Example 3b - CIA

Maxime Hervé, Florence Nicolè and Kim-Anh Lê Cao

16/10/2017

Contents

Packages to load

```
library(Hotelling)     # Function needed: clr
library(vegan)         # Functions needed: rda, stressplot
library(RVAideMemoire) # Functions needed: MVA.synt, to.dudi, MVA.plot
library(ade4)          # Functions needed: coinertia, randtest
```

Data loading

Chemical data

The first dataset composed of 32 rows (32 samples) and 4 columns (the absolute concentration of 4 metabolites).

```
tab.Chemistry <- read.table("Example 3b - Chemical data.txt",header=TRUE)
```

Microbial data

The second dataset is composed of the same 32 rows and 24 columns (the relative proportion of 24 bacterial OTUs).

```
tab.Microbio <- read.table("Example 3b - Microbial data.txt",header=TRUE)
```

As the the microbioal community data are compositional, the sum of all OTUs for a given sample is always equal to 1 (or almost because of rounding errors). We check this is the case:

```
rowSums(tab.Microbio)
```

```
 sample1  sample2  sample3  sample4  sample5  sample6  sample7  sample8
   100.1     99.9    100.0    100.0    100.1    100.1     99.9     99.9
 sample9 sample10 sample11 sample12 sample13 sample14 sample15 sample16
   100.1    100.1    100.1    100.0    100.1    100.1    100.0     99.8
sample17 sample18 sample19 sample20 sample21 sample22 sample23 sample24
    99.9     99.9     99.9    100.0    100.0     99.9    100.0    100.1
sample25 sample26 sample27 sample28 sample29 sample30 sample31 sample32
    99.9    100.0     99.9     99.8    100.0    100.0     99.9    100.1
```

Pre-treatment

Chemical data

The data are log transformed then autoscaled:

```
Chemistry.log <- log(tab.Chemistry)
Chemistry.scaled <- scale(Chemistry.log)
```

Microbial data

We transform microbial data using the Centered LogRatio method. Since zeroes are present, we add a small constant value must to the whole data, that is much lower than the minimal value of the whole data (one order of magnitude smaller).

The minimal value is:

```
min(tab.Microbio[tab.Microbio != 0])  # min value ignoring the zeroes
```

```
[1] 0.3
```

Thus we decide to add an offset of 0.01 to all values:

```
Microbio.clr <- clr(tab.Microbio + 0.01)
```

They are then autoscaled:

```
Microbio.scaled <- scale(Microbio.clr)
```

Analysis

Step 1: PCA

Chemical data

We perform the PCA:

```
PCA.Chemistry <- rda(Chemistry.scaled)
```

How much total variance does each component explain?

```
MVA.synt(PCA.Chemistry)
```

```
Criterion: total variance (%)
 Axis Proportion Cumulative
    1      66.35      66.35
    2      24.39      90.74
    3       8.45      99.19
    4       0.81     100.00
```

The first PCA component explains 66 % of the chemical variation, the second component explains 24 % (totalling 91 % of the total variance).

R-related step: transformation of the object PCA.Chemistry to be compatible with the function that performs CIA:

```
PCA.Chemistry.for.CIA <- to.dudi(PCA.Chemistry)
```

Bacterial data

We perform the PCA:

```
PCA.Microbio <- rda(Microbio.scaled)
```

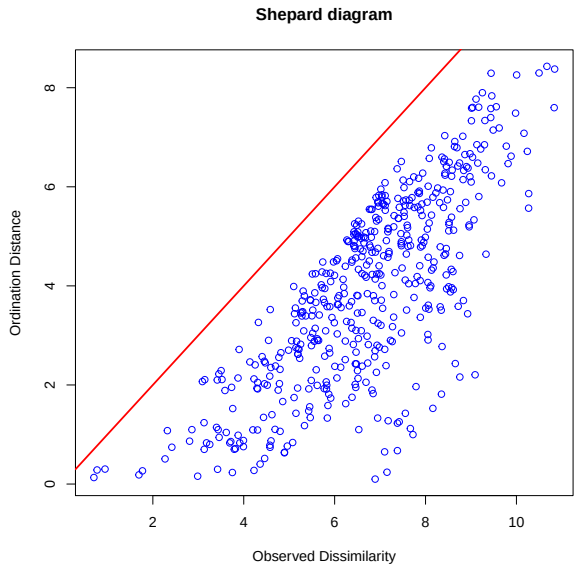
How much total variance does each component explain?

```
MVA.synt(PCA.Microbio)
```

```
Criterion: total variance (%)
 Axis Proportion Cumulative
    1      20.72      20.72
    2      18.89      39.62
    3      11.03      50.64
    4       8.51      59.15
    5       7.80      66.95
```

The first component of the PCA explains 21 % of the microbial variation, the second component explains 19 %. If the total amount of explained variance (91 %) is not considered large enough, we can inspect the Shepard diagram:

```
stressplot(PCA.Microbio,main="Shepard diagram")
```



The scatter of points follows a clear linear trend. Therefore, real sample-to-sample distances are well preserved in the PCA.

R-related step: transformation of the object PCA.Microbio to be compatible with the function that performs CIA:

```
PCA.Microbio.for.CIA <- to.dudi(PCA.Microbio)
```

Step 2: CIA

We perform the PCIA. The model includes 4 components, the maximum value since there are 4 chemical compounds

```
CIA <- coinertia(PCA.Chemistry.for.CIA,PCA.Microbio.for.CIA,scannf=FALSE,nf=4)
```

We test for the concordance between the two datasets:

```
randtest(CIA)
```

```
Monte-Carlo test
Call: randtest.coinertia(xtest = CIA)

Observation: 0.2773589

Based on 999 replicates
Simulated p-value: 0.002
Alternative hypothesis: greater

    Std.Obs Expectation    Variance
3.816991702 0.134584585 0.001399128
```

The concordance is significant, indicating that at least some OTUs' relative proportions coorelate with some metabolites.

How much co-inertia (common information) does each par of components explain?

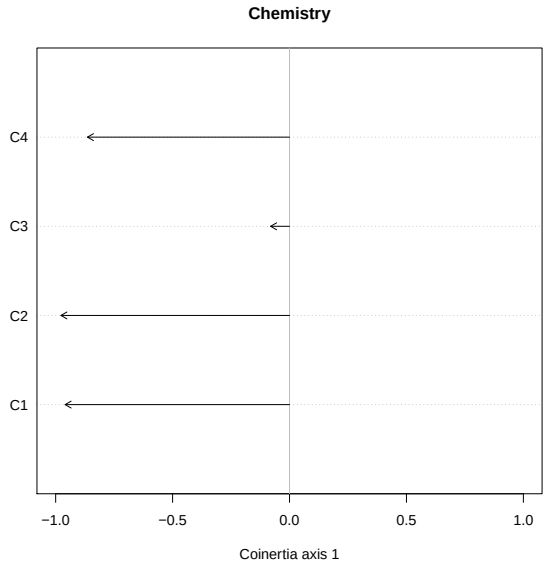
```
MVA.synt(CIA)
```

```
RV coefficient: 0.2774

Criterion: co-inertia (%)
 Axes Proportion Cumulative
    1      65.68      65.68
    2      29.13      94.81
    3       4.70      99.51
    4       0.49     100.00

Criterion: correlation between pairs of axes
 Axes Correlation
    1      0.6551
    2      0.6955
    3      0.6554
    4      0.5810
```

The first pair of components explains 66 % of the shared information between the two datasets, whereas the second pair explains 29 %. Both pairs of components are quite highly positively correlated.

We draw a first plot where we display the correlation between chemical compounds and the first chemical co-inertia component:

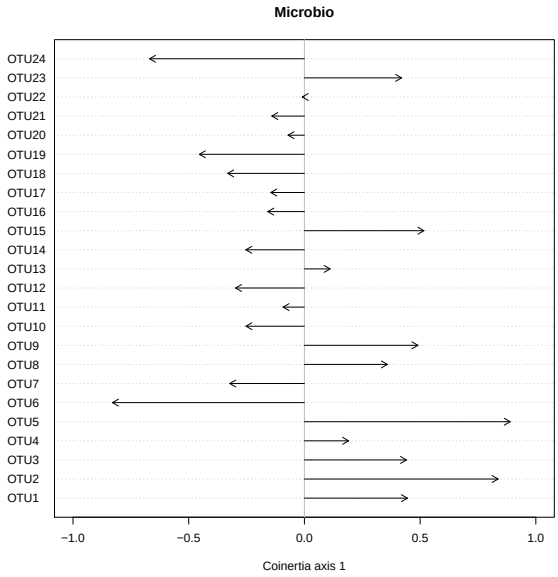
```
MVA.plot(CIA,"corr",space=1,main="Chemistry",yax=NULL)
```



Here compouds C1, C2 and C4 are highly negatively correlated to the component.

We do the same for the microbial OTUs:

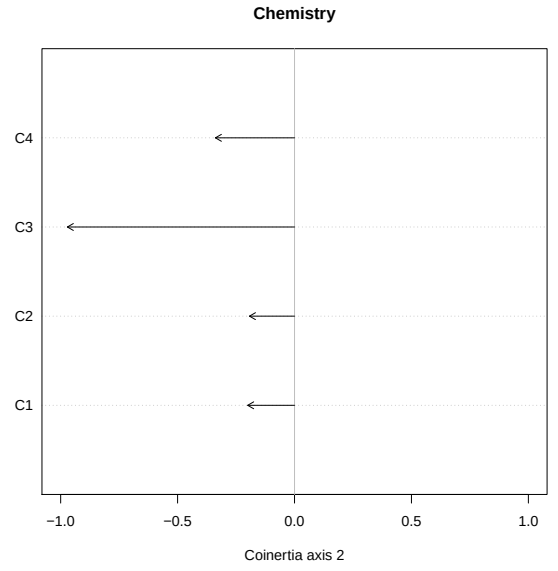
```
MVA.plot(CIA,"corr",space=2,main="Microbio",yax=NULL,cex=0.8)
```



OTUs 6 and 24 are well positively correlated to the component whereas OTUs 2 and 5 are well negatively correlated to the component. Together, this plot and the previous plot indicate that compounds C1, C2 and C4 are well positively correlated to OTUs 6 and 24, and negatively correlated to OTUs 2 and 5.

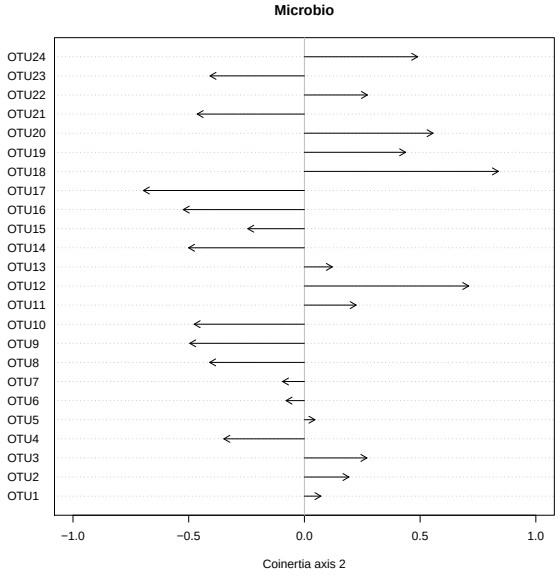
We draw a second plot where we display the correlation between chemical compounds and the second chemical co-inertia component:

```
MVA.plot(CIA,"corr",space=1,main="Chemistry",yax=NULL,xax=2)
```



We do the same for the microbial OTUs:

```
MVA.plot(CIA,"corr",space=2,main="Microbio",yax=NULL,xax=2,cex=0.8)
```



Together, this plot and the previous plot indicate that compound C3 is positively correlated to OTU 17 and negatively correlated to OTUs 12 and 18.

Information on the current R session

```
sessionInfo()
```

```
R version 3.4.0 (2017-04-21)
Platform: x86_64-w64-mingw32/x64 (64-bit)
Running under: Windows 7 x64 (build 7601) Service Pack 1

Matrix products: default

locale:
[1] LC_COLLATE=French_France.1252  LC_CTYPE=French_France.1252
[3] LC_MONETARY=French_France.1252 LC_NUMERIC=C
[5] LC_TIME=French_France.1252

attached base packages:
[1] stats     graphics  grDevices utils     datasets  methods   base

other attached packages:
[1] ade4_1.7-8           RVAideMemoire_0.9-68 vegan_2.4-4
[4] lattice_0.20-35      permute_0.9-4        Hotelling_1.0-4
[7] corpcor_1.6.9        knitr_1.17

loaded via a namespace (and not attached):
 [1] Rcpp_0.12.13       cluster_2.0.6      magrittr_1.5
 [4] splines_3.4.0      MASS_7.3-47        minqa_1.2.4
 [7] car_2.1-5          stringr_1.2.0      tools_3.4.0
[10] pbkrtest_0.4-7     nnet_7.3-12        parallel_3.4.0
[13] grid_3.4.0         nlme_3.1-131       mgcv_1.8-22
[16] quantreg_5.33      MatrixModels_0.4-1 htmltools_0.3.6
```

```
[19] lme4_1.1-14         yaml_2.1.14         rprojroot_1.2
[22] digest_0.6.12       Matrix_1.2-11       nloptr_1.0.4
[25] evaluate_0.10.1     rmarkdown_1.6       stringi_1.1.5
[28] compiler_3.4.0      backports_1.1.1     SparseM_1.77
```