Example 1 - PCA, PCoA and PPLS-DA

Maxime Hervé, Florence Nicolè and Kim-Anh Lê Cao

16/10/2017

Contents

Packages to load 1

Data loading 1

Pre-treatment 2

Compositional data 2

Binary data 2

Analysis 1: PCA on compositional data 2

Analysis 2: PPLS-DA on compositional data 4

Analysis 3: PCoA on binary data 5

Information on the current R session 7

Packages to load

```
library(Hotelling)     # Function needed: clr
library(ade4)          # Functions needed: dist.binary, is.euclid
library(vegan)         # Functions needed: rda, stressplot
library(pls)           # Function needed: cppls
library(RVAideMemoire) # Functions needed: MVA.synt, MVA.plot, MVA.cmv, MVA.test
```

Data loading

The original dataset is composed of 17 rows (17 plants, 8-9 from each color morph) and 54 columns (the color morph and the relative proportion of 53 volatile compounds).

```
tab.compositional <- read.table("Example 1 - Osimia_compositional.txt",header=TRUE)
```

As the data are compositional, the sum of all compounds for a given sample is always equal to 100 % (or almost because of rounding errors). We check this is the case:

```
rowSums(tab.compositional[,2:54])
```

```
 sample1  sample2  sample3  sample4  sample5  sample6  sample7  sample8
    99.9    100.0    100.1    100.0    100.0    100.2    100.1    100.0
 sample9 sample10 sample11 sample12 sample13 sample14 sample15 sample16
   100.0    100.4    100.1     99.9     99.9    100.2     99.9    100.0
sample17
   100.2
```

For the PPLS-DA, we need to create a dummy-coded response from the grouping factor:

```
Groups.PPLSDA <- dummy(tab.compositional$Color)
```

The first analyses will be performed on the compositional data with PCA and PPLS-DA. The last analysis will be performed on binary data, whereby the relative proportion data were transformed into presence/absence data.

```
tab.binary <- read.table("Example 1 - Osimia_binary.txt",header=TRUE)
```

Pre-treatment

Compositional data

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

The minimal non-zero value is:

```
min(tab.compositional[,2:54][tab.compositional[,2:54] != 0])
```

```
[1] 0.1
```

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

```
Chemistry.compositional <- clr(tab.compositional[,2:54] + 0.01)
```

The data are then autoscaled:

```
Chemistry.compositional.scaled <- scale(Chemistry.compositional)
```

Binary data

The distance matrix based on binary data is computed. The distance based on the simple matching coefficient is chosen since double-zeroes are considered as similarities:

```
mat.dist.binary <- dist.binary(tab.binary[,2:54],method=2)
```

Analysis 1: PCA on compositional data

We fit the PCA:

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

How much total variance does each component explain?

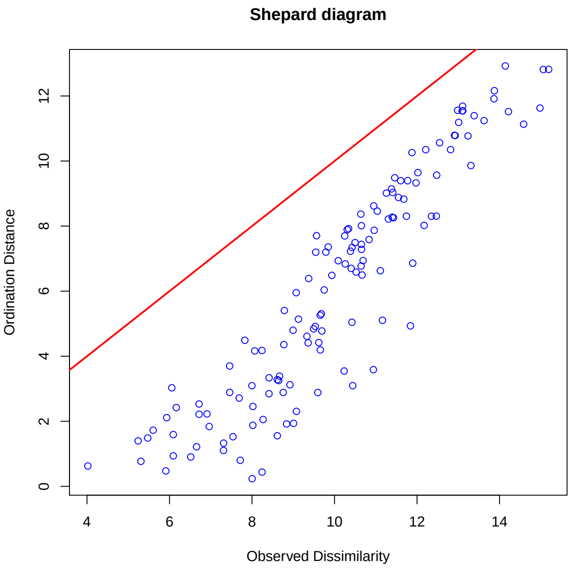
```
MVA.synt(PCA)
```

```
Criterion: total variance (%)
 Axis Proportion Cumulative
    1      35.11      35.11
    2      11.36      46.47
    3       8.99      55.46
    4       7.32      62.78
```

```
     5        6.80       69.58
```

The first PCA component explains 35 % of the variation, the second component explains 11 %. If the total amount of explained variance (46 %) is not considered large enough, we can inspect the Shepard diagram:

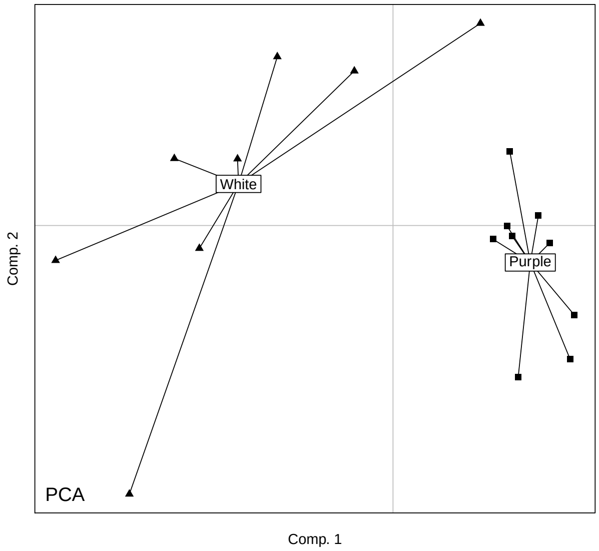
```
stressplot(PCA,main="Shepard diagram")
```



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

We draw the score plot of the PCA, where the two color morphs are well separated:

```
MVA.plot(PCA,fac=tab.compositional$Color,drawextaxes=FALSE,pch=c(15,17),main="PCA")
```



Analysis 2: PPLS-DA on compositional data

We fit the PPLS-DA model:

```
PPLSDA <- cppls(Groups.PPLSDA~Chemistry.compositional.scaled)
```

It is mandatory to validate the model before any interpretation from graphical outputs. We first estimate the classification error rate by cross-model validation (2CV). Since we have 2 groups, we probably only need 1 component. However we add a second component in order to obtain graphical 2-D visualisations:

```
MVA.cmv(Chemistry.compositional.scaled,tab.compositional$Color,model="PPLS-DA",
  crit.inn="NMC",ncomp=2)
```

```

        Cross model validation (2CV)

Model: PPLS-DA
Inner loop: 6-fold validation
Outer loop: 7-fold validation
Validation repeated 10 times
70 submodels generated (1 to 2 components)

Inner loop criterion: number of misclassifications

Mean (standard error) classification error rate (%): 8.82 (0.98)
```

We test the significance of the discrimination:

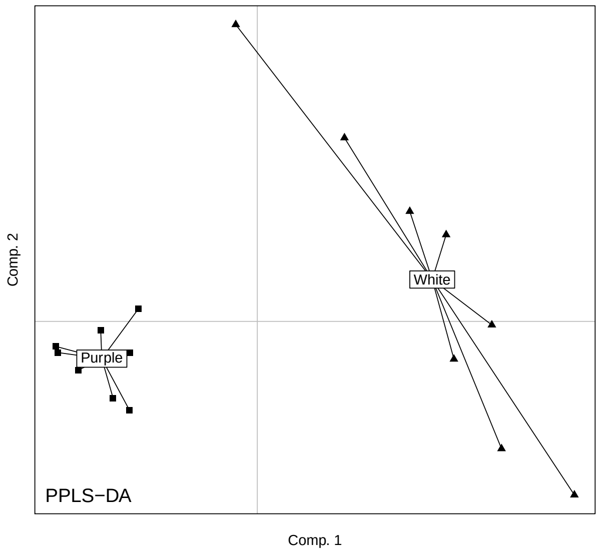
```
# This may take several mninutes to run
# Remove progress=FALSE to see computation progress
MVA.test(Chemistry.compositional.scaled,tab.compositional$Color,model="PPLS-DA",
  cmv=TRUE,ncomp=2,progress=FALSE)
```

```

    Permutation test based on cross model validation

data:  Chemistry.compositional.scaled and tab.compositional$Color
Model: PPLS-DA
2 components maximum
999 permutations
CER = 0.082353, p-value = 0.002
```

The p-value from the permutation test is significant, indicating that there is a significant difference between the two color morphs.

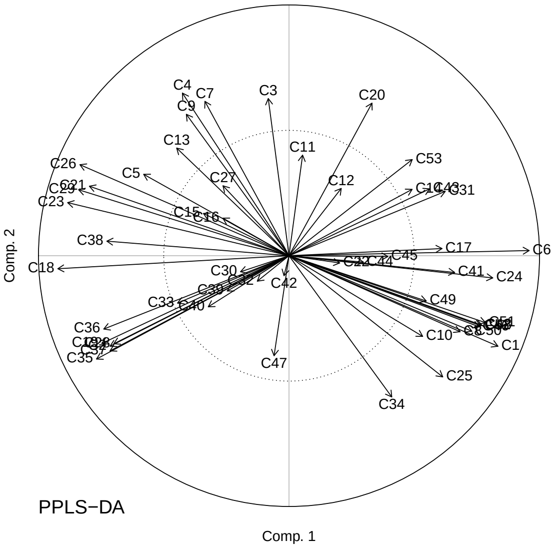
The score plot from the PPLS-DA shows a strong discrimination between the two color morphs:

```
MVA.plot(PPLSDA,fac=tab.compositional$Color,drawextaxes=FALSE,pch=c(15,17),main="PPLS-DA")
```



We draw the correlation circle plot to visualize the discriminant compounds, and how they contribute to each component:

```
MVA.plot(PPLSDA,"corr",set=1,main="PPLS-DA")
```



Analysis 3: PCoA on binary data

We first need to check wether the distance matrix on the binary data has Euclidian properties:

```
is.euclid(mat.dist.binary)
```

```
[1] TRUE
```

As it is the case, we perform the PCoA:

```
PCoA <- dbrda(mat.dist.binary~1)
```

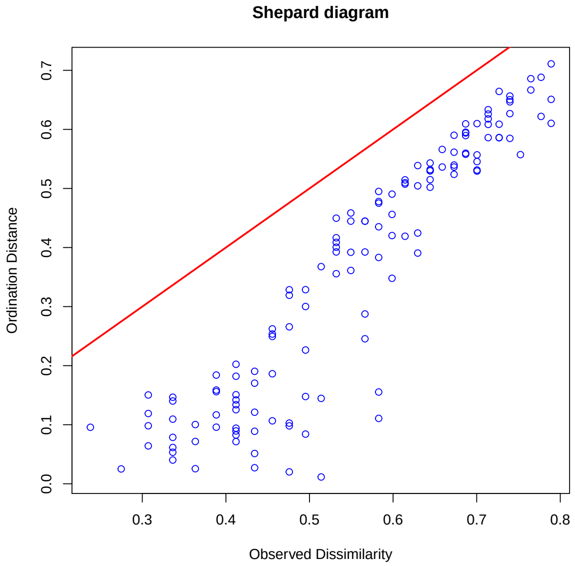
How much total variance does each component explain?

```
MVA.synt(PCoA)
```

```
Criterion: total variance (%)
 Axis Proportion Cumulative
    1      44.22      44.22
    2       9.31      53.53
    3       8.25      61.78
    4       6.61      68.39
    5       5.27      73.66
```

The first PCoA component explains 44 % of the variation, the second component explains 9 %. If the total amount of explained variance (54 %) is not considered large enough, we can inspect the Shepard diagram:

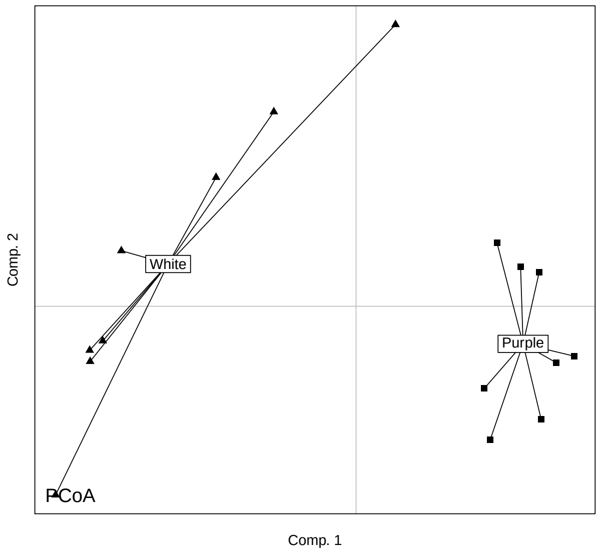
```
stressplot(PCoA,main="Shepard diagram")
```



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

We draw the score plot of the PCoA, where the two color morphs are well separated:

```
MVA.plot(PCoA,fac=tab.compositional$Color,drawextaxes=FALSE,pch=c(15,17),main="PCoA")
```



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] RVAideMemoire_0.9-68 pls_2.6-0            vegan_2.4-4
[4] lattice_0.20-35      permute_0.9-4        ade4_1.7-8
[7] Hotelling_1.0-4      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
```