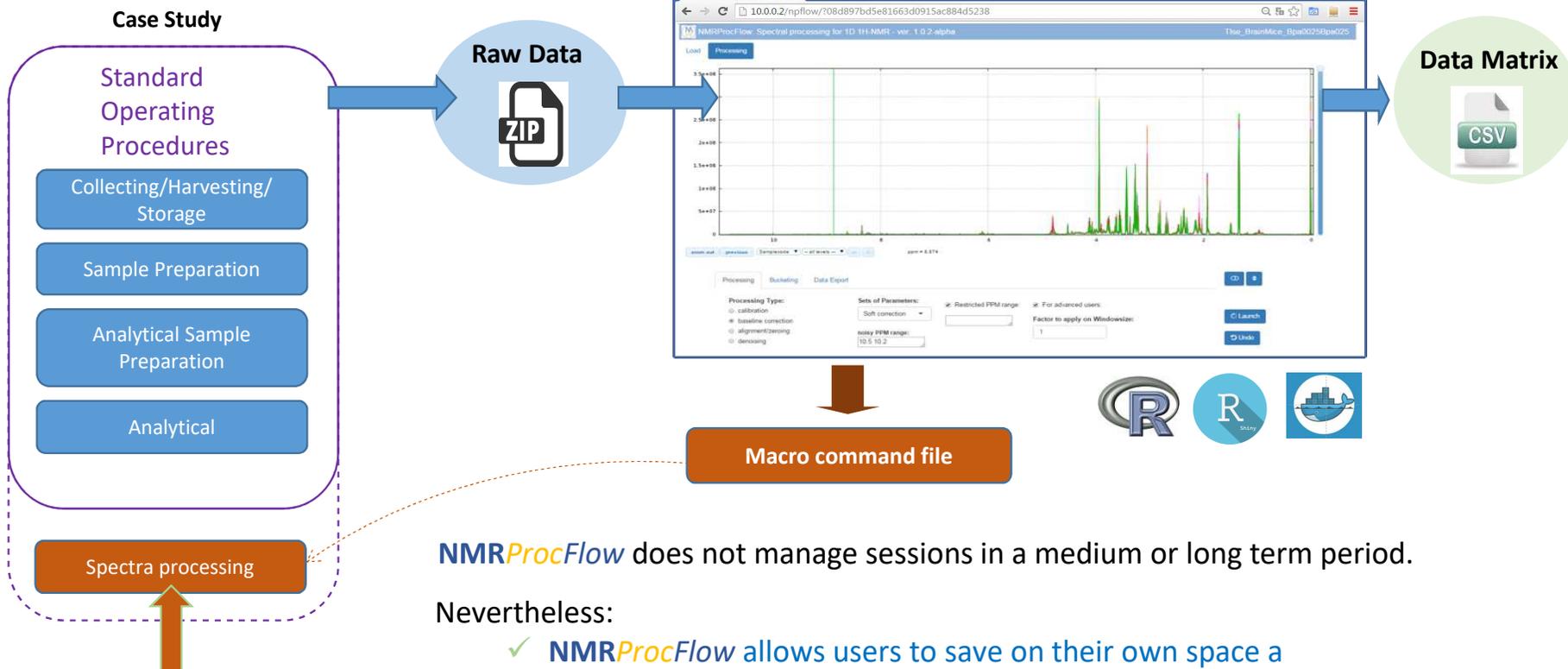


## Interactive mode



### Completes the SOP list

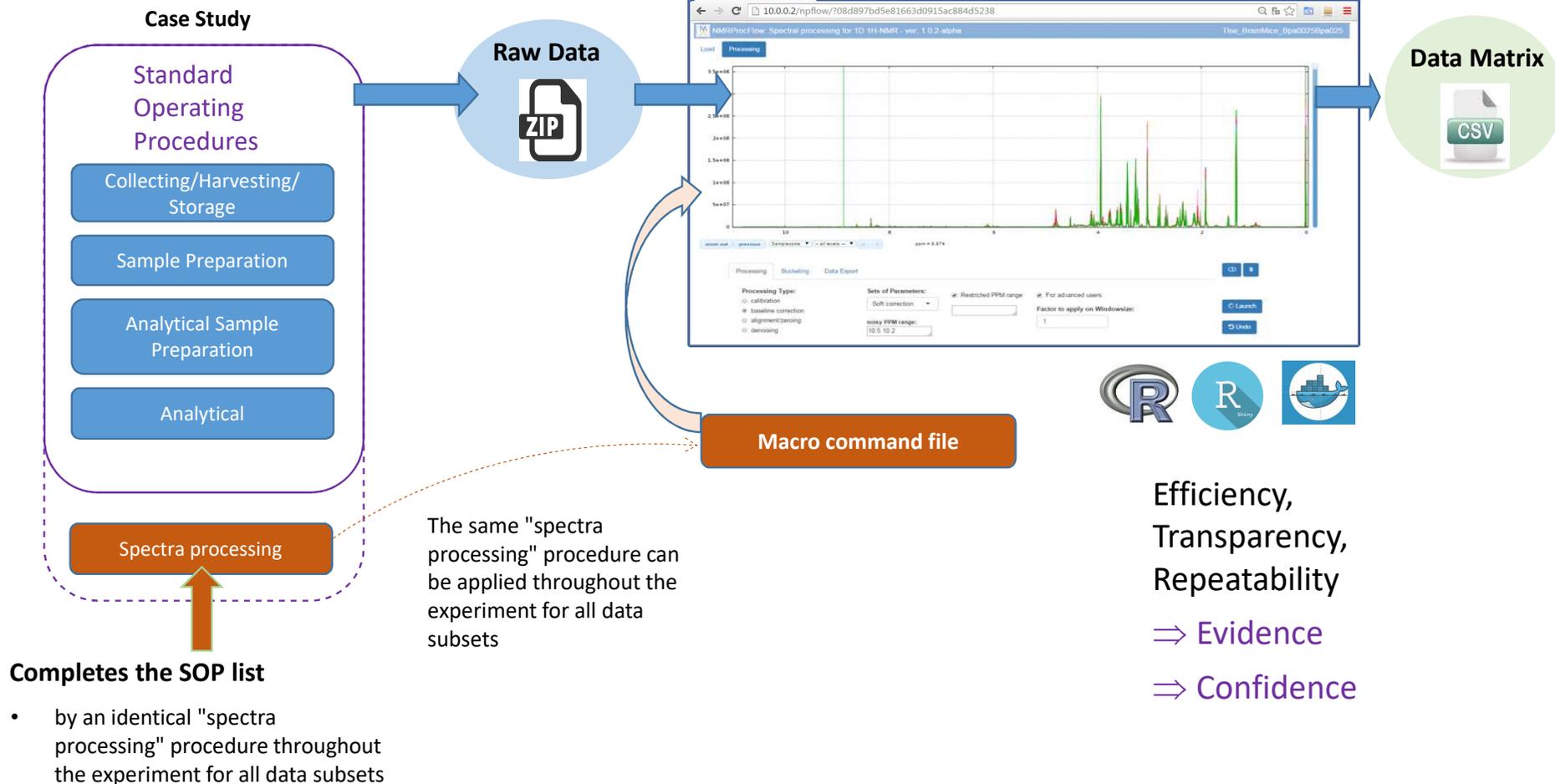
- by an identical "spectra processing" procedure throughout the experiment for all data subsets

NMRProcFlow does not manage sessions in a medium or long term period.

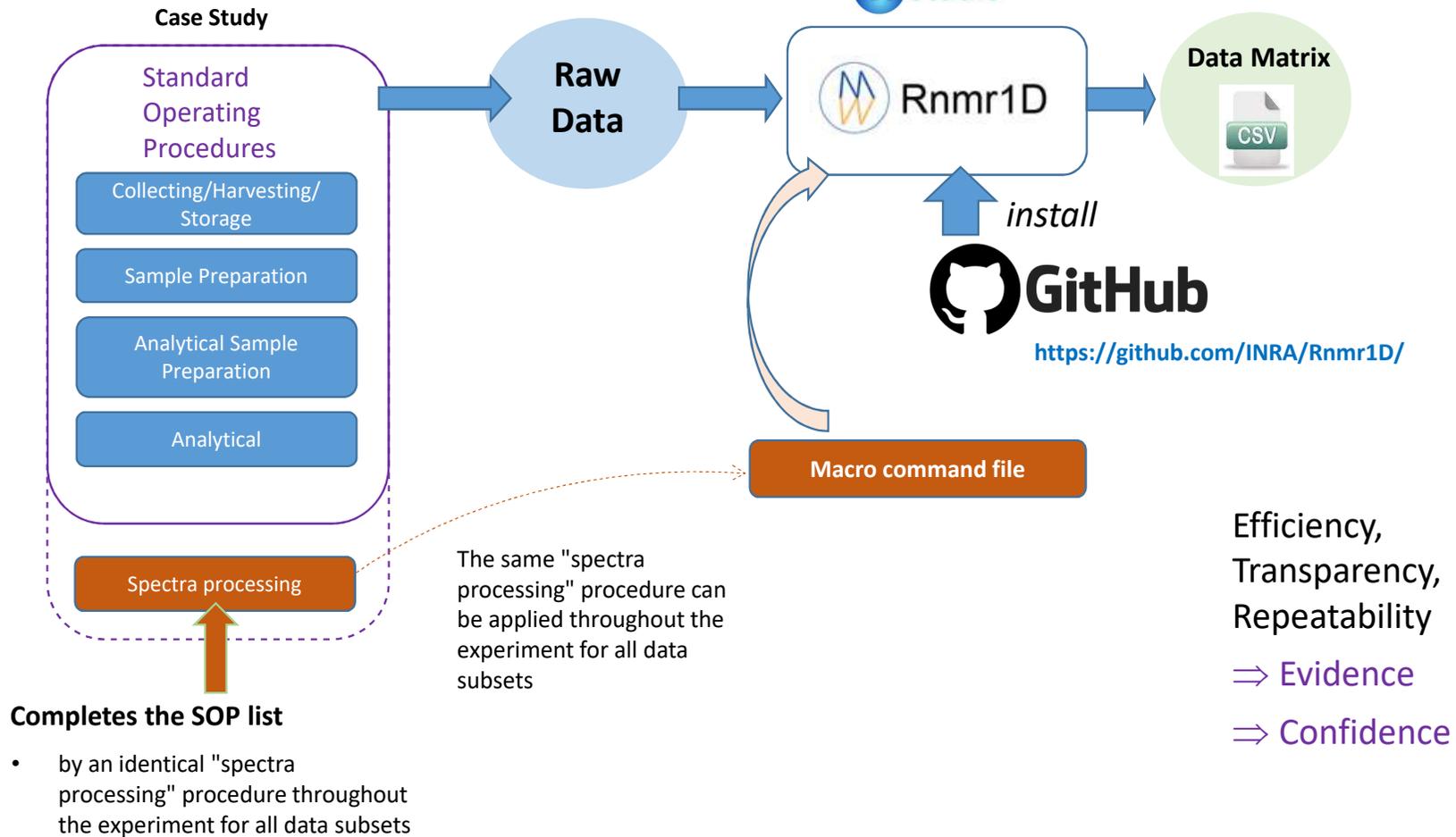
Nevertheless:

- ✓ NMRProcFlow allows users to save on their own space a minimal set of small files in order to recover / replay their session

## Interactive mode



**Batch mode**



# Example of Input Data

## Bruker raw spectra folders

## Macro command file

NP\_macro\_cmd.txt

```
###Vendor=bruker;Type=fid;LB=0.3;ZF=2;BLPHC=FALSE;PHC1=FALSE;FP=0;TSP=TRUE;

#
#Normalisation(.CSN.)of.the.Intensities.based.on.the.selected.PPM.ranges...
#
normalisation.CSN
5.204.9.435
0.607.4.578
EOL

#
#Global.Baseline.Correction:PPM.Range=(.0.499909071543063.,.10.9999706891186.)
#
gbaseline.10.2.10.5.-0.499909071543063.10.9999706891186.50.30.

#
#Baseline.Correction:PPM.Range=(.0.644.,.2.927.)
#
airpls.0.644.2.927.4.

#
#Zeroing.the.selected.zones...
#
zero
2.736.2.79
2.235.2.289
1.547.1.639
1.264.1.359
0.856.0.94
EOL

#
#Baseline.Correction:PPM.Range=(.2.924.,.5.529.)
#
airpls.2.924.5.529.2.6.
```

## Samples.txt

Optional (see <https://nmrprocflow.org/c1>)

	A	B	C	D	E	F
1	Spectrum	Samplecode	EXPNO	PROCNO	Genotype	Treatment
2	MMBBI_14P05-01	G1_A_01	1	1	G1	A
3	MMBBI_14P05-02	G1_A_02	1	1	G1	A
4	MMBBI_14P05-03	G1_A_03	1	1	G1	A
5	MMBBI_14P05-04	G1_B_01	1	1	G1	B
6	MMBBI_14P05-05	G1_B_02	1	1	G1	B
7	MMBBI_14P05-06	G1_B_03	1	1	G1	B
8	MMBBI_14P05-07	G2_A_01	1	1	G2	A
9	MMBBI_14P05-08	G2_A_02	1	1	G2	A
10	MMBBI_14P05-09	G2_A_03	1	1	G2	A
11	MMBBI_14P05-10	G2_B_01	1	1	G2	B
12	MMBBI_14P05-11	G2_B_02	1	1	G2	B
13	MMBBI_14P05-12	G2_B_03	1	1	G2	B



Macro-command Reference Guide

<https://nmrprocflow.org/themes/pdf/Macrocommand.pdf>



## Installation of the R package Rnmr1D

```
> require(devtools)
Le chargement a nécessité le package : devtools
Warning message:
le package 'devtools' a été compilé avec la version R 3.5.1

> install_github("INRA/Rnmr1D", dependencies = TRUE)
Downloading GitHub repo INRA/Rnmr1D@master
from URL https://api.github.com/repos/INRA/Rnmr1D/zipball/master
Installing Rnmr1D
"C:/PROGRA~1/R/R-35~1.0/bin/x64/R" --no-site-file --no-envIRON --no-save --no-restore --quiet CMD INSTALL \
"C:/Users/djaco/AppData/Local/Temp/Rtmp84Kixv/devtools5ec25123d01/INRA-Rnmr1D-1db1cb8" \
--library="C:/Users/djaco/Documents/R/win-library/3.5" --install-tests

* installing *source* package 'Rnmr1D' ...

** libs
c:/Rtools/mingw_64/bin/g++ -I"C:/PROGRA~1/R/R-35~1.0/include" -DNDEBUG -I"C:/Users/djaco/Documents/R/win-
library/3.5/Rcpp/include"
-O2 -Wall -mtune=generic -c RcppExports.cpp -o RcppExports.o
c:/Rtools/mingw_64/bin/g++ -I"C:/PROGRA~1/R/R-35~1.0/include" -DNDEBUG -I"C:/Users/djaco/Documents/R/win-
library/3.5/Rcpp/include"
-O2 -Wall -mtune=generic -c libCspec.cpp -o libCspec.o
c:/Rtools/mingw_64/bin/g++ -shared -s -static-libgcc -o Rnmr1D.dll tmp.def RcppExports.o libCspec.o -LC:/PROGRA~1/R/R-
35~1.0/bin/x64 -lR
installing to C:/Users/djaco/Documents/R/win-library/3.5/Rnmr1D/libs/x64
** R
** data
** inst
** byte-compile and prepare package for lazy loading
** help
*** installing help indices
converting help for package 'Rnmr1D'
finding HTML links ... fini
** building package indices
** installing vignettes
** testing if installed package can be loaded
* DONE (Rnmr1D)
```



<https://github.com/INRA/Rnmr1D/>



## Example of use

```
> library(Rnmr1D)
```

```
> data_dir <- system.file("extra", package = "Rnmr1D")  
> RAWDIR <- file.path(data_dir, "MMBBI_14P05")  
> CMDFILE <- file.path(data_dir, "NP_macro_cmd.txt")  
> SAMPLEFILE <- file.path(data_dir, "Samples.txt")
```

```
> detectCores()  
[1] 8
```

```
> out <- Rnmr1D::doProcessing(RAWDIR, cmdfile=CMDFILE, samplefile=SAMPLEFILE, ncpu=detectCores())
```

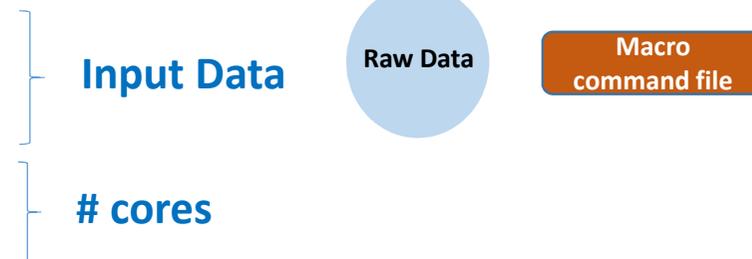
```
Rnmr1D: --- READING and CONVERTING ---
```

```
Rnmr1D: Vendor=bruker, Type=fid, LB=0.3, ZF=2, BLPHC=FALSE, PHC1=FALSE, FP=0, TSP=TRUE;
```

```
Rnmr1D: Generate the 'samples' & 'factors' files from the list of raw spectra
```

```
Rnmr1D: -- Nb Spectra = 12 -- Nb Cores = 8
```

```
Rnmr1D: Generate the final matrix of spectra...
```



## Processing applied on 8 cores

...



MS Windows 7 / 64bits

"Virtualization Technology" activated in the BIOS



...

```

Rnmr1D: -----
Rnmr1D: Process the Macro-commands file
Rnmr1D: -----
Rnmr1D:
Rnmr1D: Normalisation of the Intensities based on the selected PPM ranges...
Rnmr1D:   Method =CSN
Rnmr1D: Baseline Correction: PPM Range = ( -0.499909071543063 , 10.9999706891186 )
Rnmr1D:   Type=Global - Smoothing Parameter=50 - window Size=30
Rnmr1D: Baseline Correction: PPM Range = ( 0.644 , 2.927 )
Rnmr1D:   Type=airPLS, lambda= 4
Rnmr1D: Zeroing the selected PPM ranges ...
Rnmr1D:   Zone 1 = ( 2.736 , 2.79 )
Rnmr1D:   Zone 2 = ( 2.235 , 2.289 )
Rnmr1D:   Zone 3 = ( 1.547 , 1.639 )
Rnmr1D:   Zone 4 = ( 1.264 , 1.359 )
Rnmr1D:   Zone 5 = ( 0.856 , 0.94 )
Rnmr1D: Baseline Correction: PPM Range = ( 2.924 , 5.529 )
Rnmr1D:   Type=airPLS, lambda= 2.6
Rnmr1D: Zeroing the selected PPM ranges ...
Rnmr1D:   Zone 1 = ( 4.626 , 4.978 )
Rnmr1D: Baseline Correction: PPM Range = ( 5.795 , 8.393 )
Rnmr1D:   Type=airPLS, lambda= 2
Rnmr1D: Alignment: PPM Range = ( 5.776 , 8.389 )
Rnmr1D:   CluPA - Resolution =0.03 - SNR threshold=5 - Reference=0
Rnmr1D:   --- Peak detection : nDivRange = 164
Rnmr1D:   --- Peak detection time: 3.789999999999999 sec
Rnmr1D:   --- The reference spectrum is: 1
Rnmr1D:   --- Spectra alignment to the reference: maxShift = 41
Rnmr1D:   --- Spectra alignment time: 1.180000000000001 sec
Rnmr1D: Alignment: PPM Range = ( 0.554 , 4.731 )
Rnmr1D:   CluPA - Resolution =0.03 - SNR threshold=5 - Reference=0
Rnmr1D:   --- Peak detection : nDivRange = 164
Rnmr1D:   --- Peak detection time: 5.63 sec
Rnmr1D:   --- The reference spectrum is: 1
Rnmr1D:   --- Spectra alignment to the reference: maxShift = 41
Rnmr1D:   --- Spectra alignment time: 2.039999999999999 sec
Rnmr1D: Bucketing the selected PPM ranges ...
Rnmr1D:   AIBIN - Resolution =0.3 - SNR threshold=3 - Append=0
Rnmr1D:   Zone 1 = ( 4.788 , 9.309 ), Nb Buckets = 216
Rnmr1D:   Zone 2 = ( 0.615 , 4.67 ), Nb Buckets = 341
Rnmr1D:   Total Buckets = 557
  
```

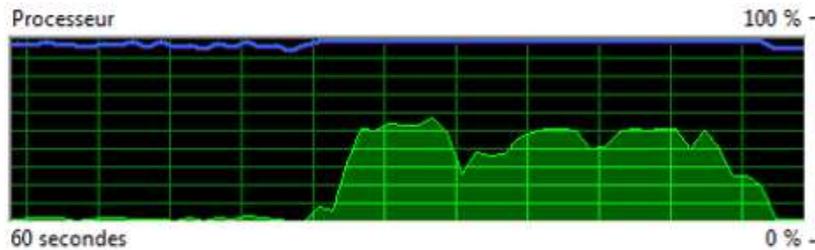
Macro command file

## Example of use

### Performance monitoring



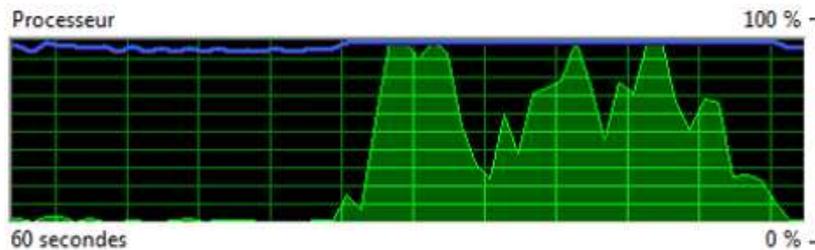
## Performance monitoring



Processing applied on 4 cores

=> 4 physical cores

```
out <- Rnmr1D::doProcessing (RAWDIR, cmdfile=CMDFILE,  
                             ncpu=detectCores(logical=FALSE))
```



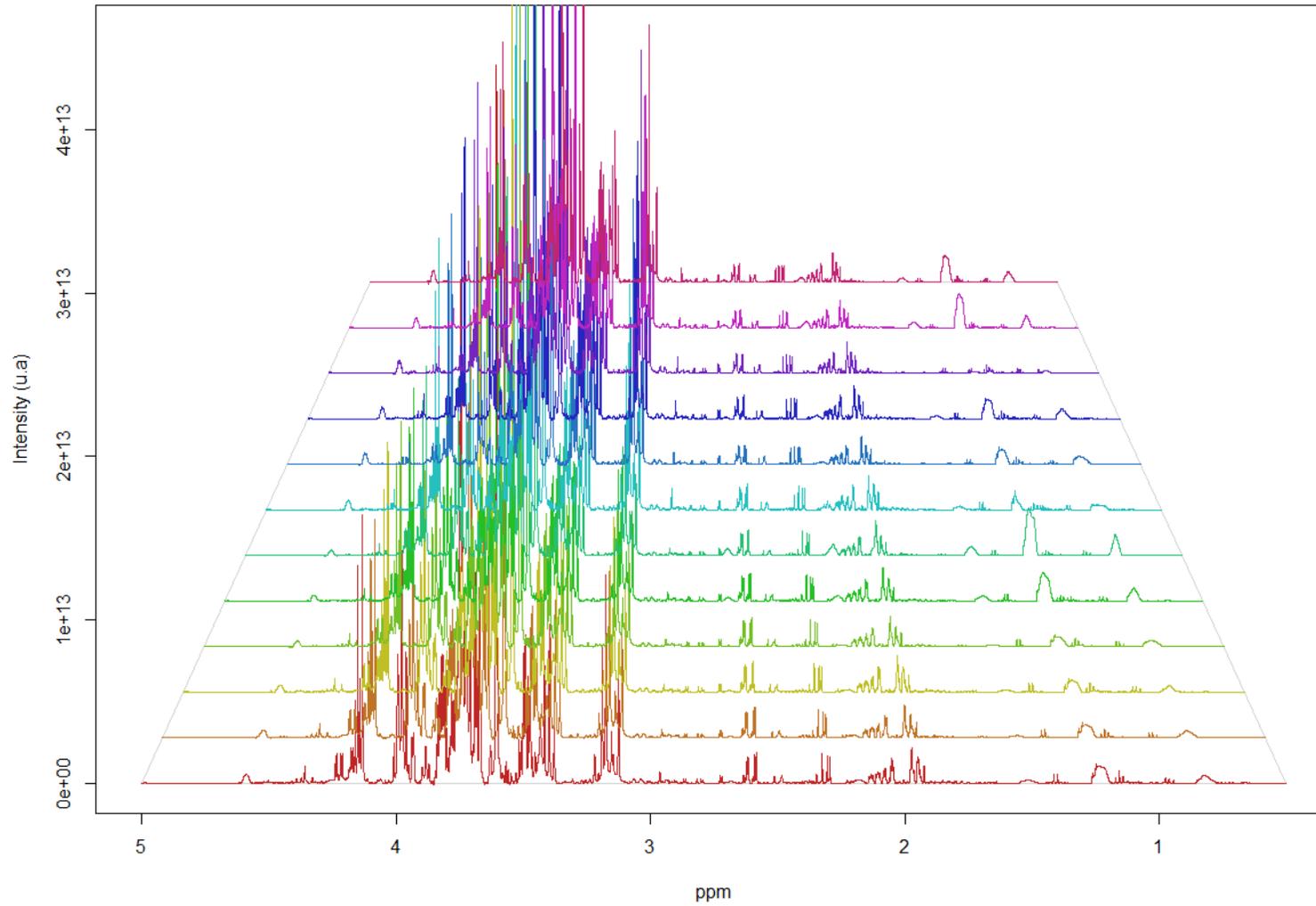
Processing applied on 8 cores

=> 4 physical cores + 4 logical cores

```
out <- Rnmr1D::doProcessing (RAWDIR, cmdfile=CMDFILE,  
                             ncpu=detectCores())
```

The time benefit is very minimal or even insignificant

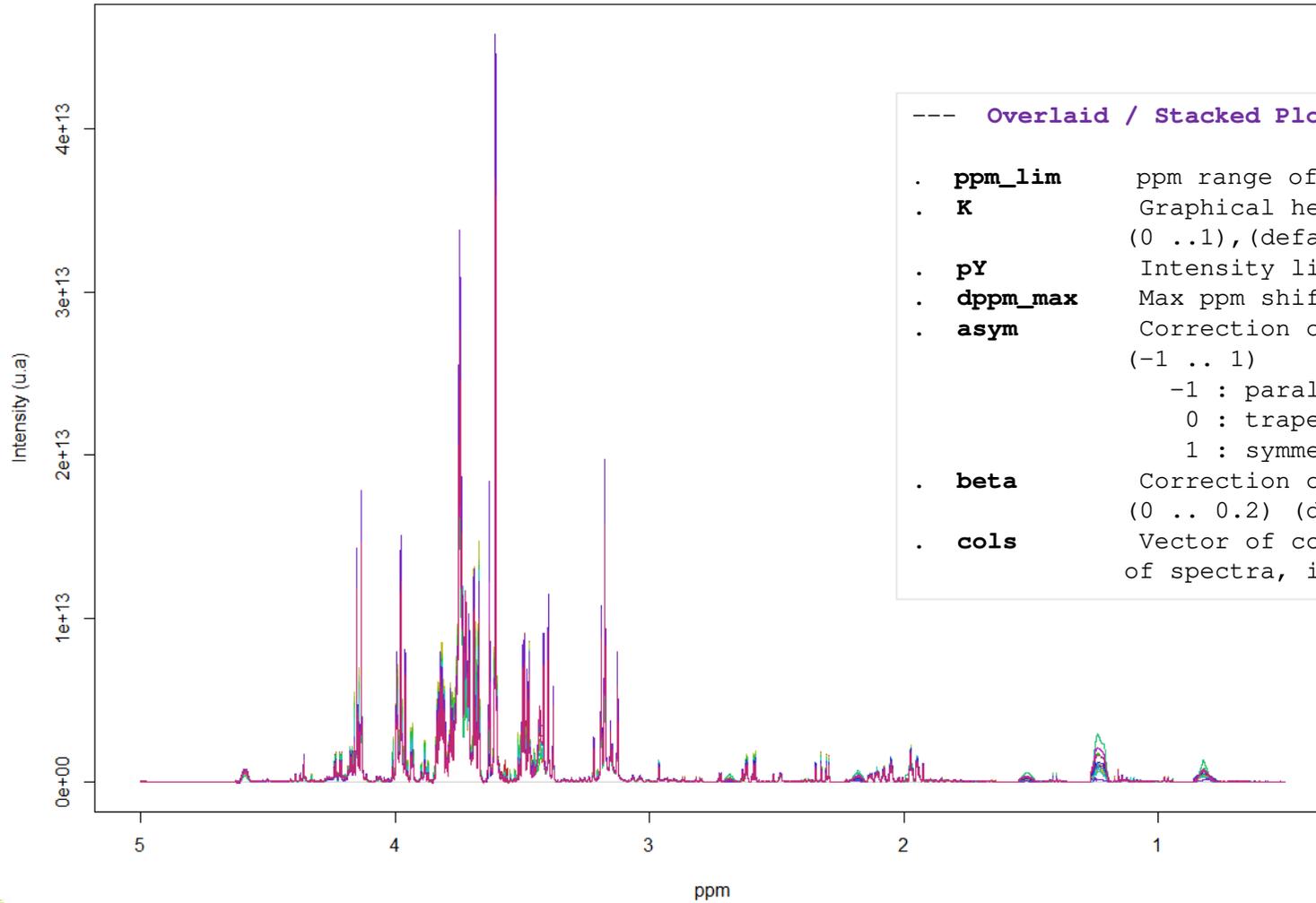
> # Stacked Plot with a perspective effect  
> plotSpecMat(out\$specMat, ppm\_lim=c(0.5,5))





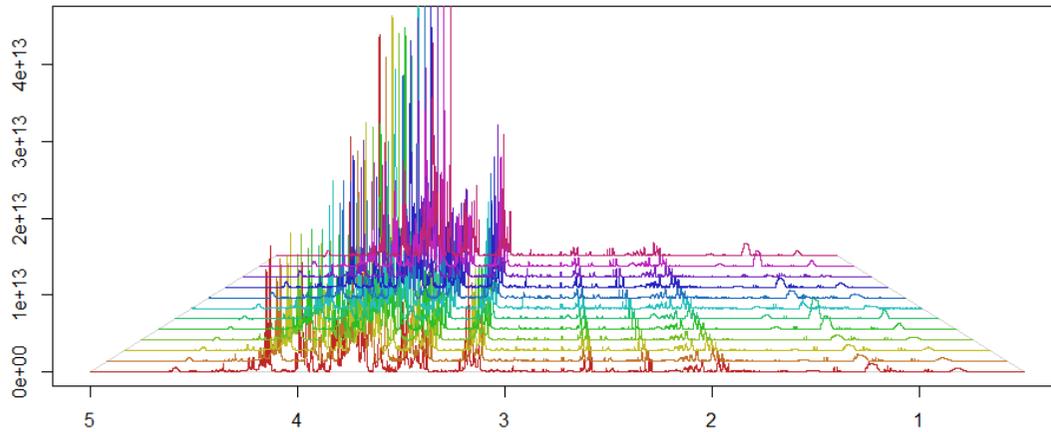
Rnmr1D

```
> # Overlaid Plot  
> plotSpecMat(out$specMat, ppm_lim=c(0.5,5), K=0)
```

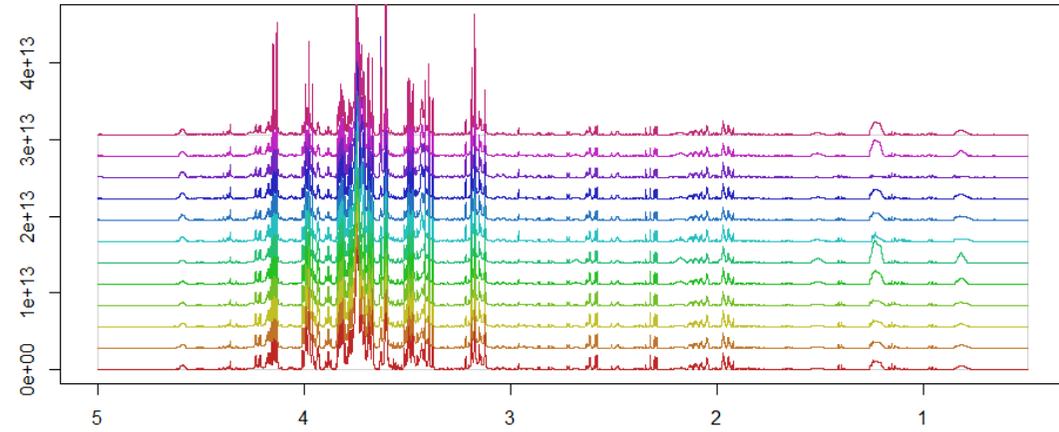


```
--- Overlaid / Stacked Plot ---  
  
. ppm_lim ppm range of the plot  
. K Graphical height of the stack  
 (0 ..1), (default=0.67)  
. pY Intensity limit factor (default 1)  
. dppm_max Max ppm shift to have a perspective effect  
. asym Correction of vertical parallax effect  
 (-1 .. 1)  
 -1 : parallelogram  
 0 : trapeze with maximum asymmetric  
 1 : symmetric trapeze  
. beta Correction of horizontal parallax effect  
 (0 .. 0.2) (default 0)  
. cols Vector of colors (same size that the number  
 of spectra, i.e dim(specmat)[1])
```

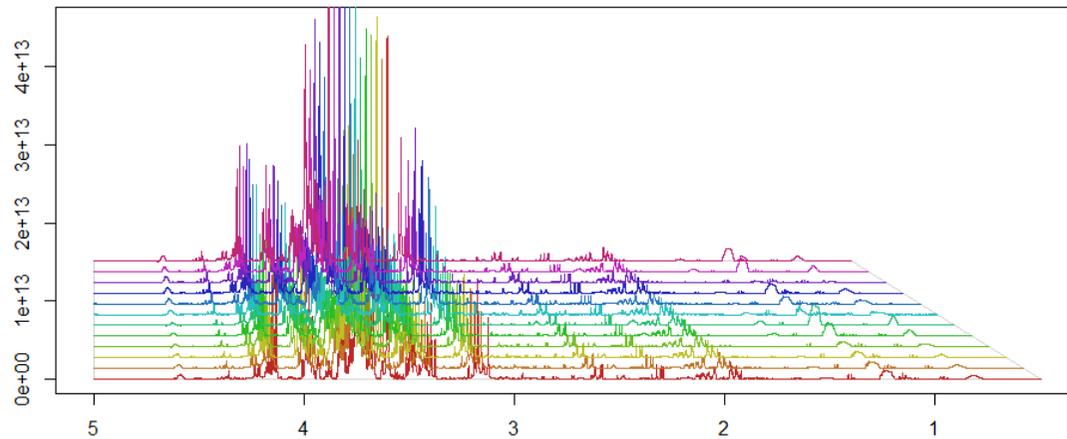
```
> plotSpecMat(out$specMat, ppm_lim=c(0.5,5), K=0.33, asym=1)
```



```
> plotSpecMat(out$specMat, ppm_lim=c(0.5,5), K=0.67, dppm_max=0)
```

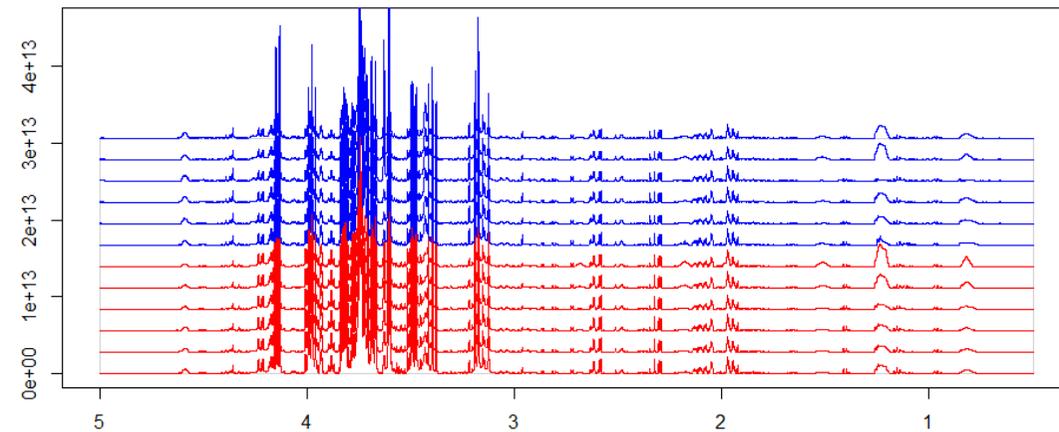


```
> plotSpecMat(out$specMat, ppm_lim=c(0.5,5), K=0.33, asym=0)
```



```
> cols <- c(rep("red",6), rep("blue",6))
```

```
> plotSpecMat(out$specMat, ppm_lim=c(0.5,5), K=0.67, dppm_max=0, cols=cols)
```



```

> out$infos
  Spectrum      Samplecode EXPNO  PROCNO  PULSE  NUC  SOLVENT  GRPDLY  PHC0          PHC1
[1,] "MMBBI_14P05-01" "G1_A_01"   "1"   "0"   "zg"   "1H"  "D2O"   "76"   "5.81985760769275" "0"
[2,] "MMBBI_14P05-02" "G1_A_02"   "1"   "0"   "zg"   "1H"  "D2O"   "76"   "5.81348967276733" "0"
[3,] "MMBBI_14P05-03" "G1_A_03"   "1"   "0"   "zg"   "1H"  "D2O"   "76"   "5.81340469257522" "0"
[4,] "MMBBI_14P05-04" "G1_B_01"   "1"   "0"   "zg"   "1H"  "D2O"   "76"   "5.80362632296259" "0"
[5,] "MMBBI_14P05-05" "G1_B_02"   "1"   "0"   "zg"   "1H"  "D2O"   "76"   "5.8091087484546"  "0"
[6,] "MMBBI_14P05-06" "G1_B_03"   "1"   "0"   "zg"   "1H"  "D2O"   "76"   "5.77728740812383" "0"
[7,] "MMBBI_14P05-07" "G2_A_01"   "1"   "0"   "zg"   "1H"  "D2O"   "76"   "5.76934290112164" "0"
[8,] "MMBBI_14P05-08" "G2_A_02"   "1"   "0"   "zg"   "1H"  "D2O"   "76"   "5.8030015185569"  "0"
[9,] "MMBBI_14P05-09" "G2_A_03"   "1"   "0"   "zg"   "1H"  "D2O"   "76"   "5.81506120257431" "0"
[10,] "MMBBI_14P05-10" "G2_B_01"   "1"   "0"   "zg"   "1H"  "D2O"   "76"   "5.8022149319462"  "0"
[11,] "MMBBI_14P05-11" "G2_B_02"   "1"   "0"   "zg"   "1H"  "D2O"   "76"   "5.80645182549586" "0"
[12,] "MMBBI_14P05-12" "G2_B_03"   "1"   "0"   "zg"   "1H"  "D2O"   "76"   "5.83602325309693" "0"

  SF          SI          SW          SWH          RELAXDELAY O1
[1,] "500.1625008" "65536" "12.0009016085441" "6002.40096038415" "20" "2500.8"
[2,] "500.1625008" "65536" "12.0009016085441" "6002.40096038415" "20" "2500.8"
[3,] "500.1625008" "65536" "12.0009016085441" "6002.40096038415" "20" "2500.8"
[4,] "500.1625008" "65536" "12.0009016085441" "6002.40096038415" "20" "2500.8"
[5,] "500.1625008" "65536" "12.0009016085441" "6002.40096038415" "20" "2500.8"
[6,] "500.1625008" "65536" "12.0009016085441" "6002.40096038415" "20" "2500.8"
[7,] "500.1625008" "65536" "12.0009016085441" "6002.40096038415" "20" "2500.8"
[8,] "500.1625008" "65536" "12.0009016085441" "6002.40096038415" "20" "2500.8"
[9,] "500.1625008" "65536" "12.0009016085441" "6002.40096038415" "20" "2500.8"
[10,] "500.1625008" "65536" "12.0009016085441" "6002.40096038415" "20" "2500.8"
[11,] "500.1625008" "65536" "12.0009016085441" "6002.40096038415" "20" "2500.8"
[12,] "500.1625008" "65536" "12.0009016085441" "6002.40096038415" "20" "2500.8"

```

```
>
> # Get the bucket table
> outBucket <- getBucketsTable(out)
>
> outBucket
  name      center      min      max      width
1  B0_6567 0.6566869 0.6547641 0.6586097 0.001922781
2  B0_7395 0.7395496 0.7346053 0.7444939 0.004944295
3  B0_7541 0.7541078 0.7507201 0.7574956 0.003387757
4  B0_7699 0.7699479 0.7666517 0.7732441 0.003296196
5  B0_7796 0.7795618 0.7732441 0.7858795 0.006317710
6  B0_8155 0.8155453 0.7858795 0.8452110 0.029665767
7  B0_8483 0.8483241 0.8452110 0.8514372 0.003113074
8  B0_8547 0.8547334 0.8514372 0.8580296 0.003296196
9  B0_9439 0.9439138 0.9389695 0.9488581 0.004944295
10 B0_9528 0.9527952 0.9488581 0.9567324 0.003937123
11 B0_9611 0.9611273 0.9567324 0.9655222 0.004394928
12 B0_9729 0.9729386 0.9655222 0.9803551 0.007416442
13 B0_9878 0.9877715 0.9843838 0.9911593 0.003387757
14 B1_0028 1.0027875 0.9984842 1.0070909 0.004303367
15 B1_0128 1.0127677 1.0109365 1.0145989 0.001831220
16 B1_0256 1.0255862 1.0233888 1.0277837 0.002197464
17 B1_0294 1.0294318 1.0277837 1.0310799 0.001648098
18 B1_0601 1.0601047 1.0545195 1.0656900 0.005585222
19 B1_0687 1.0687115 1.0656900 1.0717330 0.003021513
20 B1_0748 1.0747545 1.0717330 1.0777760 0.003021513
21 B1_0819 1.0818963 1.0777760 1.0860165 0.004120245
22 B1_0882 1.0882140 1.0860165 1.0904114 0.002197464
23 B1_0947 1.0947148 1.0904114 1.0990182 0.004303367
24 B1_1088 1.1088152 1.1024975 1.1151329 0.006317710
25 B1_1209 1.1209012 1.1151329 1.1266696 0.005768344
26 B1_1299 1.1298742 1.1266696 1.1330789 0.003204635
27 B1_1401 1.1401291 1.1330789 1.1471793 0.007050198
28 B1_1493 1.1492852 1.1471793 1.1513911 0.002105903
29 B1_1576 1.1576172 1.1513911 1.1638434 0.006226149
30 B1_1693 1.1693370 1.1638434 1.1748307 0.005493661
31 B1_1848 1.1848108 1.1808737 1.1887480 0.003937123
32 B1_2162 1.2162163 1.1887480 1.2436846 0.027468303
33 B1_2460 1.2459736 1.2436846 1.2482626 0.002289025
34 B1_2529 1.2529322 1.2482626 1.2576018 0.004669612
35 B1_2596 1.2596162 1.2576018 1.2616305 0.002014342
36 B1_2633 1.2632786 1.2616305 1.2649267 0.001648098
37 B1_3856 1.3856041 1.3788286 1.3923796 0.006775515
```

[list output truncated]

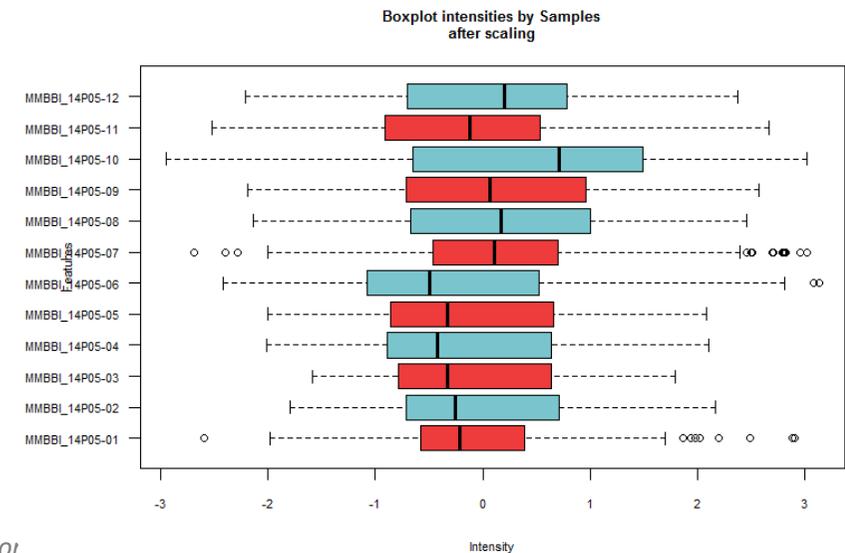
```
>
> # Get the data matrix
> outMat <- getBucketsDataset(out, norm_meth='CSN')
>
> outMat[, 1:10 ]
      BO_6586 BO_7445 BO_7575 BO_7732 BO_7859 BO_8452 BO_8514 BO_8580 BO_9489 BO_9567
MMBBI_14P05-01 2.0868060 2.0199424 2.1007711 2.7945745 26.6212010 635.7661 29.64466 5.2448412 15.9292320 3.780137
MMBBI_14P05-02 1.1431952 0.6581313 1.2469237 2.0584654 26.2619032 574.9338 34.77272 13.6654821 11.4978985 3.286773
MMBBI_14P05-03 1.7037499 0.5358581 1.7765561 1.5784654 8.1977894 456.0004 23.90461 5.1369353 15.5697662 3.619191
MMBBI_14P05-04 0.9248881 0.6907766 1.4273042 4.2105875 44.5256946 615.0048 53.04108 23.5083777 14.0492691 3.513393
MMBBI_14P05-05 2.5431175 0.6088705 1.2321557 -0.2722329 5.7977130 983.4729 41.46468 0.0000000 16.9404791 3.708823
MMBBI_14P05-06 0.9061663 0.2807962 0.4052053 0.1383976 0.5969783 1418.7925 30.10915 0.5355811 14.6727608 3.371711
MMBBI_14P05-07 1.0569201 3.1993161 2.1275805 17.4491980 72.2686120 564.9540 59.37757 26.6360799 -0.1160904 4.075282
MMBBI_14P05-08 3.3681517 3.5658065 2.4636534 22.8185308 93.0990640 892.7992 78.93366 39.2594696 8.3370352 2.883901
MMBBI_14P05-09 2.6442803 4.6951611 3.1573458 4.3021910 51.3249414 879.8324 63.22155 11.8336718 8.5829288 4.779452
MMBBI_14P05-10 2.8121621 4.1984831 3.0757877 -0.8206727 8.5941133 178.2964 17.41676 38.8074119 10.6804219 4.601391
MMBBI_14P05-11 3.4707968 3.4499198 2.4150302 1.9295123 10.3344573 955.5954 25.72136 1.1370820 8.3515091 3.290860
MMBBI_14P05-12 1.8400555 3.6281613 2.0266029 4.5196354 44.0279187 954.1062 46.79817 11.4727439 10.5812756 3.950967
```

```
> out$samples
      Spectrum Samplecode Genotype Treatment
1 MMBBI_14P05-01 G1_A_01 G1 A
2 MMBBI_14P05-02 G1_A_02 G1 A
3 MMBBI_14P05-03 G1_A_03 G1 A
4 MMBBI_14P05-04 G1_B_01 G1 B
5 MMBBI_14P05-05 G1_B_02 G1 B
6 MMBBI_14P05-06 G1_B_03 G1 B
7 MMBBI_14P05-07 G2_A_01 G2 A
8 MMBBI_14P05-08 G2_A_02 G2 A
9 MMBBI_14P05-09 G2_A_03 G2 A
10 MMBBI_14P05-10 G2_B_01 G2 B
11 MMBBI_14P05-11 G2_B_02 G2 B
12 MMBBI_14P05-12 G2_B_03 G2 B
```

```
> out$factors[,2]
[1] "Samplecode" "Genotype" "Treatment"
```

```
outMat.scaled <- scale(na.omit(outMat), center=T, scale=T)

par(las=1, mar=c(5, 6, 4, 2), ps=8)
boxplot( t(outMat.scaled),
  main=paste('Boxplot intensities by Samples\nafter scaling'),
  horizontal = T, xlab='\n\n\n\n\nIntensity', ylab='Features',
  col=c("brown2", "cadetblue3"))
```



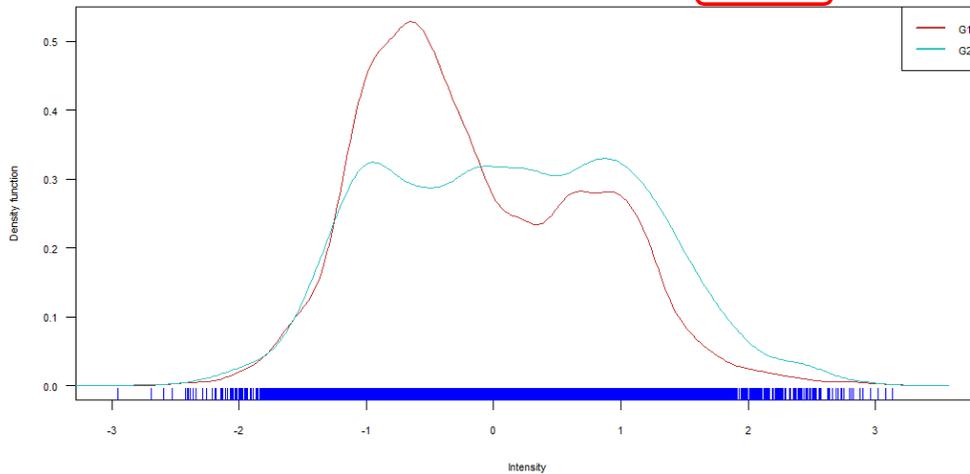


Rnmr1D

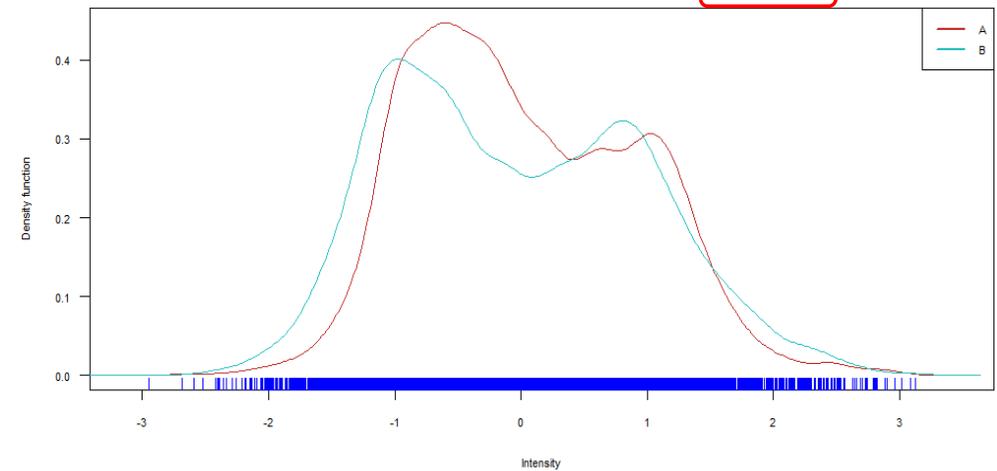
```
> require(reshape)
Le chargement a nécessité le package : reshape
> require(ks)
Le chargement a nécessité le package : ks
```

```
DensityByGroups <- function(x, groups) {
  Groups <- as.factor(groups)
  colors <- rainbow(length(levels(Groups)), s=0.8, v=0.75)
  for (i in 1:length(levels(Groups))) {
    selectRows <- (Groups==levels(Groups)[i])
    ifelse (i==1, fadd<-FALSE, fadd<-TRUE)
    subData <- x[selectRows, ]
    dataIn<-melt(subData)
    h <- hpi(x=dataIn[,3])
    fhat <- kde(x=dataIn[,3], h=h)
    plot(fhat, drawpoints=TRUE, xlab='Intensity', col=colors[i], add=fadd, cex=1)
  }
  legend("topright", legend=levels(Groups), col=colors[c(1:nlevels(Groups))], lwd=2, lty=1, horiz=FALSE, cex=1)
}
```

```
> DensityByGroups(outMat.scaled , out$samples$Genotype)
```



```
> DensityByGroups(outMat.scaled , out$samples$Treatment)
```



# Ropls - <https://bioconductor.org/packages/release/bioc/html/ropls.html>

```
> library(ropls)
```

```
> outMat.pca <- oplS(outMat)
```

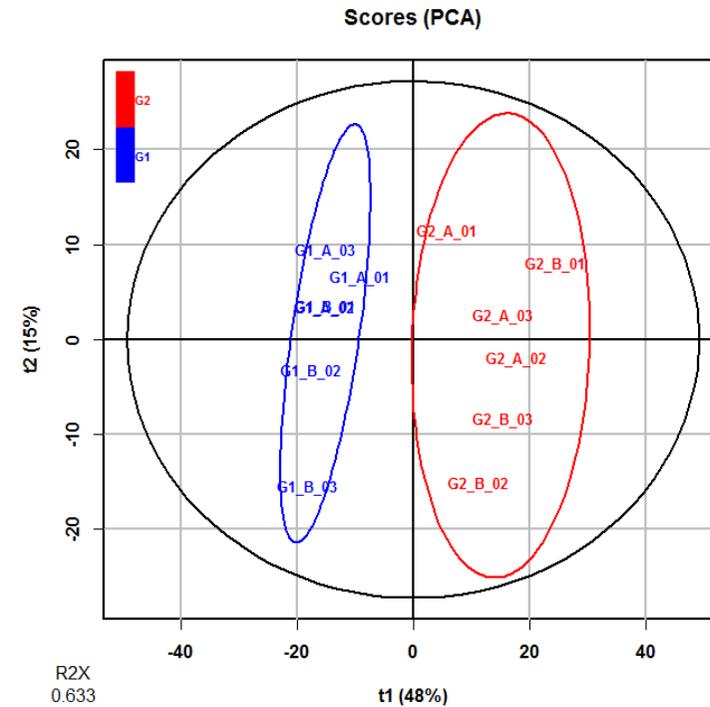
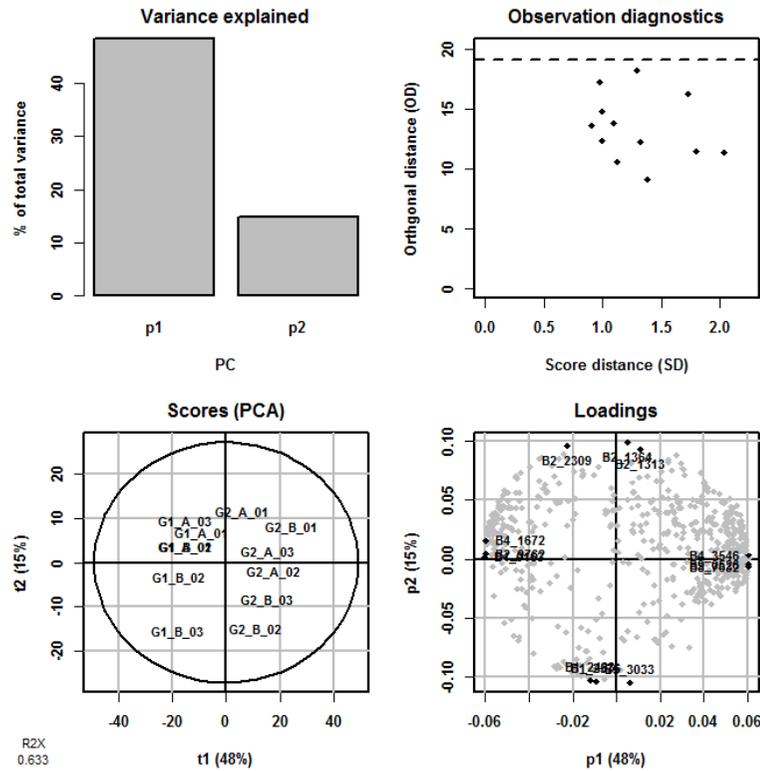
PCA

12 samples x 557 variables

standard scaling of predictors

```
R2X(cum) pre ort  
Total 0.633 2 0
```

```
> plot(outMat.pca, typeVc = "x-score",  
       parAsColFcvn = out$samples$Genotype,  
       parEllipsesL = TRUE)
```



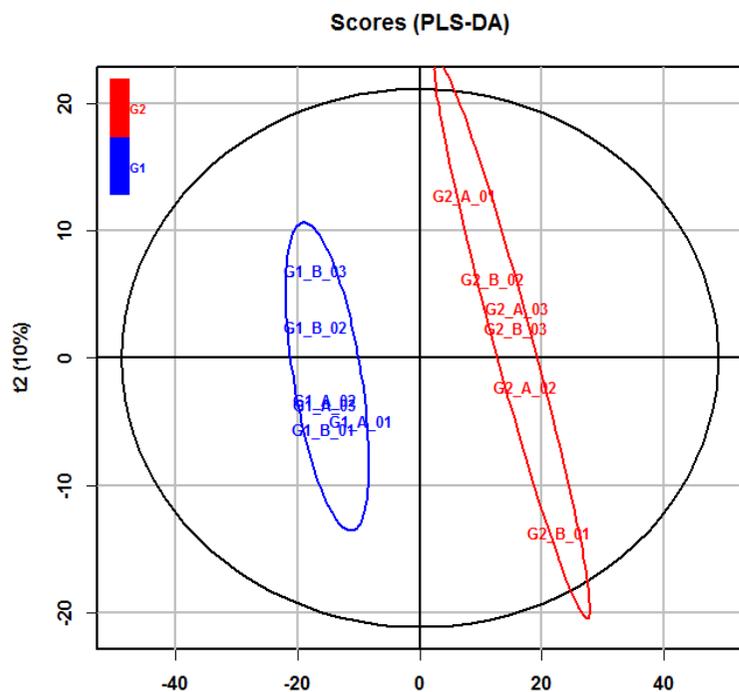
# Ropls

```
> outMat.plsda <- oplS(outMat, out$samples$Genotype)
```

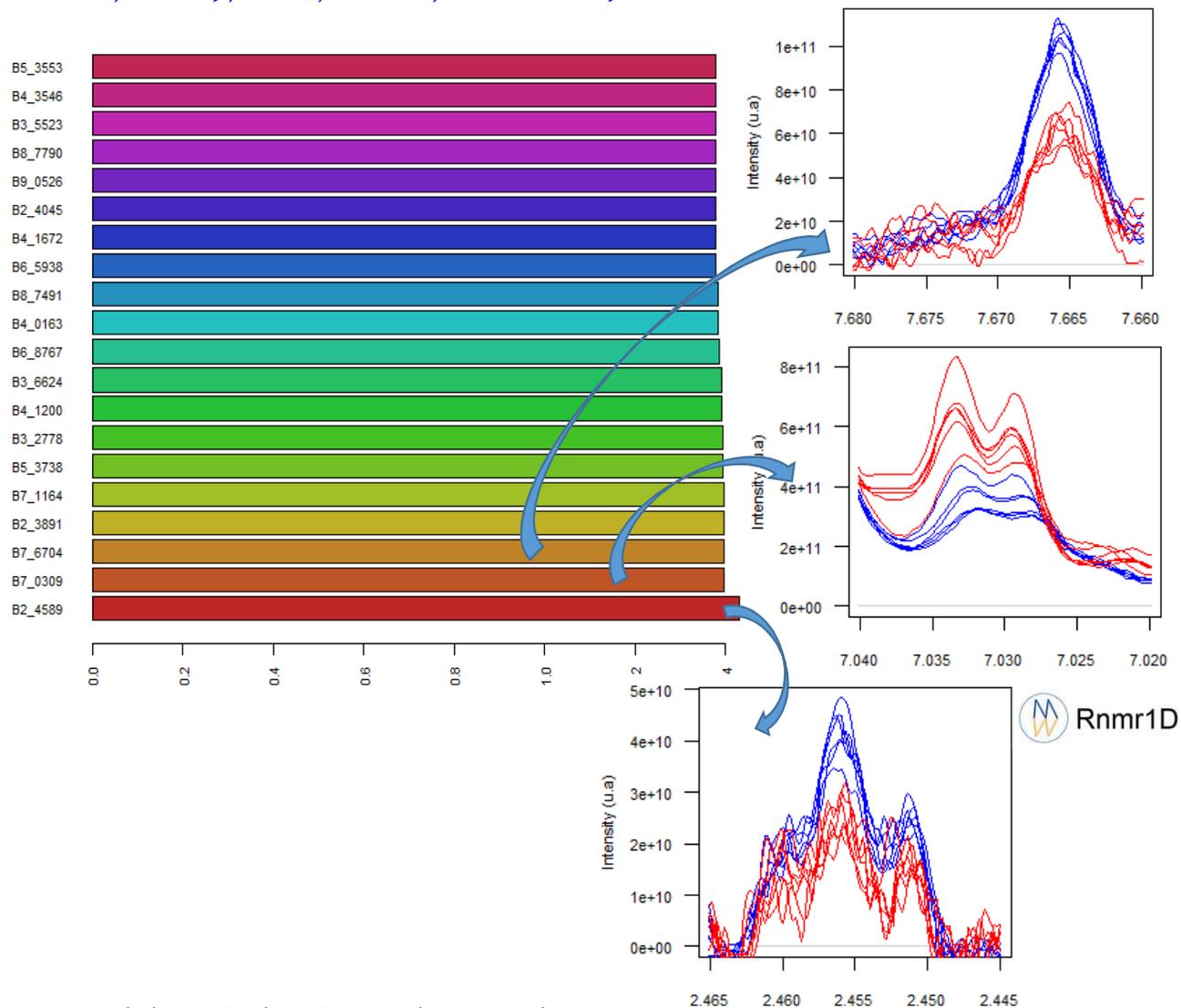
PLS-DA 12 samples x 557 variables and 1 response  
standard scaling of predictor(s) and response(s)

	R2X(cum)	R2Y(cum)	Q2(cum)	RMSEE	pre	ort	pR2Y	pQ2
Total	0.667	0.998	0.935	0.0297	3	0	0.05	0.05

```
> plot(outMat.plsda, typevc = "x-score",
       parAscColFcvn = out$samples$Genotype,
       parEllipsesL = TRUE)
```



```
> VIP <- outMat.plsda@vipvn[order(outMat.plsda@vipvn, decreasing = TRUE)][1:20]
> barplot(VIP, main="VIP", names.arg=names(VIP), horiz=TRUE, col=rainbow(20),
         s=0.8, v=0.75), las=2, cex=0.7, cex.axis=0.7)
```



Bucket Clustering based on a cut threshold applied on the buckets tree resulting of a hierarchical classification analysis. (cut threshold automatically estimated)

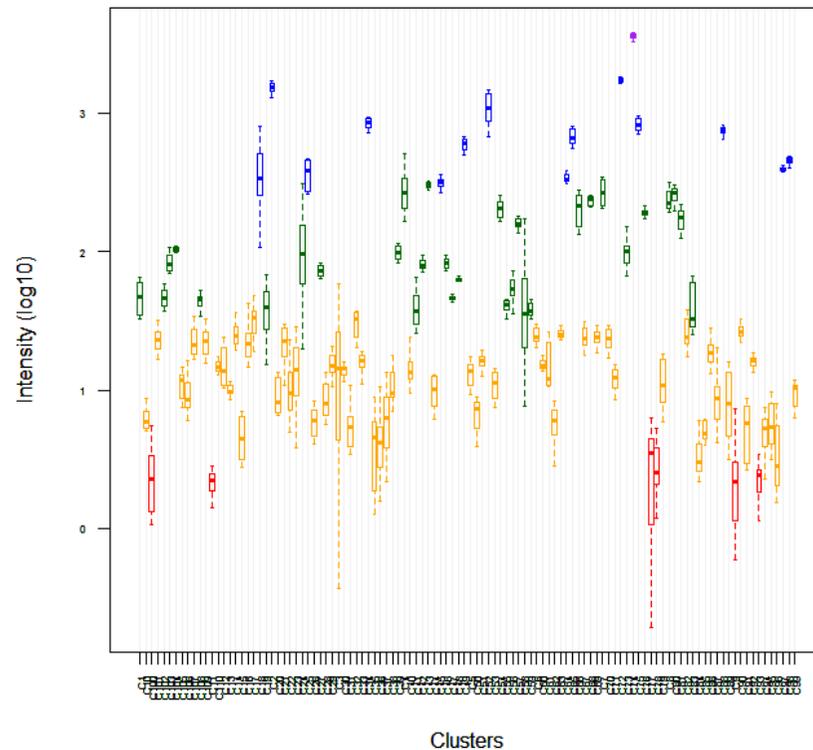
```
> outclust <- getClusters(outMat, method='hca')
```

```
#-- Clustering --
# Distance Method: euclidean
# Agglomeration Method: complete
# Cutting Tree threshold: 0.26
# Nb Clusters: 110
#
```

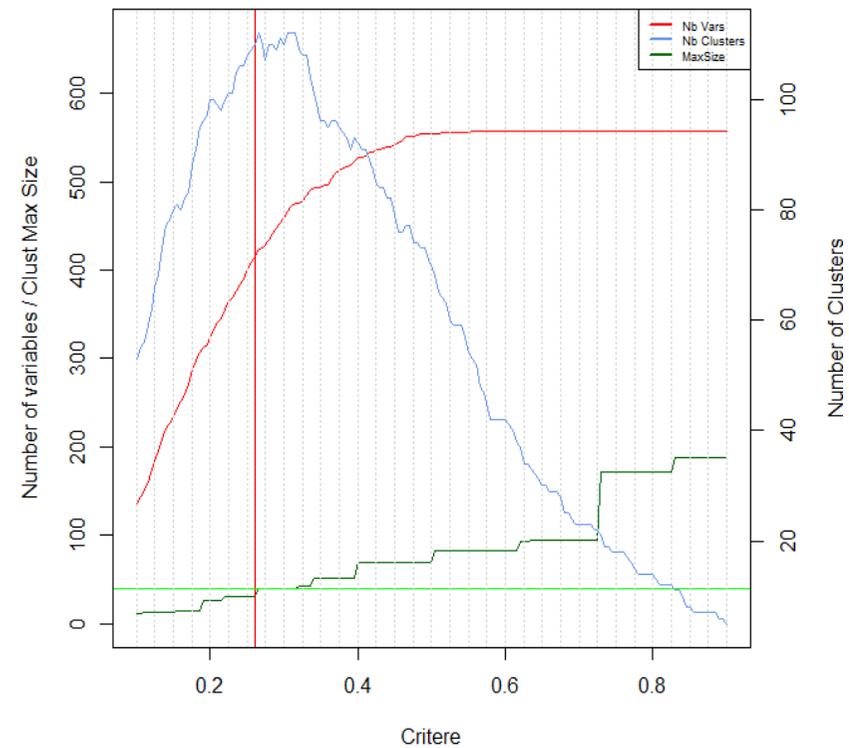
```
> plotClusters(outMat,outclust)
```

```
> plotCriterion(outclust)
```

**Boxplot by clusters (log10 transformed)**



**Critere = 0.260, Nb Clust = 110, Nb Vars = 415, Clust Max Size = 31**



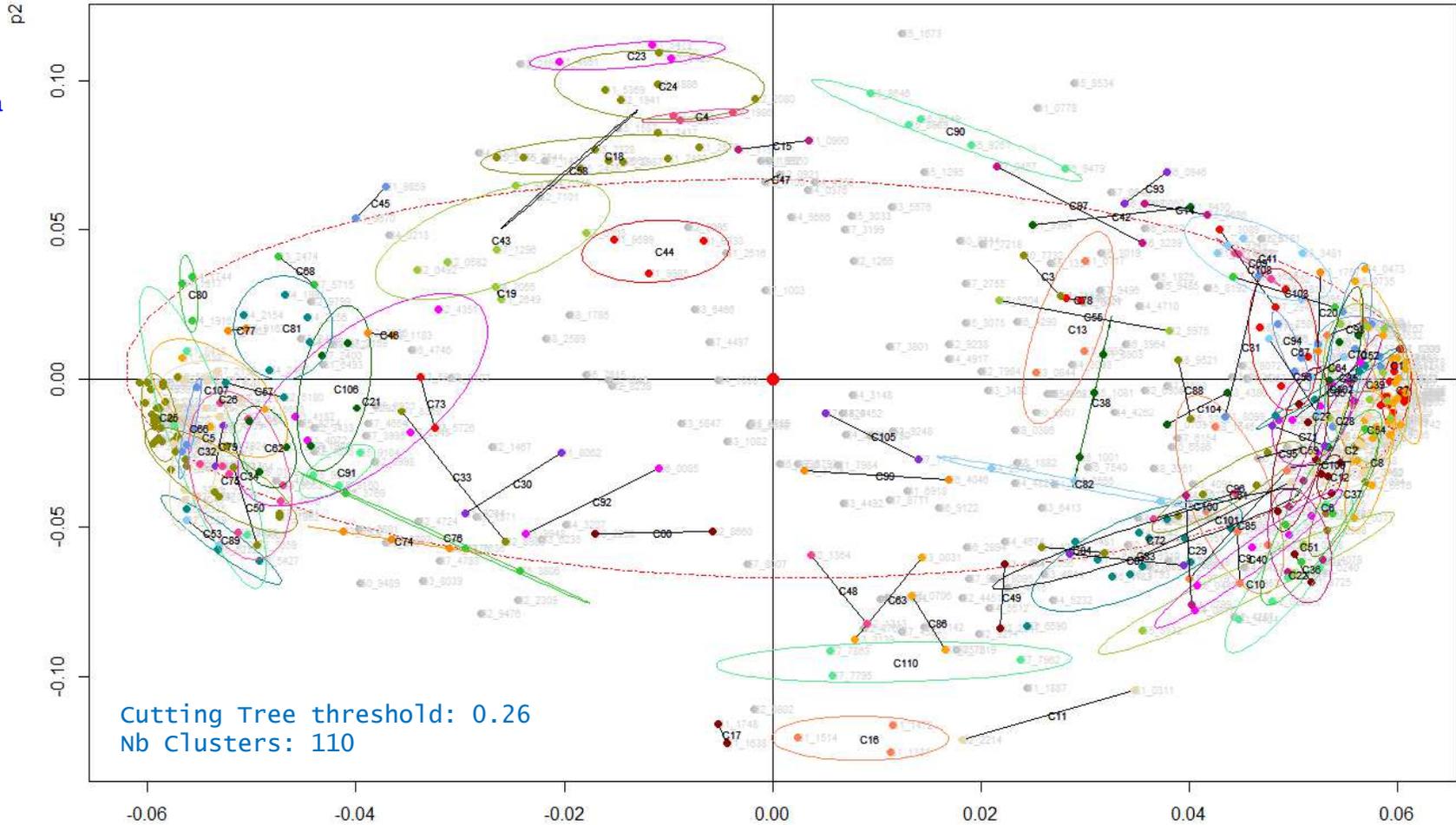
Bucket Clustering based on a cut threshold applied on the buckets tree resulting of a hierarchical classification analysis. (cut threshold automatically estimated)

```
> plotLoadings(outMat.plsda@loadingMN,1,2, outclust$clustertab, cexlabel=0.6)
```

Loadings

Ropls

outMat.plsda



p1

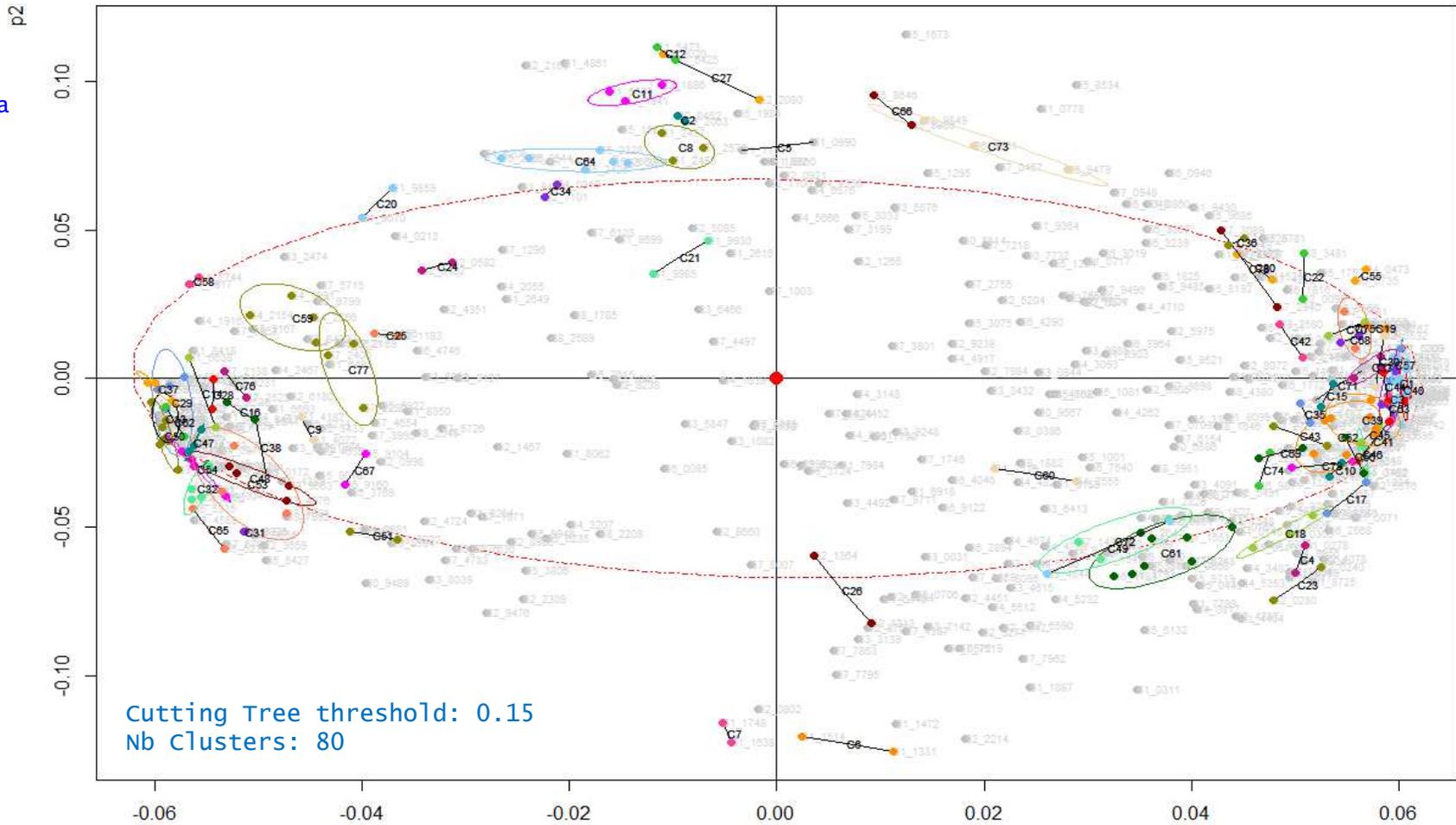
Bucket Clustering based on a cut threshold applied on the buckets tree resulting of a hierarchical classification analysis. (cut threshold automatically estimated)

```
> plotLoadings(outMat.plsda@loadingMN,1,2, outclust$clustertab, cexlabel=0.6)
```

Loadings

Ropls

outMat.plsda



## Bucket Clustering based on a lower threshold applied on correlations (cval=0 threshold automatically estimated)

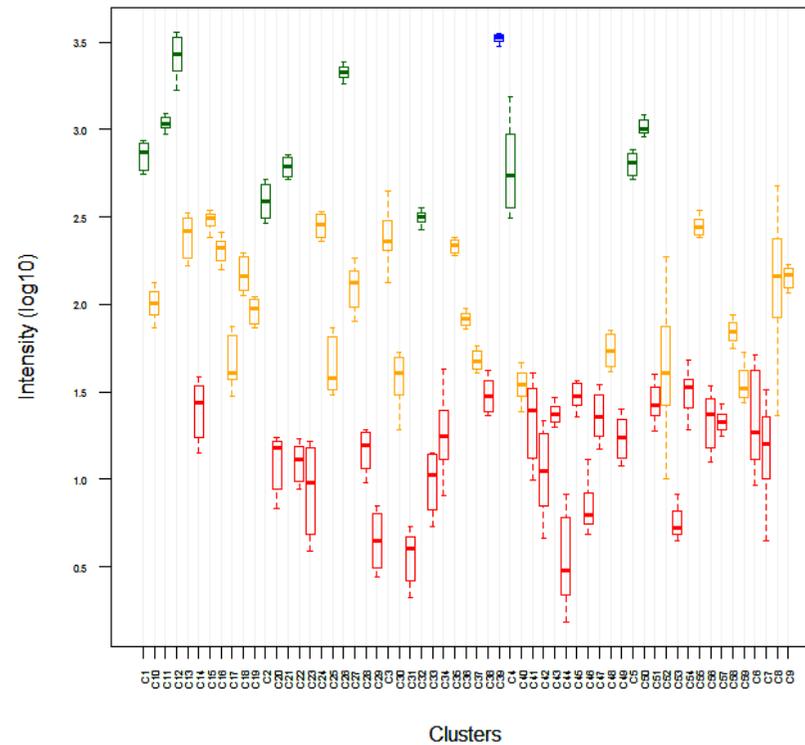
```
> outclust <- getClusters(outMat, method='corr', cval=0, dc=0.01, ncpu=4 )
```

```
##-- Clustering --
# Correlation Method: pearson
# Correlation Threshold : 0.982
# Correlation Tolerance: 0.01
# Nb Clusters: 59
#
```

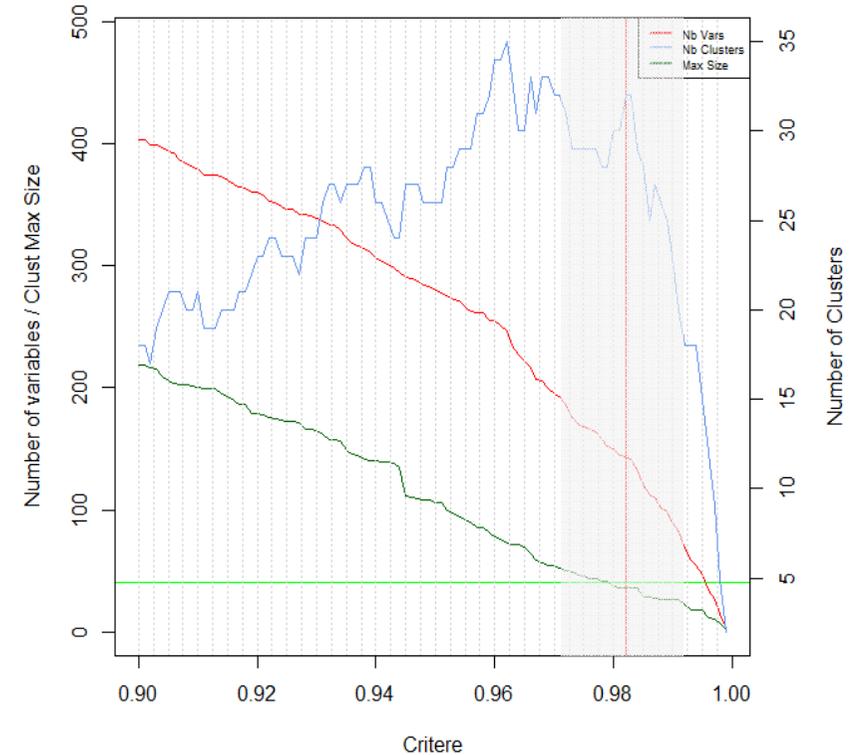
```
> plotClusters(outMat,outclust)
```

```
> plotCriterion(outclust)
```

**Boxplot by clusters (log10 transformed)**



**Critere = 0.982, Nb Clust = 32, Nb Vars = 144, Clust Max Size = 36**



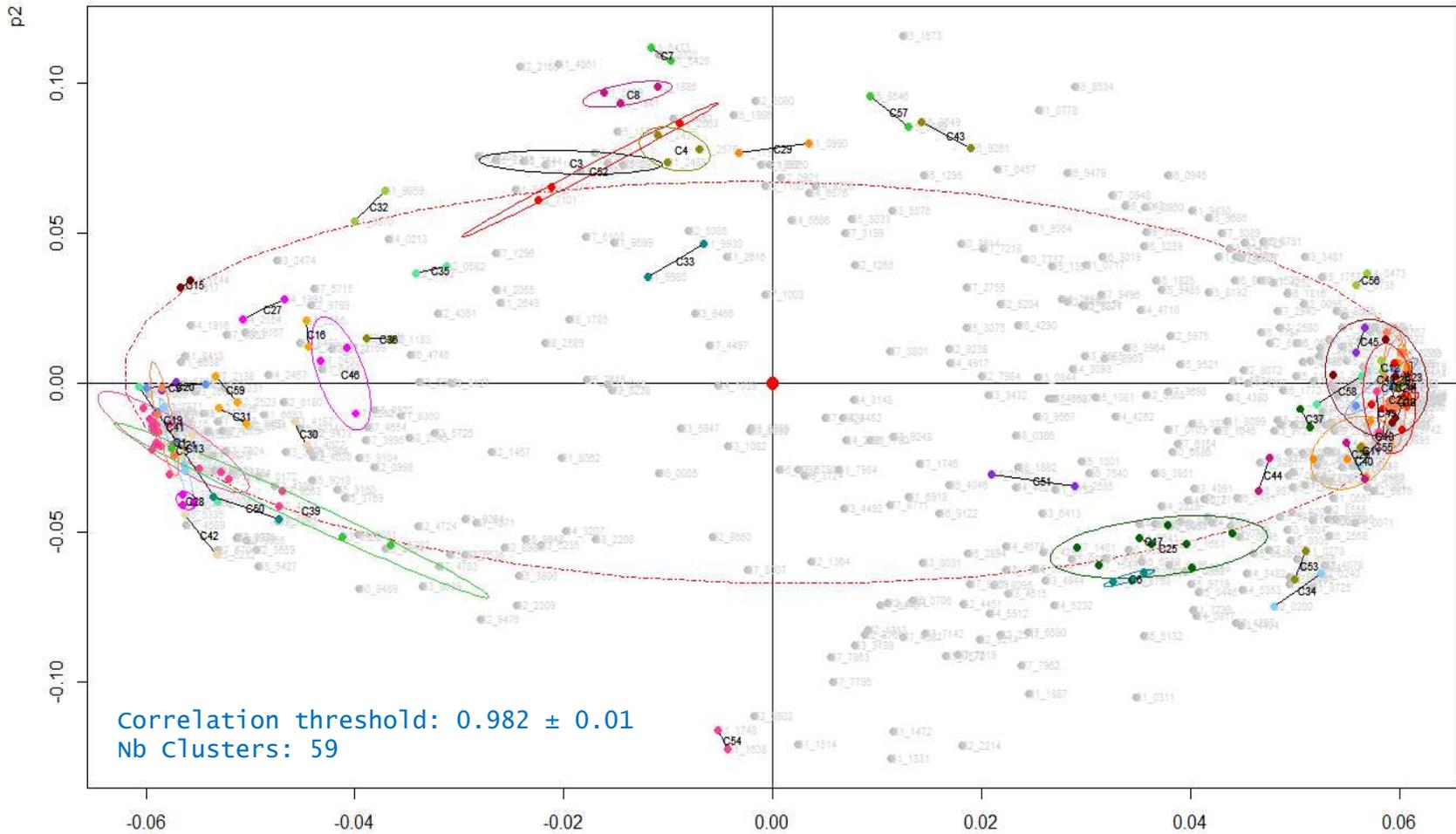
# Bucket Clustering based on a lower threshold applied on correlations

```
> plotLoadings(outMat.plsda@loadingMN,1,2, outclust$clustertab, cexlabel=0.6 )
```

Loadings

Ropls

outMat.plsda



p1

# Bucket Clustering based on a lower threshold applied on correlations

```
> plotLoadings(outMat.plsda@loadingMN,1,2, outclust$clustertab, cexlabel=0.6 )
```

Ropls

outMat.plsda

Loadings

