Constrained Ordination

Gavin Simpson

February 14, 2017

Summary

This practical will use the PONDS dataset to demonstrate direct gradient analysis (CCA) of species and environmental data. The file pondsenv.csv contains the species data (48 taxa) and pondsenv.csv contains the transformed environmental variables (15 variables) for 30 sites. You will use R and the `vegan` package to analyse these data using a variety of direct ordination graphical display techniques.

1 Canonical Correspondence Analysis

In this part of the practical you will use the `cca()` function from package `vegan` to perform a canonical correspondence analysis (CCA) of the diatom species data and the enviromental data. Begin by loading `vegan` and reading in the data sets:

```
> library(vegan)
> diat <- read.csv("ponddiat.csv")
> env <- read.csv("pondsenv.csv")
```

The ponds are numbered numerically in the form X???, the species are encoded using DIATCODE and environmental variable codes should be self explanatory:

```
> rownames(diat)

 [1] "X004" "X007" "X031" "X034" "X037" "X042" "X050" "X053"
 [9] "X057" "X058" "X065" "X069" "X073" "X074" "X076" "X079"
[17] "X082" "X083" "X085" "X086" "X098" "X100" "X101" "X105"
[25] "X107" "X108" "X112" "X113" "X114" "X120"

> names(diat)

 [1] "AC001A" "AC013A" "AC013E" "AM011A" "AM012A" "AS001A"
 [7] "AU002A" "AU003B" "CC001A" "CC002A" "CC9997" "CM004A"
[13] "CO001A" "CY002A" "CY003A" "CY009A" "CY011A" "FR001A"
[19] "FR002A" "FR002C" "FR006A" "FR006E" "FR009B" "FR018A"
[25] "FR019A" "GO013A" "NA004A" "NA007A" "NA022A" "NA042A"
[31] "NA114A" "NI009A" "NI014A" "NI015A" "NI083A" "NI196A"
[37] "NI9969" "NI9971" "OP001A" "ST001A" "ST002A" "ST010A"
[43] "SU016A" "SY002A" "SY003A" "SY003C" "SY010A" "UN9992"

> names(env)

 [1] "pH"           "Conductivity" "Alkalinity"   "TP"
 [5] "SiO2"         "NO3"          "Na"           "K"
 [9] "Mg"           "Ca"           "Cl"           "SO4"
[13] "Chla"         "Secchi"       "MaxDepth"
```

vegan has a nice formula interface, so it works in a similar way to the notation you used in the two regression practical classes. To fit a CCA model to the diatom and environmental data use the `cca()` function:

```
> ponds.cca <- cca(diat ~ ., data = env)
> ponds.cca
```

```
Call: cca(formula = diat ~ pH + Conductivity + Alkalinity
+ TP + SiO2 + NO3 + Na + K + Mg + Ca + Cl + SO4 + Chla +
Secchi + MaxDepth, data = env)

              Inertia Proportion Rank
Total           5.812      1.000
Constrained     3.273      0.563   15
Unconstrained   2.539      0.437   14
Inertia is mean squared contingency coefficient

Eigenvalues for constrained axes:
 CCA1  CCA2  CCA3  CCA4  CCA5  CCA6  CCA7  CCA8  CCA9 CCA10
0.595 0.406 0.338 0.324 0.303 0.238 0.206 0.190 0.151 0.133
CCA11 CCA12 CCA13 CCA14 CCA15
0.126 0.100 0.080 0.053 0.030

Eigenvalues for unconstrained axes:
  CA1   CA2   CA3   CA4   CA5   CA6   CA7   CA8   CA9  CA10
0.445 0.316 0.287 0.246 0.239 0.205 0.174 0.149 0.119 0.103
 CA11  CA12  CA13  CA14
0.085 0.084 0.052 0.035
```

The formula means the diatom data matrix `diat` is modelled by everything in the environmental data matrix `env`. We use the data argument to inform R of where to look for the explanatory variables. The "`.`" is an R shortcut that saves you from having to type out the full formula.

Q and A

1. What are the values of λ_1 and λ_1, the eigenvalues for constrained axes one and two?
2. What is the total variance (inertia) in the diatom data?
3. What proportion of the total variance is explained by the environmental variables?
4. What proportion of the variance remains un-explained?

The `summary()` method provides further, detailed results of the CCA:

```
> summary(ponds.cca)

Call:
cca(formula = diat ~ pH + Conductivity + Alkalinity + TP + SiO2 +      NO3 + Na + K + Mg + Ca + Cl + SO

Partitioning of mean squared contingency coefficient:
              Inertia Proportion
Total            5.81      1.000
Constrained      3.27      0.563
Unconstrained    2.54      0.437

Eigenvalues, and their contribution to the mean squared contingency coefficient

Importance of components:
                        CCA1   CCA2   CCA3   CCA4   CCA5  CCA6
Eigenvalue            0.595 0.4057 0.3380 0.3235 0.3027 0.238
....
Cumulative Proportion 0.5489 0.55800 0.5631 0.6396 0.6940 0.7433
                         CA4    CA5    CA6    CA7    CA8    CA9
Eigenvalue            0.2464 0.2391 0.2046 0.1739 0.1492 0.1191
Proportion Explained  0.0424 0.0411 0.0352 0.0299 0.0257 0.0205
Cumulative Proportion 0.7857 0.8268 0.8620 0.8919 0.9176 0.9381
                        CA10   CA11   CA12    CA13    CA14
Eigenvalue            0.1029 0.0855 0.0842 0.05183 0.03540
Proportion Explained  0.0177 0.0147 0.0145 0.00892 0.00609
```

```
Cumulative Proportion 0.9558 0.9705 0.9850 0.99391 1.00000

Accumulated constrained eigenvalues
Importance of components:
....
GO013A  0.24656 -0.3739 -0.4477 -0.6555  0.13995  0.5494
NA004A  0.23280 -0.3111  0.2488 -0.3424 -0.29339  0.1527
NA007A  0.29271 -0.7319  0.1229 -0.3692  0.02761  0.5166
NA022A  0.52985 -0.6343 -0.0954 -0.3280 -0.56960  0.0716
NA042A  0.38264 -0.5795  0.4819 -0.2279 -0.12173  0.4246
NA114A  0.21048 -0.4050  0.1963 -0.4150  0.32425  0.2338
NI009A  0.61745 -0.3647  0.1407 -0.0338  0.03198 -0.2747
....
X053 -0.1725 -0.5003  0.3191 -0.5665 -0.4865  0.65538
X057  0.9142 -1.1549  1.3368 -1.2761  3.5393 -1.56745
X058  0.5654 -0.1994  0.2256 -1.1450 -0.8957 -0.43419
X065 -1.6211 -0.4601  1.9133 -0.4862  0.5156 -0.20015
X069 -0.7930 -0.7882  0.5006 -0.0760  0.3343  0.37013
X073 -0.0949 -0.9835  0.2648 -0.3931  0.6266  0.23900
X074 -0.2656 -0.8484 -0.2802 -1.2193 -0.2225  1.15417
....
X053 -0.1218 -1.6078  1.2768  0.0236 -0.4576  0.509
X057  0.8768 -0.8923  0.8277 -0.5631  2.4478 -0.955
X058  0.2281 -0.2825  0.1884 -0.8765 -0.4436  0.167
X065 -1.8824 -0.2748  1.6694 -0.3020  0.3299 -0.772
X069 -0.3668 -1.1997  0.3089 -0.1174  0.0795  0.897
X073  0.0323 -0.0351  0.6231 -0.9520 -0.3776  0.313
X074 -0.2434 -0.6792 -0.5866 -1.5369  0.1345  0.175
```

Q and A

1. Why are there two sets of site scores?
2. Look at the biplot scores in the summary output. Suggest which variables are important on CCA axes 1 and on CCA axis 2?

The `plot()` method is used to produce a triplot/biplot of the ordination results. Plot a triplot of the CCA of the `ponds` data (Figure 1).

```
> plot(ponds.cca)
```

Q and A

1. Using the triplot, the biplot scores of the enviromental variables and the ordination axes, interpret the axes in terms of environmental gradients.
2. Indicate which species are characteristic of particular types of water.

1.1 Comparison with un-constrained methods

Perform a CA and a DCA of the ponds diatom data:

```
> ponds.ca <- cca(diat)
> ponds.ca

Call: cca(X = diat)

              Inertia Rank
Total            5.81
Unconstrained    5.81   29
Inertia is mean squared contingency coefficient
```

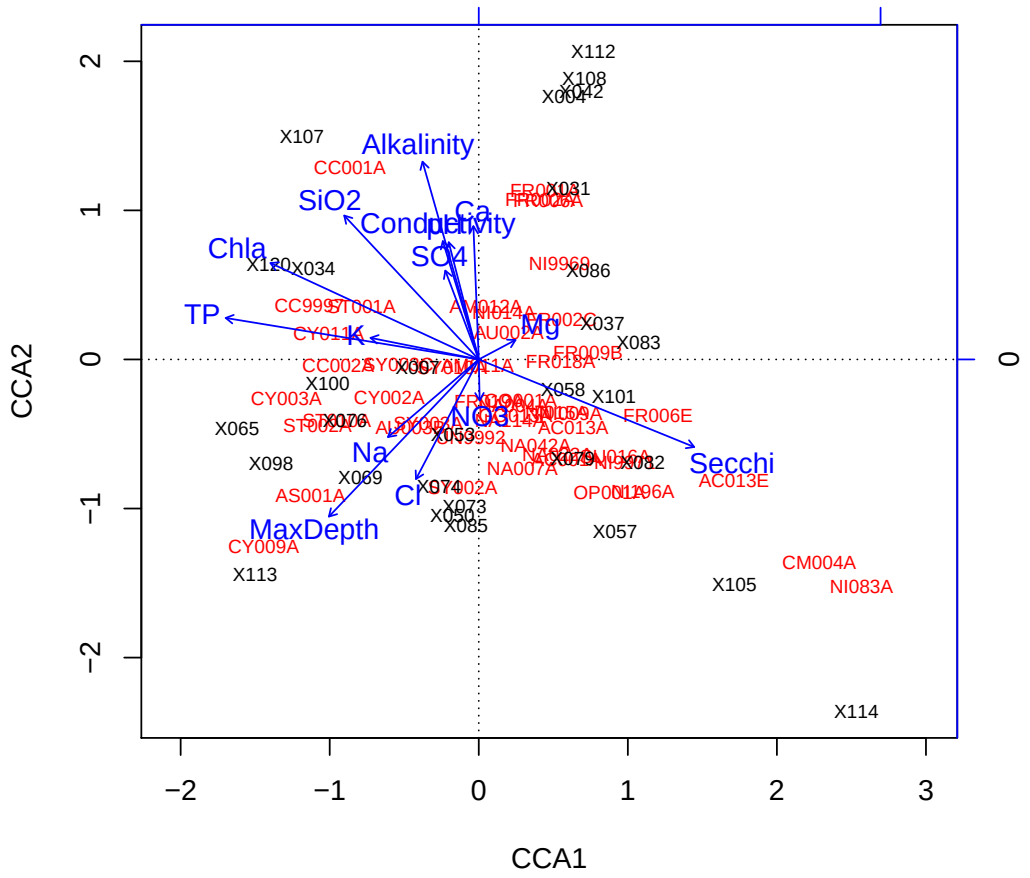

Figure 1: Triplot of the CCA of the Ponds diatom and hydrochemistry data.

```
Eigenvalues for unconstrained axes:
  CA1   CA2   CA3   CA4   CA5   CA6   CA7   CA8
0.678 0.573 0.486 0.402 0.397 0.386 0.363 0.312
(Showed only 8 of all 29 unconstrained eigenvalues)

> ponds.dca <- decorana(diat)
> ponds.dca

Call:
decorana(veg = diat)

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

                 DCA1  DCA2  DCA3  DCA4
Eigenvalues     0.675 0.366 0.331 0.291
Decorana values 0.678 0.441 0.315 0.208
Axis lengths    3.963 3.440 2.771 2.548
```

Refer to the handout on indirect ordination for hints and answer the following question:

Q and A

1. How does the result of the CCA compare to the results of the CA and DCA?

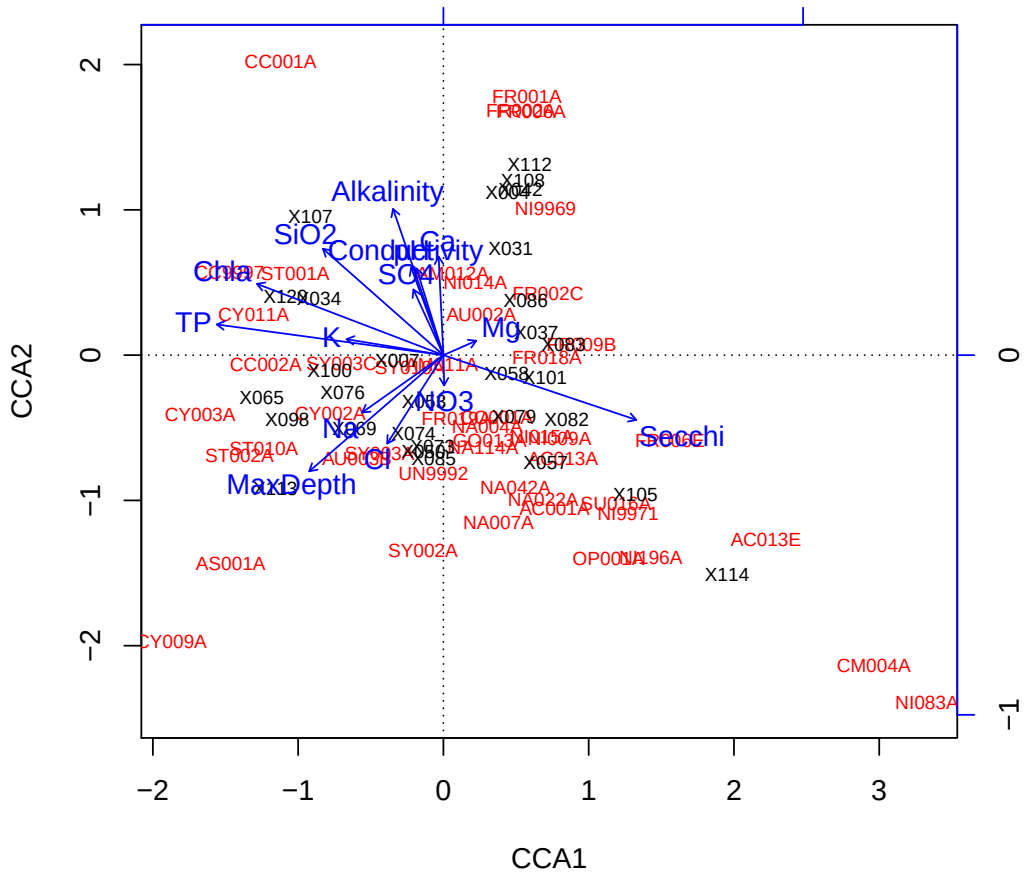

Figure 2: Triplot of the CCA of the Ponds diatom and hydrochemistry data using `scaling = 1`.

2. Plot the CA/DCA biplots are compare the configuration of sites in these biplots to the one shown in the CCA triplot. Do they suggest that our measured environmental variables explain the main floristic gradients in the diatom data?

So far, you have used the default scaling (`scaling = 2`) for the plots and summaries. Redraw the triplot using `scaling = 1`, to draw a triplot where the site scores are weighted averages of the species scores:

```
> plot(ponds.cca, scaling = 1)
```

Q and A

1. What effect does the choice of scaling have on the ordination plots?

1.2 Interpretting the CCA results

There is a lot more that can be done to interpret the results of the CCA and explore relationships between the diatom species and the environmental variables, as well as determining model performance for the CCA itelf.

1.2.1 Outliers?

One useful diagnostic for the configuration is to identify outlier or "odd" sites or species. Plotting the Hills's N_2 values for both species and samples can help visualise outliers. We can produce biplots using N_2 values for species and sites easily in R using `renyi()` in `vegan`. First we calculate the N_2 values for sites and species:

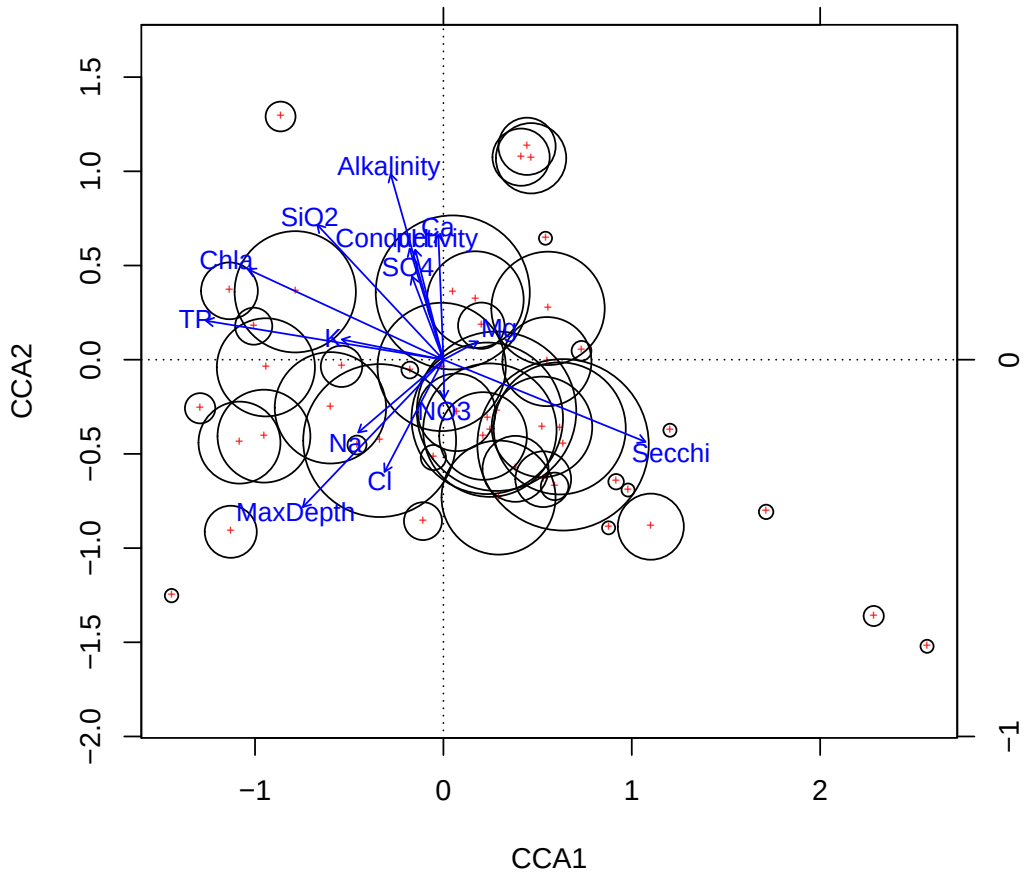

Figure 3: Biplot showing the species:environment relationships. Species symbols are scaled by Hill's N_2.

```
> diat.n2 <- renyi(t(diat), scales = 2, hill = TRUE)
> ponds.n2 <- renyi(diat, scales = 2, hill = TRUE)
```

Then we use these values to scale the plotting symbol used to display the sites of the species and use `identify()` to label the outlier spescies/sites (remember to click on some species [red crosses] to label them and right click on the graph to finish). Firstly for the species:

```
> sppN2 <- plot(ponds.cca, display = "species", type = "n")
> points(ponds.cca, display = "species", pch = "+", col = "red", cex = 0.5)
> symbols(scores(ponds.cca)$species, circles = diat.n2,
+         add = TRUE, inches = 0.5)
> text(ponds.cca, display = "bp", arrow.mul = 2,
+      col = "blue", cex = 0.9)
> identify(sppN2, what = "species", ps = 10)
```

... and for the sites:

```
> siteN2 <- plot(ponds.cca, display = "sites", type = "n")
> points(ponds.cca, display = "sites", pch = "+", col = "red", cex = 0.5)
> symbols(scores(ponds.cca)$sites, circles = ponds.n2,
+         add = TRUE, inches = 0.5)
> text(ponds.cca, display = "bp", arrow.mul = 2,
+      col = "blue", cex = 0.9)
> identify(siteN2, what = "sites", ps = 10)
```

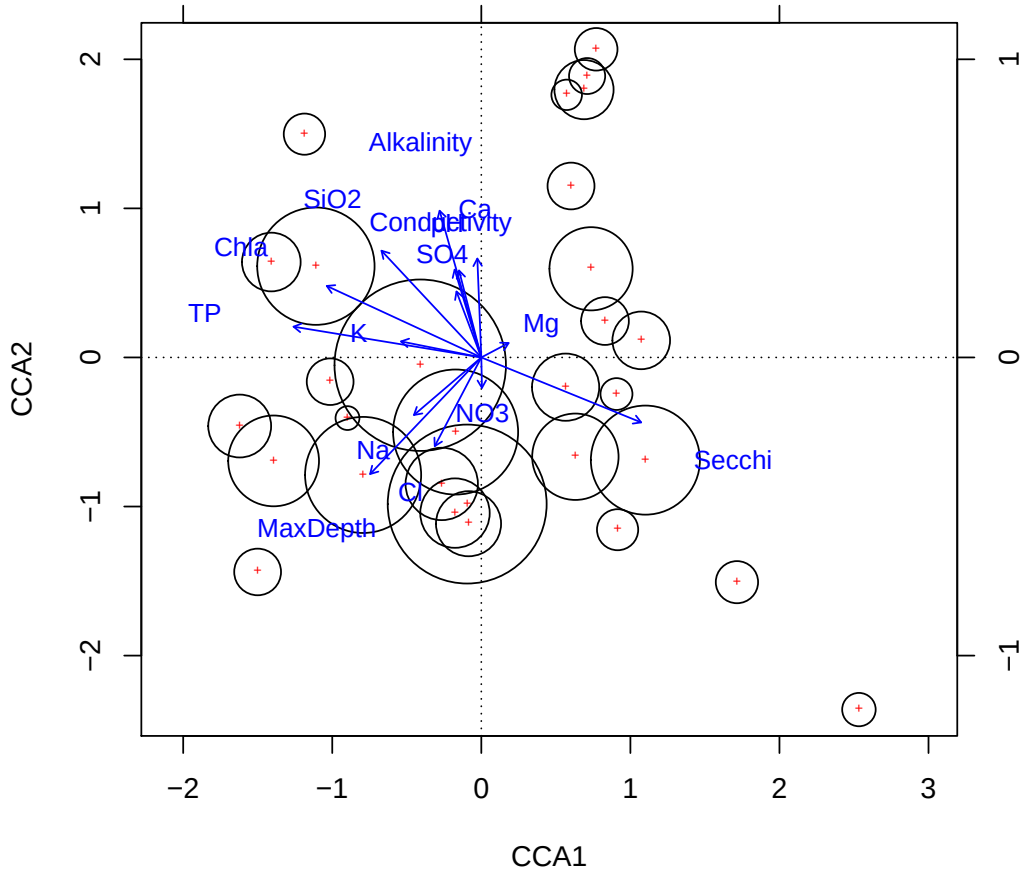

Figure 4: Biplot showing the site:environment relationships. Sites symbols are scaled by Hill's N_2.

To help interpret these plots, we add the species/site labels to the species/site Hill's N_2 values and print them to the screen.

```
> names(diat.n2) <- colnames(diat)
> sort(diat.n2, decreasing = TRUE)

AC013A CO001A AM012A SY003A NA004A GO013A NI009A AM011A ST001A
13.087 12.172 11.761 11.671 11.565 10.186 10.082  9.795  9.259
NA007A FR002C CY002A NI015A CC002A NI014A ST010A FR018A NA114A
 8.703  8.655  8.503  7.677  7.491  7.379  7.088  6.799  6.691
ST002A FR019A FR006A NI196A NA042A FR001A FR002A CC9997 NA022A
 6.257  5.921  5.371  5.045  5.035  4.369  4.368  4.337  4.258
AS001A AU002A SY003C SY002A CY011A CY003A CC001A AC001A UN9992
 3.973  3.551  3.168  2.901  2.837  2.317  2.284  2.112  1.928
CM004A FR009B AU003B SY010A SU016A AC013E CY009A FR006E NI083A
 1.541  1.510  1.449  1.284  1.183  1.103  1.029  1.000  1.000
NI9969 NI9971 OP001A
 1.000  1.000  1.000

> names(ponds.n2) <- rownames(diat)
> sort(ponds.n2, decreasing = TRUE)

  X007   X073   X053   X034   X069   X082   X098   X079   X086
14.872 13.789 10.847 10.180 10.070  9.463  7.877  7.493  7.219
  X074   X050   X058   X085   X065   X042   X120   X083   X037
```

```
 6.278  6.022  5.830  5.647  5.454  5.133  5.091  5.001  4.144
  X031   X113   X100   X112   X105   X107   X057   X108   X114
 4.070  4.060  4.057  3.705  3.669  3.589  3.575  3.132  2.905
  X101   X004   X076
 2.772  2.650  2.074
```

Q and A

1. Using the Hill's N_2 plots and the actual N_2 values for the sites and species, which species are abundant and which are rare in the Ponds diatom data set?
2. Which of the sites have low species diversity and which high diversity?

1.2.2 How significant are the constraints?

The CCA model we have built is a weighted, multivariate multiple regression and just as in regression, we want to achieve as parsimonious a model as possible, one that adequately describes the species environmental relationships without being overly complex. Whilst it is common for users to throw as many constraints as possible at a CCA this has the effect of *reducing* the contraints on the ordination (it becomes more like the CA the more constraints you use) and of building an overly complex model that is over fitted to that particular data set. In this section you will look at some of the model building/selection tools available within R and `vegan`.

Firstly, we should look for redundant constraints—environmental variables that are highly corellated with each other are prime candidates for exclusion from the model. Produce a corellation matrix of the environmental data set and calculate the variance inflation factors for each variable.

```
> cor(env) # output not shown in handout
```

```
> vif.cca(ponds.cca)
```

```
          pH Conductivity   Alkalinity           TP          SiO2
       7.239       30.262       16.345       10.386        5.019
         NO3           Na            K           Mg           Ca
       2.180       43.888        8.870       23.695        6.633
          Cl          SO4         Chla      Secchi     MaxDepth
      36.753       18.664        3.755        2.166        2.169
```

Q and A

1. Suggest which variables might be redundant and therefore dropped from the CCA model?

We should also check the significance of the full CCA model we have fit. This is done using the `anova()` function:

```
> anova(ponds.cca)
```

```
Permutation test for cca under reduced model
Permutation: free
Number of permutations: 999

Model: cca(formula = diat ~ pH + Conductivity + Alkalinity + TP + SiO2 + NO3 + Na + K + Mg + Ca + Cl + S
          Df ChiSquare   F Pr(>F)
Model     15      3.27 1.2   0.02 *
Residual 14      2.54
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Note that this uses random permutations so your P-value may vary.

Q and A

1. Is the full model significant at the 0.01 level?

Canoco also reports the species:environment correlation—the correlation between the sites scores that are weighted averages of the species scores and the site score that are linear combinations of the environmental data. Function `spenvcor()` calculates the correlation between the two sets of site scores.

```
> spenvcor(ponds.cca)

  CCA1   CCA2   CCA3   CCA4   CCA5   CCA6   CCA7   CCA8   CCA9
0.9714 0.8872 0.9369 0.8923 0.8996 0.8931 0.9119 0.8690 0.7703
 CCA10  CCA11  CCA12  CCA13  CCA14  CCA15
0.8615 0.7336 0.6773 0.7731 0.6087 0.6064
```

Q and A

1. Are there high correlations between the two sets of site scores?

2. What does this tell you about the relationships between the species and the environmental data?

1.2.3 Fowards selection and backwards elimination

Whilst automated model building methods are not the panacea that many people think they are, they can be a useful aid when model building with lots of environmental variables.

The model selection tools available in `vegan` are different to those available in CANOCO, and are based on the concept of AIC, a fairly new concept for CCA as CCA is not based on concepts of deviance and log likelihoods (from which AIC was derived). Instead features of the CCA results are converted into a deviance by calculating the Chi-square of the residual data matrix after fitting constraints (in RDA, deviance is taken to be the residual sum of squares instead). From here an AIC statistic can be calculated, the details of which are given in the reference quoted in the help page for `deviance.cca()` (type `?deviance.cca` at the R prompt to read this page if you so wish).

Before we begin, note that the author of `vegan`, Jari Oksanen, is not convinced about all aspects of this approach, and advocates checking the results manually—which is good advice seeing as you should not be relying on automated model selection tools anyway!

To begin, define a null model to which we will sequentially add variables in order of added importance:

```
> mod0 <- cca(diat ~ 1, data = env)
> mod0

Call: cca(formula = diat ~ 1, data = env)

              Inertia Rank
Total            5.81
Unconstrained    5.81   29
Inertia is mean squared contingency coefficient

Eigenvalues for unconstrained axes:
  CA1   CA2   CA3   CA4   CA5   CA6   CA7   CA8
0.678 0.573 0.486 0.402 0.397 0.386 0.363 0.312
(Showed only 8 of all 29 unconstrained eigenvalues)
```

As you can see, this is an unconstrained model or a CA. Function `step.cca()` is used to *step* forwards or backwards through a series of nested models adding or dropping an explanatory variable at each iteration[1]. To use `step()` we need to define an upper and lower scope for the stepping to place over. We will use `mod0` as the lower scope and `ponds.cca` (the full model) as the upper scope when performing forward selection—this is reversed when performing backwards elimination.

```
> mod <- step(ponds.cca, scope = list(lower = formula(mod0),
+                             upper = formula(ponds.cca)))
```

You should have seen lots of output false across the screen. At each stage, the effect of adding/dropping a variable is evaluated in terms of AIC and the variables ordered by AIC. Low AIC values are preferred, and if a lower AIC can be achieved by adding or removing a variable at a stage then this variable is added/deleted and the procedure repeats, this time using the new formula as the starting point. In the above example, we used both forwards and backwards elmination at each step.

Print out the record of the steps:

[1] `step()` is a generic function and `step` methods can be written for different modelling functions. This means you only need to use the generic `step()` and R take care of finding and using the correct method for the object you are running `step()` on. Another example of a generic is `anova()`, which you used earlier—what you actually used was `anova.cca()`

```
> mod$anova

             Step Df Deviance Resid. Df Resid. Dev   AIC
1                 NA       NA        14       5817 190.0
2            - Ca  1    272.9        15       6090 189.4
3  - Conductivity  1    314.1        16       6404 188.9
4          - Chla  1    325.5        17       6730 188.4
5    - Alkalinity  1    447.4        18       7177 188.3
6            - Na  1    492.7        19       7670 188.3
7            - Cl  1    313.1        20       7983 187.5
8           - SO4  1    473.5        21       8456 187.2
9          - SiO2  1    420.0        22       8876 186.7
10          - NO3  1    518.2        23       9395 186.4
11           - Mg  1    534.7        24       9929 186.1
12            - K  1    469.4        25      10399 185.4
13           - pH  1    528.3        26      10927 184.9
14           - TP  1    671.5        27      11599 184.7
```

We see that we started with the full model and calcium was dropped from the full model. Next Conductivity was dropped and so on, with TP being the last variable dropped. At no stage was a variable added back into the model. To view the final model simply type:

```
> mod

Call: cca(formula = diat ~ Secchi + MaxDepth, data = env)

              Inertia Proportion Rank
Total           5.812      1.000
Constrained     0.749      0.129    2
Unconstrained   5.063      0.871   27
Inertia is mean squared contingency coefficient

Eigenvalues for constrained axes:
 CCA1  CCA2
0.461 0.289

Eigenvalues for unconstrained axes:
  CA1   CA2   CA3   CA4   CA5   CA6   CA7   CA8
0.595 0.496 0.446 0.395 0.385 0.323 0.296 0.280
(Showed only 8 of all 27 unconstrained eigenvalues)
```

The final model contains two variables—secchi disk depth and maximum pond depth. Test this model and see how significant the effects of the constraints are:

```
> anova(mod)

Permutation test for cca under reduced model
Permutation: free
Number of permutations: 999

Model: cca(formula = diat ~ Secchi + MaxDepth, data = env)
         Df ChiSquare  F Pr(>F)
Model     2      0.75  2  0.001 ***
Residual 27      5.06
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Q and A

1. Is this model better than full model?
2. How much of the total inertia is explained by the two constraints?

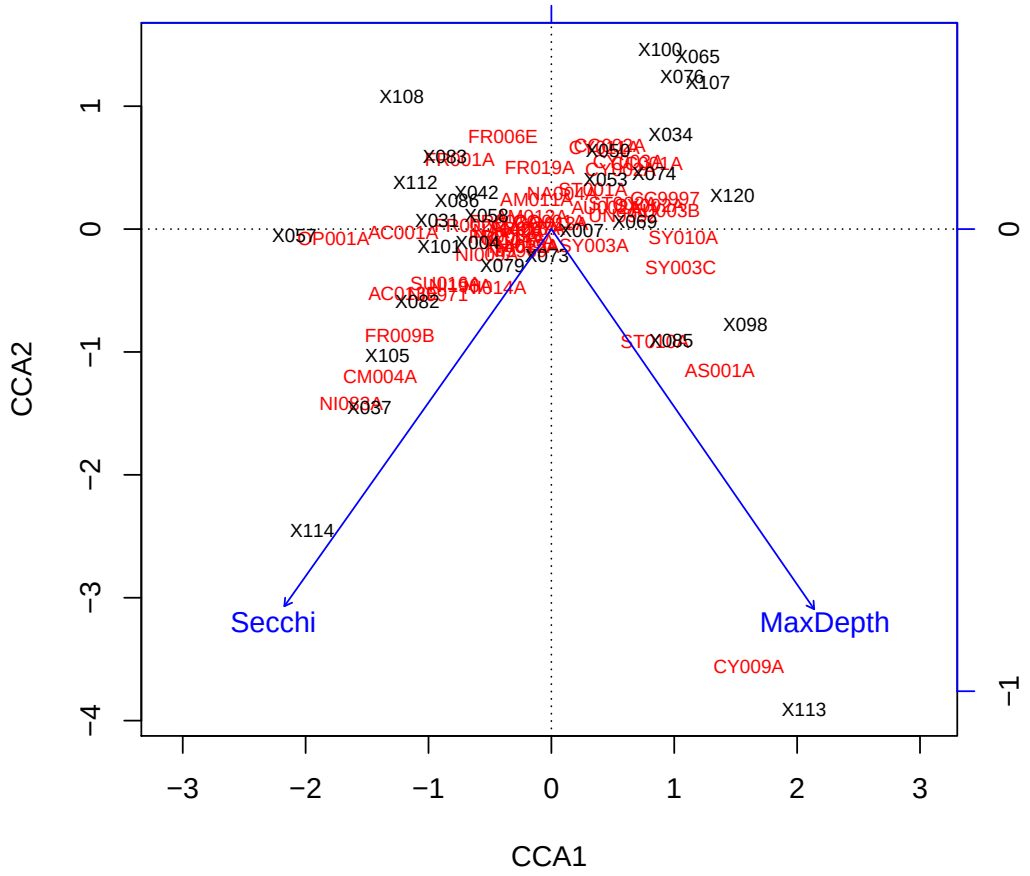

Figure 5: Triplot of the CCA of the Ponds diatom and hydrochemistry data after forwards selection and backwards elimination.

Produce a triplot of this model:

```
> plot(mod)
```

The triplot suggests that there is a strong outlier site in terms of maximum depth (Pond X113). We might wish to investigate how the CCA model might change if we deleted this observation. We delete this observation and build new null and full CCA models

```
> no.need <- which(rownames(diat) == "X113")
> diat2 <- diat[-no.need, ]
> env2 <- env[-no.need, ]
> mod0 <- cca(diat2 ~ 1, data = env2)
> cca.delete <- cca(diat2 ~ ., data = env2)
```

We can now retry the automatic stepping model selection and plot the resulting triplot:

```
> mod.delete <- step(cca.delete, scope = list(lower = formula(mod0),
+                    upper = formula(cca.delete)))
> plot(mod.delete)
```

Q and A

1. How does this model compare to the model with MaxDepth and Secchi only?

A further thing we should check is whether we get different models whether we do forward selection, backward elimination or both. The default for `step()` is to evaluate both forward and backward steps. If we wish to perform forward selection only, we need to tell R to start from the null model:

```
> mod.fwd<- step(mod0, scope = list(lower = formula(mod0),
+                    upper = formula(cca.delete)))
> plot(mod.fwd)
```

Q and A

1. Which variables has forward selection chosen?

This highlights one of the problems with automatic model building tools. As a description of the data, `mod.delete` seems a nicer plot, but it retains a number environmental variables that are very correlated. Forward selection produces a model with a single environmental variables. So which to use? And therein lies the problem. There is no substitution for rolling up ones sleeves and getting involved in building and checking lots of candidate models.

As a starter, we could look at the significance of the terms in `mod.delete`:

```
> anova(mod.delete, by = "terms")

Permutation test for cca under reduced model
Terms added sequentially (first to last)
Permutation: free
Number of permutations: 999

Model: cca(formula = diat2 ~ pH + Alkalinity + TP + NO3 + Na + K + Mg + Cl + SO4 + Secchi + MaxDepth, da
           Df ChiSquare    F Pr(>F)
pH          1     0.239 1.37  0.071 .
Alkalinity  1     0.229 1.31  0.119
TP          1     0.368 2.11  0.001 ***
NO3         1     0.235 1.34  0.080 .
Na          1     0.165 0.94  0.538
K           1     0.181 1.03  0.397
Mg          1     0.206 1.18  0.233
Cl          1     0.243 1.39  0.060 .
SO4         1     0.190 1.09  0.351
Secchi      1     0.332 1.90  0.003 **
MaxDepth    1     0.223 1.27  0.107
Residual   17     2.971
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Here, the significance of terms are assessed sequentially from first to last. A number of the environmental variables are not significant under this test. As a strategy for producing a parsimonius model, we might proceed by removing the variable that contributes the least here, Na.

Q and A

1. As an exercise if you have time, try dropping out terms and rerun `anova` to try to produce a parsimonious model.

1.3 Partial CCA models

There are occaisions where we might wish to fit a model to our species data after controlling for the effects of one or more environmental variables. These models are known as partial constrained ordinations—the effect of the one or more environmental variables are partialled out, and a CCA/RDA model is applied to explain the residual variation.

In `vegan` partial models are fitted using the `Condition()` function within the model formula describing the model you wish to fit. The `Condition()` function is used to *condition* the model on the set of covariables and fit a model to the residuals of the conditioned model. Multiple variables can be included within `Condition()`, separated by a "+". Partial models can also be used to evaluate the significance of adding a new variable to a

model already containing one or more variables—partial out the existing variables and fit a model with the new variable of interest, using `anova()` to assess the effect of adding this new variable.

Say we were interested in investigating the effects of the hydrochemical variables on diatom distributions in the Ponds dataset, after controlling for the effects of MaxDepth and Secchi, we would fit this model in R like so:

```
> partial.mod <- cca(diat ~ . + Condition(MaxDepth + Secchi), data = env)
> partial.mod

Call: cca(formula = diat ~ pH + Conductivity + Alkalinity
+ TP + SiO2 + NO3 + Na + K + Mg + Ca + Cl + SO4 + Chla +
Secchi + MaxDepth + Condition(MaxDepth + Secchi), data =
env)

              Inertia Proportion Rank
Total           5.812      1.000
Conditional     0.749      0.129    2
Constrained     2.524      0.434   13
Unconstrained   2.539      0.437   14
Inertia is mean squared contingency coefficient
Some constraints were aliased because they were collinear (redundant)

Eigenvalues for constrained axes:
 CCA1  CCA2  CCA3  CCA4  CCA5  CCA6  CCA7  CCA8  CCA9 CCA10
0.397 0.369 0.328 0.277 0.206 0.205 0.163 0.151 0.146 0.105
CCA11 CCA12 CCA13
0.080 0.054 0.042

Eigenvalues for unconstrained axes:
  CA1   CA2   CA3   CA4   CA5   CA6   CA7   CA8   CA9  CA10
0.445 0.316 0.287 0.246 0.239 0.205 0.174 0.149 0.119 0.103
 CA11  CA12  CA13  CA14
0.085 0.084 0.052 0.035

> anova(partial.mod)

Permutation test for cca under reduced model
Permutation: free
Number of permutations: 999

Model: cca(formula = diat ~ pH + Conductivity + Alkalinity + TP + SiO2 + NO3 + Na + K + Mg + Ca + Cl + S
         Df ChiSquare    F Pr(>F)
Model    13      2.52 1.07   0.21
Residual 14      2.54
```

Q and A

1. Do the remaining environmental variables explain significant amounts of the variance in the species data after controlling for MaxDepth and Secchi?
2. How much of the variance is explained by the Conditional variables?
3. How much of the variance is explained by the constraints?
4. How much is left unexplained?

Finally, plot a triplot for this model:

```
> plot(partial.mod)
```

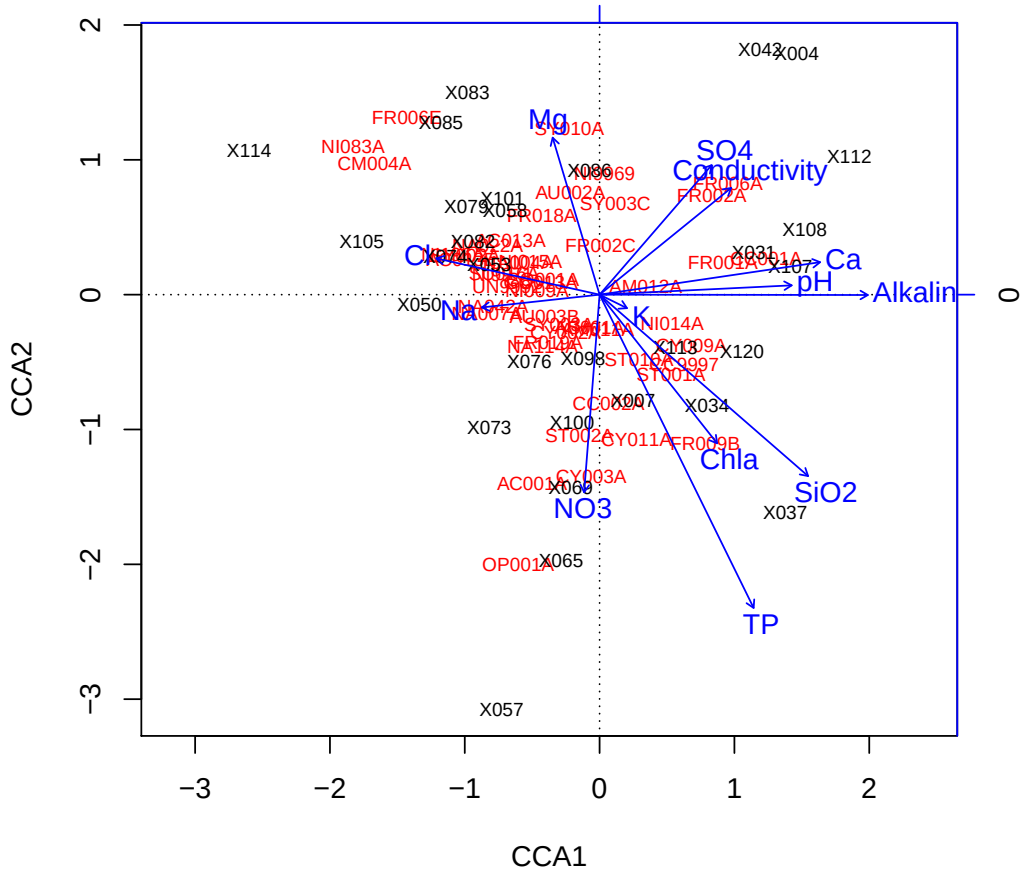

Figure 6: Triplot of the partial CCA of the Ponds diatom and hydrochemistry data after controlling for the effects of MaxDepth and Secchi.

2 Canonical Analysis of Principal Coordinates

`rda()` tries to preserves the Euclidean distances between sites in low dimensional ordination space. Euclidean distances may not always be appropriate for the type of data or analysis you may wish to fit. Canonical Analysis of Principal Coordinates (CAP) allows you to use any dissimilarity matrix in the place of the Euclidean distance. One potential use of this method is to fit models to species data where you have rare or strange sites which may upset CCA. Another potential use might be to analyse many environmental data in relation to physical constraints with permutation tests being used to test significance of the contraints—such an analysis could potentially be performed using multiple regression, but if the data do not met the assumptions of least squares, for example, CAP can be used to analyse these data.

Here we fit a CAP model to the same Ponds diatom data set using the hydrochemical data as constraints, but use Bray-Curtis dissimilarities instead of the Euclidean distances of RDA.

```
> diat.cap <- capscale(diat ~ ., data = env, distance = "bray",
+                      add = TRUE)
> diat.cap

Call: capscale(formula = diat ~ pH + Conductivity +
Alkalinity + TP + SiO2 + NO3 + Na + K + Mg + Ca + Cl +
SO4 + Chla + Secchi + MaxDepth, data = env, distance =
"bray", add = TRUE)

              Inertia Proportion Rank
Total          12.631      1.000
```

```
Constrained     7.492      0.593   15
Unconstrained   5.138      0.407   14
Inertia is Lingoes adjusted squared Bray distance

Eigenvalues for constrained axes:
 CAP1  CAP2  CAP3  CAP4  CAP5  CAP6  CAP7  CAP8  CAP9 CAP10
1.926 1.227 0.646 0.569 0.504 0.463 0.390 0.340 0.292 0.263
CAP11 CAP12 CAP13 CAP14 CAP15
0.246 0.198 0.167 0.146 0.114

Eigenvalues for unconstrained axes:
 MDS1  MDS2  MDS3  MDS4  MDS5  MDS6  MDS7  MDS8  MDS9 MDS10
1.128 0.576 0.494 0.469 0.417 0.371 0.265 0.258 0.243 0.232
MDS11 MDS12 MDS13 MDS14
0.205 0.178 0.166 0.135

Constant added to distances: 0.1091
```

> anova(diat.cap)

```
Permutation test for capscale under reduced model
Permutation: free
Number of permutations: 999

Model: capscale(formula = diat ~ pH + Conductivity + Alkalinity + TP + SiO2 + NO3 + Na + K + Mg + Ca +
          Df SumOfSqs    F Pr(>F)
Model     15     7.49 1.36  0.003 **
Residual 14     5.14
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

> plot(diat.cap)

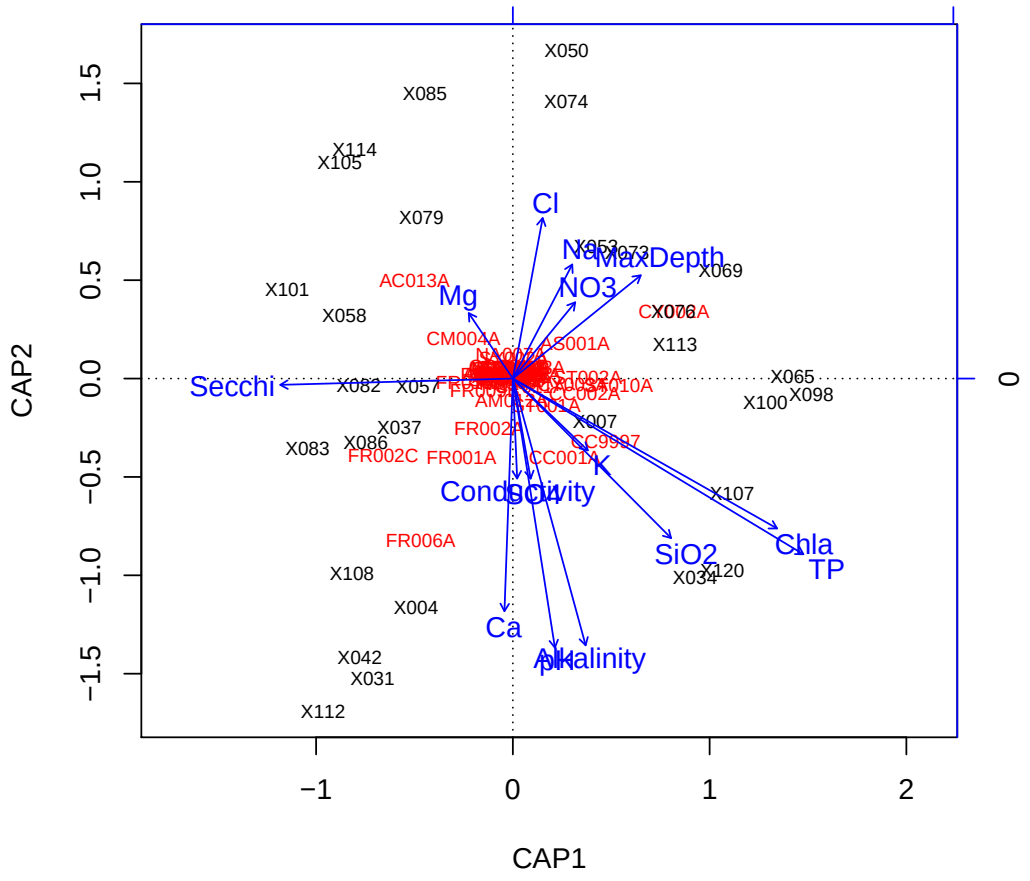

Figure 7: Canonical Analysis of Principal Coordinates of the Ponds diatom and hydrochemistry data.