

# MetMSLine Workflow

WMB Edmands

November 12, 2015

The following illustrates the MetMSLine workflow with example data:

## 1. Read in peak table and co-variate information and pre-process the data.

```
# file path example peakTable in comma delimited csv file
# (see ?example_Exp_MS1features for details).
peakTable <- system.file("extdata", "MS1features_example.csv", package = "MetMSLine")
peakTable <- read.csv(peakTable, header=T, stringsAsFactors=F)

# load co-variates table in comma delimited csv file
coVariates <- system.file("extdata", "coVariates.csv", package = "MetMSLine")
coVariates <- read.csv(coVariates, header=T)
# observation names (i.e. sample names)
obsNames <- colnames(peakTable)[grep('ACN_|MeOH_', colnames(peakTable))]

# zero fill
peakTable <- zeroFill(peakTable, obsNames)
## zero filling with half the minimum non-zero value

# Normalize (median fold change/ probabilistic quotient), total ion signal
# also available ?signNorm
peakTable <- signNorm(peakTable, obsNames, method="medFC")
## Median fold change normalization...

# data deconvolution based on retention time and interfeature correlation
# calculation of weighted mean (see ?weigthed.mean) within each pseudospectral
# cluster (i.e. the sum of mass spectral intensities across all samples are used
# to weight the contribution of each feature to the average).
wMeanPeakTable <- rtCorrClust(peakTable, obsNames, rtThresh=2, corrThresh=0.9,
                             minFeat=1)
## hierarchical clustering peak group retention times...
##
## intra RT group correlation clustering 45 rt groups...
##
## Calculating weighted mean for 2724 pseudospectra accounting for 3720 of 3720 total features

# extract weighted mean pseudospectra table
wMeanPspec <- wMeanPeakTable$wMeanPspec

# log transform (base 2)
wMeanPspec <- logTrans(wMeanPspec, obsNames, base=2)
## log transforming to the base 2...
```

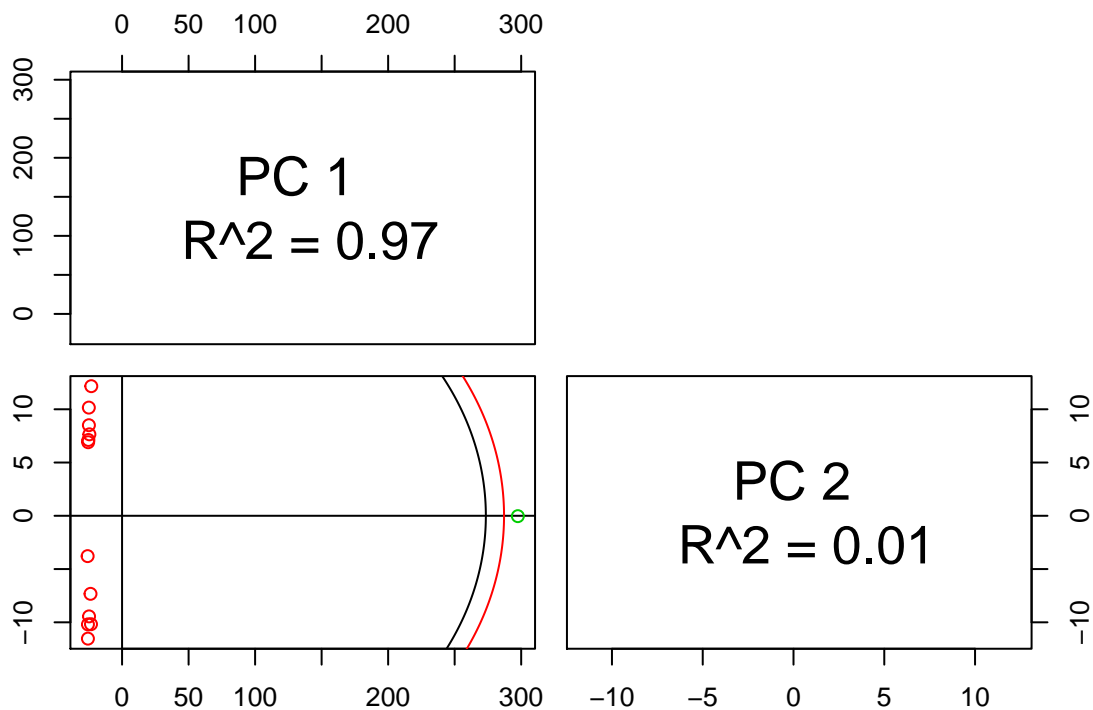
## 2. PCA projection, automatic outlier removal and score plot cluster identification.

```
# add dummy blank to illustrate pca outlier detection
wMeanPspec$blank_1 <- 0.0001
# observation names (i.e. sample names)
obsNames <- colnames(wMeanPspec)[grep('ACN_|MeOH_|blank_', colnames(wMeanPspec))]

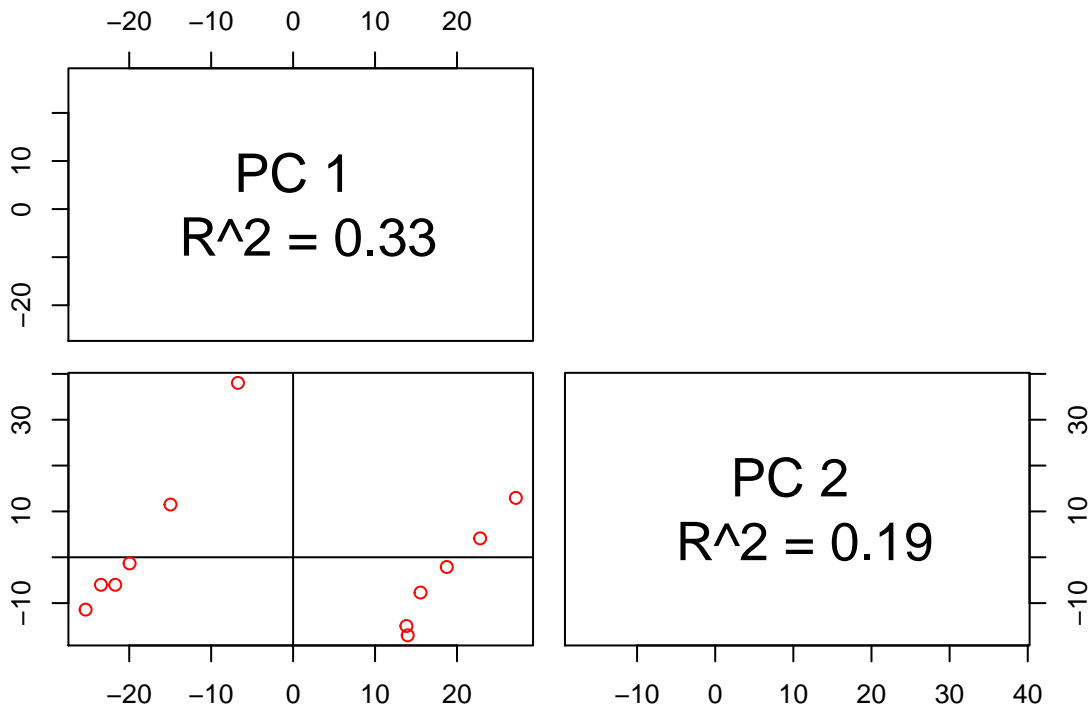
# PCA projection using pca of pcaMethods and automatic outlier removal based
# on proportional expansion of the Hotellings T2 ellipse
pcaOutResults <- pcaOutId(wMeanPspec, obsNames, cv="q2", outTol=1.05,
                          scale="pareto")
```

```
## Calculating PCA model 1...
## 1 outliers identified PCA model 1
## Calculating PCA model 2...
```

```
# Plot PCA displaying any outliers and expanded Hotelling's ellipse, colour according
# to any potential outliers detected. function modified from pcaMethods ?plotPcs.
plotPcsEx(pcaOutResults$pcaResults[[1]]$pcaResult,
          pcaOutResults$pcaResults[[1]]$exHotEllipse, type="scores",
          col=pcaOutResults$pcaResults[[1]]$possOut+2)
```



```
# plot second PCA model iteration after outlier removal
plotPcsEx(pcaOutResults$pcaResults[[2]]$pcaResult,
  pcaOutResults$pcaResults[[2]]$exHotEllipse, type="scores",
  col=pcaOutResults$pcaResults[[2]]$posOut+2)
```



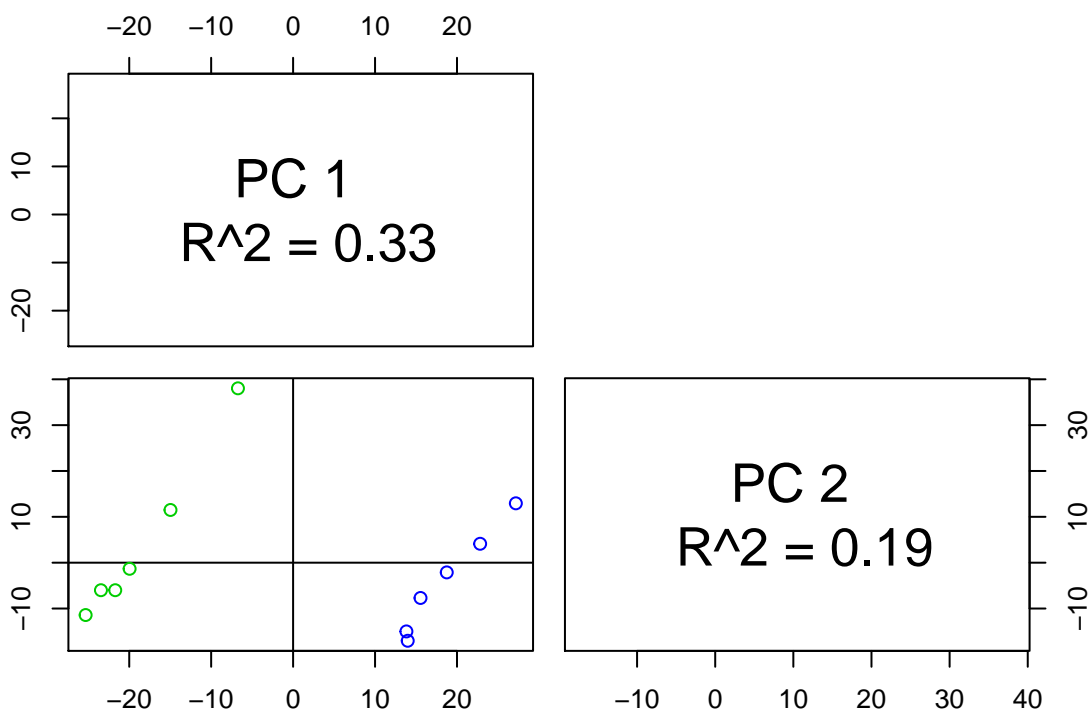
```
# show PCA results iteration 2
pcaOutResults$pcaResults[[2]]$pcaResult
```

```
## svd calculated PCA
## Importance of component(s):
##           PC1    PC2
## R2          0.3347 0.1865
## Cumulative R2 0.3347 0.5211
## 2724      Variables
## 12       Samples
## 0        NAs ( 0 %)
## 2        Calculated component(s)
## Data was mean centered before running PCA
## Data was scaled before running PCA
## Scores structure:
## [1] 12  2
## Loadings structure:
## [1] 2724  2
```

```
# show Q2 cross-validation statistic
pcaOutResults$pcaResults[[2]]$pcaResult@cvstat
```

```
##      PC 1      PC 2
## 0.1918085 0.1230653
```

```
# label by extraction type using co-variables table
plotPcsEx(pcaOutResults$pcaResults[[2]]$pcaResult,
          pcaOutResults$pcaResults[[2]]$exHotEllipse, type="scores",
          col=as.numeric(as.factor(coVariates$extractionType)) + 2)
```



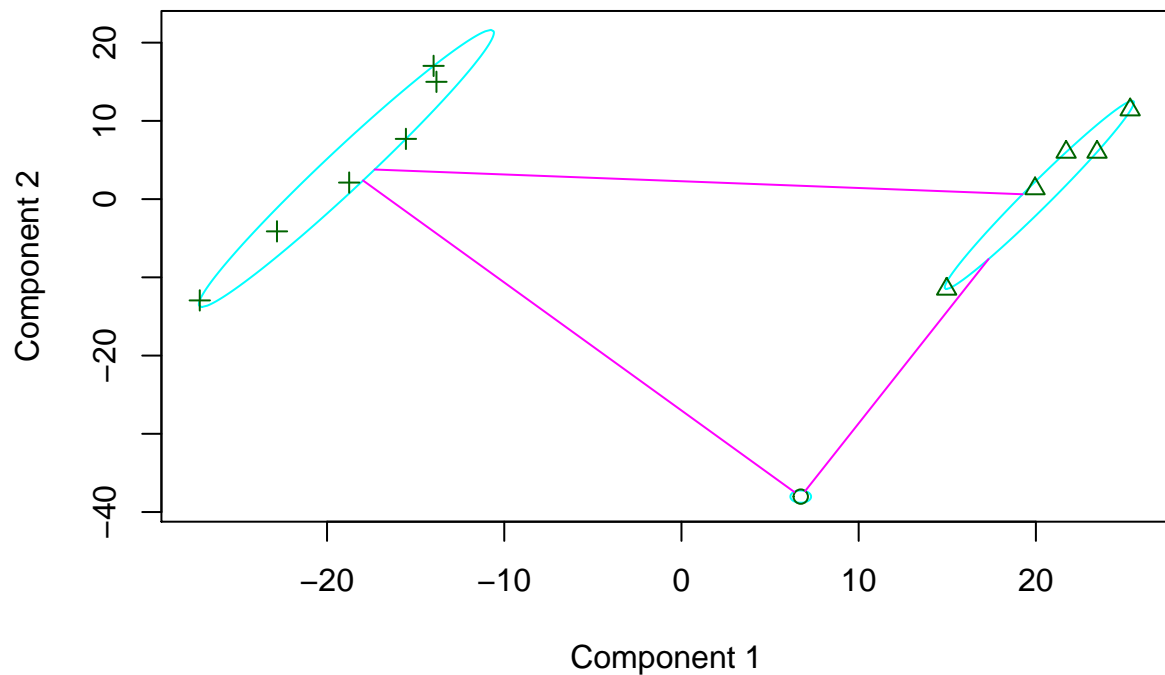
```
# Automatically identify potential cluster membership given the table of co-variables
finalPca <- pcaOutResults$pcaResults[[length(pcaOutResults$pcaResults)]]$pcaResult
clustIdentity <- pcaClustId(finalPca, coVarTable=coVariates)
```

```
## 3 PCA scores clusters identified by PAM
```

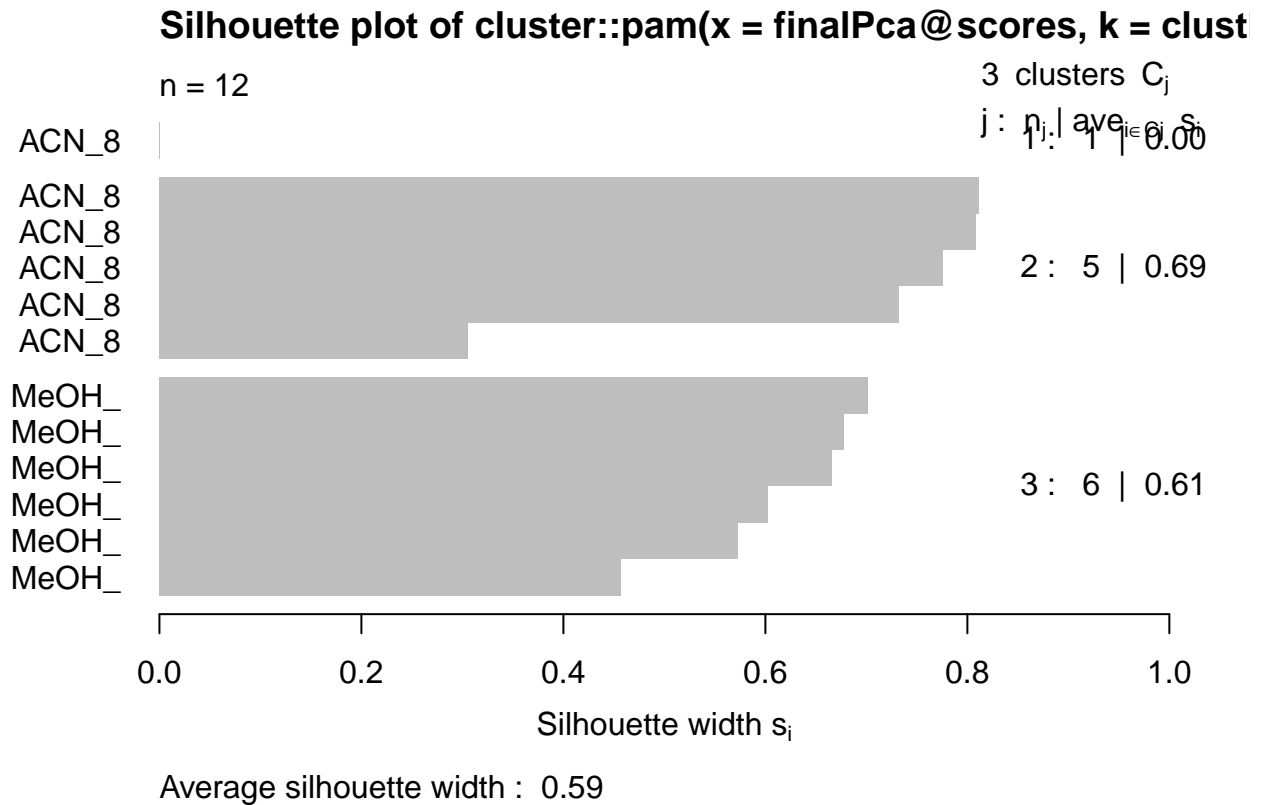
```
## largest coefficient of determination (r2) from linear modelling of all
## co-variables and their two-factor interactions to PCA scores clusters:
## "extractionType" + "runOrder"
## "0.8668"
```

```
# plot pam cluster model (partitioning around the medoids), minimisation of  
# dissimilarities.  
plot(cluster::pam(finalPca@scores, clustIdentity[[1]]$nc))
```

**clusplot(cluster::pam(x = finalPca@scores, k = clustIdentity[[1]]\$nc**



These two components explain 100 % of the point variability.



### 3. Univariate statistical analysis by co-variate based automatic test type selection.

The most appropriate univariate statistical method is selected based on frequency of factor levels of a co-variate (y-variable) supplied. This provides objective and automatable means of test selection. Multiple comparison adjustment can also be performed (e.g. Bonferroni).

```
# outliers removed peak table from pcaOutId output
outRemPeakTable <- pcaOutResults$outRem
obsNames <- colnames(outRemPeakTable)[grep('ACN_|MeOH_', colnames(outRemPeakTable))]

# automatic univariate statistical method selection and mean/median fold calculation
statResult <- coVarTypeStat(outRemPeakTable, obsNames,
                           coVariate=coVariates$extractionType,
                           Logged=T, base=2)

# volcanoPlot
volcanoPlot(log2(statResult[[5]]$FoldChange), statResult[[5]]$p.value)
```

