

Application of limma to peptide intensities

Michiel van Ooijen, Victor L. Jong, Marinus J Eijkemans, Albert J.R. Heck, Arno C. Andeweg, Nadine A. Binai, Henk-Jan van den Ham

Supplementary material as part of:

‘Identification of differentially expressed peptides in high-throughput proteomics data’

This is a tutorial on how to apply limma to publically available data. For this purpose we use a dataset consisting of human colon biopsies that have been process with different protein preparation protocols. The purpose of this experiment was to observe the differences in proteomes obtained from samples processed using 4 different protein preservation protocols. The data have been processed with MaxQuant. For details, please refer to the original publication and dataset PXD002029, available from the PRIDE database (<http://www.ebi.ac.uk/pride/archive/projects/PXD002029>).

Step 0: Prerequisites

The application of limma to a dataset requires a working installation of R (<https://www.r-project.org/>).

Step 1: Reading in the data

We read in the peptide level data that is provided by the authors of the study in the `peptides.txt` file that is in the `biopsies_combined_search.zip` PRIDE project file. We select the columns containing the intensity values per sample, and the columns containing peptide annotation.

```
txt = read.delim('peptides.txt', stringsAsFactors=F)
## annotation columns & intensity columns
annc = c('Sequence', 'Gene.names', 'Protein.names')
intc = grep('Intensity.', colnames(txt), value=T)
```

By parsing the column names, we construct an annotation file for the samples.

```
spl = sapply(intc, function(x) strsplit(x, '_'))
targets = data.frame(intc, do.call('rbind', lapply(spl, '[' , 1:4))[,2:4])
colnames(targets) = c('SampleID', 'Participant', 'Protocol', 'replicate')
head(targets)
```

```
##                               SampleID Participant Protocol replicate
## Intensity.04_TB_N2_1 Intensity.04_TB_N2_1          TB         N2         1
## Intensity.05_TB_N2_2 Intensity.05_TB_N2_2          TB         N2         2
## Intensity.06_TB_N2_3 Intensity.06_TB_N2_3          TB         N2         3
## Intensity.08_TB_FF_1 Intensity.08_TB_FF_1          TB         FF         1
## Intensity.09_TB_FF_2 Intensity.09_TB_FF_2          TB         FF         2
## Intensity.10_TB_FF_3 Intensity.10_TB_FF_3          TB         FF         3
```

Step 2: Preprocessing the peptide data

We now process the intensities such that they do not contain only measured values, and set all other values to zero. The data are log₂-transformed before the application of limma - an increase or decrease of one unit therefore represents a two-fold increase or decrease, respectively.

```
intensities = log2(data.matrix(txt[,intc]))
intensities[intensities == -Inf] = NA
```

In total, there are 25123 peptides in this dataset. Out of the 602952 measurements, 333145 values (just over half the datapoints) are actually missing.

Step 3: Defining limma design & contrasts

After loading the limma library, we use the sample information to construct a design matrix for applying limma to this dataset. Please note that the limma documentation is extensively documented and we refer to the limmaUserGuide() for further details.

```
library(limma)
# limmaUserGuide()
des.donor = sapply(unique(targets$Participant), function(x) 'VA' %in% x)
des.protocol = sapply(unique(targets$Protocol), function(x) targets$Protocol %in% x)
colnames(des.protocol) = unique(targets$Protocol)
design = cbind(VA=des.donor, des.protocol)+0
head(design)
```

```
##      VA N2 FF RL FC
## [1,]  0  1  0  0  0
## [2,]  1  1  0  0  0
## [3,]  0  1  0  0  0
## [4,]  1  0  1  0  0
## [5,]  0  0  1  0  0
## [6,]  1  0  1  0  0
```

We now define which groups we would like to compare. For the purpose of this analysis, all pairwise comparisons between the Protocol groups have been made.

```
contr = makeContrasts(
  FC - N2, FF - N2, RL - N2,
  FC - FF, RL - FC, RL - FF,
  levels = design)
```

Step 4: Fitting peptide models with limma

We now fit a linear model to every peptide using the design we created (for details, refer to the limma user manual). We define contrasts of treatment we want to compare. In this case, that is the different treatments.

```
fit = lmFit(intensities, design)
```

```
## Warning: Partial NA coefficients for 8945 probe(s)
```

```
fit2 = contrasts.fit(fit, contr)
fit2 = eBayes(fit2)
```

fit2 now contains the results of limma for all peptides. Out of the 25123 peptides in this dataset, limma was able to perform a test for 12161; all other peptides have too little data available for performing a test.

Step 5: Generating limma results

The results from a limma analysis can be extracted in several ways. One method is to view the differentially expressed peptides, sorted by significance:

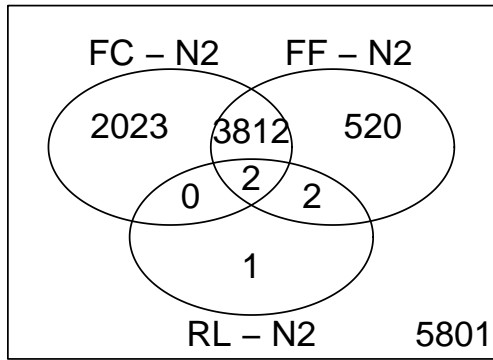
```
topTable(fit2, coef = 'FF - N2', lfc=1, genelist = txt$Gene.names, n=5)
```

```
##          ID      logFC AveExpr      t      P.Value      adj.P.Val
## 13852     HPX  -4.631747 24.78451 -22.11781 1.284410e-15 1.561971e-11
## 10623   HSPD1 -3.774267 24.86940 -17.39352 8.517936e-15 4.041107e-11
## 4609   S100A10 -3.052875 27.57334 -17.22640 1.049049e-14 4.041107e-11
## 22354     ALB -4.735461 30.05581 -17.03817 1.329202e-14 4.041107e-11
## 11253   HADHB -3.301556 24.69495 -16.78481 1.834448e-14 4.461745e-11
##                B
## 13852 25.21900
## 10623 23.71253
## 4609  23.51844
## 22354 23.29747
## 11253 22.99599
```

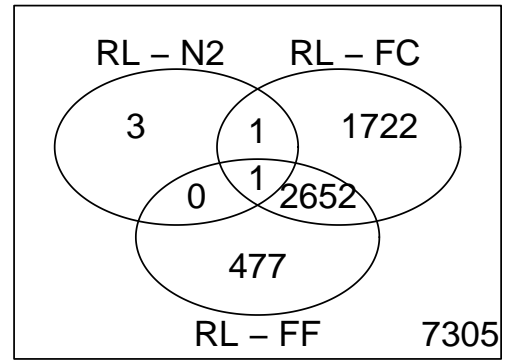
Another is to generate creating Venn diagrams of up- and down-regulated peptides.

```
res = decideTests(fit2, lfc=1, p.value = 0.05 )
par(mfrow=c(2,2))
vennDiagram(res[,1:3], main='All vs N2')
vennDiagram(res[,c(3,5:6)], main='All vs RL')
vennDiagram(res[,c(1,4,5)], main='All vs FC')
vennDiagram(res[,c(2,4,6)], main='All vs FF')
```

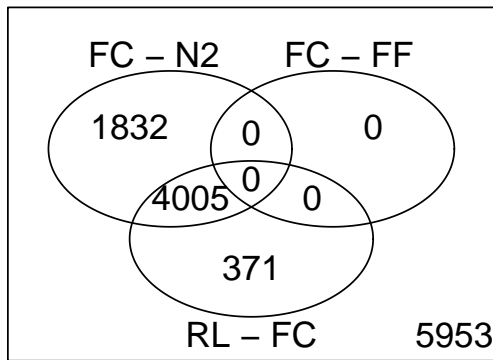
All vs N2



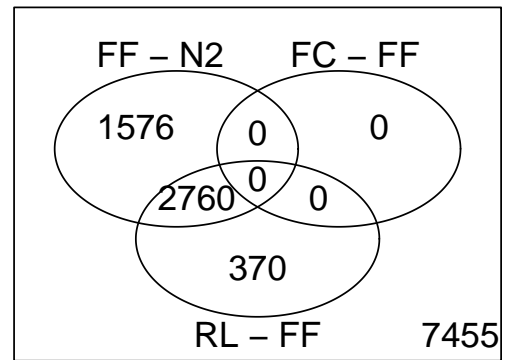
All vs RL



All vs FC



All vs FF



Conclusions

From this analysis, we can conclude that processing protocol can have a large influence on the outcome of proteome profiling. Processing the samples after snap-freezing (N2) or RNAlater (RL), compared to the two FFPE protocols (FF, FC) yields many differentially expressed peptides. However, storage at before FFP does not seem to affect the proteome, since there are no differentially expressed peptides in this comparison. Similarly, comparing snap-freezing and RNAlater indicate that there are few statistically significant differences in between these protocols.