Analysis of community ecology data in R

Jinliang Liu (刘金亮)

Institute of Ecology, College of Life Science
Zhejiang University
Email: jinliang.liu@foxmail.com
http://jinliang.weebly.com

R packages

```
### load Packages
library(vegan)
library(picante)
library(packfor)
library(ecodist)
```

Table to Data frame

Community data--Abundance data

```
> head(gb)
                    Species Quantity Sample    Abbr Max.Ht Habitat
1 Pterostichus melanarius          4     E5 Pte.mel    2.5    Edge
2     Leistus spinibarbis          2     E6 Lei.spi    2.0    Edge
3       Carabus violaceus          1     E6 Car.vio    2.0    Edge
4      Platynus assimilis          4     E5 Pla.ass    2.5    Edge
5         Agonum muelleri          1     E5 Ago.mue    2.5    Edge
6  Calathus rotundicollis         94     E5 Cal.rot    2.5    Edge
```

- gb.biol=xtabs(Quantity~Sample+Abbr, data=gb)
- head(gb.biol)
- class(gb.biol)="matrix"
- gb.biol=as.data.frame(gb.biol)
- head(gb.biol)

```
      Abbr
Sample Aba.par Acu.dub Ago.afr Ago.ful Ago.mue
    E1     388       0       0       0       0
    E2     325       0       0       0       0
    E3     295       0       0       0       0
    E4     350       0       0       0       2
    E5     407       0       0       0       1
    E6     381       0       0       0       0
```

```
> head(gb.biol)
   Aba.par Acu.dub Ago.afr Ago.ful Ago.mue Ago.vid Ama.aen
E1     388       0       0       0       0       0       0
E2     325       0       0       0       0       0       0
E3     295       0       0       0       0       0       0
E4     350       0       0       0       2       0       0
E5     407       0       0       0       1       0       0
E6     381       0       0       0       0       0       0
```

Community data--0/ 1 data

- gb.pa=table(gb$Abbr, gb$Sample)
- table(gb$Sample, gb$Abbr)
- table(gb$Abbr,gb$Sample,gb$Habitat)

```
> table(gb$Abbr, gb$Sample)
          E1 E2 E3 E4 E5 E6 G1 G2
Aba.par    1  1  1  1  1  1  1  1
Acu.dub    0  0  0  0  0  0  1  0
Ago.afr    0  0  0  0  0  0  1  1
Ago.ful    0  0  0  0  0  0  1  0
Ago.mue    0  0  0  1  1  0  0  0
```

```
> table(gb$Sample, gb$Abbr)
   Aba.par Acu.dub Ago.afr Ago.ful Ago.mue Ago.vid
E1       1       0       0       0       0       0
E2       1       0       0       0       0       0
E3       1       0       0       0       0       0
E4       1       0       0       0       1       0
E5       1       0       0       0       1       0
E6       1       0       0       0       0       0
```

```
> table(gb$Abbr,gb$Sample,gb$Habitat)
, ,  = Edge
          E1 E2 E3 E4 E5 E6 G1 G2 G3 G4 (
Aba.par    1  1  1  1  1  1  0  0  0  0
Acu.dub    0  0  0  0  0  0  0  0  0  0
Ago.afr    0  0  0  0  0  0  0  0  0  0
Ago.ful    0  0  0  0  0  0  0  0  0  0
Ago.mue    0  0  0  1  1  0  0  0  0  0
Ago.vid    0  0  0  0  0  0  0  0  0  0

, ,  = Grass
          E1 E2 E3 E4 E5 E6 G1
Aba.par    0  0  0  0  0  0  1
Acu.dub    0  0  0  0  0  0  1
Ago.afr    0  0  0  0  0  0  1
Ago.ful    0  0  0  0  0  0  1
Ago.mue    0  0  0  0  0  0  0
Ago.vid    0  0  0  0  0  0  1

, ,  = Wood
          E1 E2 E3 E4 E5 E6 (
Aba.par    0  0  0  0  0  0
Acu.dub    0  0  0  0  0  0
Ago.afr    0  0  0  0  0  0
Ago.ful    0  0  0  0  0  0
Ago.mue    0  0  0  0  0  0
Ago.vid    0  0  0  0  0  0
```

Environmental data

```
› gb.site=aggregate(Quantity~Sample+Habitat,  FUN=sum,data=gb)
› head(gb.site)
› rownames(gb.site)=gb.site$Sample
› gb.site=gb.site[,-1]
› head(gb.site)
```

	Sample	Habitat	Quantity
1	E1	Edge	715
2	E2	Edge	734
3	E3	Edge	646
4	E4	Edge	867
5	E5	Edge	971
6	E6	Edge	901

	Habitat	Quantity
E1	Edge	715
E2	Edge	734
E3	Edge	646
E4	Edge	867
E5	Edge	971
E6	Edge	901

Alpha diversity

```
› gb.pa=table(gb$Abbr,gb$Sample)
› head(gb.pa)
#### the frequency of the species; the sum of each row
› rowSums(gb.pa)
› apply(gb.pa,MARGIN=1,FUN=sum)
### Individuals of each species
› tapply(gb$Quantity, INDEX=gb$Abbr,FUN=sum)
### Number of species in each site; the sum of each column
› colSums(gb.pa)
› apply(gb.pa,MARGIN=2,FUN=sum)
```

```
> head(gb.pa)
         E1 E2 E3 E4 E5 E6 G1
Aba.par   1  1  1  1  1  1  1
Acu.dub   0  0  0  0  0  0  1
Ago.afr   0  0  0  0  0  0  1
Ago.ful   0  0  0  0  0  0  1
Ago.mue   0  0  0  1  1  0  0
```

```
> rowSums(gb.pa)
Aba.par Acu.dub
     18       3
Ama.sim Bad.bul
      1       3
Cal.rot Car.vio
     18      12
```

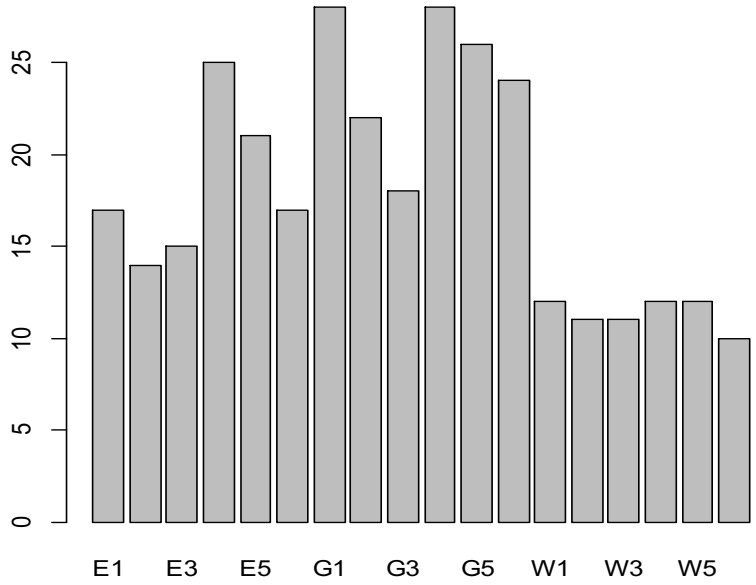
```
> colSums(gb.pa)
E1 E2 E3 E4 E5 E6 G1 G2
17 14 15 25 21 17 28 22
```

```
> tapply(gb$Quantity,INDEX=gb$Abbr,FUN=sum)
Aba.par Acu.dub Ago.afr Ago.ful Ago.mue
   4868       3      20       3       3
Ama.sim Bad.bul Bem.big Bem.gut Bem.lam
      1       4      30       3      18
Cal.rot Car.vio Cli.fos Cur.aul Cyc.car
    765      33      22       1       9
```

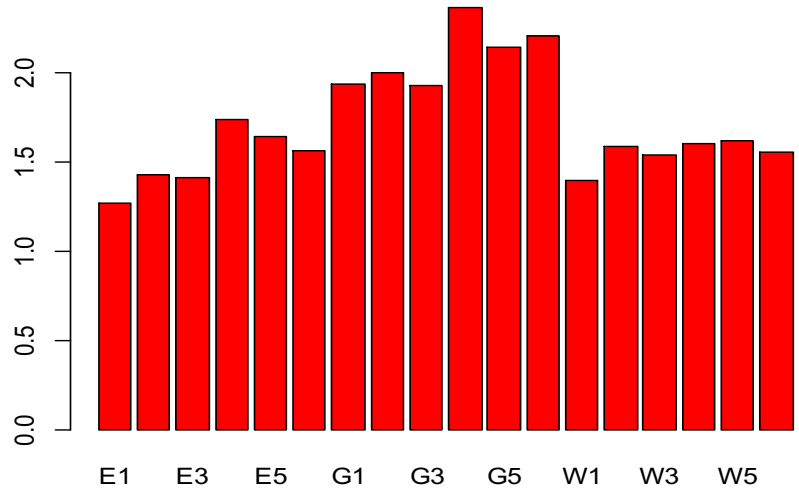
Alpha diversity

Vegan package

```
› library(vegan)
› specnumber(gb.biol)    ###species richness in each site
› length(specnumber(gb.biol,MARGIN=2)>0)   ###species richness in all sites
› diversity(gb.biol,"shannon")     ####Shannon-Winer  diversity index
› diversity(gb.biol,"simpson")     ####Simpson  diversity index
```



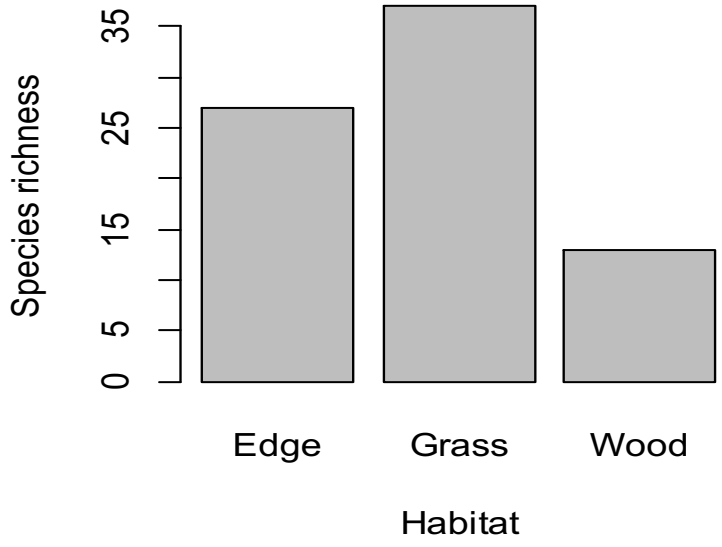
barplot(specnumber(gb.biol))

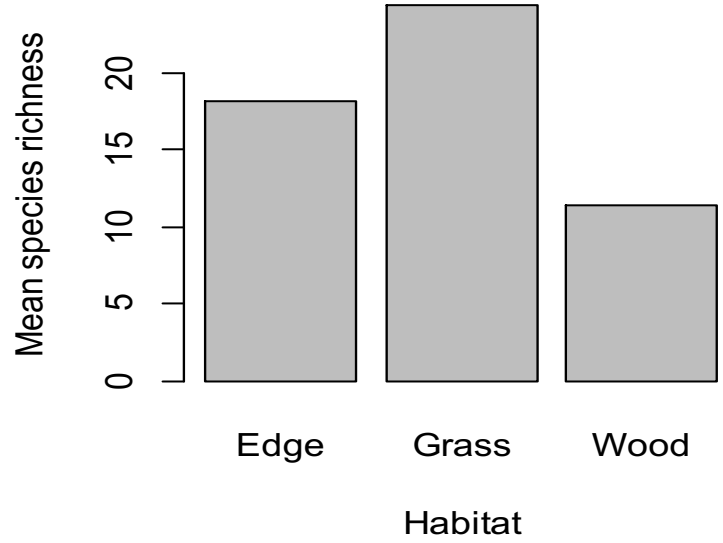


barplot(diversity(gb.biol),col="red")

Alpha diversity

```
› specnumber(gb.biol,  gb.site$Habitat)   ####Species richness in each habitat
› tapply(specnumber(gb.biol),gb.site$Habitat,  FUN=mean) ###Mean species richness
                                                          in each habitat
```






```
barplot(specnumber(gb.biol,
gb.site$Habitat),xlab="Habitat",ylab="Species
richness")
```

```
barplot(tapply(specnumber(gb.biol),gb.site$Habitat,FU
N=mean),xlab="Habitat",ylab="Mean  species richness")
```

Beta diversity

R functions

- dist-offers just limited number of distance measures-those ecological sensible are euclidean, canbera and manhattan. The result is the distance matrix, object of the class dist.
- vegdist (library vegan) - default distance used in this function is Bray-Curtis distance, which is (contrary to Euclidean distance) considered as very suitable for ecological data (it's quantitative measure derived from Sørensen index of similarity)
- dsvdis (library labdsv) - similar to vegdist, just different library.
- designdist (library vegan) - you can design virtually any distance measure using the formula for their calculation.
- daisy (library cluster) - offers euclidean, manhattan and gower distance.
- distance (library ecodist) - contains seven distance measures, but the function more than for practical use is for demonstration of the script (for larger matrices the calculation is rather long).

Beta diversity

——Presence-absence data

$$\beta = \gamma / \alpha$$

```
› gamma=specnumber(gb.biol,gb.site$Habitat)
› alpha=tapply(specnumber(gb.biol),gb.site$Habitat,FUN=mean)
› beta=gamma/alpha
› barplot(gamma/alpa,xlab="Habitat",ylab="Beta  diversity")  ###  “” is not same with "" in R
```

Beta diversity

——Presence-absence data

Vegan packages

betadiver (x, method=NA, order=FALSE, help=FALSE)
vegdist (x,method="bray", binary=F)

```
> betadiver(help=T)
1 "w" = (b+c)/(2*a+b+c)
2 "-1" = (b+c)/(2*a+b+c)
3 "c" = (b+c)/2
4 "wb" = b+c
5 "r" = 2*b*c/((a+b+c)^2-2*b*c)
6 "I" = log(2*a+b+c) - 2*a*log(2)/(2*a+b+c) - ((a+b)*log(a+b) + (a+c)*log(a+c)) / (2*a+b+c)
7 "e" = exp(log(2*a+b+c) - 2*a*log(2)/(2*a+b+c) - ((a+b)*log(a+b) + (a+c)*log(a+c)) / (2*a+b+c))-1
8 "t" = (b+c)/(2*a+b+c)
9 "me" = (b+c)/(2*a+b+c)
10 "j" = a/(a+b+c)
11 "sor" = 2*a/(2*a+b+c)
12 "m" = (2*a+b+c)*(b+c)/(a+b+c)
13 "-2" = pmin(b,c)/(pmax(b,c)+a)
14 "co" = (a*c+a*b+2*b*c)/(2*(a+b)*(a+c))
15 "cc" = (b+c)/(a+b+c)
16 "g" = (b+c)/(a+b+c)
17 "-3" = pmin(b,c)/(a+b+c)
18 "l" = (b+c)/2
19 "19" = 2*(b*c+1)/((a+b+c)^2+(a+b+c))
20 "hk" = (b+c)/(2*a+b+c)
21 "rlb" = a/(a+c)
22 "sim" = pmin(b,c)/(pmin(b,c)+a)
23 "gl" = 2*abs(b-c)/(2*a+b+c)
24 "z" = (log(2)-log(2*a+b+c)+log(a+b+c))/log(2)
```

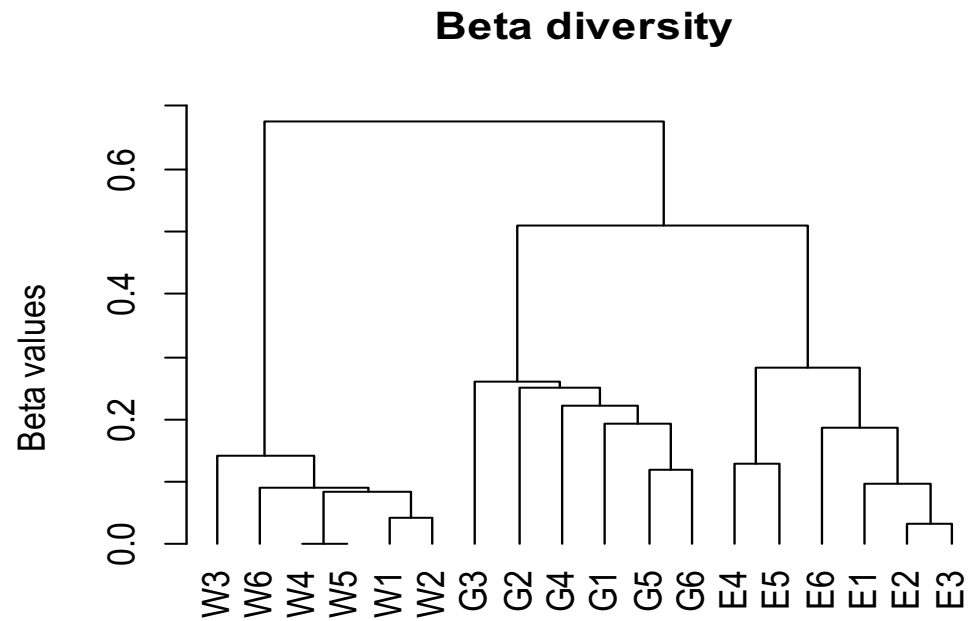
betadiver(gb.biol,"sor")

Cautions
'sor' and 'jac' indices calculated by *betadiver* are similarity indices. The '1-sor' in *betediver* equal to the 'bray' in *vegdist*.

Visualising beta diversity

Cluster dendrogram

```
› gb.beta=betadiver(gb.biol,method="w")
› gb.clus=hclust(gb.beta)
› plot(gb.clus,hang=-1,main="Beta
  diversity",ylab="Beta values",xlab="Bettle
  community sample", sub="Complete joining
  cluster")
```



Ordination analysis

	Raw-data based		Distance-based
	Linear	**Unimodal**	
Unconstrained	PCA(tb-PCA)	CA, DCA	PCoA, NMDS
Constrained	RDA(tb-RDA)	CCA	db-RDA

Transformation-based methods (tb-PCA and tb-RDA) represent analysis using raw species-site data, pre-transformed using e.g. Hellinger transformation (which, combined with Euclidean distance implicit for PCA/RDA, creates Hellinger distance).

To decide whether to apply **linear or unimodal ordination method** on your data, you can use the rule of thumb introduced by Lepš & Šmilauer (2003): first, calculate DCA (detrended by segments) on your data, and check the length of the *first* DCA axis. If the length is > 4, data are heterogeneous and you should go for unimodal methods, if the length is < 3, data are homogeneous and you can use linear methods (in the grey zone between 3 and 4 both methods are OK). Or, if your data are heterogeneous, but you still want to use linear constrained ordination (RDA), you can calculate PCA/RDA using Hellinger's transformation of species data (as recommended e.g. by Borcard et al. 2011).

R code: decorana ()

unconstrained ordination: indirect gradient analysis, ordination axes are not constrained by environmental factors

- ✓ to uncover the main compositional gradients in the species data, structuring the community, and these gradients can be interpreted by known (estimated, measured) environmental factors. Environmental variables does not enter the ordination algorithm, but they are used post hoc, after the analysis.
- ✓ primarily descriptive method, used to uncover and describe the pattern in multivariate data. It generates hypotheses, but cannot test them.

constrained ordination: direct gradient analysis, ordination axes are constrained by environmental factors

- ✓ it relates the species composition directly to the environmental variables and extracts the variance in species composition which is directly related to these variables.
- ✓ to test directly the hypotheses about the influence of environmental factors on species composition.
- ✓ step-wise selection of important environmental variables (and excluding those which are not relevant for species composition), test of significance of the variance explained by environmental factors (Monte Carlo permutation test) and partitioning of the variance explained by particular environmental variables.

Unconstrained ordination

PCA (Principal Component Analysis)

- PCA axes are sorted in descending order according to the amount of variance they extract – eigenvalues. How to decide, which axes are important and representative, e.g. for visualization of data? There are two options:

✓ Kaiser-Guttman criterion – calculate the mean of all eigenvalues and interpret only axes with eigenvalues larger than this mean;

✓ broken stick model – randomly divides the stick of unit length into the same number of pieces as there is PCA axes and then sorts these pieces from the longest to the shortest. Repeats this procedure many times and averages the results of all permutations (analytical solution to this problem is also known). Broken stick model represents eigenvalues, which would occur by random. One may want to interpret only those PCA axes with eigenvalues larger than values generated by broken stick model.

R code: ev=PCACAeig; evplot(ev) or sig= PCAsignificance(PCA,axes= 14); sig

Unconstrained ordination

PCA (Principal Component Analysis)

- R functions

✓ prcomp (stats)

✓ rda (vegan)

✓ pca (labdsv)

✓ evplot: offers visual methods to decide the importance of ordination axes, using Keiser-Guttman criterion and broken stick model.

✓ PCAsignificance (BiodersityR):calculates broken-stick model for PCA axes

Examples

```
library(vegan)
chem=read.delim("chem")
stand.chem=scale(chem)###standardized the variables
pca=rda(stand.chem)
biplot(pca,display="species")
```

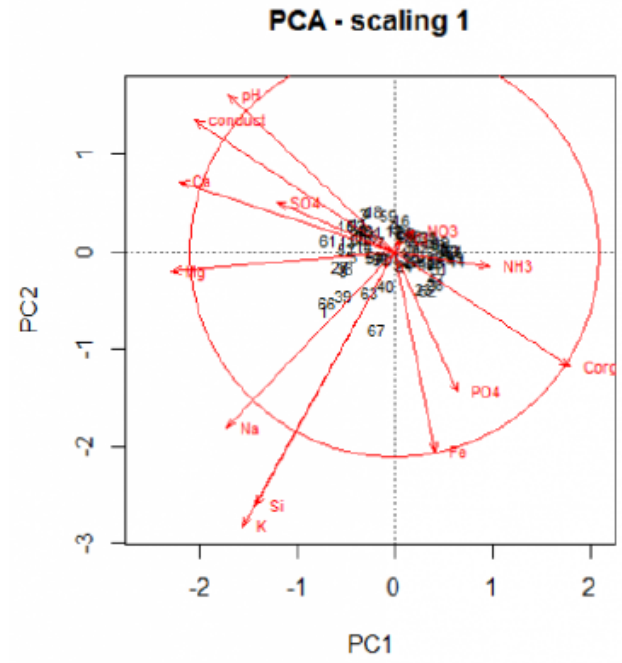
Note that in this specific case, when we are analyzing dataset of environmental variables, data had to be standardized ahead of analysis – all variables should be brought to the same scale, otherwise the variables with large values will have too high influence in the analysis.

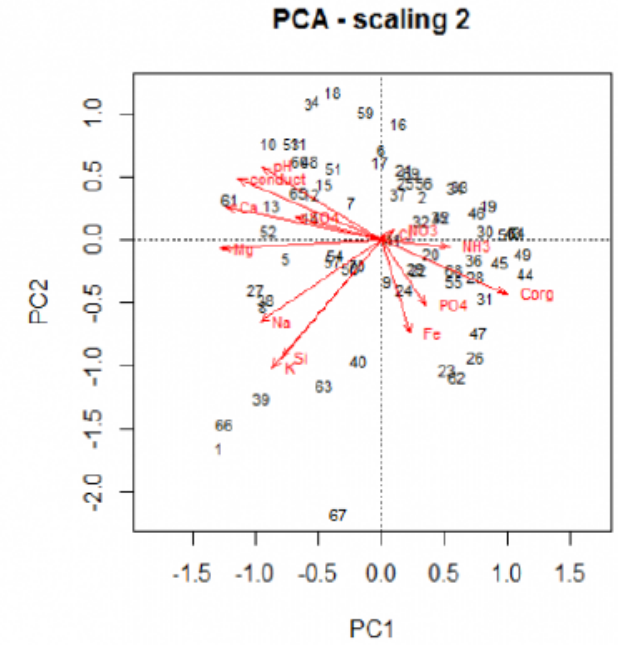
PCA

Scaling

scaling in this case refers to the way ordination results are projected in the reduced space for graphical display

- Scaling 1 - distances among objects (sites) in the biplot
- Scaling 2 - distances among objects in the biplot



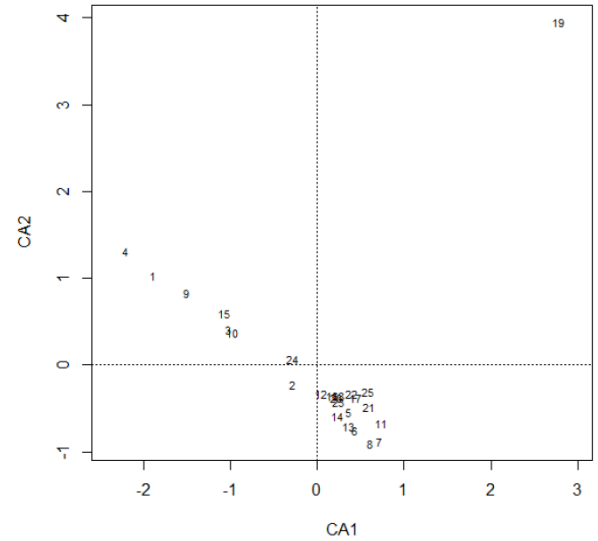


Unconstrained ordination

CA (Correspondence Analysis)

- ✓ Unimodal method of unconstrained ordination
- ✓ Suffers from artefact called *arch effect*, which is caused by non-linear correlation between first and higher axes
- ✓ Popular, even though clumsy way how to remove this artefact is to use detrending in DCA.

```
R code:
library(vegan)
CA<-cca(sp)
Ordiplot(CA)
```



Unconstrained ordination

DCA (Detrended Correspondence Analysis)

- ✓ Detrended version of Correspondence Analysis, removing the arch effect from ordination
- ✓ the length of the first axis (in SD units) refers to the heterogeneity or homogeneity of the dataset (a sort of beta diversity measure)

```
R code:
library(vegan)
DCA<-decorana(sp)
```

Unconstrained ordination

PCoA (Principal Coordinate Analysis)

- ✓ This method is also known as MDS (Metric Multidimensional Scaling)
- ✓ PCA preserves Euclidean distances among samples and CA chi-square distances, PCoA provides Euclidean representation of a set of objects whose relationship is measured by any similarity or distance measure chosen by the user

› pcoa(D, correction="none", rn=NULL)

- › D: A distance matrix of class dist or matrix
- › Correction: Correction methods for negative eigenvalues: "lingoes" and "cailliez". Default value: "none".

Unconstrained ordination

NMDS (Non-metric Multidimensional Scaling)

- Non-metric alternative to PCoA analysis - it can use any distance measure among samples, and the main focus is on projecting the relative position of sample points into low dimensional ordination
- distance based, not eigenvalue based

✓ metaMDS (library vegan)

✓ stressplot (library vegan)

✓ goodness (library vegan)

R code:

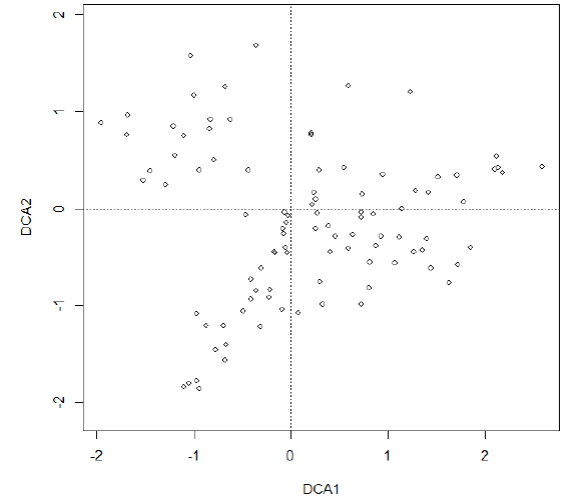
```
NMDS <- metaMDS (vltava.spe)
NMDS
ordiplot (NMDS, type = 't')
stressplot (NMDS)
plot (NMDS, display = 'sites', type = 't', main = 'Goodness of fit')
points (NMDS, display = 'sites', cex = goodness (NMDS)*200)
```

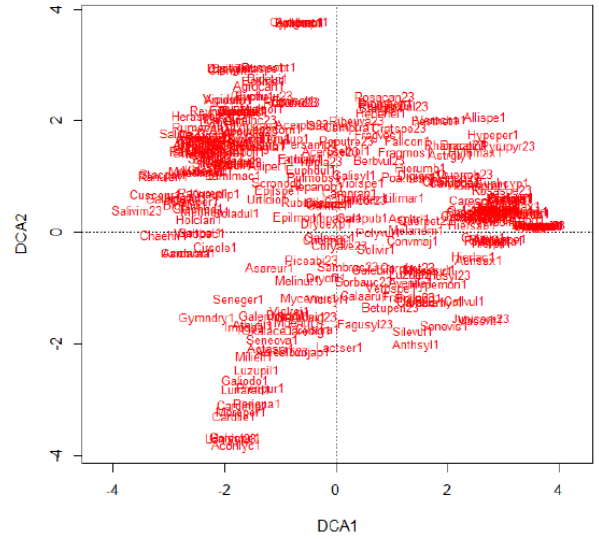
Unconstrained ordination

Ordination diagrams

Ordiplot (library vegan)

- veg.data
- env.data
- library (vegan)
- DCA <- decorana (veg = log1p (veg.data))
- ordiplot (DCA, display = 'sites', type = 'p')
- ordiplot (DCA, display = 'species', type = 't')





Unconstrained ordination

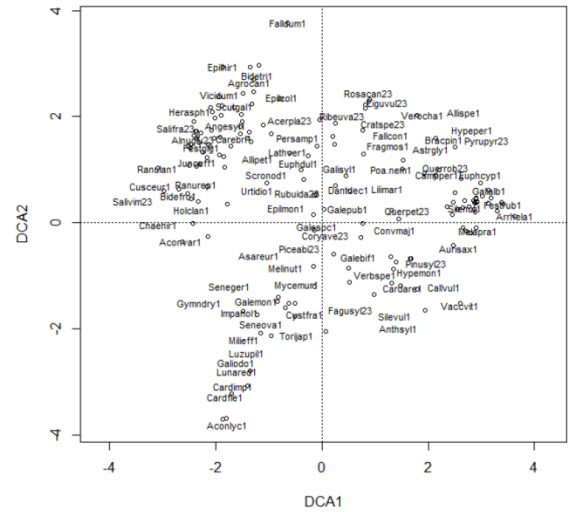
Ordination diagrams

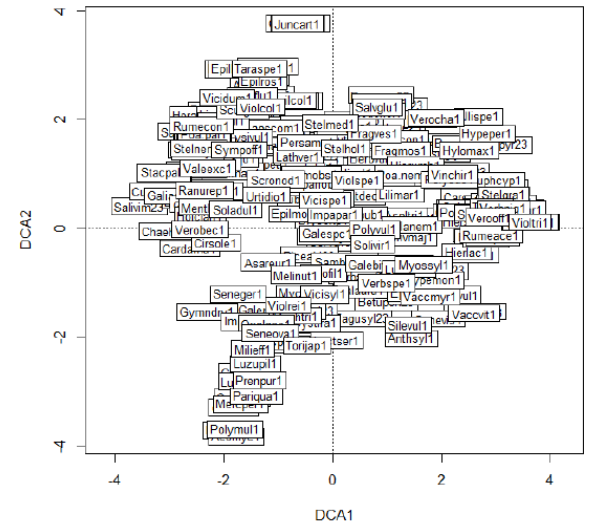
- Orditorp (library vegan)

› veg.data
› env.data
› library (vegan)
› DCA <- decorana (veg = log1p (veg.data))
› ordiplot (DCA, display = 'sp', type = 'n')
› orditorp (DCA, display = 'sp')

- Ordilable (library vegan)

› ordiplot (DCA, display = 'sp', type = 'n')
› ordilabel (DCA, display = 'sp')





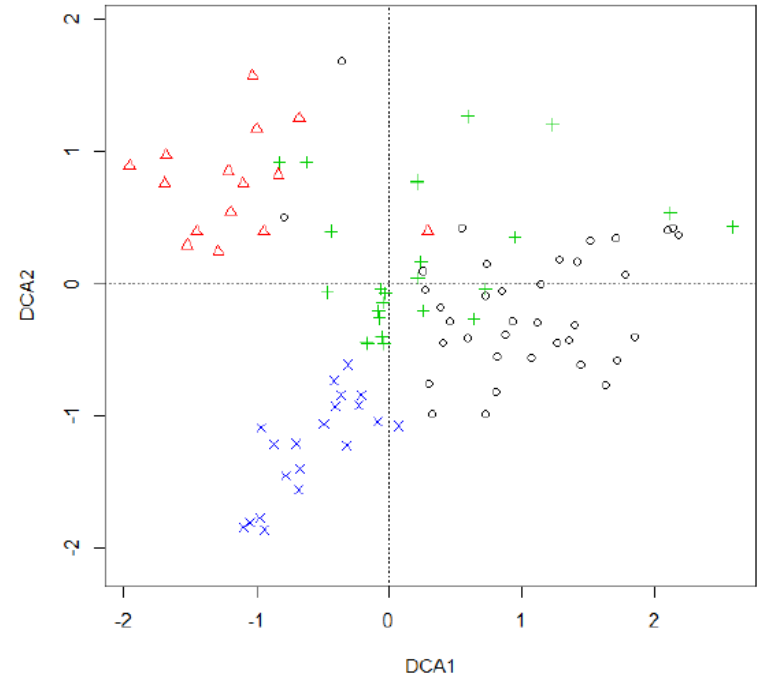
Unconstrained ordination

Ordination diagrams

- points (library vegan)

you can control their color, symbol, size etc. The following examples use the classification of samples (done by cluster analysis) into four groups, which is stored in variable GROUP in env.data data frame.

```
› veg.data
› env.data
› library (vegan)
› DCA <- decorana (veg = log1p (veg.data))
› ordiplot (DCA, display = 'si', type = 'n')
› points (DCA, col = env.data$GROUP, pch =
  env.data$GROUP )
```

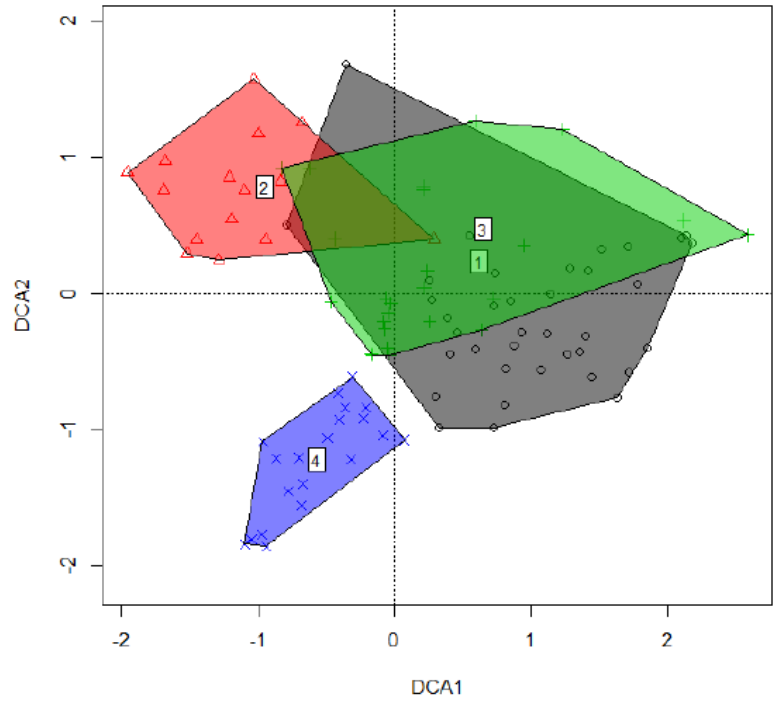


Unconstrained ordination

Ordination diagrams

- ordispider and ordihull (library vegan)

› ordiplot (DCA, display = 'si', type = 'n')
› points (DCA, col = env.data$GROUP, pch = env.data$GROUP)
› for (i in unique (env.data$GROUP)) ordihull (DCA, groups = env.data$GROUP, show.group = i, col = i, draw = 'polygon', label = T)



Unconstrained ordination

Supplementary variables in unconstrained ordination

- Environmental variables can be used as supplementary to interpret gradients in species composition recovered by axes of unconstrained ordination

R functions

- `envfit` (vegan) - fits supplementary variables on ordination scores, using multiple regression. Tests the significance of each variable using permutation test[1]. Wrapper for functions `vectorfit` and `factorfit` applied on quantitative or qualitative environmental variables, respectively. Results can be plotted onto ordination diagram using `plot` function.
- `scores` (vegan) - extract scores of samples or species on ordination axes.
- `ordisurf` (vegan) - projects supplementary environmental variable as a non-linear surface onto ordination diagram (using GAM model).
- `p.adjust.envfit` - home-made function (author: D. Zeleny) to adjust the p-values in the object created by the function `envfit`, using `p.adjust` function. Definition is here.

Unconstrained ordination

- vltava.spe
- vltava.env
- library (vegan)
- PCA <- rda (X = decostand (log1p (vltava.spe), method = 'hellinger'))
- ef <- envfit (PCA, vltava.env [, c('ASPSSW', 'SOILDPT', 'pH')], perm = 999)
- ef

```
ef.adj <- ef
pvals.adj <- p.adjust (ef$vectors$pvals, method = 'bonferroni')
ef.adj$vectors$pvals <- pvals.adj
ef.adj
```

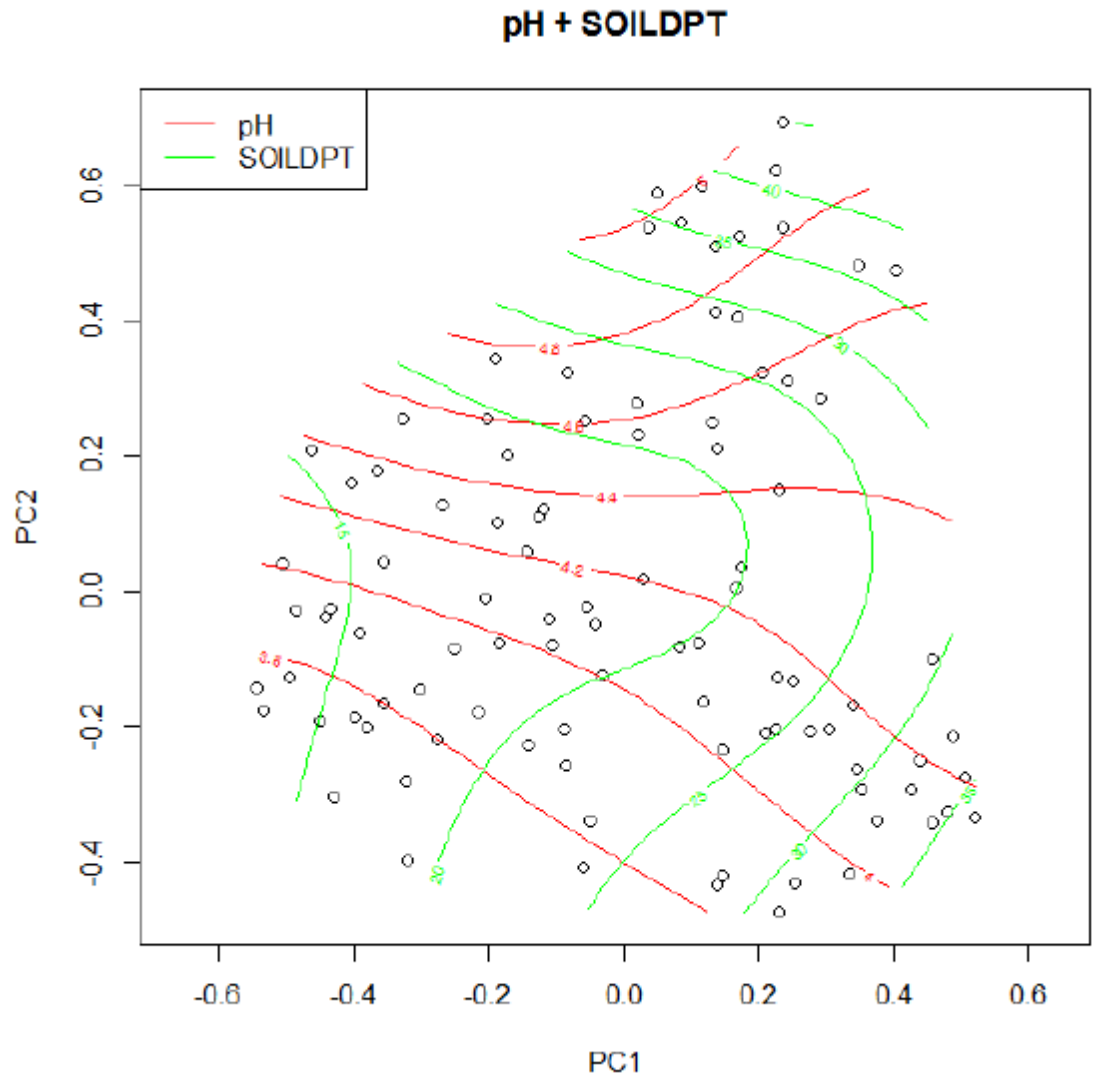
```
***VECTORS

            PC1      PC2     r2 Pr(>r)
ASPSSW  -0.87822  0.47826 0.4769  0.001 ***
SOILDPT  0.94893  0.31549 0.2891  0.001 ***
pH       0.38093  0.92460 0.3885  0.001 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
Permutation: free
Number of permutations: 999
```

The variable the most strongly related to the first two ordination axes is aspect, followed by pH and soil depth (judged by the value of r^2). pH is strongly related to the second ordination axis, while soil depth and xericity to the first (according to the coefficients in PC1 and PC2 columns).

Unconstrained ordination

Projecting environmental variable onto ordination as nonlinear surface

- ordisurf
  - ordisurf(PCA, vltava.env[,'pH'], main = 'pH + SOILDPT')
  - ordisurf(PCA, vltava.env[,'SOILDPT'], add = T, col = 'green')
  - legend('topleft', col = c('red','green'), lty = 1, legend = c('pH','SOILDPT'))



Unconstrained ordination

- **Use of mean Ellenberg indicator values as supplementary variables**

Constrained ordination

RDA (Redundancy Analysis)

- **matrix syntax**

✓ RDA = rda (Y, X, W)

✓ where Y is the response matrix (species composition), X is the explanatory matrix (environmental factors) and W is the matrix of covariables

- **formula syntax**

✓ RDA = rda (Y ~ var1 + factorA + var2*var3 + Condition (var4), data = XW)

✓ as explanatory are used: quantitative variable var1, categorical variable factorA, interaction term between var2 and var3, whereas var4 is used as covariable and hence partialled out.

Constrained ordination

- RDA (Redundancy Analysis)

```
› vasc <- read.delim()
› chem <- read.delim()
› library (vegan)
› vasc.hell <- decostand (vasc, 'hell')
› rda.vasc <- rda (vasc.hell ~ ., chem)
› RsquareAdj (rda.vasc)
```

Constrained ordination

- CCA (Canonical Correspondence Analysis)

R functions

- **`cca`** - this function calculates CCA if matrix of environmental variables is supplied (if not, it calculates CA).
- **`RsquareAdj`** - in case of CCA, it extracts only the value of R^2, while values of adjusted R^2 are not available (these need to be calculated by permutations and it is not available in R yet).
- **`anova.cca`** - tests the significance of the variation in species composition explained by explanatory variables, using Monte Carlo permutation test.

Constrained ordination

Monte Carlo permutation test

- The function of permutation test is to evaluate whether the variation explained by the model (variables) is higher than would be variation explained by the same number of randomly generated variables.

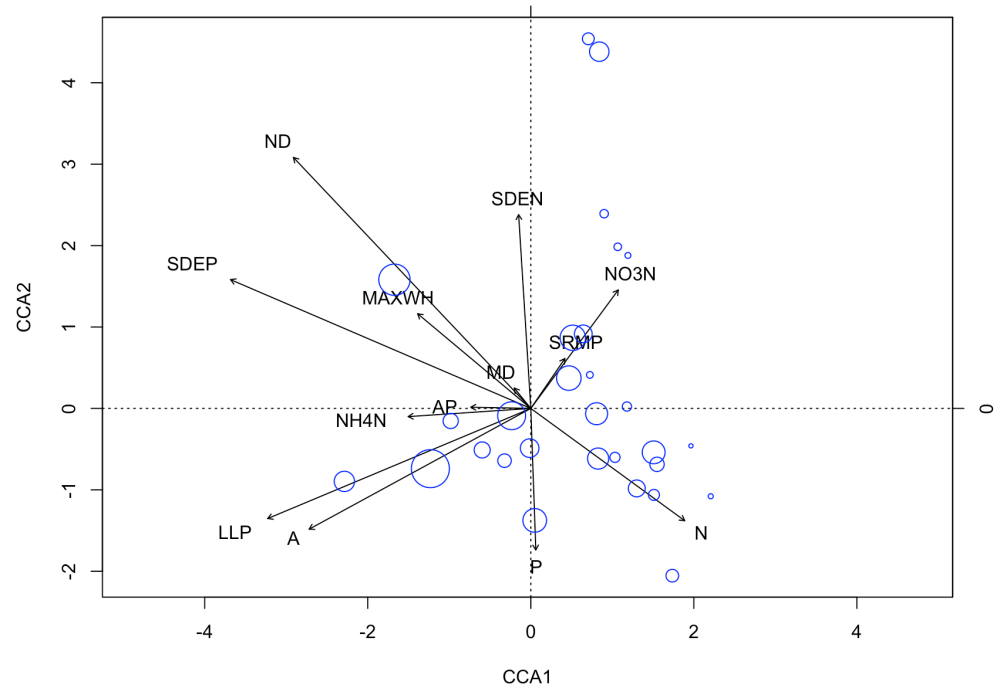
R functions

- **anova.cca** (library vegan) - tests the significance of the variation in species composition explained by explanatory variables, using Monte Carlo permutation test. It can test the significance of the global model (default), only the first constrained axis (adding argument `first = TRUE`), or significance of all axes (`by = "axis"`), significance of individual terms (explanatory variables) added sequentially (`by = "terms"`) or significance of variation explained by individual explanatory variables after removing variation of all other variables in the model (`by = "margin"`).

- anova.cca(rda.vasc)
- anova (rda.vasc, first = TRUE)
- anova (rda.vasc, by = 'axis', parallel = 4)
- anova (rda.vasc, by = 'terms', parallel = 4)
- anova (rda.vasc, by = 'margin', parallel = 4)

Plot CCA

```
plot(env.cca.all,type='n')
text(env.cca.all, display = 'cn') ###environmental vaiables
env2=read.table("env2.txt",header = T)
points(env.cca.all, pch=1,cex=env2$AG, col="blue") #sites with different shape and color
text(env.cca.all,'species') #species names
```



Variable selection

- **PCNM for spatial variables**

used to transform (spatial) distances to rectangular data that suitable for constrained ordination or regression.

- **Environmental variables**

✓ Removed correlated environmental factors

e.g. highly correlated with other factors ($r > 0.6$)

✓ PCA

- Forward selection

Forward selection was performed using a double stopping criteria in which new variables added to the model had to achieve a 0.05 α-level and the cumulative adjusted R^2 of the model could not exceed the adjusted R^2 of the model created from all variables (Blanchet et al. 2008). This reduced each set of variables to a more parsimonious set that retained nearly all of the original explanatory power (Blanchet et al. 2008, Borcard et al. 2011).

› 'forward.sel' in package *packfor*
› *ordistep (library vegan)*
› *ordiR2step (library vegan)*

PCNM

- The PCNM functions are used to express distances in rectangular form that is similar to normal explanatory variables used in, e.g., constrained ordination (rda, cca and capscale) or univariate regression (lm) together with environmental variables
- This is regarded as a more powerful method than forcing rectangular environmental data into distances and using them in partial mantel analysis (mantel.partial) together with geographic distances

- R code: pcnm() in vegan
  - pcnm(GeoDist)
  - pcnm(GeoDist)$vectors

Forward selection

forward.sel(Y, X, K = nrow(X) – 1, …)

- Y: A matrix of n lines and m columns that contains (numeric) response variable
- X: A matrix of n lines and p columns that contains (numeric) explanatory variables.

```
› library(packfor)
› library(picante)
› forward.sel(pcoa(Beta.bray)$vectors,pcnm(GeoDist)$vectors)###
› forward.sel(pcoa(Beta.bray)$vectors,Env[,1:7])
```

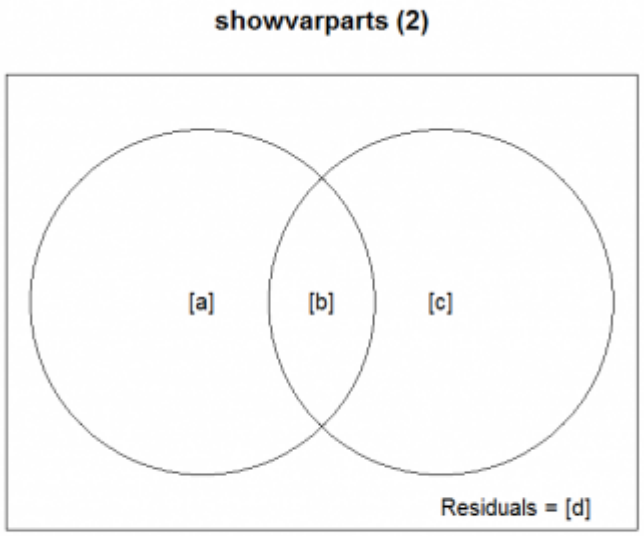
Constrained ordination

Variance partition

The explanatory environmental and PCNM variables retained in forward selection were then used to partition variation

R : 'varpart' in package *vegan*

›varpart (fertil.spe, ~ dose, ~ cover, data = fertil.env)
›plot (varp, digits = 2)




```
# fractions [a+b+c]:
rda.all <- rda (fertil.spe ~ dose + cover, data = fertil.env)
# fraction [a]:
rda.dose.cover <- rda (fertil.spe ~ dose + Condition (cover), data =
fertil.env)
# fraction [c]:
rda.cover.dose <- rda (fertil.spe ~ cover + Condition (dose), data =
fertil.env)
```

```
# fractions [a+b]:
rda.dose <- rda (fertil.spe ~ dose, data = fertil.env)
# fractions [b+c]:
rda.cover <- rda (fertil.spe ~ cover, data = fertil.env)
```

```
## fraction [a+b+c]
RsquareAdj (rda.all)
# $r.squared
# [1] 0.1430381
#
# $adj.r.squared
# [1] 0.1286354
```

```
anova (rda.all)

## fraction [b+c]
anova (rda.cover)
```

dbRDA

Distance-based redundancy analysis

- a method for carrying out constrained ordinations on data using non-Euclidean distance measures

① a distance matrix is calculated using the distance measure of choice

② a principle coordinates analysis (PCoA) is done on the matrix

③ the eigenvalues obtained in the PCoA are plugged into an RDA

- The proportion of variation explained is given as the adjusted R^2 of the explanatory variable set in the dbRDA, which is an unbiased estimator that corrects for the number of explanatory variables (Peres-Neto et al. 2006).

dbRDA

```
library(vegan)
dbRDA=capscale(species001 ~ MAT+TD+lnMAP+lnMSP, data=env, dist="bray")
##or dbRDA=capscale(beta ~ MAT+TD+lnMAP+lnMSP, data=env)
plot(dbRDA)
anova(dbRDA) ## overall test of the significance of the analysis
anova(dbRDA, by="axis", perm.max=500) ## test axes for significance
anova(dbRDA, by="terms", permu=200) ## test for sig. environ. Variables
RsquareAdj(capscale(beta ~ MAT+Condition(TD+lnMAP+lnMSP),
data=env))##Variance partition
```

You may want to do this analysis using a CCA instead of a RDA. The “capscale” function has a RDA set as the default but you can opt to use a CCA instead by adding “.cca” to your code. For example, “anova.cca(dbRDA)” instead of “anova(dbRDA)”. “?capscale” in R for the details.

MRM

Multiple Regression on distance Matrices

- permutation tests of significance for regression coefficients and R-squared.

```
› library(ecodist)
› MRM(Beta.jac~envdist+GeoDist+HisDist)#####R square is multiple value
› summary(lm(Beta.jac~envdist+GeoDist+HisDist)) ####Adusted R square
```

MFA

Multiple Factor Analysis

- ✓ similar to PCA but this method can treat variables of different groups with the same mathematical type
- ✓ infer the relative importance of each group of variables instead of single variables by using the RV coefficient

- "FactoMineR" package
- ✓ dedicated to multivariate data analysis

Numerical classification

Hierarchical agglomerative classification

- hclust

```
library (vegan)
dis <- vegdist (sqrt (vltava.spe), method = 'bray') # percentage cover data
are transformed by square root
cluster.single <- hclust (d = dis, method = 'single')
cluster.complete <- hclust (dis, 'complete')
cluster.average <- hclust (dis, 'average')
plot (cluster.single, main = 'Single linkage')
rect.hclust (cluster.single, k = 3) ####Divides dendrogram into given number of groups
```

Numerical classification

TWINSPAN: hierarchical divisive classification

```
› library (twinspanR)
› library (vegan)
› data (danube)
› res <- twinspan (danube$spe, modif = TRUE, clusters = 4)
› k <- cut (res)
› dca <- decorana (danube$spe)
› par (mfrow = c(1,2))
› ordiplot (dca, type = 'n', display = 'si', main = 'Modified TWINSPAN')
› points (dca, col = k)
› for (i in c(1,2,4)) ordihull (dca, groups = k, show.group = i, col = i,
› draw = 'polygon', label = TRUE)
› ordiplot (dca, type = 'n', display = 'si', main = 'Original assignment\n
› (Ellenberg 1954)')
› points (dca, col = danube$env$veg.type)
› for (i in c(1:3)) ordihull (dca, groups = danube$env$veg.type,
› show.group = unique (danube$env$veg.type)[i], col = i, draw = 'polygon', label = TRUE)
```

References

- David Zelený. **Analysis of community ecology data in R** http://www.davidzeleny.net/anadat-r/
- Mark Gardener. **Community Ecology**: Analytical Methods Using R and Excel
- Borcard D., Gillet G., Legendre P. **Numerical Ecology with R**