ropls: PCA, PLS(-DA) and OPLS(-DA) for multivariate analysis and feature selection of omics data

Etienne A. Thévenot

March 25, 2016

Contents

1 The ropls package

The *ropls* R package implements the PCA, PLS(-DA) and OPLS(-DA) approaches with the original, NIPALS-based, versions of the algorithms [1, 2]. It includes the R2 and Q2 quality metrics [3, 4], the permutation diagnostics [5], the computation of the VIP values [1], the score and orthogonal distances to detect outliers [6], as well as many graphics (scores, loadings, predictions, diagnostics, outliers, etc).

The functionalities from *ropls* can also be accessed via a graphical user interface in the *Multivariate* module from the *Workflow4Metabolomics.org* online resource for computational metabolomics (http://workflow4metabolomics.org/) which provides a user-friendly, web-based environment for data pre-processing, statistical analysis, and annotation [7].

2 Context

2.1 Orthogonal Partial Least-Squares

Partial least-squares, which is a latent variable regression method based on covariance between the predictors and the response, has been shown to efficiently handle datasets with multi-collinear predictors, as in the case of spectrometry measurements [1]. More recently, Trygg and Wold introduced the orthogonal projection to latent structures (OPLS) algorithm to model separately the variations of the predictors correlated and orthogonal to the response [2].

OPLS has a similar predictive capacity compared to PLS and improves the interpretation of the predictive components and of the systematic variation [8]. In particular, OPLS modeling of single responses only requires one predictive component.

Diagnostics such as the Q2Y metrics and permutation testing are of high importance to avoid overfitting and assess the statistical significance of the model. The variable importance in projection (VIP), which reflects both the loading weights for each component and the variability of the response explained by this component [8, 9] is often used for feature selection [2, 8].

2.2 OPLS software

OPLS is available in the SIMCA-P commercial software (Umetrics, Umeå, Sweden; [3]). In addition, the kernel-based version of OPLS [10] is available in the open-source R statistical environment [11], and a single implementation of the linear algorithm in R has been described recently [12].

3 The sacurine metabolomics dataset

3.1 Study objective

The objective was to study the influence of age, body mass index and gender on metabolite concentrations in urine, by analysing 183 samples from a cohort of adults with liquid chromatography coupled to high-resolution mass spectrometry ([13]).

3.2 Pre-processing and annotation

Urine samples were analyzed by using an LTQ Orbitrap in the negative ionization mode. A total of 109 metabolites were identified or annotated at the MSI level 1 or 2. After retention time alignment with XCMS, peaks were integrated with Quan Browser. Signal drift and batch effect were corrected, and each urine profile was normalized to the osmolality of the sample. Finally, the data were log10 transformed.

3.3 Covariates

The volunteers' *age*, *body mass index*, and *gender* were recorded.

4 Hands-on

4.1 Loading

We first load the *ropls* package:

```
> library(ropls)
```

We then load the *sacurine* dataset which contains:

1. The *dataMatrix* matrix of numeric type containing the intensity profiles (log10 transformed)
2. The *sampleMetadata* data frame containg sample metadata
3. The *variableMetadata* data frame containg variable metadata

```
> data(sacurine)
> names(sacurine)
```

```
[1] "dataMatrix"       "sampleMetadata"   "variableMetadata"
```

We attach *sacurine* to the search path and display a summary of the content of the *dataMatrix*, *sampleMetadata* and *variableMetadata* with the *strF* **F**unction of the *ropls* package (see also *str*)

```
> attach(sacurine)
> strF(dataMatrix)
```

```
       dim  class    mode typeof   size NAs  min mean median max
 183 x 109 matrix numeric double 0.2 Mb   0 -0.3  4.2    4.3   6
       (2-methoxyethoxy)propanoic acid isomer (gamma)Glu-Leu/Ile ...
HU_011                             3.019766011        3.888479324 ...
HU_014                              3.81433889        4.277148905 ...
...                                        ...                ... ...
HU_208                             3.748127215        4.523763202 ...
HU_209                             4.208859398        4.675880567 ...
       Valerylglycine isomer 2  Xanthosine
HU_011             3.889078716 4.075879575
HU_014             4.181765852 4.195761901
...                        ...         ...
HU_208             4.634338821 4.487781609
HU_209              4.47194762 4.222953354
```

```
> strF(sampleMetadata)
```

```
     age     bmi gender
 numeric numeric factor
 nRow nCol size NAs
  183    3 0 Mb   0
        age   bmi gender
HU_011   29 19.75      M
HU_014   59 22.64      F
...     ...   ...    ...
HU_208   27 18.61      F
HU_209 17.5 21.48      F
```

```
> strF(variableMetadata)
```

```
 msiLevel      hmdb chemicalClass
  numeric character     character
 nRow nCol size NAs
  109    3 0 Mb   0
                                       msiLevel      hmdb chemicalClass
(2-methoxyethoxy)propanoic acid isomer        2                  Organi
(gamma)Glu-Leu/Ile                            2                  AA-pep
...                                         ...       ...           ...
Valerylglycine isomer 2                       2           AA-pep:AcyGly
Xanthosine                                    1 HMDB00299        Nucleo
```

4.2 Principal Component Analysis (PCA)

We perform a PCA on the *dataMatrix* matrix (samples as rows, variables as columns):

```
> sacurine.pca <- opls(dataMatrix)
```

```
PCA
183 samples x 109 variables
standard scaling of predictors
      R2X(cum) pre ort
Total    0.501   8   0
```

A summary of the model (8 components were selected) is printed. In addition the default `summary` figure is displayed (Figure 1).

Note:

1. Since *dataMatrix* does not contain missing value, the singular value decomposition was used by default; NIPALS can be selected with the `algoC` argument specifying the **algo**rithm (**C**haracter),
2. The `predI = NA` default number of **pred**ictive components (**I**nteger) means that only components with a variance superior to the mean variance of all components are kept (note that this rule requires all components to be computed and can be quite time-consuming for large datasets with the NIPALS algorithm; in such cases, one may specify a limited number of components with the `predI` parameter).

Let us see if we notice any partition according to gender, by labeling/coloring the samples according to the gender and drawing the Mahalanobis ellipses for the male and female subgroups (Figure 2).

```
> genderFc <- sampleMetadata[, "gender"]
> plot(sacurine.pca, typeVc = "x-score",
+ parAsColFcVn = genderFc, parEllipsesL = TRUE)
```

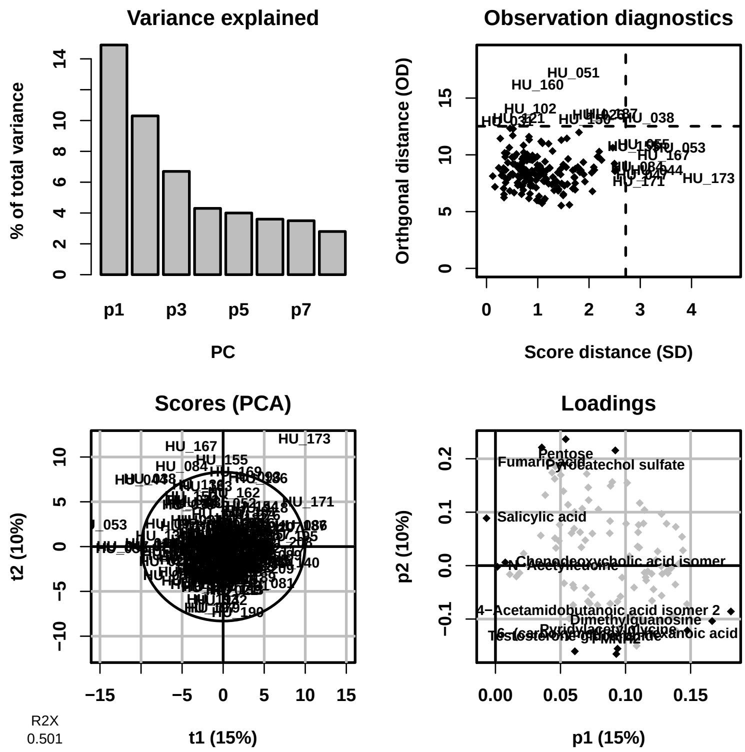

Figure 1: PCA summary plot. Top left `overview`: the *scree* plot (i.e., inertia barplot) suggests that 3 components may be sufficient to capture most of the inertia; Top right `outlier`: this graphics shows the distances within and orthogonal to the projection plane ([6]): the name of the samples with a high value for at least one of the distances are indicated; Bottom left `x-score`; Bottom right `x-loading`.

Note that the plotting **par**ameter to be used **As Col**ors (**F**actor of **c**haracter type or **V**ector of **n**umeric type) has a length equal to the number of rows of the dataMatrix matrix (ie of samples) and that this qualitative or quantitative variable is converted into colors (by using an internal palette or color scale, respectively). We could have visualized the *age* of the individuals by specifying `parAsColFcVn = sampleMetadata[, "age"]`.

4.3 Partial least-squares: PLS and PLS-DA

For PLS (and OPLS), the **Y** response(s) must be provided. **Y** can be either a numeric vector (respectively matrix) for single (respectively multiple) (O)PLS regression, or a character factor for (O)PLS-DA classification as in the following example (Figure 3).

```
> sacurine.plsda <- opls(dataMatrix, genderFc)
```

```
PLS-DA
183 samples x 109 variables and 1 response
standard scaling of predictors and response(s)
      R2X(cum) R2Y(cum) Q2(cum) RMSEE pre ort pR2Y  pQ2
Total    0.275     0.73   0.584 0.262   3   0 0.05 0.05
```

Note:

1. When set to NA (as in the default), the number of components `predI` is determined automatically as follows ([3]): A new component *h* is added to the model if :

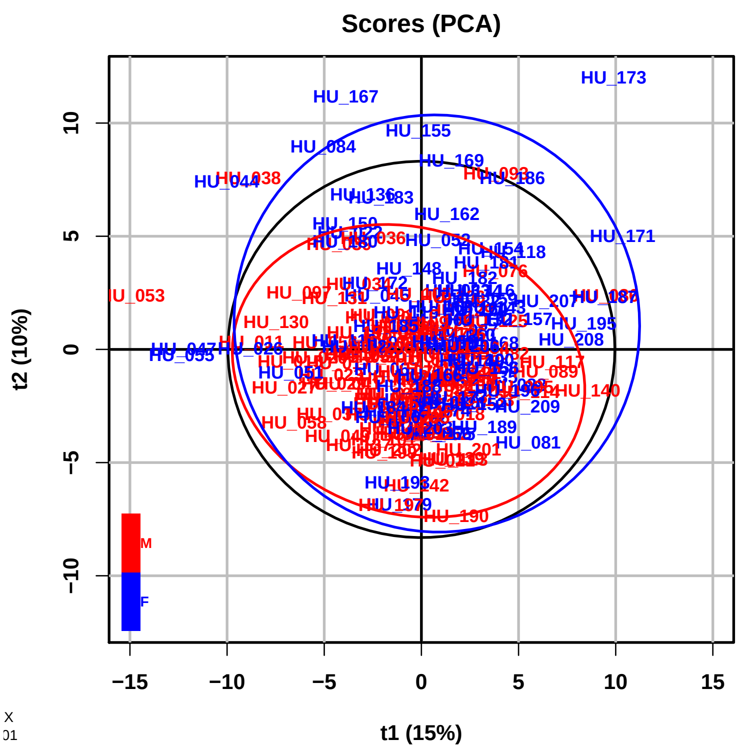

Figure 2: PCA score plot colored according to *gender*. The displayed components can be specified with `parCompVi` (plotting **par**ameter specifying the **Comp**onents: **V**ector of 2 **i**ntegers)

(a) $R2Y_h \geq 1\%$, i.e., the percentage of Y dispersion (i.e., sum of squares) explained by component *h* is more than 1%, and

(b) $Q2Y_h = 1 - PRESS_h/RSS_{h-1} \geq 0$ for PLS (or 5% when the number of samples is less than 100) or 1% for OPLS: $Q2Y_h \geq 0$ means that the predicted residual sum of squares ($PRESS_h$) of the model including the new component *h* estimated by 7-fold cross-validation is less than the residual sum of squares (RSS_{h-1}) of the model with the previous components only (with RSS_0 being the sum of squared Y values).

2. The predictive performance of the full model is assessed by the cumulative Q2Y metric: $Q2Y = 1 - \prod_{h=1}^{r}(1 - Q2Y_h)$. We have $Q2Y \in [0, 1]$, and the higher the Q2Y, the better the performance. Models trained on datasets with a larger number of features compared with the number of samples can be prone to overfitting: in that case, high Q2Y values are obtained by chance only (see section 4.5.1). To estimate the significance of Q2Y (and R2Y) for single response models, permutation testing can be used [5]: models are built after random permutation of the Y values, and $Q2Y_{perm}$ are computed. The *p*-value is equal to the proportion of $Q2Y_{perm}$ above $Q2Y$ (the default number of permutations is 20 as a compromise between quality control and computation speed; it can be increased with the `permI` parameter, e.g. to 1,000, to assess if the model is significant at the 0.05 level),

3. The NIPALS algorithm is used for PLS (and OPLS); *dataMatrix* matrices with (a moderate amount of) missing values can thus be analysed.

We see that our model with 3 predictive (`pre`) components has significant and quite high R2Y and Q2Y values.

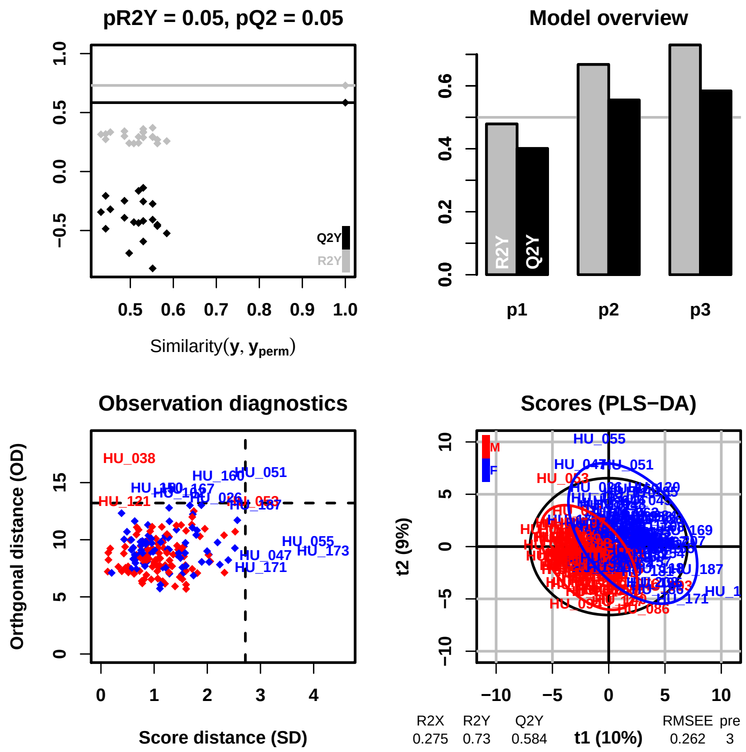

Figure 3: PLS-DA model of the *gender* response. The default `overview` figure displays: Top left: *significance* diagnostic: the $R2Y$ and $Q2Y$ of the model are compared with the corresponding values obtained after random permutation of the **y** response; Top right: *inertia* barplot: the graphic here suggests that 3 orthogonal components may be sufficient to capture most of the inertia; Bottom left: *outlier* diagnostics (see Figure 1); Bottom right: *X-score* plot: the number of components and the cumulative $R2X$, $R2Y$ and $Q2Y$ are indicated below the plot.

4.4 Orthogonal partial least squares: OPLS and OPLS-DA

To perform OPLS(-DA), we set `orthoI` (number of components which are **ortho**gonal; **I**nteger) to either a specific number of orthogonal components, or to NA. Let us build an OPLS-DA model of the *gender* response (Figure 4). Note that for OPLS modeling of a single response, the number of predictive component is 1.

```
> sacurine.oplsda <- opls(dataMatrix, genderFc,
+ predI = 1, orthoI = NA)
```

```
OPLS-DA
183 samples x 109 variables and 1 response
standard scaling of predictors and response(s)
      R2X(cum) R2Y(cum) Q2(cum) RMSEE pre ort pR2Y  pQ2
Total    0.275     0.73   0.602 0.262   1   2 0.05 0.05
```

Let us assess the predictive performance of our model. We first train the model on a subset of the samples (here we use the odd subset value which splits the data set into two halves with similar proportions of samples for each class; alternatively, we could have used a specific subset of indices for training):

```
> sacurine.oplsda <- opls(dataMatrix, genderFc, predI = 1, orthoI = NA, subset = "odd")
```

```
OPLS-DA
92 samples x 109 variables and 1 response
```

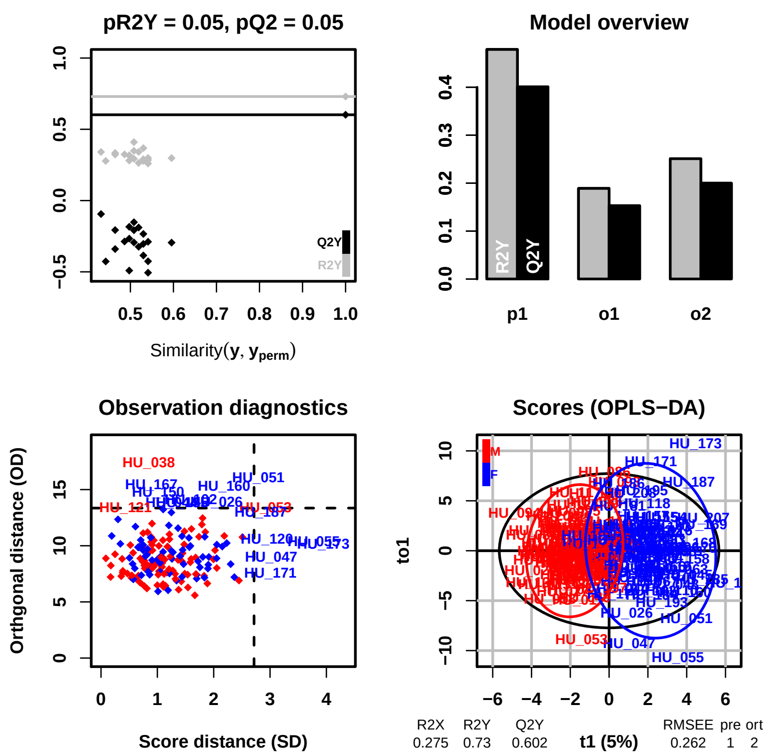

Figure 4: OPLS-DA model of the *gender* response. In the (`score` plot), the predictive component is displayed as abscissa and the (selected; default = 1) orthogonal component as ordinate.

```
standard scaling of predictors and response(s)
      R2X(cum) R2Y(cum) Q2(cum) RMSEE RMSEP pre ort
Total     0.26     0.825    0.608 0.213 0.341   1   2
```

We compute the performances on the training subset:

```
> trainVi <- getSubsetVi(sacurine.oplsda)
> table(genderFc[trainVi], fitted(sacurine.oplsda))
```

```
     M  F
  M 50  0
  F  0 42
```

We then compute the performances on the test subset:

```
> table(genderFc[-trainVi],
+       predict(sacurine.oplsda, dataMatrix[-trainVi, ]))
```

```
     M  F
  M 43  7
  F  7 34
```

As expected, the predictions on the test subset are (slightly) lower. The classifier however still achieves 91% of correct predictions.

4.5 Comments

4.5.1 Overfitting

Overfitting (i.e., building a model with good performances on the training set but poor performances on a new test set) is a major caveat of machine learning techniques applied to data sets with more variables than samples. A simple simulation of a random **X** data set and a **y** response shows that perfect PLS-DA classification can be achieved as soon as the number of variables exceeds the number of samples, as detailed in the example below (Figure 5; [14]). It is therefore essential to check that the $Q2Y$ value of the model is significant by random permutation of the labels.

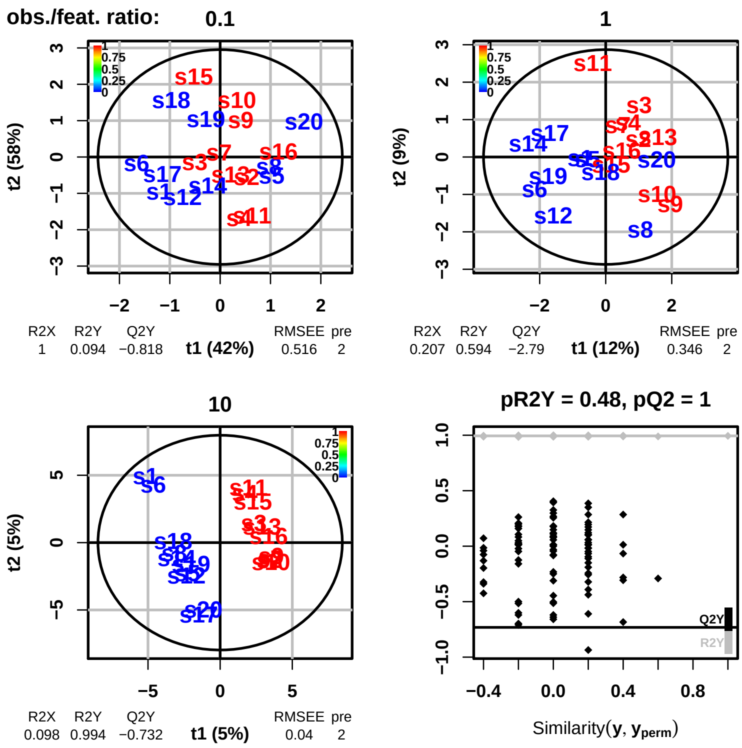

Figure 5: PLS overfit when the number of features exceeds the number of observations. A random matrix **X** of 20 observations $\times$ 200 features was generated by sampling from the uniform distribution $U(0,1)$. A random **y** response was obtained by sampling (without replacement) from a vector of 10 zeros and 10 ones. Top left, top right, and bottom left: the X-score plots of the PLS modeling of **y** by the (sub)matrix of **X** restricted to the first 2, 20, or 200 features, are displayed (i.e., the observation/feature ratios are 0.1, 1, and 10, respectively). Despite the good separation obtained on the bottom left score plot, we see that the $Q2Y$ estimation of predictive performance is low (negative). Bottom right: a significant proportion of the models (in fact here all models) trained after random permutations of the labels have a higher $Q2Y$ value than the model trained with the true labels, confirming that PLS cannot specifically model the **y** response with the **X** predictors, as expected.

```
> set.seed(123)
> obsI <- 20
> featVi <- c(2, 20, 200)
> featMaxI <- max(featVi)
> xRandMN <- matrix(runif(obsI * featMaxI), nrow = obsI)
> yRandVn <- sample(c(rep(0, obsI / 2), rep(1, obsI / 2)))
> layout(matrix(1:4, nrow = 2, byrow = TRUE))
> for(featI in featVi) {
+     randPlsi <- opls(xRandMN[, 1:featI], yRandVn,
+                      predI = 2,
```

```
+                      permI = ifelse(featI == featMaxI, 100, 0),
+                      plotL = FALSE)
+       plot(randPlsi, typeVc = "x-score", parDevNewL = FALSE,
+            parCexN = 1.3, parTitleL = FALSE)
+       mtext(featI/obsI, font = 2, line = 2)
+       if(featI == featMaxI)
+            plot(randPlsi, typeVc = "permutation", parDevNewL = FALSE,
+              parCexN = 1.3)
+       }
> mtext(" obs./feat. ratio:", adj = 0, at = 0, font = 2, line = -2, outer = TRUE)
```

4.5.2 VIP from OPLS models

The classical VIP metric is not useful for OPLS modeling of a single response since ([15, 13]):

1. VIP values remain identical whatever the number of orthogonal components selected,
2. VIP values are univariate (i.e., they do not provide information about interactions between variables).

In fact, when features are standardized, we can demonstrate a mathematical relationship between VIP and *p*-values from a Pearson correlation test ([13]), as illustrated by the code below (Figure 6).

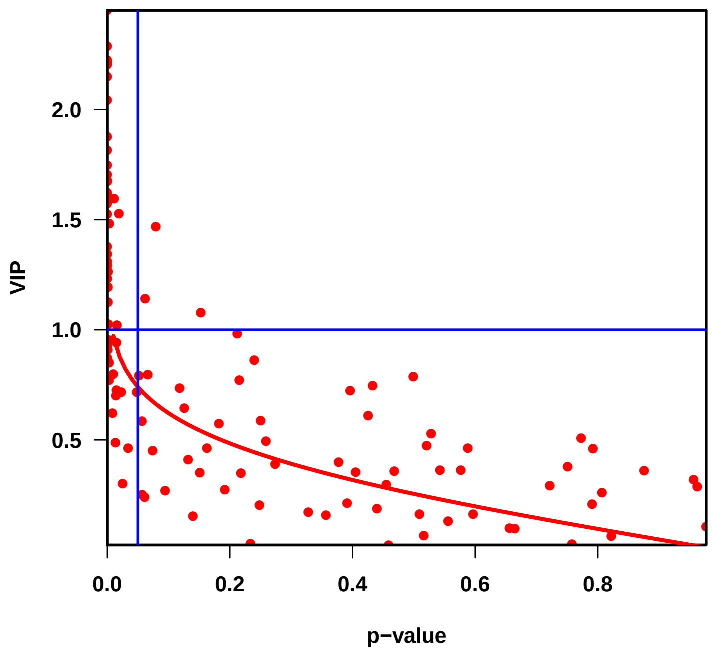

Figure 6: Relationship between VIP from one-predictive PLS or OPLS models with standardized variables, and *p*-values from Pearson correlation test. The (p_j, VIP_j) pairs corresponding respectively to the VIP values from OPLS modelling of the *age* response with the *sacurine* dataset, and the *p*-values from the Pearson correlation test are shown as *red* dots. The $y = \Phi^{-1}(1 - x/2)/z_{rms}$ curve is shown in *red* (where Φ^{-1} is the inverse of the probability density function of the standard normal distribution, and z_{rms} is the quadratic mean of the z_j quantiles from the standard normal distribution; $z_{rms} = 2.6$ for the *sacurine* dataset and the *age* response). The vertical (resp. horizontal) *blue* line corresponds to univariate (resp. multivariate) thresholds of $p = 0.05$ and $VIP = 1$, respectively ([13]).

```
> ageVn <- sampleMetadata[, "age"]
> pvaVn <- apply(dataMatrix, 2,
+                function(feaVn) cor.test(ageVn, feaVn)[["p.value"]])
> vipVn <- getVipVn(opls(dataMatrix, ageVn, predI = 1, orthoI = NA, plot = FALSE))
> quantVn <- qnorm(1 - pvaVn / 2)
> rmsQuantN <- sqrt(mean(quantVn^2))
> par(font = 2, font.axis = 2, font.lab = 2, las = 1,
+     mar = c(5.1, 4.6, 4.1, 2.1),
+     lwd = 2, pch = 16)
> plot(pvaVn, vipVn,
+      col = "red",
+      pch = 16,
+      xlab = "p-value", ylab = "VIP", xaxs = "i", yaxs = "i")
> box(lwd = 2)
> curve(qnorm(1 - x / 2) / rmsQuantN, 0, 1, add = TRUE, col = "red", lwd = 3)
> abline(h = 1, col = "blue")
> abline(v = 0.05, col = "blue")
```

The VIP properties above result from:

1. OPLS models of a single response have a single predictive component,
2. in the case of one-predictive component (O)PLS models, the general formula for VIPs can be simplified to $VIP_j = \sqrt{m} \times |w_j|$ for each feature j, were m is the total number of features and **w** is the vector of loading weights,
3. in OPLS, **w** remains identical whatever the number of extracted orthogonal components,
4. for a single-response model, **w** is proportional to **X′y** (where ′ denotes the matrix transposition),
5. if **X** and **y** are standardized, **X′y** is the vector of the correlations between the features and the response.

Galindo-Prieto et al. (2014) have recently suggested new VIP metrics for OPLS, VIP_{pred} and VIP_{ortho}, to separately measure the influence of the features in the modeling of the dispersion correlated to, and orthogonal to the response, respectively ([15]).

For OPLS(-DA) models, you can therefore get from the model generated with *opls*:

- the predictive VIP vector (which corresponds to the $VIP_{4,pred}$ metric measuring the variable importance in prediction) with `getVipVn(model)`,
- the orthogonal VIP vector which is the $VIP_{4,ortho}$ metric measuring the variable importance in orthogonal modeling with `getVipVn(model, orthoL = TRUE)`.

As for the classical VIP, we still have the mean of VIP^2_{pred} (and of VIP^2_{ortho}) which, each, equals 1.

4.5.3 (Orthogonal) Partial Least Squares Discriminant Analysis: (O)PLS-DA

Two classes When the **y** response is a factor of 2 levels (character vectors are also allowed), it is internally transformed into a vector of values $\in \{0, 1\}$ encoding the classes. The vector is centered and unit-variance scaled, and the (O)PLS analysis is performed.

Brereton and Lloyd (2014) have demonstrated that when the sizes of the 2 classes are unbalanced, a bias is introduced in the computation of the decision rule, which penalizes the class with the highest size ([16]). In this case, an external procedure using resampling (to balance the classes) and taking into account the class sizes should be used for optimal results.

Multiclass In the case of more than 2 levels, the **y** response is internally transformed into a matrix (each class is encoded by one column of values $\in \{0, 1\}$). The matrix is centered and unit-variance scaled, and the PLS analysis is performed.

In this so-called 'PLS2' implementation, the proportions of 0 and 1 in the columns is usually unbalanced (even in the case of balanced size of the classes) and the bias described previously occurs ([16]). The multiclass PLS-DA results from *ropls* are therefore indicative only, and we recommend to set an external procedure where each column of the matrix is modeled separately (as described above) and the resulting probabilities are aggregated (see for instance [17]).

4.6 Closing session

Before closing this example session, we detach *sacurine* from the search path:

```
> detach(sacurine)
```

4.7 Session information

```
> sessionInfo()
```

- R version 3.2.3 (2015-12-10), `x86_64-w64-mingw32`
- Locale: `LC_COLLATE=French_France.1252`, `LC_CTYPE=French_France.1252`, `LC_MONETARY=French_France.1252`, `LC_NUMERIC=C`, `LC_TIME=French_France.1252`
- Base packages: base, datasets, graphics, grDevices, methods, stats, utils
- Other packages: ropls 1.3.20
- Loaded via a namespace (and not attached): BiocStyle 1.9.7, tools 3.2.3

5 Pre-processing and annotation of mass spectrometry data

To illustrate how *dataMatrix*, *sampleMetadata* and *variableMetadata* can be obtained from raw mass spectra file, we use the LC-MS data from the *faahKO* package [18]. We will pre-process the raw files with the *xcms* package [19] and annotate isotopes and adducts with the *CAMERA* package [20], as described in the corresponding vignettes (all these packages are from *bioconductor*).

Let us start by getting the paths to the 12 raw files (6 KO and 6 WT mice) in the ".cdf" open format. The files are grouped in two sub-directories ("KO" and "WT") since *xcms* can use sample class information when grouping the peaks and correcting retention times.

```
> library(faahKO)
> cdfpath <- system.file("cdf", package = "faahKO")
> cdffiles <- list.files(cdfpath, recursive = TRUE, full.names = TRUE)
> basename(cdffiles)
```

```
 [1] "ko15.CDF" "ko16.CDF" "ko18.CDF" "ko19.CDF" "ko21.CDF" "ko22.CDF" "wt15.CDF"
 [8] "wt16.CDF" "wt18.CDF" "wt19.CDF" "wt21.CDF" "wt22.CDF"
```

Next, *xcms* is used to pre-process the individual raw files, as described in the vignette.

```
> library(xcms)
```

```
> xset <- xcmsSet(cdffiles)
```

```
> xset
```

```
An "xcmsSet" object with 12 samples

Time range: 2506.1-4147.7 seconds (41.8-69.1 minutes)
Mass range: 200.1-599.3338 m/z
Peaks: 4721 (about 393 per sample)
Peak Groups: 0
Sample classes: KO, WT

Peak picking was performed on MS1.
Profile settings: method = bin
                  step = 0.1

Memory usage: 0.713 MB
```

```
> xset <- group(xset)
```

```
262 325 387 450 512 575
```

```
> xset2 <- retcor(xset, family = "symmetric", plottype = "mdevden")
```

```
Retention Time Correction Groups: 133
```

```
> xset2 <- group(xset2, bw = 10)
```

```
262 325 387 450 512 575
```

```
> xset3 <- fillPeaks(xset2)
```

Finally, the *annotateDiffreport* from *CAMERA* annotates isotopes and adducts and builds a peak table containing the peak intensities and the variable metadata.

```
> library(CAMERA)
> diffreport <- annotateDiffreport(xset3, quick=TRUE)
```

```
Start grouping after retention time.
Created 128 pseudospectra.
Generating peak matrix!
Run isotope peak annotation
 % finished: 10  20  30  40  50  60  70  80  90  100
Found isotopes: 81
```

```
> diffreport[1:4, ]
```

```
                name      fold     tstat       pvalue    mzmed    mzmin    mzmax    rtmed
300.2/3390 M300T3390  5.693594 -14.44368 5.026336e-08 300.1898 300.1706 300.2000 3390.324
301.2/3390 M301T3390  5.876588 -15.57570 6.705719e-08 301.1879 301.1659 301.1949 3389.627
298.2/3187 M298T3187  3.870918 -11.93891 3.310025e-07 298.1508 298.1054 298.1592 3186.803
491.2/3397 M491T3397 24.975703 -16.83986 4.463361e-06 491.2000 491.1877 491.2063 3397.160
              rtmin    rtmax npeaks KO WT      ko15      ko16      ko18      ko19
300.2/3390 3386.765 3396.335     12  6  6 4534353.6 4980914.5 5290739.1 4564262.9
301.2/3390 3386.765 3392.101      7  6  1  962353.4 1047934.1 1109303.0  946943.4
298.2/3187 3184.124 3191.312      4  4  0  180780.8  203927.0  191015.9  190626.8
491.2/3397 3367.123 3424.681      6  6  0  432037.0  332159.1  386966.8  334951.5
                ko21      ko22       wt15      wt16      wt18      wt19        wt21
300.2/3390 4733236.1 3931592.6 349660.885 491793.18 645526.70 634108.85 1438254.446
301.2/3390  984787.2  806171.5  86450.412 120096.52 143007.95 137319.69  218483.143
298.2/3187  156869.1  220288.6  16269.096  43677.78  54739.13  76318.01   54726.115
```

```
491.2/3397  294816.2  373577.6   7643.138  10519.94  26472.29  33598.32    8030.467
                  wt22  isotopes adduct pcgroup
300.2/3390 1364627.84    [9][M]+             20
301.2/3390  291392.97 [9][M+1]+             20
298.2/3187   49679.94                       102
491.2/3397       0.00                        28
```

We then build the dataMatrix, sampleMetadata and variableMetadata matrix and dataframes as follows:

```
> sampleVc <- grep("^ko|^wt", colnames(diffreport), value = TRUE)
> dataMatrix <- t(as.matrix(diffreport[, sampleVc]))
> dimnames(dataMatrix) <- list(sampleVc, diffreport[, "name"])
> sampleMetadata <- data.frame(row.names = sampleVc,
+ genotypeFc = substr(sampleVc, 1, 2))
> variableMetadata <- diffreport[, !(colnames(diffreport) %in% c("name", sampleVc))]
> rownames(variableMetadata) <- diffreport[, "name"]
```

The data can now be analysed with the *ropls* package as described in the previous section (i.e. by performing a PCA and an OPLS-DA):

```
> library(ropls)
> opls(dataMatrix)
```

```
PCA
12 samples x 400 variables
standard scaling of predictors
      R2X(cum) pre ort
Total    0.588   2   0
```

```
> opls(dataMatrix, sampleMetadata[, "genotypeFc"], orthoI = NA)
```

```
OPLS-DA
12 samples x 400 variables and 1 response
standard scaling of predictors and response(s)
      R2X(cum) R2Y(cum) Q2(cum)  RMSEE pre ort pR2Y  pQ2
Total    0.739    0.993   0.823 0.0554   1   3  0.1 0.05
```

6 Other datasets

In addition to the *sacurine* dataset presented above, the package contains the following datasets to illustrate the functionalities of PCA, PLS and OPLS (see the examples in the documentation of the opls function):

aminoacids Amino-Acids Dataset. Quantitative structure property relationship (QSPR; [1]).

cellulose NIR-Viscosity example data set to illustrate multivariate calibration using PLS, spectral filtering and OPLS (Multivariate calibration using spectral data. Simca tutorial. Umetrics, Sweden).

cornell Octane of various blends of gasoline: Twelve mixture component proportions of the blend are analysed [4].

foods Food consumption patterns accross European countries (FOODS). The relative consumption of 20 food items was compiled for 16 countries. The values range between 0 and 100 percent and a high value corresponds to a high consumption. The dataset contains 3 missing data [3].

linnerud Three physiological and three exercise variables are measured on twenty middle-aged men in a fitness club [4].

lowarp A multi response optimization data set (LOWARP) [3].

mark Marks obtained by french students in mathematics, physics, french and english. Toy example to illustrate the potentialities of PCA [21].

References

[1] S. Wold, M. Sjöström, and L. Eriksson. Pls-regression: a basic tool of chemometrics. *Chemometrics and Intelligent Laboratory Systems*, 58:109–130, 2001. URL: http://dx.doi.org/10.1016/S0169-7439(01)00155-1.

[2] J. Trygg and S. Wold. Orthogonal projection to latent structures (o-pls). *Journal of Chemometrics*, 16:119–128, 2002. URL: http://dx.doi.org/10.1002/cem.695.

[3] I. Eriksson, E. Johansson, N. Kettaneh-Wold, and S. Wold. *Multi- and megavariate data analysis. Principles and applications*. Umetrics Academy, 2001.

[4] M. Tenenhaus. *La regression PLS : theorie et pratique*. Editions Technip, 1998.

[5] Ewa Szymanska, Edoardo Saccenti, AgeK. Smilde, and JohanA. Westerhuis. Double-check: validation of diagnostic statistics for pls-da models in metabolomics studies. *Metabolomics*, 8(1):3–16, 2012. URL: http://dx.doi.org/10.1007/s11306-011-0330-3.

[6] M. Hubert, PJ. Rousseeuw, and K. Vanden Branden. Robpca: a new approach to robust principal component analysis. *Technometrics*, 47:64–79, 2005. URL: http://dx.doi.org/10.1198/004017004000000563.

[7] Franck Giacomoni, Gildas Le Corguillé, Misharl Monsoor, Marion Landi, Pierre Pericard, Mélanie Pétéra, Christophe Duperier, Marie Tremblay-Franco, Jean-François Martin, Daniel Jacob, Sophie Goulitquer, Etienne A. Thévenot, and Christophe Caron. Workflow4metabolomics: a collaborative research infrastructure for computational metabolomics. *Bioinformatics*, 31(9):1493–1495, 2015. URL: http://dx.doi.org/10.1093/bioinformatics/btu813.

[8] Rui Climaco Pinto, Johan Trygg, and Johan Gottfries. Advantages of orthogonal inspection in chemometrics. *Journal of Chemometrics*, 26(6):231–235, 2012. URL: http://dx.doi.org/10.1002/cem.2441.

[9] Tahir Mehmood, Kristian Hovde Liland, Lars Snipen, and Solve Saebo. A review of variable selection methods in partial least squares regression. *Chemometrics and Intelligent Laboratory Systems*, 118(0):62–69, 2012. URL: http://dx.doi.org/10.1016/j.chemolab.2012.07.010.

[10] Max Bylesjo, Mattias Rantalainen, Jeremy Nicholson, Elaine Holmes, and Johan Trygg. K-opls package: Kernel-based orthogonal projections to latent structures for prediction and interpretation in feature space. *BMC Bioinformatics*, 9(1):106, 2008. URL: http://dx.doi.org/10.1186/1471-2105-9-106.

[11] R Development Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2008. ISBN 3-900051-07-0. URL: http://www.R-project.org.

[12] R. Gaude, F. Chignola, D. Spiliotopoulos, A. Spitaleri, M. Ghitti, JM. Garcia-Manteiga, S. Mari, and G. Musco. muma, an r package for metabolomics univariate and multivariate statistical analysis. *Current Metabolomics*, 1:180–189, 2013. URL: http://dx.doi.org/10.2174/2213235X11301020005.

[13] Etienne A. Thévenot, Aurélie Roux, Xu Ying, Eric Ezan, and Christophe Junot. Analysis of the human adult urinary metabolome variations with age, body mass index and gender by implementing a comprehensive workflow for univariate and opls statistical analyses. *Journal of Proteome Research*, 14(8):3322–3335, 2015. URL: http://dx.doi.org/10.1021/acs.jproteome.5b00354.

[14] R. Wehrens. *Chemometrics with R: multivariate data analysis in the natural sciences and life sciences*. Springer, 2011.

[15] Beatriz Galindo-Prieto, Lennart Eriksson, and Johan Trygg. Variable influence on projection (vip) for orthogonal projections to latent structures (opls). *Journal of Chemometrics*, 28(8):623–632, 2014. URL: http://dx.doi.org/10.1002/cem.2627.

[16] Richard G. Brereton and Gavin R. Lloyd. Partial least squares discriminant analysis: taking the magic away. *Journal of Chemometrics*, 28(4):213–225, 2014. URL: http://dx.doi.org/10.1002/cem.2609.

[17] M Bylesjo, M Rantalainen, O Cloarec, J Nicholson, E Holmes, and J Trygg. Opls discriminant analysis: combining the strengths of pls-da and simca classification. *Journal of Chemometrics*, 20:341–351, 2006. URL: http://dx.doi.org/10.1002/cem.1006.

[18] Alan Saghatelian, Sunia A. Trauger, Elizabeth J. Want, Edward G. Hawkins, Gary Siuzdak, and Benjamin F. Cravatt. Assignment of endogenous substrates to enzymes by global metabolite profiling. *Biochemistry*, 43(45):14332–14339, November 2004. URL: http://dx.doi.org/10.1021/bi0480335.

[19] C. A. Smith, E. J. Want, G. O'Maille, R. Abagyan, and G. Siuzdak. Xcms: Processing mass spectrometry data for metabolite profiling using nonlinear peak alignment, matching, and identification. *Analytical Chemistry*, 78(3):779–787, 2006. URL: http://dx.doi.org/10.1021/ac051437y.

[20] Carsten Kuhl, Ralf Tautenhahn, Christoph Bottcher, Tony R. Larson, and Steffen Neumann. Camera: An integrated strategy for compound spectra extraction and annotation of liquid chromatography/mass spectrometry data sets. *Analytical Chemistry*, 84(1):283–289, 2012. URL: http://dx.doi.org/10.1021/ac202450g, `arXiv:http://pubs.acs.org/doi/pdf/10.1021/ac202450g`.

[21] A. Baccini. Statistique descriptive multidimensionnelle (pour les nuls), 2010.