GENOMICS, PROTEOMICS, EVOLUTION

Microarray Data Analysis Project

Sex and aging effect on skeletal muscle transcriptome in humans

Olivier Boes

Université Libre de Bruxelles – January 2015

Introduction

The article on which this project is based, *"Microarray Analysis Reveals Novel Features of the Muscle Aging Process in Men and Women"* [1], investigates the effect of aging on gene expression in male and female muscle (biceps brachii) cells. The authors used *Affymetrix Human Genome U133 Plus 2.0* chips to obtain whole-genome gene expression profiling in 22 different human subjects: 11 women (7 young, 4 old) and 11 men (7 young, 4 old). Differential gene expression analysis was then conducted on the microarray data (available online through *Array Express* [2] or *Gene Expression Omnibus* [3]) using R/Bioconductor. In particular, they used a logistic regression-based method (LRpath [4]) in combination with intensity-based Bayesian moderated t-tests (IBMT [5]). The authors were especially interested in transcriptional differences between Old Women and Young Women, Old Men and Young Men, and between Old Women and Old Men.

In this project, differential gene expression analysis is conducted on the same microarray data, but with tools used during the practicals, such as LIMMA [6]. In the last section of this document, we compare our results with those of the original article. Due to (human and computational) time constraints, we will mostly discuss and compare transcriptional differences between Old Women and Young Women.

An automated script (`session.R`) able to download the data, process it, produce the plots and conduct the analyses described in this document was developed as part of this project. Any further insight into the actual R code used can thus be obtained by looking at `session.R`. Lists of differentially expressed genes and enriched GO terms are also available in files `OWvsYW.txt`, `OMvsYM.txt`, and `OMvsOW.txt` (they will be recreated on execution of the script).

1 Obtaining the data

The raw microarray data, along with sample descriptions, could be downloaded directly from the R console using the `ArrayExpress` or `GEOquery` Bioconductor libraries (this is what `session.R` does). But here, for simplicity, we will assume that the 22 raw CEL files were manually downloaded and placed in our working folder. Reading the sample descriptions (for example the `.sdrf` file if using *Array Express* [2]), each sample (i.e. each CEL file) belongs to one of four categories (see figure 1). We further assume that this information is encoded in a R data frame (`session.R` generates it from the downloaded data), as shown below.

Young Women	Young Men	Old Women	Old Men
GSM948623.CEL	GSM948630.CEL	GSM948637.CEL	GSM948641.CEL
GSM948624.CEL	GSM948631.CEL	GSM948638.CEL	GSM948642.CEL
GSM948625.CEL	GSM948632.CEL	GSM948639.CEL	GSM948643.CEL
GSM948626.CEL	GSM948633.CEL	GSM948640.CEL	GSM948644.CEL
GSM948627.CEL	GSM948634.CEL		
GSM948628.CEL	GSM948635.CEL		
GSM948629.CEL	GSM948636.CEL		

Figure 1: 4 types of samples (this YW/YM/OW/OM color code will be used in later plots).

```
> info
                ID TYPE
GSM948623.CEL YW1   YW
GSM948624.CEL YW2   YW
GSM948625.CEL YW3   YW
GSM948626.CEL YW4   YW
GSM948627.CEL YW5   YW
GSM948628.CEL YW6   YW
GSM948629.CEL YW7   YW
GSM948630.CEL YM1   YM
GSM948631.CEL YM2   YM
GSM948632.CEL YM3   YM
GSM948633.CEL YM4   YM
GSM948634.CEL YM5   YM
GSM948635.CEL YM6   YM
GSM948636.CEL YM7   YM
GSM948637.CEL OW1   OW
GSM948638.CEL OW2   OW
GSM948639.CEL OW3   OW
GSM948640.CEL OW4   OW
GSM948641.CEL OM1   OM
GSM948642.CEL OM2   OM
GSM948643.CEL OM3   OM
GSM948644.CEL OM4   OM
```

The raw data is then read using `ReadAffy`, which produces an `AffyBatch` object.

```
> library(affy)
> affy <- ReadAffy(filenames=rownames(info), phenoData=info)
```

2 Preprocessing and quality control

Raw intensities values were preprocessed and normalized by robust multi-array average (RMA) expression measure i.e. the same procedure used by the authors of the original analysis [1, p. 1037]. This gives us an `ExpressionSet` object, which encapsulates a 54675×22 matrix containing $\log_2$-intensity values for each probeset/gene (row) and each array/sample (column).

```
> eset <- rma(affy)
> dim(eset)
Features  Samples
   54675       22
```

As a first quality control check, we draw box-plots and histograms of the intensity values before and after preprocessing, and we see that after RMA the distributions become much more similar (figure 2).

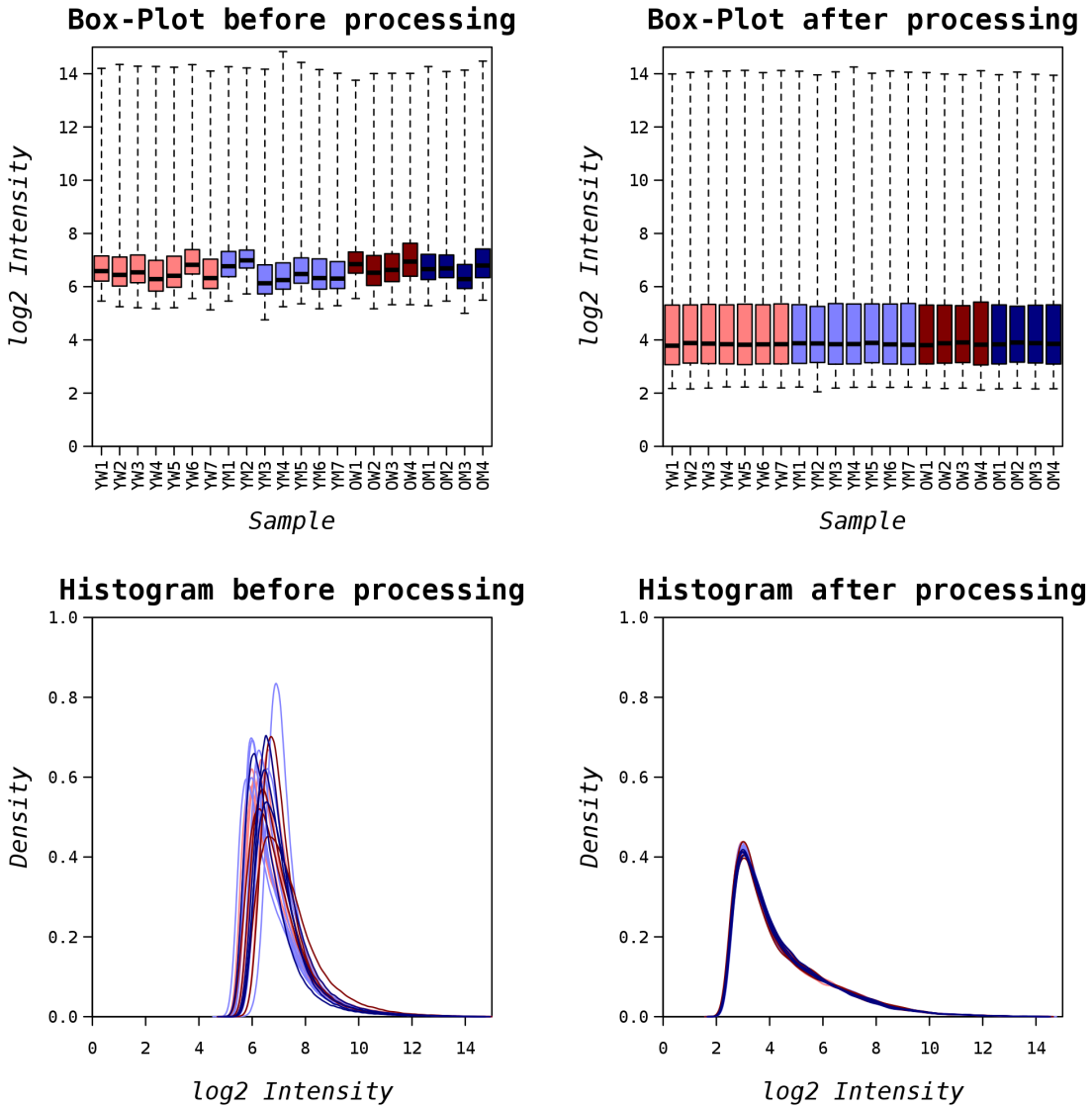

Figure 2: Distributions of intensity values before and after using RMA.

MA-plots are another way to visualize distributions discrepancy across the arrays/samples. For a given array/sample, the intensity ratios between this sample and the mean of all other samples (M) are plotted against average intensities (A). Such scatter plots were drawn for 4 different samples (figure 3), and we see that the points are distributed around the $y = 0$ horizontal line, indicating good normalization.

Finally, we actually have access to the processed data the original authors normalized and used in their research: the file containing it can also be downloaded. So in that particular case we can compare our pro-

cessed data to the original processed data, assuming the original authors knew what they did. Some quick correlation check is performed below, but I did not further use the already-processed data.

```
> orig <- readExpressionSet(orig.file)
> corr <- diag(cor(exprs(eset),exprs(orig)))
> names(corr) <- info$ID
> corr
  YW1   YW2   YW3   YW4   YW5   YW6   YW7   YM1   YM2   YM3   YM4
0.999 0.998 0.998 0.999 0.999 0.998 0.998 0.998 0.998 0.999 0.998
  YM5   YM6   YM7   OW1   OW2   OW3   OW4   OM1   OM2   OM3   OM4
0.998 0.999 0.999 0.998 0.998 0.998 0.999 0.998 0.998 0.998 0.999
```

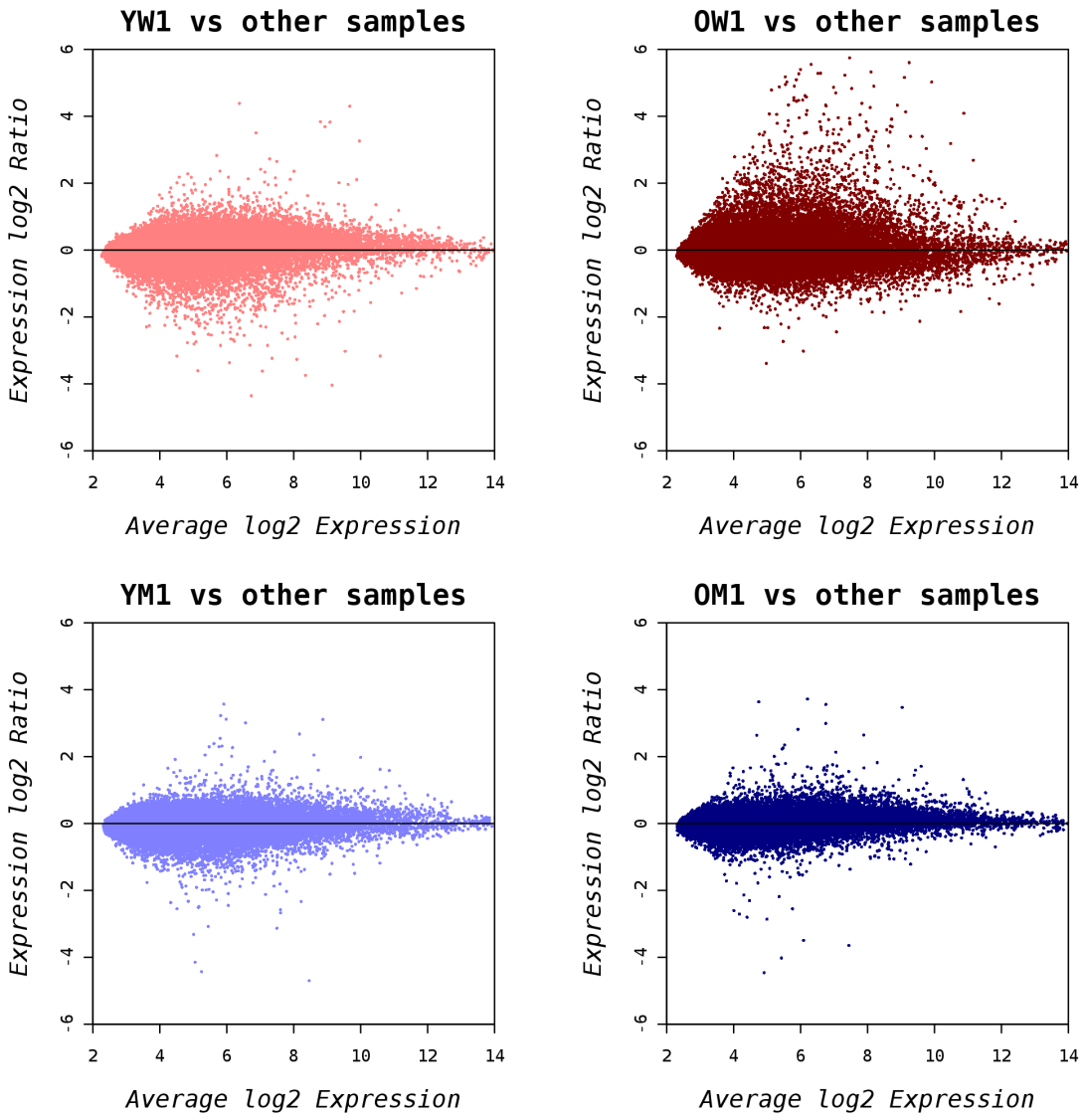

Figure 3: MA-plots for samples YW1, OW1, YM1, and OM1.

3 Setting up the LIMMA model

We will now fit the intensity values into a linear model using LIMMA [6], but before that we will filter out some genes. According to the LIMMA user's guide this filtering step is recommended, but we have to be careful with it:

> *We generally recommend that [filtering] is done before the linear modelling and empirical Bayes step, but after normalization. [...] Note that filtering methods involving variances should not be used. The limma algorithm analyses the spread of the genewise variances. Any filtering method based on genewise variances will change the distribution of variances, will interfere with the limma algorithm and hence will give poor results.*

The `genefilter` library will be used for filtering, however its default filter removes genes with low interquartile range, which interferes with the variance. Therefore we will rather use a filter which removes genes with low mean intensity, i.e. those with an intensity close to the background intensity. We also filter out genes lacking an Entrez identifier, but we keep duplicate genes. Once this filtering step is done, we are left with only 37386 genes (instead of 54675).

```
> library(genefilter)
> eset <- featureFilter(eset, require.entrez=TRUE, remove.dupEntrez=FALSE)
> eset <- varFilter(eset, var.func=mean, var.cutoff=0.1)
> dim(eset)
Features  Samples
   37386       22
```

We may now fit the intensity values into a linear model with four coefficients, one for each possible sample type: OM, OW, YM, and YW. The required design matrix is built using the `info$TYPE` character vector. We are interested in three contrasts: OW-YW, OM-YM, and OM-OW: those are specified using a contrasts matrix.

```
> library(limma)
> mat <- model.matrix(~0+info$TYPE) # design matrix
> colnames(mat) <- substring(colnames(mat), 10) # cols: OM OW YM YW
> rownames(mat) <- info$ID                      # rows: YW1 YW2 ... OM3 OM4
> fit <- lmFit(eset, mat)
> mat <- makeContrasts(OW-YW, OM-YM, OM-OW, levels=mat) # contrasts matrix
> colnames(mat) <- sub(" - ", "vs", colnames(mat))
> fit <- contrasts.fit(fit, mat)
> fit <- eBayes(fit)
```

The returned object, `fit`, contains the design matrix and the contrasts matrix:

```
> fit$design
    OM OW YM YW
YW1  0  0  0  1
YW2  0  0  0  1
YW3  0  0  0  1
YW4  0  0  0  1
YW5  0  0  0  1
YW6  0  0  0  1
YW7  0  0  0  1
YM1  0  0  1  0
YM2  0  0  1  0
YM3  0  0  1  0
```

```
YM4  0  0  1  0
YM5  0  0  1  0
YM6  0  0  1  0
YM7  0  0  1  0
OW1  0  1  0  0
OW2  0  1  0  0
OW3  0  1  0  0
OW4  0  1  0  0
OM1  1  0  0  0
OM2  1  0  0  0
OM3  1  0  0  0
OM4  1  0  0  0
attr(,"assign")
[1] 1 1 1 1
attr(,"contrasts")
attr(,"contrasts")$`info$TYPE`
[1] "contr.treatment"

> fit$contrasts
      Contrasts
Levels OWvsYW OMvsYM OMvsOW
    OM      0      1      1
    OW      1      0     -1
    YM      0     -1      0
    YW     -1      0      0
```

4 Computing fold changes

The `limma` library has a function `topTable` which may be used to find genes with large ($\log_2$-)fold changes.

```
> top10 <- topTable(fit, "OWvsYW", sort.by="logFC")
> top10
                logFC  AveExpr         t      P.Value  adj.P.Val            B
207175_at     4.390614 3.903407  4.655807 1.297359e-04 0.02437339  1.189895424
224918_x_at   3.787628 5.568567  4.760160 1.009723e-04 0.02123274  1.418692852
231736_x_at   3.748317 5.217157  4.946333 6.465367e-05 0.01888392  1.825240579
200832_s_at   3.671929 3.905373  4.118163 4.741164e-04 0.03930461  0.005890609
205913_at     3.411547 4.931039  4.042102 5.695358e-04 0.04258533 -0.161596532
219140_s_at   3.277798 3.921454  3.572340 1.757546e-03 0.06448913 -1.188784851
1565162_s_at  3.244035 3.489473  3.669230 1.394710e-03 0.05832510 -0.978421718
205440_s_at   3.013657 3.298222  3.596606 1.658785e-03 0.06283217 -1.136198395
225420_at     2.897431 6.755220  4.564742 1.615152e-04 0.02602762  0.989807138
202917_s_at  -2.816974 7.673897 -3.099358 5.349337e-03 0.10092707 -2.195991680
```

The meaning of `logFC` in that data frame has to be explained. In general, a fold change measures how much a quantity changes when going from an initial value to a final value: if it goes from A to B the fold change is the ratio B/A. In the case of multiple initial values $a_1, \ldots, a_m$ and final values $b_1, \ldots, b_n$, the fold change is usually understood as being the ratio B/A of the means $A = \text{mean}(a_1, \ldots, a_m)$ and $B = \text{mean}(b_1, \ldots, b_n)$.

However in the present case, all values in `eset` are on the $\log_2$-scale, and similarly the LIMMA algorithm does everything on the $\log_2$-scale. For this reason, when it computes the fold change, it uses the *geometric* mean rather than the *arithmetic* mean. Moreover, the `logFC` appearing in `topTable` results are in fact the values found in `fit$coefficients`. We can verify this ourselves (using the fact that the geometric mean is `2^mean(log2(x))` where `mean(x)` is the arithmetic mean).

```
> OW <- exprs(eset)[rownames(top10), grep("OW", info$TYPE)] # log2-intensity, OW columns
> YW <- exprs(eset)[rownames(top10), grep("YW", info$TYPE)] # log2-intensity, YW columns
> AlogFC <- log2( rowMeans(2^OW) / rowMeans(2^YW) ) # arithmetic logFC
> GlogFC <- rowMeans(OW) - rowMeans(YW)             # geometric logFC
> cbind(top=top10$logFC, coef=fit$coefficients[rownames(top10),"OWvsYW"], GlogFC, AlogFC)
                  top      coef    GlogFC    AlogFC
207175_at     4.390614  4.390614  4.390614  7.027618
224918_x_at   3.787628  3.787628  3.787628  6.123829
231736_x_at   3.748317  3.748317  3.748317  6.226807
200832_s_at   3.671929  3.671929  3.671929  6.556062
205913_at     3.411547  3.411547  3.411547  6.206911
219140_s_at   3.277798  3.277798  3.277798  6.774913
1565162_s_at  3.244035  3.244035  3.244035  6.564360
205440_s_at   3.013657  3.013657  3.013657  5.218346
225420_at     2.897431  2.897431  2.897431  4.293458
202917_s_at  -2.816974 -2.816974 -2.816974 -3.256315
```

Therefore, if we want to know the usual "arithmetic" `logFC`, we have to compute it ourselves.

As an illustration, volcano plots were drawn using both `logFC` (figure 4).

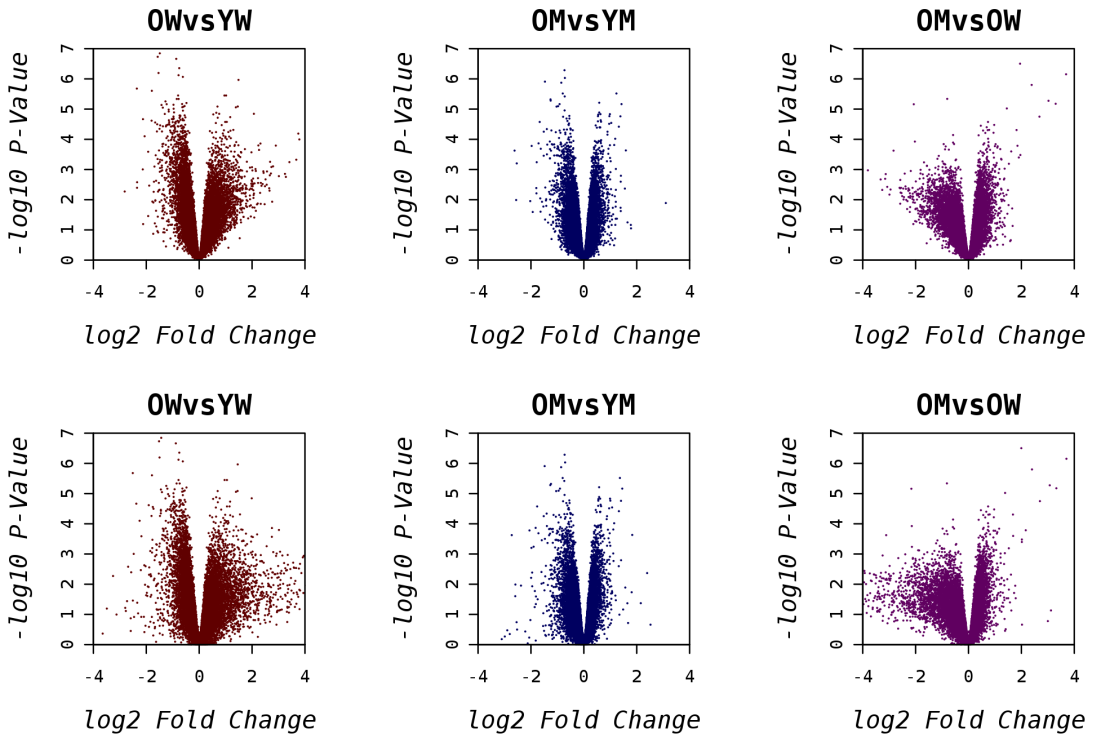

Figure 4: Volcano plots: first line use geometric `logFC`, second line uses arithmetic `logFC`.

5 Finding differentially expressed genes

The `decideTests` function from `limma` can be used to perform multiple hypothesis testing. Its return value is a matrix with a row for each gene and a column for each contrast. A value of 1 means increased expression, -1 means decreased, 0 means no change.

```
> sel <- decideTests(fit, method="global", p.value=0.05)
> sel
TestResults matrix
         Contrasts
          OWvsYW OMvsYM OMvsOW
  1053_at      0      0      0
  117_at       0      0      0
  121_at       0      0      0
  1316_at      0      0      0
  1320_at      0      0      0
37381 more rows ...
```

A Venn diagram of differentially expressed genes is shown in figure 5 and a heatmap with clustering in figure 6. We then annotated the differentially expressed genes for each contrast, removed the duplicates, and wrote the resulting list to a file for later use.

```
> library(annotate)
> library(hgu133plus2.db)
> genes <- which(sel[,"OWvsYW"] != 0)
> genes <- mget(names(genes), hgu133plus2SYMBOL)
> genes <- unique(unlist(genes))
> write(genes, file="OWvsYW.txt")
```

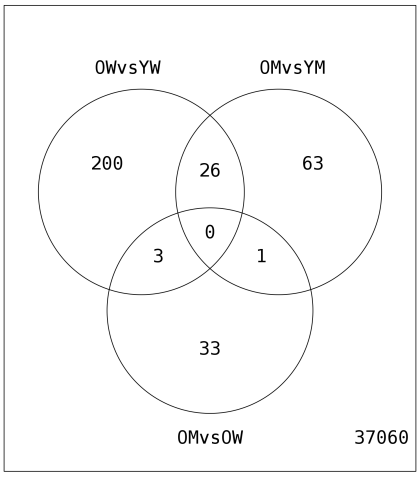

Figure 5: Venn diagram of differentially expressed genes.

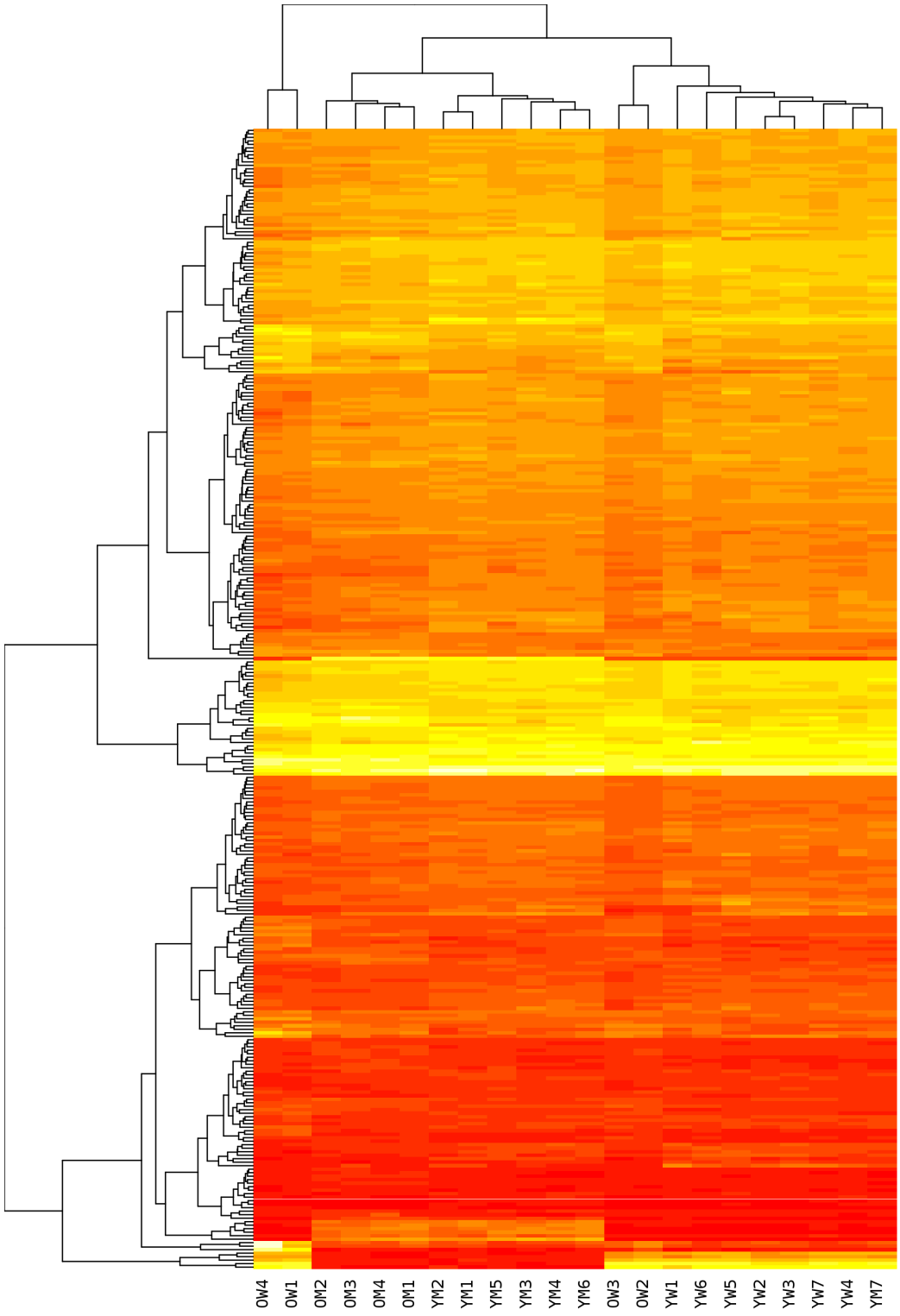

Figure 6: Heatmap and clustering for the selected genes.

6 Finding enriched gene sets

The GOstats package was used to find enriched Gene Ontology (GO) terms. GO in fact covers three domains: cellular components (CC), molecular function (MF), and biological process (BP). Due to the long computational time involved, we only searched for enriched GO BP terms. These terms were then written to a file. Up-regulated genes and down-regulated genes were handled separately. The piece of code below is an example for terms enriched with up-regulated genes in old women compared to young women.

```
> library(GOstats)
> genes.all <- mget(rownames(eset), hgu133plus2ENTREZID)
> genes.all <- unique(unlist(genes.all))
> genes <- which(sel[,"OWvsYW"] == 1) # up-regulated genes
> genes <- mget(names(genes), hgu133plus2ENTREZID)
> genes <- unique(unlist(genes))
> go.par <- new(
+   "GOHyperGParams",
+   geneIds = genes,                # selected genes
+   universeGeneIds = genes.all,    # universe genes
+   annotation = "hgu133plus2.db",  # annotation package
+   ontology = "BP",                # BP domain only
+   pvalueCutoff = 0.05,            # cutoff value
+   conditional = TRUE,             # use the DAG structure of GO
+   testDirection = "over"          # look for overrepresented terms
> )
> go.res <- hyperGTest(go.par) # this step is computationally expensive!
> go.res <- summary(go.res)
> write.table(go.res[,c(1,7)], file="OWvsYW.txt")
```

We also submitted the gene list for OWvsYW to the DAVID server [7]. GO terms obtained in this manner can be found in the DAVID-OWvsYW.txt file.

7 Conclusion

Regarding differential expression between Old Women and Young Women, we found some results close to those described in the original article. For example, with Old Women, genes involved in extra-cellular matrix remodeling are up-regulated while genes involved in mitochondrial structure and function are down-regulated. We also discovered a drastic up-regulation of gene ADIPOQ, a claim also found in the original paper.

References

[1] DongmeiLiu, Maureen A.Sartor, Gustavo A.Nader, Emidio E.Pistilli, LeahTanton, CharlesLilly, LaurieGutmann, Heidi B.IglayReger, Paul S.Visich, Eric P.Hoffman, Paul M.Gordon *Microarray Analysis Reveals Novel Features of the Muscle Aging Process in Men and Women* (2013).
http://biomedgerontology.oxfordjournals.org/content/68/9/1035

[2] Array Express E-GEOD-38718: *Sex and aging effect on skeletal muscle transcriptome in humans*
http://www.ebi.ac.uk/arrayexpress/experiments/E-GEOD-38718/

[3] Gene Expression Omnibus GSE38718: *Sex and aging effect on skeletal muscle transcriptome in humans*
http://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE38718

[4] Maureen A. Sartor, George D. Leikauf, Mario Medvedovic *LRpath: a logistic regression approach for identifying enriched biological groups in gene expression data* (2009).
http://www.ncbi.nlm.nih.gov/pubmed/19038984

[5] Maureen A. Sartor, Craig R. Tomlinson, Scott C. Wesselkamper, Siva Sivaganesan, George D. Leikauf, Mario Medvedovic *Intensity-based hierarchical Bayes method improves testing for differentially expressed genes in microarray experiments* (2006).
http://www.biomedcentral.com/1471-2105/7/538

[6] Gordon K. Smyth *LIMMA: Linear Models for Microarray Data* (2005).
http://bioconductor.org/packages/release/bioc/html/limma.html

[7] DAVID Functional Annotation Tool
http://david.abcc.ncifcrf.gov/