yaxs="i")

rect(30, -3, 31.9, 3, density = 75, angle = 45, col = "lightgray", border =
FALSE, lty = 1, lwd = 0.5)

points(bio5, prsd3, col="darkgray", cex=0.3, pch=19)

points(gamline3, col="black", type="l", lwd=3)

mtext("(c)", side=3, adj=0, cex=1.5)

arrows(24, 1, y1=1, x1=27.8, length=0.1, angle=15)
arrows(24, 0.5, y1=0.5, x1=24.9, length=0.1, angle=15)
arrows(24, 0.75, y1=0.75, x1=25.2, length=0.1, angle=15)
arrows(24, 0.25, y1=0.25, x1=24.4, length=0.1, angle=15)

text(24.4, 0.9, "2070 high", cex=0.9)
text(24.4, 0.65, "2030 high", cex=0.9)
text(24.4, 0.4, "2070 low", cex=0.9)
text(24.4, 0.15, "2030 low", cex=0.9)

```

```

##BIO14

fv4 <- predict(GAM4, type="terms")
prsd4 <- residuals(GAM4, type="working")+fv4[,10] # column 11 is the smooth term
in this case
gamline4 <- cbind(bio14, fv4[,10])
gamline4 <- gamline4[order(gamline4[, 1]), ] # so in order for line

plot(prsd4 ~ bio14, cex=0.3, pch=19, xlab="Mean rainfall driest month (mm)",
ylab="s(rainfall driest month)", cex.lab=1.2, cex.axis=1.3, col="darkgray",
bty="n", xaxs="i", yaxs="i", ylim=c(-2, 2))

rect(18, -3, 23, 3, density = 75, angle = 45, col = "lightgray", border = FALSE,
lty = 1, lwd = 0.5)

points(bio14, prsd4, col="darkgray", cex=0.3, pch=19)

points(gamline4, col="black", type="l", lwd=3)

mtext("(d)", side=3, adj=0, cex=1.5)

arrows(35, -1.5, y1=-1.5, x1=24.5, length=0.1, angle=15)
arrows(35, -1, y1=-1, x1=34.65, length=0.1, angle=15)
arrows(35, -1.25, y1=-1.25, x1=31.85, length=0.1, angle=15)

text(31, -1.4, "2070 high", cex=0.9)
text(31, -1.15, "2030 high", cex=0.9)
text(31, -0.9, "2070 low", cex=0.9)

-----

#alternative method - find break.points in linear models
LM <- lm(SCORES$CCA1~bio1)
LM2 <- lm(SCORES2$CCA1~bio12)
LM3 <- lm(SCORES3$CCA1~bio5)
LM4 <- lm(SCORES4$CCA1~bio14)

SEG <- segmented(LM,~bio1, psi=c(15, 17))
SEG2 <- segmented(LM2, ~bio12, psi=c(400, 600))
SEG3 <- segmented(LM3, ~bio5, psi=c(30, 32))
SEG4 <- segmented(LM4, ~bio14, psi=c(18,23))

-----

##phylogenetic node correlations

###see R source code in 'phylo.endemism.R' available at
http://davidnipperess.blogspot.com.au/2012/07/phyloendemism-r-function-for.html
-- #there are functions here and in packages 'ape'/'caper' (plus more generic R
matrix operations) to convert a phylogenetic tree to a binary matrix and to
intersect this matrix with a species composition dataset, to obtain a
phylogenetic nodes versus plots matrix --#you can then use the entire matrix, or

```

as here, pick out e.g. family level nodes. We used the sum of occurrences of nodes within plots.

```
test <- cbind(SCORES$CCA1, families)
rcorr.adjust(test, type="pearson")

#results saved, load and remove families with non-significant correlation:
vectors <- read.table(file="correlation_data_CCA_biol.csv", sep="\t",
header=TRUE)
vectors <- vectors[-c(6, 10, 16, 28, 37, 40, 42, 54, 55, 56),]

#plot
par(mfrow=c(1,2), mar=c(3,0,1,0))

linestack(vectors$r, labels=vectors$family, axis=TRUE, air=1.8, cex=0.7, hoff=7,
font=6, at=-0.3)

plot(0.5, 0.5, ylim=c(-1,1),xlim=c(0,1), cex=0, xaxt="n", xlab="", bty="n",
yaxt="n", ylab="")

arrows(0.1, 0.2, x1=0.1, y1=0.8, lwd=2, lty=1, angle=10)
arrows(0.1, -0.2, x1=0.1, y1=-0.8, lwd=2, angle=10)

text(0.15, -0.4, "cooler", srt=270, cex=1.5)
text(0.15, 0.4, "warmer", srt=270, cex=1.5)
```

---