msmsTests package
Blocks design to compensate batch effects

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

May 1, 2024

Contents

1 Introduction

This vignette exemplifies the use of the packages msmsEDA and msmsTests in discovering and correcting batch effects in label-free LC-MS/MS data based on spectral counts. Label-free experiments are specially sensitive to these effects as each condition has to be measured separately and may be influenced by uncontrolled factors in a different extend.

2 Dataset

This dataset [1] is the result of spiking experiments, showing real LC-MS/MS data. Samples of 500 micrograms of a standard yeast lysate are spiked with 200 and 600fm of a complex mix of 48 equimolar human proteins (UPS1, Sigma-Aldrich). The dataset comes with the package msmsEDA [2], and was used to evidence batch effects by exploratory

data analysis tools [3]. The dataset consists in an instance of the *MSnSet* class, defined in the MSnbase package [4], a S4 class [5] [6]. 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 by vignette("ExpressionSetIntroduction",package="Biobase") [7])

```
> library(msmsTests)
> 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
```

```
> msms.counts <- exprs(msms.dataset)
> dim(msms.counts)
```

```
[1] 697  14
```

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

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

Although the mix is equimolar the signal strength of each protein is markedly different, allowing to cover a wide range of SpC values, what makes it specially worth in this sort of experiments:

```
> idx <- grep("HUMAN",featureNames(msms.dataset))
> mSpC <- t( apply(msms.counts[idx,],1,function(x)
                    tapply(x,pData(msms.dataset)$treat,mean)) )
> apply(mSpC,2,summary)
```

```
              U200      U600
Min.     0.2857143  1.142857
1st Qu.  2.4642857  6.071429
Median   4.5714286 12.214286
Mean     5.9378882 15.161491
3rd Qu.  6.9642857 20.500000
Max.    18.8571429 47.000000
```

3 Batch effects

Real life LC-MS/MS experiments use to be complicated enough to be able to get all required technical or biological replicates in a single batch run. Commonly a dataset collects results from multiple batches. The batches may be influenced by factors which escape our control capacity, and typically these datasets show the so known '*batch effects*' when the runs where obtained in different dates. The confounding caused by these effects is easily evidenced by multidimensional tools like Principal Components Analysis (PCA) or Hierarchical Clustering (HC), when the samples cluster by batches instead of by treatment [3] [8] .

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

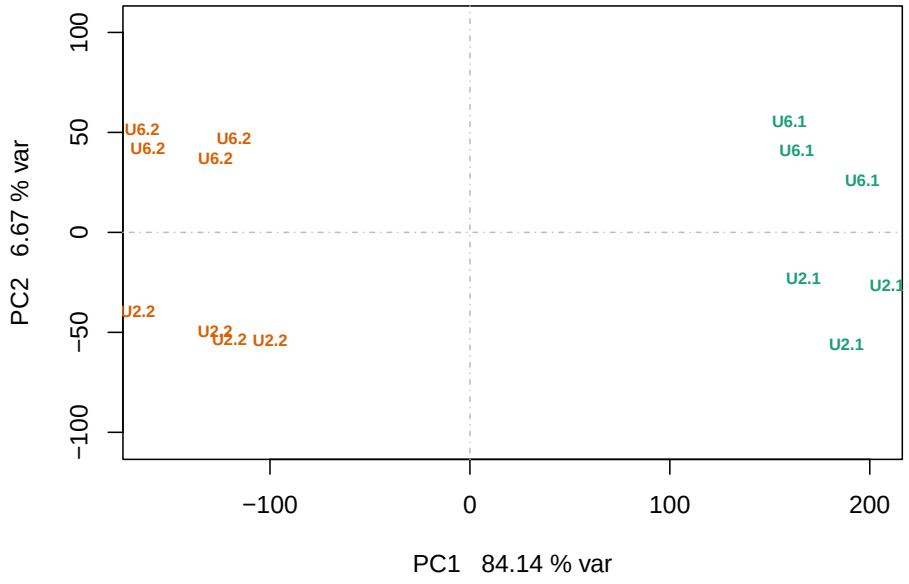

Figure 1: Principal Components Analysis

4 Results on a single batch

The next code shows the results obtained from the data of a single batch with four replicates in each condition. The statistic test used for differential expression is the quasi-likelihood GLM [9] [10], and the p-values are adjusted with FDR control by the Benjamini-Hochberg [11] method. The quality of the results is given by a truth table.

```
> ###  Subset and pre-process dataset
> fl <- pData(msms.dataset)$batch=="2502"
> e <- msms.dataset[,fl]
> e <- pp.msms.data(e)
> ### Null and alternative model
```

```
> null.f <- "y~1"
> alt.f <- "y~treat"
> ### Normalizing condition
> counts <- exprs(e)
> div <- apply(counts,2,sum)
> ### Quasi-likelihood GLM
> ql.res <- msms.glm.qlll(e,alt.f,null.f,div=div)
> ### Adjust p-values with FDR control.
> adjp <- p.adjust(ql.res$p.value,method="BH")
> ###  Truth table
> nh <- length(grep("HUMAN",featureNames(e)))
> ny <- length(grep("HUMAN",featureNames(e),invert=TRUE))
> tp <- length(grep("HUMAN",rownames(ql.res)[adjp<=0.05]))
> fp <- sum(adjp<=0.05)-tp
> (tt.ql1 <- data.frame(TP=tp,FP=fp,TN=ny-fp,FN=nh-tp))

  TP FP  TN FN
1 24  4 571 22
```

These results may be polished by a post-test filter, so that only relevant features are accepted as differentially expressed, and the false positives are further restricted [1].

```
> ###  Post-test filter
> ql.tbl <- test.results(ql.res,e,pData(e)$treat,"U600","U200",div,
                     alpha=0.05,minSpC=2,minLFC=1,method="BH")$tres
> ql.nms <- rownames(ql.tbl)[ql.tbl$DEP]
> ###  Truth table
> ridx <- grep("HUMAN",ql.nms)
> tp <- length(ridx)
> fp <- length(ql.nms)-length(ridx)
> (tt.ql11 <- data.frame(TP=tp,FP=fp,TN=ny-fp,FN=nh-tp))

  TP FP  TN FN
1 17  0 575 29
```

5 Results on the global dataset

With a higher number of replicates the tests become more sensitive, and a higher number of differentially expressed features may be identified. The next code explores the full dataset, composed of two batches and seven replicates of each condition. Again the quality of the results is given by a truth table.

```
> ###  Pre-process dataset
> gble <- pp.msms.data(msms.dataset)
> ### Null and alternative model
> null.f <- "y~1"
```

```
> alt.f <- "y~treat"
> ### Normalizing condition
> div <- apply(exprs(gble),2,sum)
> ### Quasi-likelihood GLM
> ql.res <- msms.glm.qlll(gble,alt.f,null.f,div=div)
> ### Adjust p-values with FDR control.
> adjp <- p.adjust(ql.res$p.value,method="BH")
> ###  Truth table
> nh <- length(grep("HUMAN",featureNames(gble)))
> ny <- length(grep("HUMAN",featureNames(gble),invert=TRUE))
> tp <- length(grep("HUMAN",rownames(ql.res)[adjp<=0.05]))
> fp <- sum(adjp<=0.05)-tp
> (tt.ql2 <- data.frame(TP=tp,FP=fp,TN=ny-fp,FN=nh-tp))

  TP FP  TN FN
1 32  1 628 14
```

Applying a post-test filter, as before, the results become:

```
> ###  Post-test filter
> ql.tbl <- test.results(ql.res,gble,pData(gble)$treat,"U600","U200",div,
                    alpha=0.05,minSpC=2,minLFC=1,method="BH")$tres
> ql.nms <- rownames(ql.tbl)[ql.tbl$DEP]
> ###  Truth table
> ridx <- grep("HUMAN",ql.nms)
> tp <- length(ridx)
> fp <- length(ql.nms)-length(ridx)
> (tt.ql22 <- data.frame(TP=tp,FP=fp,TN=ny-fp,FN=nh-tp))

  TP FP  TN FN
1 29  0 629 17
```

6 Results on the global dataset with a blocking factor

When the batches are balanced in the treatment conditions the presence of confounding factors translates into bigger variance and lower sensitivity. We may account for this extra variability by introducing the batches into the model, as a blocking factor. The next code explores the corresponding results.

```
> ### Null and alternative model
> null.f <- "y~batch"
> alt.f <- "y~treat+batch"
> ### Quasi-likelihood GLM
> ql.res <- msms.glm.qlll(gble,alt.f,null.f,div=div)
> ### Adjust p-values with FDR control.
> adjp <- p.adjust(ql.res$p.value,method="BH")
```

```
> ###  Truth table
> nh <- length(grep("HUMAN",featureNames(gble)))
> ny <- length(grep("HUMAN",featureNames(gble),invert=TRUE))
> tp <- length(grep("HUMAN",rownames(ql.res)[adjp<=0.05]))
> fp <- sum(adjp<=0.05)-tp
> (tt.ql3 <- data.frame(TP=tp,FP=fp,TN=ny-fp,FN=nh-tp))

  TP FP  TN FN
1 41 18 611  5
```

The correction improved the number of true positives, but significantly increased the number of false positives. This may be polished by the post-test filter to remove the non relevant features:

```
> ###  Post-test filter
> ql.tbl <- test.results(ql.res,gble,pData(gble)$treat,"U600","U200",div,
                      alpha=0.05,minSpC=2,minLFC=1,method="BH")$tres
> ql.nms <- rownames(ql.tbl)[ql.tbl$DEP]
> ###  Truth table
> ridx <- grep("HUMAN",ql.nms)
> tp <- length(ridx)
> fp <- length(ql.nms)-length(ridx)
> (tt.ql33 <- data.frame(TP=tp,FP=fp,TN=ny-fp,FN=nh-tp))

  TP FP  TN FN
1 35  0 629 11
```

7 Comparison of results

The following table collects the results obtained so far, where we see how increasing the number of replicates we improve the sensitivity, how the use of a post-test filter helps in restricting the number of false positives, and how blocking helps to remove the extra variability introduced by batch effects.

	TP	FP	TN	FN
subset	24	4	571	22
subset filtered	17	0	575	29
global	32	1	628	14
global filtered	29	0	629	17
blocking	41	18	611	5
blocking filtered	35	0	629	11

Table 1: Truth tables

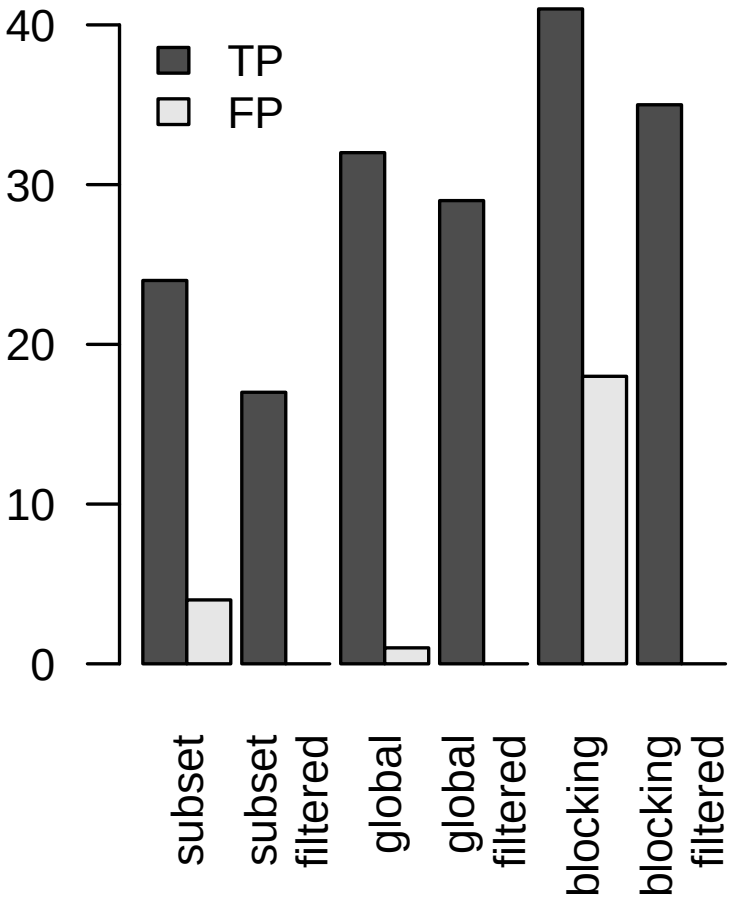

Figure 2: Comparison of results

References

[1] Gregori J., Villareal L., Sanchez A., Baselga J., Villanueva J., *An Effect Size Filter Improves the Reproducibility in Spectral Counting-based Comparative Proteomics.* Journal of Proteomics 2013, http://dx.doi.org/10.1016/j.jprot.2013.05.030

[2] Gregori J., Sanchez A. and Villanueva J. (2013). *msmsEDA: Exploratory Data Analysis of LC-MS/MS data by spectral counts.* R package version 1.0.

[3] 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

[4] 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).

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

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

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

[8] Luo, J. et al. *A comparison of batch effect removal methods for enhancement of prediction performance using MAQC-II microarray gene expression data.* The Pharmacogenomics Journal 2010, 10, 278-291.

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

[10] Li, M.; Gray, W.; Zhang, H.; Chung, C. H.; Billheimer, D.; Yarbrough, W. G.; Liebler, D. C.; Shyr, Y.; Slebos, R. J. C. *Comparative shotgun proteomics using spectral count data and quasi-likelihood modeling.* J Proteome Res 2010, 9, 4295-4305

[11] Benjamini, Y., and Hochberg, Y. (1995). *Controlling the false discovery rate: a practical and powerful approach to multiple testing.* Journal of the Royal Statistical Society Series B, 57, 289-300.