msmsEDA
LC-MS/MS Exploratory Data Analysis

Josep Gregori, Alex Sanchez, and Josep Villanueva
Vall Hebron Institute of Oncology &
Statistics Dept. Barcelona University
josep.gregori@gmail.com

October 26, 2021

1 Introduction

In biomarker discovery, label-free differential proteomics is based on comparing the expression of proteins between different biological conditions [1] [2]. The experimental design involved for these experiments requires of randomization and blocking [12]. Working with balanced blocks in which all biological conditions are measured by a number of technical and/or biological replicates, and within the shortest time window possible, helps to reduce experimental bias. Unfortunately, there are many factors that may bias the results in a systematic manner: different operators, different chromatographic columns, different protein digestions, an eventual repair of the LC-MS system, and different laboratory environment conditions. Blocking and randomization help to control to some extent these factors. However, even using the best experimental design some uncontrolled variables may still interfere with differential proteomics experiments. These uncontrolled variables may be responsible for the manifestation of batch effects. Effects which are usually evidenced when samples do not cluster by their biological condition using multidimensional unsupervised techniques such as Principal Components Analysis (PCA) or Hierarchical Clustering (HC) [3]. The most benign consequence of batch effects is an increase in the observed variability with a decreased sensitivity to detect biological differences. In the worst scenario, batch effects can mask completely the underlying biology in the experiment.

Exploratory Data Analysis (EDA) helps in evidencing confounding factors and eventual outliers. In front of any '-omics' experiment it is always wise to perfom an EDA before any differential expression statistical analysis [4] [5]. The results of this exploratory analysis, visualized by PCA maps, HC trees, and heatmaps, will inform about the extent of batch effects and the presence of putative outliers. As a result the analyst may decide on the inclusion of a blocking factor to take into account the batch effects, or about the exclusion of a bad conditioned sample.

2 An example LC-MS/MS dataset

The dataset of this example [5] is the result of an spiking experiment, showing LC-MS/MS data obtained in ideal conditions, and optimal to detect batch effects. Samples of 500 micrograms of a standard yeast lysate are spiked either with 200fm or 600fm of a complex mix of 48 human proteins (UPS1, Sigma-Aldrich®). The measures were done in two different runs, separated by a year time. The first run consisted of four replicates of each condition, and the second run consisted of three replicates of each condition.
The dataset consists in an instance of the *MSnSet* class, defined in the MSnbase package [6], a S4 class [7] [8]. This *MSnSet* object contains a spectral counts (SpC) matrix in the *assayData* slot, and factors treatment and batch in the *phenoData* slot. (See also the expressionSet vignette [9])

```
> library(msmsEDA)
> data(msms.dataset)
> msms.dataset

MSnSet (storageMode: lockedEnvironment)
assayData: 697 features, 14 samples
  element names: exprs
protocolData: none
phenoData
  sampleNames: U2.2502.1 U2.2502.2 ... U6.0302.3 (14 total)
  varLabels: treat batch
  varMetadata: labelDescription
featureData: none
experimentData: use 'experimentData(object)'
  pubMedIds: http://www.ncbi.nlm.nih.gov/pubmed/22588121
Annotation:
- - - Processing information - - -
 MSnbase version: 1.8.0

> dim(msms.dataset)

[1] 697  14

> head(pData(msms.dataset))

          treat batch
U2.2502.1  U200  2502
U2.2502.2  U200  2502
U2.2502.3  U200  2502
U2.2502.4  U200  2502
U6.2502.1  U600  2502
U6.2502.2  U600  2502

> table(pData(msms.dataset)$treat)
```

```
U200 U600
   7    7
```

```
>   table(pData(msms.dataset)$batch)
```

```
0302 2502
   6    8
```

```
>   table(pData(msms.dataset)$treat, pData(msms.dataset)$batch)
```

```
       0302 2502
  U200    3    4
  U600    3    4
```

The aim of the exploratory data analysis, in this case, is to evidence the existence of batch effects between the two runs. And eventually to check the opportunity of a simple batch effects correction.

Before proceeding to the EDA, a data pre-processing is required to solve NAs and to remove improper rows in the spectral counts matrix. The NAs, common when joining datasets in which not exactly the same proteins are identified, should be substituted by 0. By improper rows to be removed, we mean: i) the rows with all zeroes, which could come from the subsetting from a bigger SpC matrix. ii) The rows belonging to artefactual identifications of '-R' proteins.

```
> e <- pp.msms.data(msms.dataset)
> processingData(e)
```

```
- - - Processing information - - -
Subset [697,14][675,14] Tue Oct 26 21:31:43 2021
Applied pp.msms.data preprocessing [Tue Oct 26 21:31:43 2021]
 MSnbase version: 1.8.0
```

```
> dim(e)
```

```
[1] 675  14
```

```
> setdiff(featureNames(msms.dataset), featureNames(e))
```

```
 [1] "YER160C-R"    "YJR066W-R"    "YEL061C-R"    "YJL190C (+1)" "YLL057C"
 [6] "YBR189W"      "YDR227W"      "YER103W"      "YGR029W"      "YNR032C-A"
[11] "YMR172W"      "YJL200C"      "YDL126C"      "YGL173C"      "YML037C"
[16] "YHR102W"      "YMR165C"      "YJL138C (+1)" "YBL046W"      "YJL123C"
[21] "YOR123C"      "YGR276C"
```

3 SpC distribution

A first glance to the contents of the spectral counts matrix is given by the distribution of SpC by sample, including the number of proteins identified and the total spectral counts by sample.

```
> tfvnm <- count.stats(e)
> { cat("\nSample statistics after removing NAs and -R:\n\n")
   cat("SpC matrix dimension:",dim(e),"\n\n")
   print(tfvnm)
 }

Sample statistics after removing NAs and -R:

SpC matrix dimension: 675 14

          proteins counts min lwh med hgh max
U2.2502.1      590   5398   0   2   3 8.0 183
U2.2502.2      592   5501   0   2   3 7.0 205
U2.2502.3      586   5477   0   1   3 8.0 202
U2.2502.4      586   5251   0   1   3 7.0 203
U6.2502.1      582   5692   0   1   3 8.5 194
U6.2502.2      577   5686   0   1   3 8.0 208
U6.2502.3      578   5552   0   1   3 8.0 215
U6.2502.4      560   5601   0   1   3 8.0 217
U2.0302.1      512   5629   0   1   2 7.0 409
U2.0302.2      499   5840   0   0   2 7.0 384
U2.0302.3      513   5726   0   1   2 7.0 364
U6.0302.1      491   5975   0   0   2 7.5 395
U6.0302.2      474   5739   0   0   2 7.5 358
U6.0302.3      474   5891   0   0   2 8.0 355
```

Three graphical means will contribute to visualize this distribution:

i) A barplot of total SpC by sample scaled to the median. Ideally when the same amount of total protein is measured in each sample, the same total number of spectral counts will be observed. A good quality experiment will show all the bars near 1.

ii) A set of SpC boxplots by sample. As most proteins in a sample may show very low signal, 0 or 1 SpC, resulting in sparce SpC vectors, more informative boxplots are obtained when removing the values below a given threshold. Here the proteins with SpC below `minSpC` are excluded of a sample. The boxplots show the distribution of the remaining log_2 transformed SpC values by sample.

iii) Superposed SpC density plots by sample. As before the proteins with SpC below `minSpC` are excluded of each sample and the SpC are log_2 transformed.

```
> layout(mat=matrix(1:2,ncol=1),widths=1,heights=c(0.35,0.65))
> spc.barplots(exprs(e),fact=pData(e)$treat)
> spc.boxplots(exprs(e),fact=pData(e)$treat,minSpC=2,
              main="UPS1 200fm vs 600fm")
```

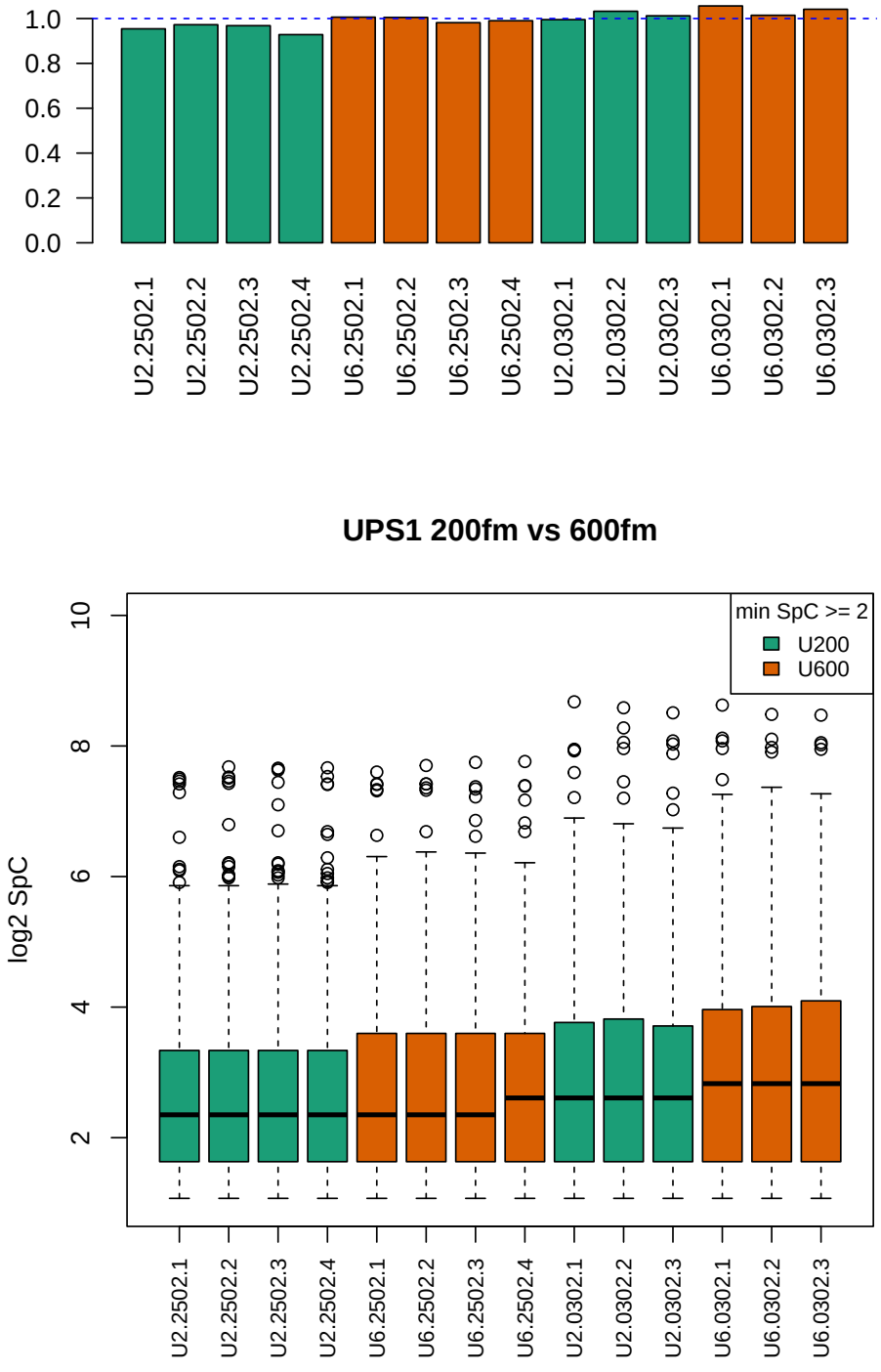

Figure 1: A) Total SpC by sample scaled to the median. B) Samples SpC boxplots.

```
> spc.densityplots(exprs(e),fact=pData(e)$treat,minSpC=2,
                   main="UPS1 200fm vs 600fm")
```

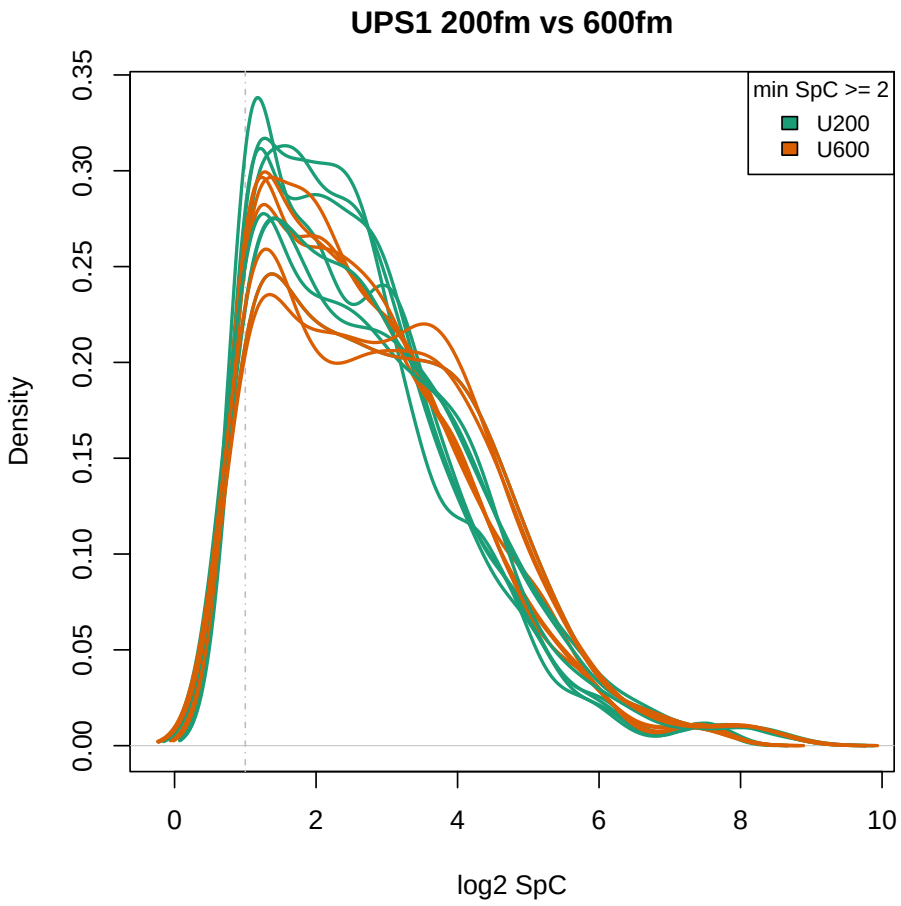

Figure 2: Samples SpC density plots.

4 Principal Components Analysis

A plot on the two principal components of the SpC matrix visualizes the clustering of samples. Ideally the samples belonging to the same condition should cluster together. Any mixing of samples of different conditions may indicate the influence of some confounding factors.

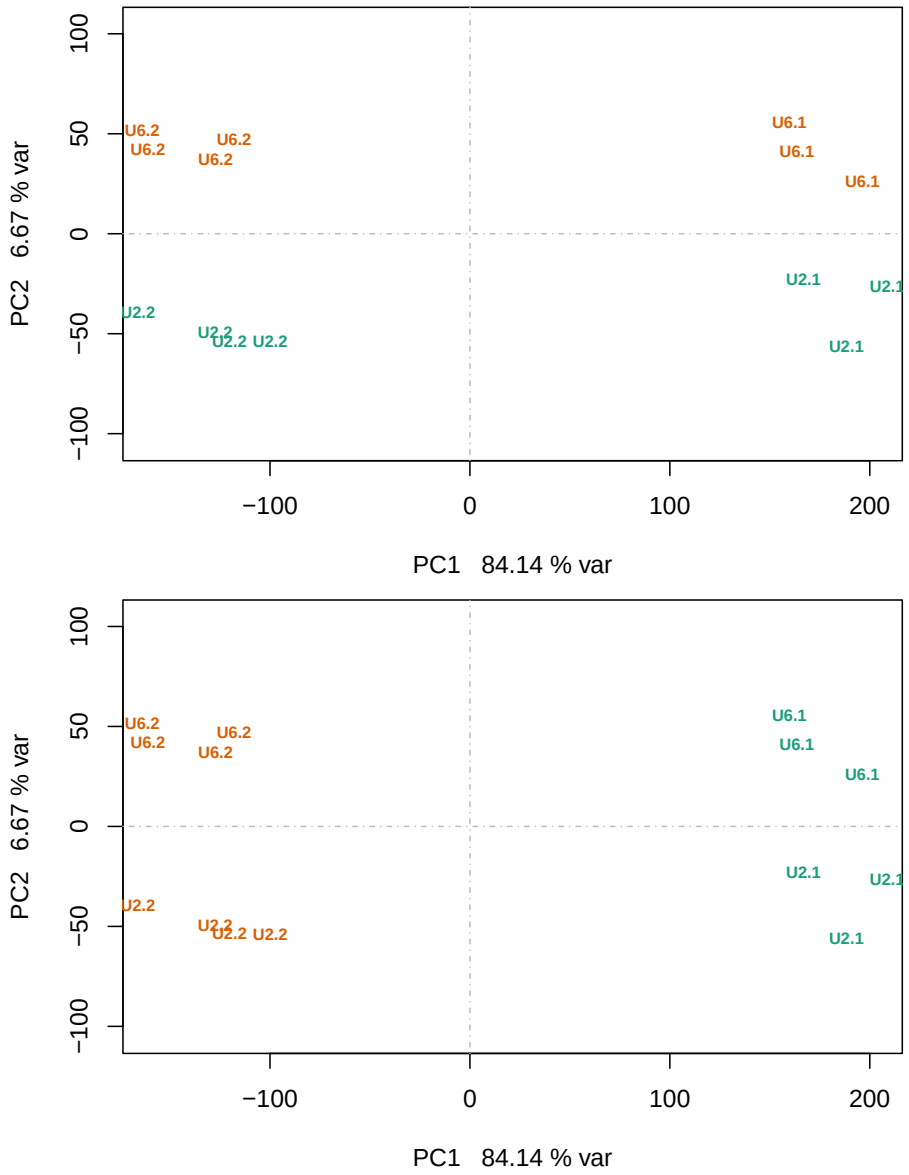

Figure 3: PCA plot on PC1/PC2, showing confounding. The labels are colored by treatment level on top, and by batch number bellow. Labels themselves are selfexplicative of treatment condition and batch.

```
> facs <- pData(e)
> snms <- substr(as.character(facs$treat),1,2)
> snms <- paste(snms,as.integer(facs$batch),sep=".")
> pcares <- counts.pca(e)
```

```
> smpl.pca <- pcares$pca
> { cat("Principal components analisis on the raw SpC matrix\n")
   cat("Variance of the first four principal components:\n\n")
   print(summary(smpl.pca)$importance[,1:4])
 }

Principal components analisis on the raw SpC matrix
Variance of the first four principal components:

                            PC1      PC2      PC3      PC4
Standard deviation     163.82011 46.11827 32.26952 22.75380
Proportion of Variance   0.84136  0.06668  0.03265  0.01623
Cumulative Proportion    0.84136  0.90804  0.94068  0.95691
```

Note how in these plots the samples tend to cluster by batch instead of by treatment, this is evidence of a confounding factor. Something uncontrolled contributes globally to the results in a higer extend than the treatment itself.

5 Hierarchical clustering

The hierarchical clustering of samples offers another view of the same phenomenon:

```
> counts.hc(e,facs=pData(e)[, "treat", drop = FALSE])
```

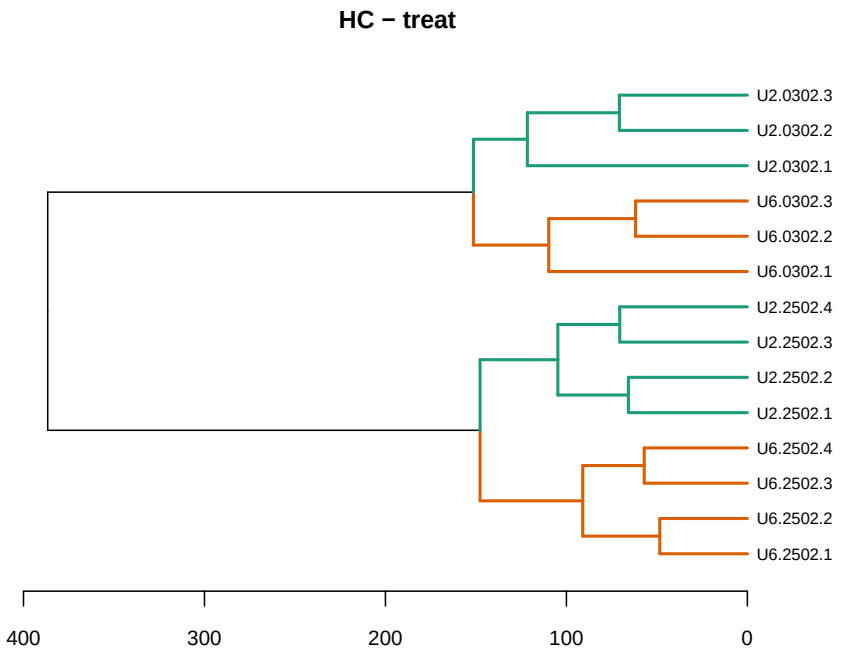

Figure 4: Hirearchical clustering of samples, showing confounding. The labels are colored by treatment level.

```
> counts.hc(e,facs=pData(e)[, "batch", drop = FALSE])
```

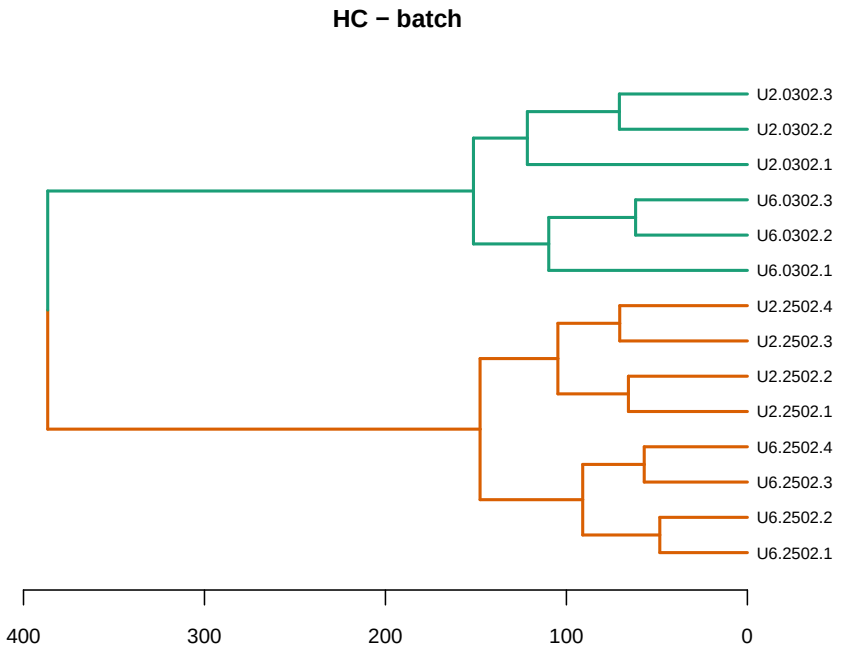

Figure 5: Hirearchical clustering of samples, showing confounding. The labels are colored by batch number.

6 Heatmap

A heatmap may be more informative than the dendrogram of samples, in the sense that it allows to identify the proteins most sensitive to this confounding. In this case we need a heatmap heigh enough to allow room for all the protein names. The function *counts.heatmap* provides two heatmaps, the first one is fitted on an A4 page and offers a general view, the second one is provided in a separate pdf file and is drawn with 3mm high rows to allow a confortable identification of each protein and its expression profile in the experiment.

```
> counts.heatmap(e,etit="UPS1",fac=pData(e)[, "treat"])
```

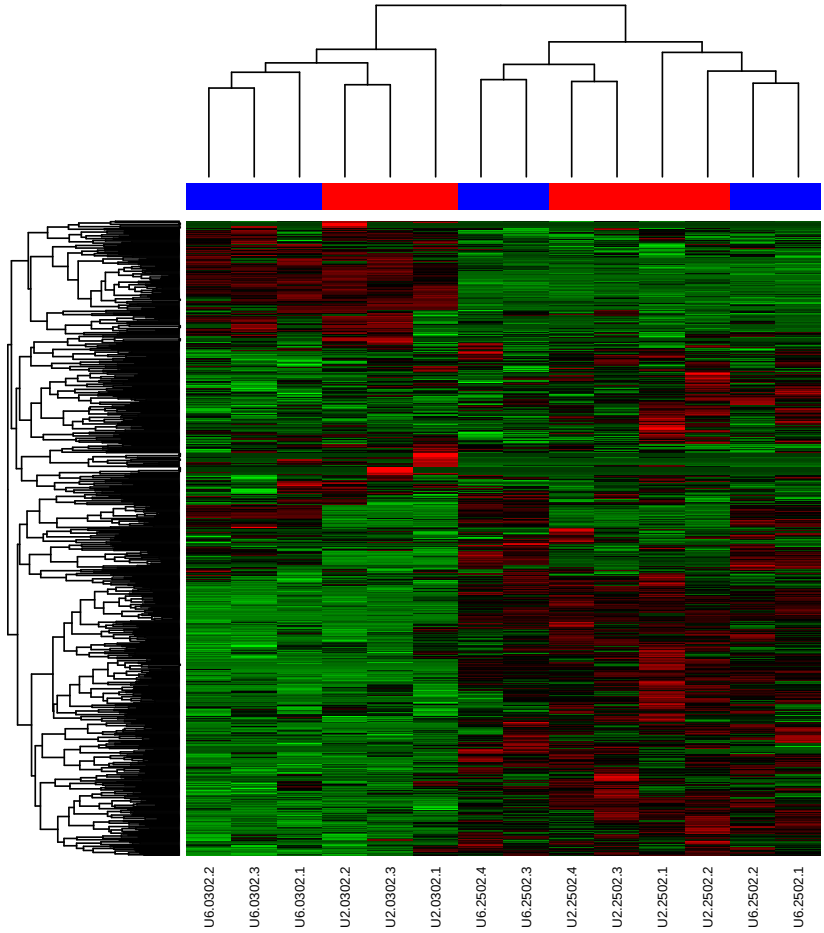

Figure 6: Global view heatmap. The column color bar is colored as per treatment levels.

7 Batch effects correction

When counfounding is detected due to batch effects, as in this case, and the runs are balanced in the two conditions to be compared, blocking may help in the reduction of the residual variance, improving the sensitivity of the statistical tests. When dealing with spectral counts the usual model for the differential expression tests is a Generalized Linear Model (GLM) [10] based on the Poisson distribution, the negative binomial, or the quasi-likelihood, and these models admit blocking as the usual ANOVA [11] when dealing with normally distributed continous data.

The visualization of the influence of a batch effects correction, in the exploratory data analysis step, is easily carried out by the so called *mean centering* approach [4], by which the centers of each batch are made to coincide concealing the observed bias between the different batches. The PCA on the batch mean centered expression matrix will then show the level of improvement.

```
> data(msms.dataset)
```

```
> msnset <- pp.msms.data(msms.dataset)
> spcm <- exprs(msnset)
> fbatch <- pData(msnset)$batch
> spcm2 <- batch.neutralize(spcm, fbatch, half=TRUE, sqrt.trans=TRUE)

> ###  Plot the PCA on the two first PC, and colour by treatment level
> ###  to visualize the improvement.
> exprs(msnset) <- spcm2
> facs <- pData(e)
> snms <- substr(as.character(facs$treat),1,2)
> snms <- paste(snms,as.integer(facs$batch),sep=".")
> par(mar=c(4,4,0.5,2)+0.1)
> counts.pca(msnset, facs=facs$treat, do.plot=TRUE, snms=snms)
```

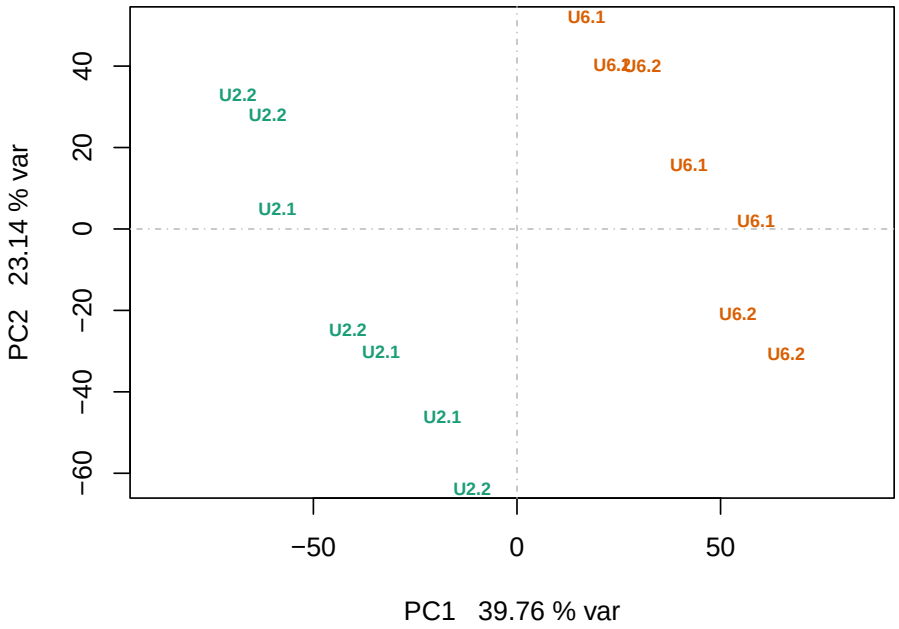

Figure 7: PCA plot on the batch mean centered expression matrix, showing now a better clustering by treatment level.

This plots shows a clear improvement in the clustering of samples by treatment condition, and suggest that a model with batch as block factor will give better results than a model just including the treatment factor. The incidence of the correction may be evaluated as:

```
> ###  Incidence of the correction
> summary(as.vector(spcm-spcm2))

      Min.    1st Qu.     Median       Mean    3rd Qu.       Max.
-115.55183   -0.47569   -0.00694    0.21003    0.88057  137.91051

> plot(density(as.vector(spcm-spcm2)))
```

8 Dispersion

The simplest distribution used to explain the observed counts in sampling is the Poisson distribution. With this distribution the variance is equal to the mean, so that the dispersion coefficient -the ratio variance to mean- is one. When there are other sources of variation, apart of the sampling, such as the usual variability among biological replicates, we talk of overdispersion. In this situation the coefficient of dispersion is greater than one and the Poisson distribution is unable to explain this extra source of variance. Alternative GLM models able to explain overdispersion are based on the negative binomial distribution, or on the quasilikelihood [10].

An EDA of a dataset based on counts should include an exploration of the residual coefficients of dispersion for each of the factors in the experimental design. This will help in deciding the model to use in the inference step.

The *disp.estimates* function plots the distribution of residual dispersion coefficients, and the scatterplot of residual variances vs mean SpC for each of the factors in the parameter *facs*, if this parameter is NULL then the factors are taken as default from the *phenoData* slot of the MSnSet object.

```
> dsp <- disp.estimates(e)
> signif(dsp,4)
```

```
        0.25    0.5  0.75   0.9  0.95  0.99      1
treat 0.4471 0.7500 1.111 1.677 2.474 12.97 67.210
batch 0.2619 0.4264 0.689 1.279 2.071  5.64  8.273
```

This function returns silently the quartiles and the quantiles at 0.9, 0.95 and 0.99 of the residual dispersion of each factor. With technical replicates it is not uncommon to observe most of the dispersion coefficients lower that one. This is a situation of underdispersion, most likely due to the competitive sampling at the MS system entrance.

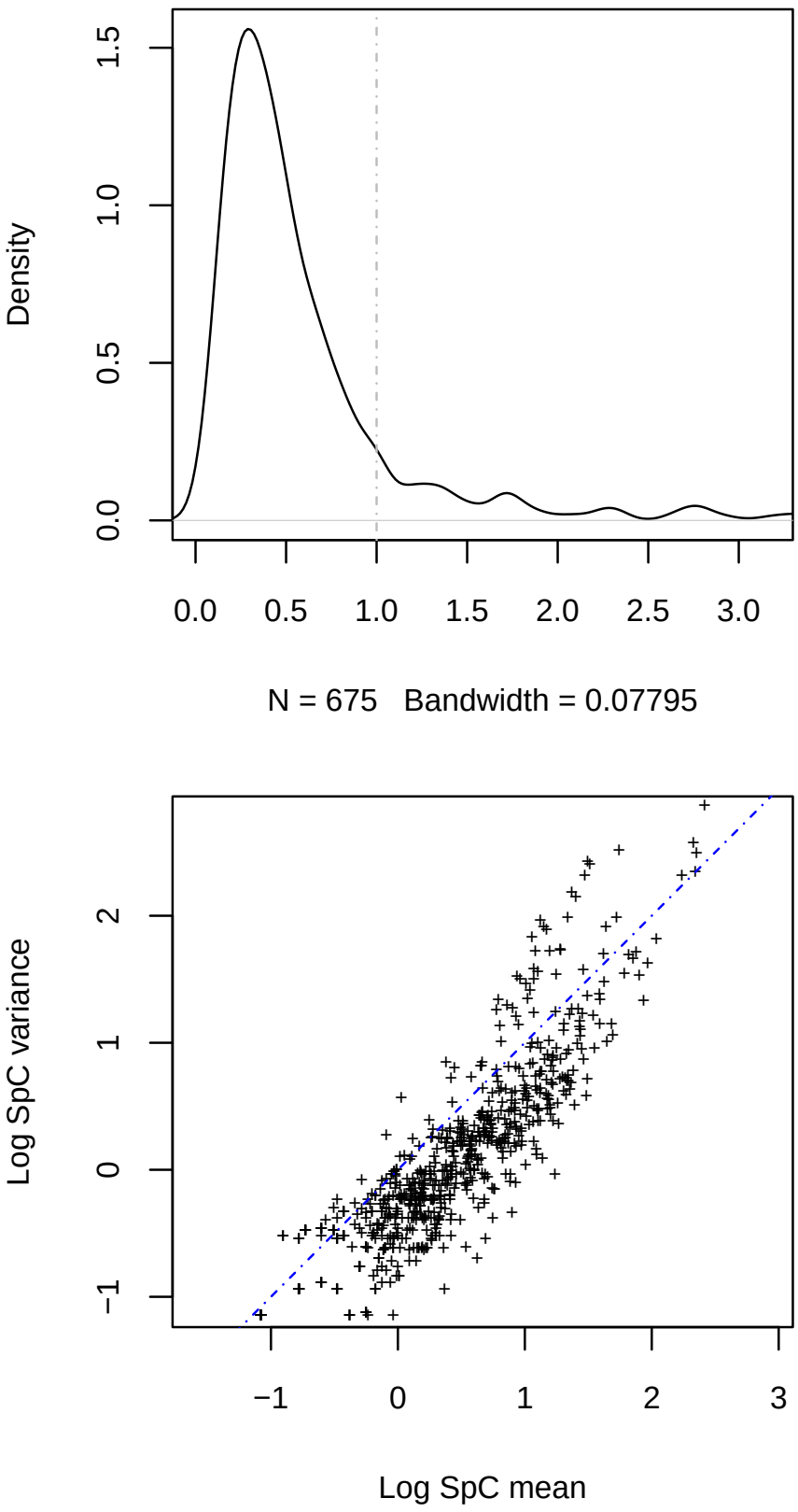

Figure 8: Residual dispersion density plot, and residual variance vs mean scatterplot in log10 scale, of the batch factor.

9 Informative features

In the border between EDA and inference we may explore the number and distribution of informative features. We mean by informative features those proteins showing a high enough signal in the most abundant condition and with an absolute log fold change between treatment levels above a given threshold. The following plot shows in red the features with signal not bellow 2 SpC in the most abundant condition, and with `minLFC` of 1. To improve the plot two transformations are available, either '`log2`' or '`sqrt`', although '`none`' is also accepted.

```
> spc.scatterplot(spcm2,facs$treat,trans="sqrt",minSpC=2,minLFC=1,
                  main="UPS1 200fm vs 600fm")
```

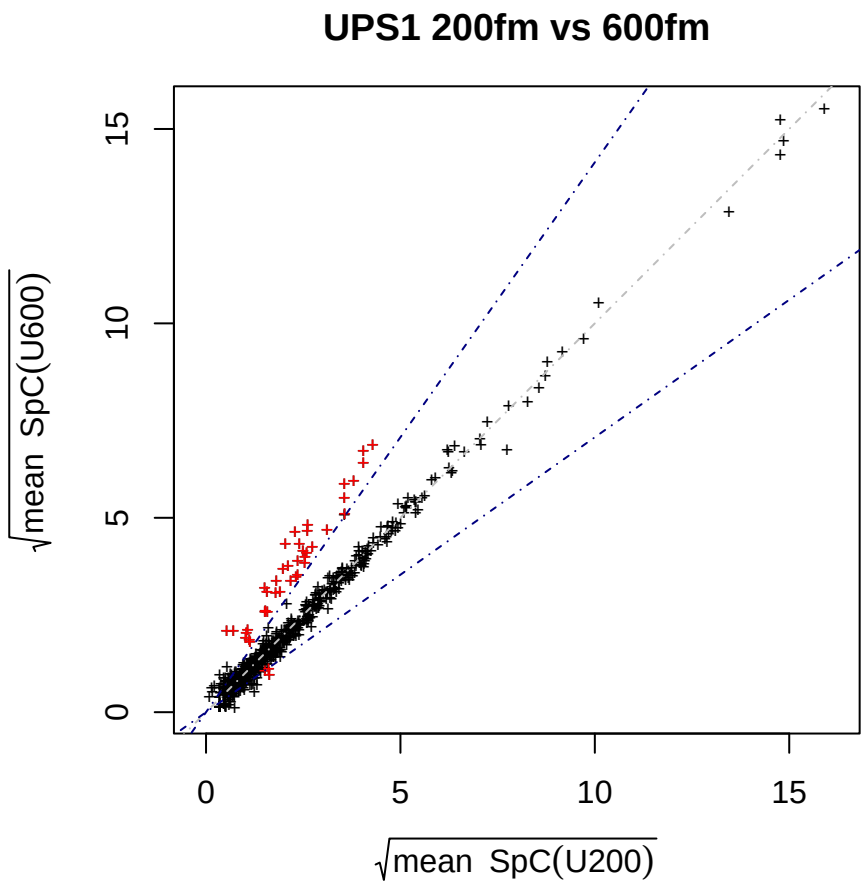

Figure 9: Scatterplot with informative features in red, showing the borders of fold change 2 and 0.5 as blue dash-dot lines.

References

[1] Mallick P., Kuster B. *Proteomics: a pragmatic perspective.* Nat Biotechnol 2010;28:695-709.

[2] Neilson K.A., Ali N.A., Muralidharan S., Mirzaei M., Mariani M., Assadourian G., et al. *Less label, more free: approaches in label-free quantitative mass spectrometry.* Proteomics 2011;11:535-53.

[3] Husson F., Le S., Pages J. *Exploratory Multivariate Analysis by Example Using R.* CRC Press; 2010.

[4] Luo J., Schumacher M., Scherer A., Sanoudou D., Megherbi D., Davison T., et al. *A comparison of batch effect removal methods for enhancement of prediction performance using MAQC-II microarray gene expression data.* Pharmacogenomics J 2010, 10(4): 278-291

[5] Gregori J., Villareal L., Mendez O., Sanchez A., Baselga J., Villanueva J., *Batch effects correction improves the sensitivity of significance tests in spectral counting-based comparative discovery proteomics*, Journal of Proteomics, 2012, 75, 3938-3951

[6] Laurent Gatto and Kathryn S. Lilley, MSnbase - an R/Bioconductor package for isobaric tagged mass spectrometry data visualization, processing and quantitation, Bioinformatics 28(2), 288-289 (2012).

[7] Chambers J.M. *Software for data analysis: programming with R*, 2008 Springer

[8] Genolini C. *A (Not So) Short Introduction to S4* (2008)

[9] Falcon S., Morgan M., Gentleman R. *An Introduction to Bioconductor's ExpressionSet Class* (2007)

[10] Agresti A., *Categorical Data Analysis*, Wiley-Interscience, Hoboken NJ, 2002

[11] Kutner M.H., Nachtsheim C.J., Neter J., Li W., *Applied Linear Statistical Models*, Fifth Edition, McGraw-Hill Intl. Edition 2005

[12] Quinn G.P., Keough M.J. *Experimental Design and Data Analysis for Biologists* Cambridge University Press; 1st Edition, Cambridge 2002.