Lab 04 - Bioinformatics

Thijs Quast (thiqu264), Lennart Schilling (lensc874), Mariano Maquieira Mariani (marma330)

12-12-2018

Contents

Question 1

Using the *getGEOSuppFiles*-function of the *GEOquery*-package, the data for the series *GSE20986* will be downloaded. The data includes twelve different samples of human umbilical vein endothelial cells (HUVEC), human iris, retinal and choroidal cells. RNA extracts of cells were hybridised to *Affymetrix HGU133plus2* which represents the complete coverage of the *Human Genome U133 Set* plus 6,500 additional genes for analysis of over 47,000 transcripts.

In the first step, the data (series *GSE20986* will be downloaded from the website of the *National Center for Biotechnology Information* using the *GEOquery*-package. Afterwards, the data will be transformed to the correct format. Doing so, it will be untarred and gunzipped.

```
library(GEOquery)
x = getGEOSuppFiles("GSE20986")
untar("GSE20986/GSE20986_RAW.tar", exdir = "data")
cels = list.files("data/", pattern = "[gz]")
sapply(paste("data", cels, sep = "/"), gunzip)
```

```
## data/GSM524662.CEL.gz data/GSM524663.CEL.gz data/GSM524664.CEL.gz
##              13555726              13555055              13555639
## data/GSM524665.CEL.gz data/GSM524666.CEL.gz data/GSM524667.CEL.gz
##              13560122              13555663              13557614
## data/GSM524668.CEL.gz data/GSM524669.CEL.gz data/GSM524670.CEL.gz
##              13556090              13560054              13555971
## data/GSM524671.CEL.gz data/GSM524672.CEL.gz data/GSM524673.CEL.gz
##              13554926              13555042              13555290
```

After unpacking, a data.frame with the different samples (including target of each sample) will be created and saved as a txt.file. This file be used within the *read.affy*-function as an input parameter to read the unzipped files.

```
phenodata = matrix(rep(list.files("data"), 2), ncol =2)
phenodata <- as.data.frame(phenodata)
colnames(phenodata) <- c("Name", "FileName")
phenodata$Targets <- c("iris",
                       "retina",
                       "retina",
                       "iris",
                       "retina",
                       "iris",
                       "choroid",
                       "choroid",
                       "choroid",
                       "huvec",
                       "huvec",
                       "huvec")
write.table(phenodata, "data/phenodata.txt", quote = F, sep = "\t", row.names = F)
knitr::kable(phenodata)
```

Name	FileName	Targets
GSM524662.CEL	GSM524662.CEL	iris
GSM524663.CEL	GSM524663.CEL	retina
GSM524664.CEL	GSM524664.CEL	retina
GSM524665.CEL	GSM524665.CEL	iris
GSM524666.CEL	GSM524666.CEL	retina

Name	FileName	Targets
GSM524667.CEL	GSM524667.CEL	iris
GSM524668.CEL	GSM524668.CEL	choroid
GSM524669.CEL	GSM524669.CEL	choroid
GSM524670.CEL	GSM524670.CEL	choroid
GSM524671.CEL	GSM524671.CEL	huvec
GSM524672.CEL	GSM524672.CEL	huvec
GSM524673.CEL	GSM524673.CEL	huvec

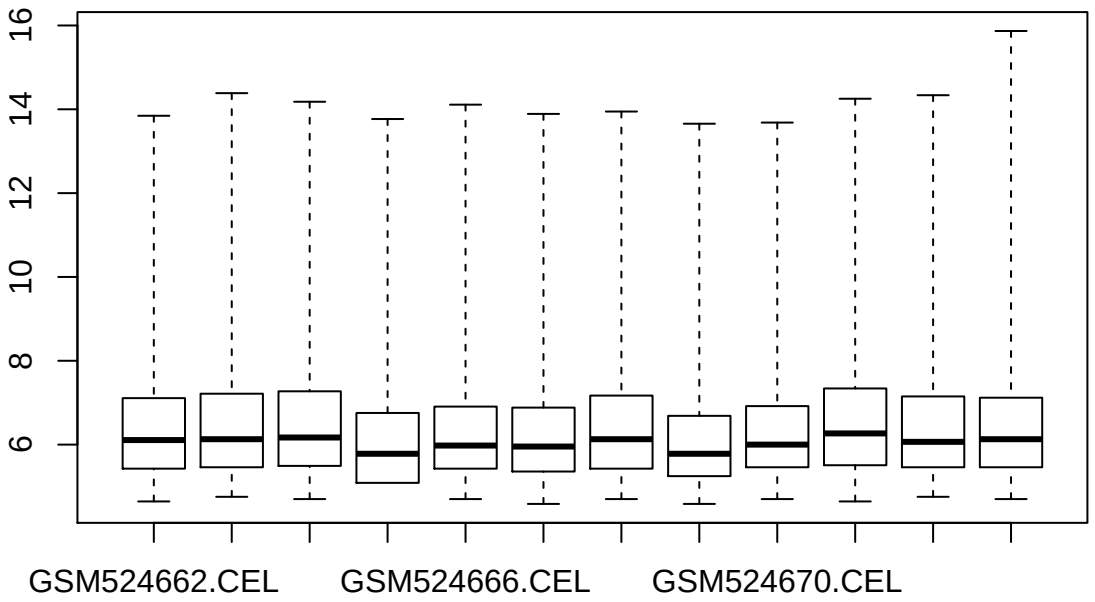
The downloaded and prepared data will be read. The created .txt-file will be integrated in the returned *AffyBatch*-object.

```
library(simpleaffy)
celfiles <- read.affy(covdesc = "phenodata.txt", path = "data")
```

Boxplots for each sample will be created:

```
suppressWarnings(boxplot(celfiles))
```

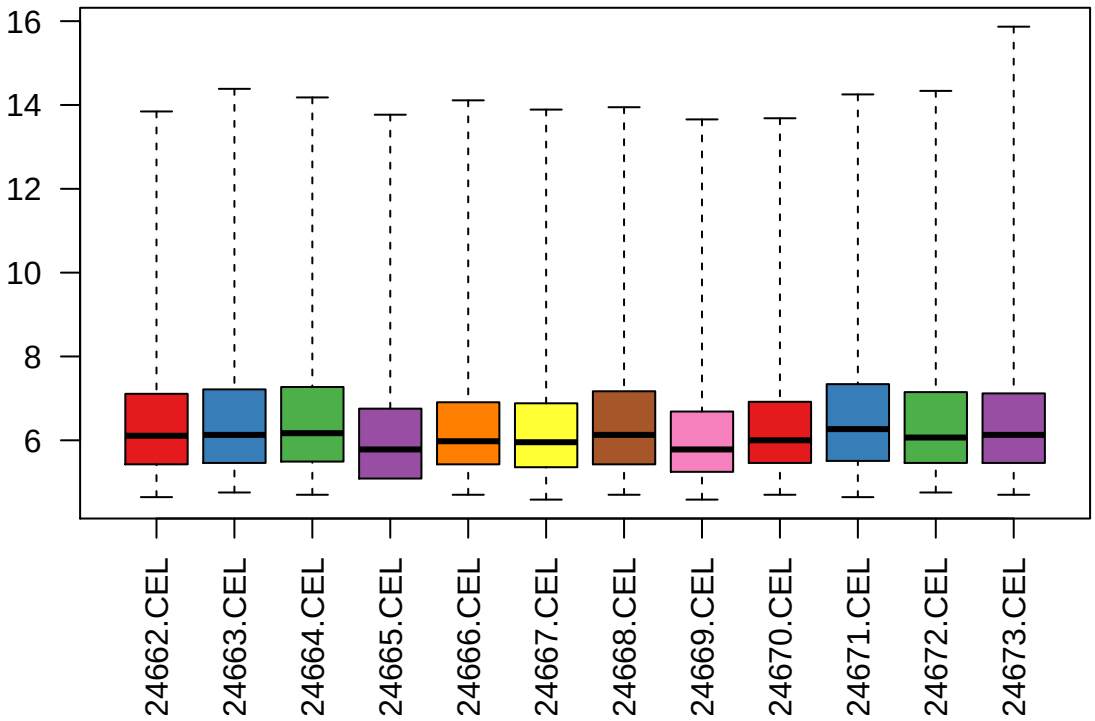
```
##
```



Each boxplot represents the distribution of each sample related to how well the RNAs fit to the delivered genes of the *Affymetrix Human Genome U133 Plus 2.0 Array*.

In the next step, the samples will be coloured differently.

```
library(RColorBrewer)
cols = brewer.pal(8, "Set1")
boxplot(celfiles, col = cols, las = 2)
```



The resulting boxplot is characterized by the same boxes as before. The colouring lead to fill the in total twelve boxes/samples with eight different colours. For us, this process does not deliver any bigger insights.

The data of each sample will be extracted. The samples will be named using the specified targets. As a result, a data frame with twelve columns and 1354896 rows will be extracted.

```
samples <- celfiles$Targets
eset <- exprs(celfiles)
colnames(eset) <- samples
```

Having a look at the first six rows of the extracted data, each column represents one sample. Each row value shows, how strong the RNA of the sample binded to the specific DNA spot. A low value indicates that a sample did not bind with a high intensity to the DNA spot. The higher the value the more binding intensity could be identified.

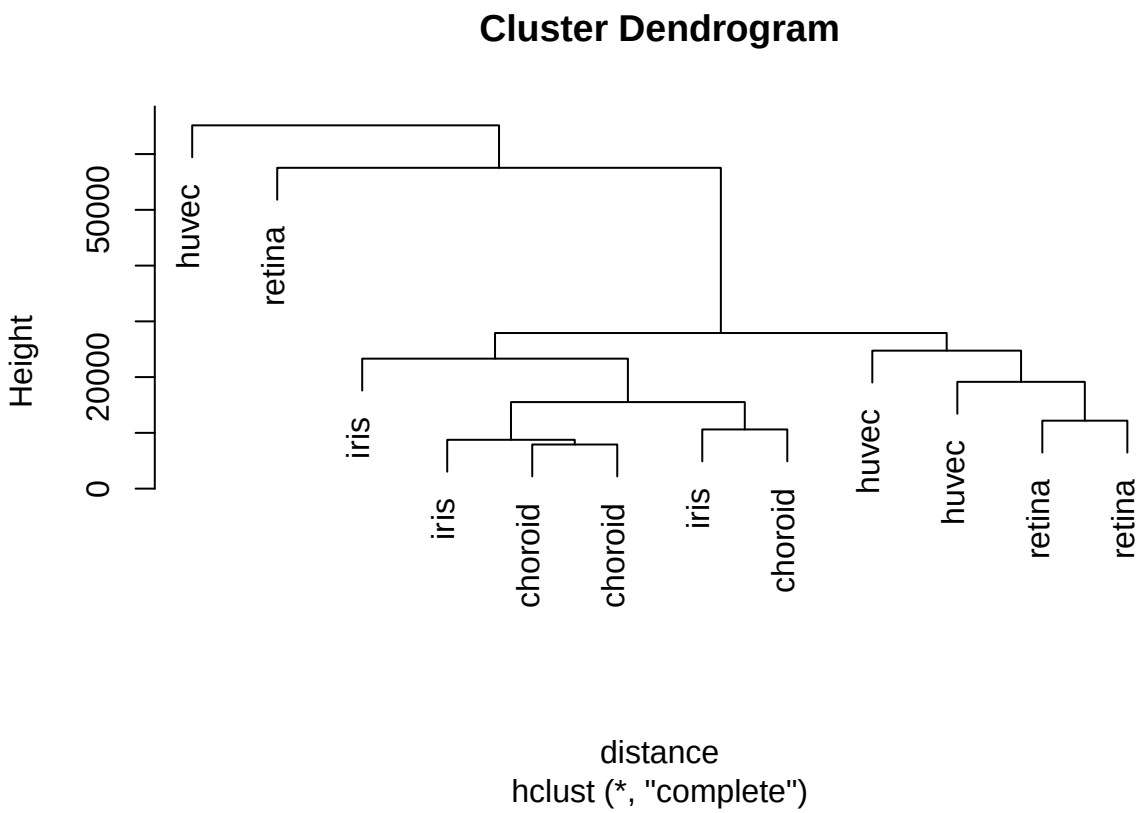
```
head(eset)
```

```
##    iris retina retina iris retina  iris choroid choroid choroid huvec
## 1    67    110     83   58     81    61      56      50      64   106
## 2 11835  12382  13641 8984  14521 10840   14401   13408   13640 13277
## 3    87    134    119   74     94    73     143      97     115   138
## 4 12199  12731  13762 9928  14816 11134   14505   13316   13375 13530
## 5    52     66     54   46     89    49      47      50      67    52
## 6    64     90     86   38     78    52      75      57      45    80
##   huvec huvec
## 1    99    78
## 2 12344 16459
## 3   140   108
## 4 12553 17324
## 5    62    77
## 6    55    79
```

Using these values, euclidian distances between the samples can be calculated. With these distance results, hierarchical clustering can be performed. This gives a visual overview about how similar or non-similar the

different samples are.

```
distance <- dist(t(eset), method = "maximum")
clusters <- hclust(distance)
plot(clusters)
```



It can be seen that on the on hand samples of *huvec* and *retina* and on the other hand samples of *iris* and *choroid* are clustered together. A point which has to be mentioned is that for *huvec* and *retina*, two different clusters can be found.

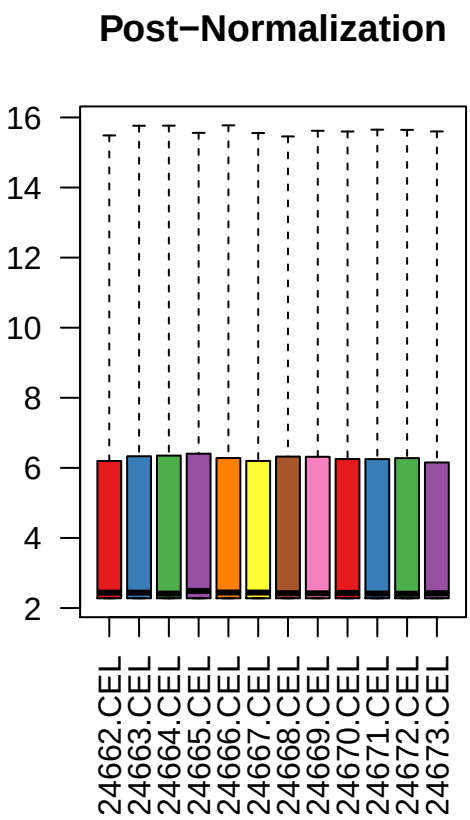
So far, the data was not normalized yet. The *affyPLM*-package can be used to correct the optical noise and non-specific binding.

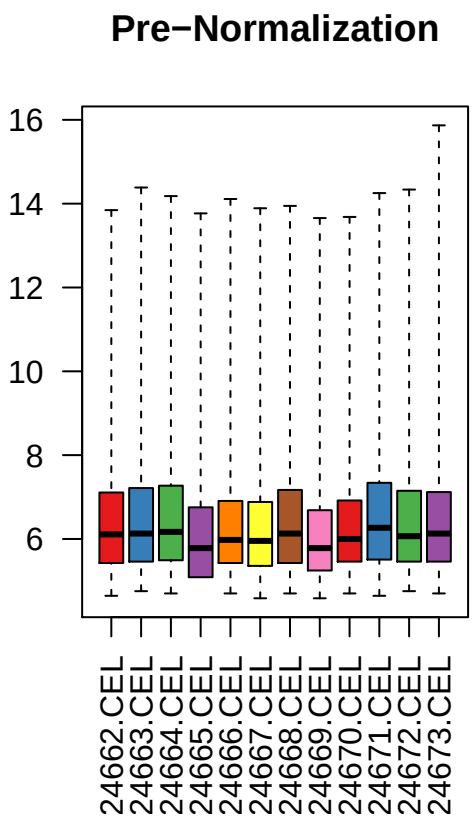
```
library(affyPLM)
celfiles.gcrma = gcrma(celfiles)
```

```
## Adjusting for optical effect.....Done.
## Computing affinities.Done.
## Adjusting for non-specific binding.....Done.
## Normalizing
## Calculating Expression
```

Comparing the results of the boxplots before and after the nomalization, bigger differences can be seen:

```
par(mfrow=c(1,2))
boxplot(celfiles.gcrma, col = cols, las = 2, main = "Post-Normalization");
boxplot(celfiles, col = cols, las = 2, main = "Pre-Normalization")
```





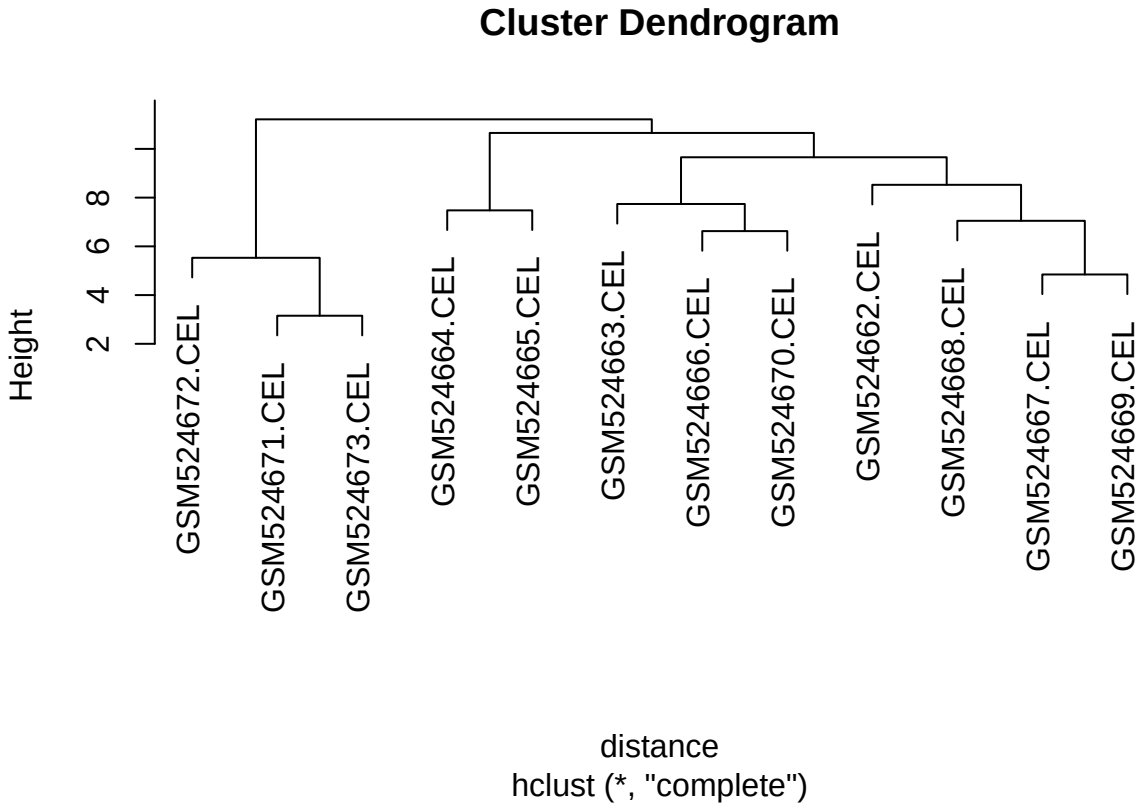

```
dev.off()
```

```
## null device
##           1
```

The microarray data for the samples seem to be much more similar distributed after the normalization.

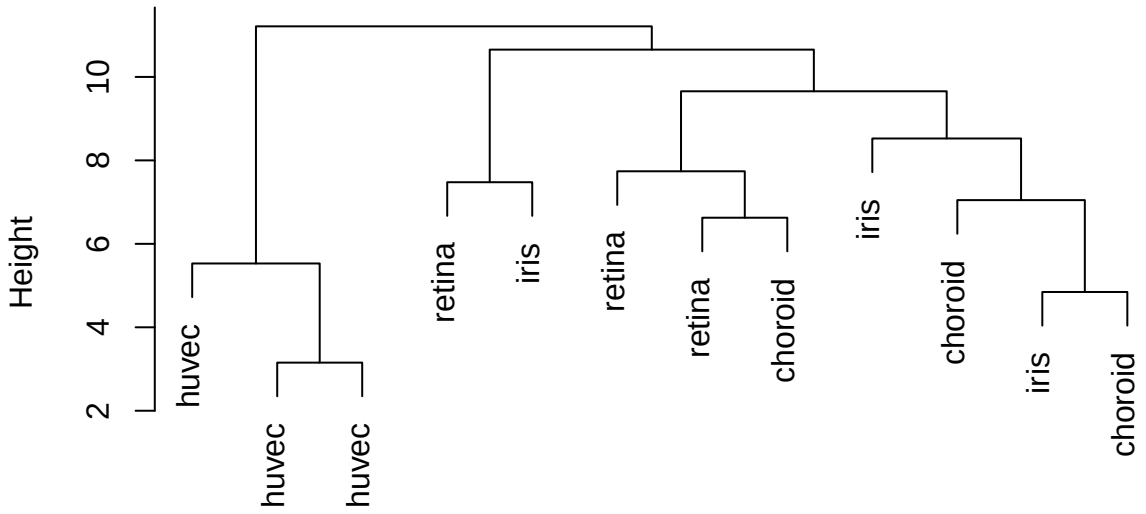
The calculation of the distances between the normalized samples lead to the following hierarchical clustering:

```
distance <- dist(t(exprs(celfiles.gcrma)), method = "maximum")
clusters <- hclust(distance)
plot(clusters)
```



Within this plot, the different sample names were not adjusted before. That is why it is hard to tell if the clustering changed related to the previous clustering (not normalized). Manually, we adjust the samples names and repeat the clustering.

```
esetNormalized = exprs(celfiles.gcrma)
colnames(esetNormalized) = celfiles$Targets
clusters = hclust(dist(t(esetNormalized), method = "maximum"))
plot(clusters)
```



As a result, it leads to the result that all *huvec*-samples are clutered together - separated from the other eye cells.

In the following step, a design matrix will be created using the different specified sample names (*iris*, *retina*, *choroid* and *huvec*). The result is a matrix consisting of the four dummy variables and the twelve samples as rows. The first row for example has a value of 1 within the *iris*-variable and 0 elsewhere. Thus, this row represents an *iris*-sample

```
samples <- as.factor(samples)
design <- model.matrix(~0+samples)
colnames(design) <- c("choroid", "huvec", "iris", "retina")
knitr::kable(design)
```

choroid	huvec	iris	retina
0	0	1	0
0	0	0	1
0	0	0	1
0	0	1	0
0	0	0	1
0	0	1	0
1	0	0	0
1	0	0	0
1	0	0	0
0	1	0	0
0	1	0	0
0	1	0	0

The first row for example has a value of 1 within the *iris*-variable and 0 elsewhere. Thus, this row represents

an *iris*-sample

Using the created design matrix, a contrast matrix will be also constructed calculating the contrasts between the *huvec*-samples and the other three sample types.

```
library(limma)
contrast.matrix = makeContrasts(
  huvec_choroid = huvec - choroid,
  huvec_retina = huvec - retina,
  huvec_iris = huvec - iris,
  levels = design)
```

Both design matrix and contrast matrix will be used to fit a linear model to the normalized data.

```
fit = lmFit(celfiles.gcrma, design)
huvec_fit <- contrasts.fit(fit, contrast.matrix)
huvec_ebay <- eBayes(huvec_fit)
```

Within the next steps, a plot will created to explain the differences between the *huvec*-samples and the other samples visually. A gene will be classified as upregulated or downregulated only if the adjusted p-value is lower than 0.5. Then, these genes are more (upregulated) or less (downregulated) included within the *huvec*-samples compared to the other samples.

```
library(hgu133plus2.db)
library(annotate)
probenames.list <- rownames(topTable(huvec_ebay, number = 100000))
getsymbols <- getSYMBOL(probenames.list, "hgu133plus2")
results <- topTable(huvec_ebay, number = 100000, coef = "huvec_choroid")
results <- cbind(results, getsymbols)
summary(results)
```

```
##      logFC             AveExpr             t                P.Value
##  Min.   :-9.19111   Min.   : 2.279   Min.   :-39.77473   Min.   :0.0000
##  1st Qu.:-0.05967   1st Qu.: 2.281   1st Qu.: -0.70649   1st Qu.:0.1523
##  Median : 0.00000   Median : 2.480   Median :  0.00000   Median :0.5079
##  Mean   :-0.02353   Mean   : 4.375   Mean   :  0.07441   Mean   :0.5346
##  3rd Qu.: 0.03986   3rd Qu.: 6.241   3rd Qu.:  0.67455   3rd Qu.:1.0000
##  Max.   : 8.67086   Max.   :15.541   Max.   :296.84201   Max.   :1.0000
##
##    adj.P.Val            B             getsymbols
##  Min.   :0.0000   Min.   :-7.710   YME1L1  :   22
##  1st Qu.:0.6036   1st Qu.:-7.710   HFE     :   15
##  Median :1.0000   Median :-7.451   CFLAR   :   14
##  Mean   :0.7436   Mean   :-6.582   NRP2    :   14
##  3rd Qu.:1.0000   3rd Qu.:-6.498   ARHGEF12:   13
##  Max.   :1.0000   Max.   :21.290   (Other) :41857
##                                    NA's    :12740
```

```
results$threshold <- "1"
a <- subset(results, adj.P.Val < 0.05 & logFC > 5)
results[rownames(a), "threshold"] <- "2"
b <- subset(results, adj.P.Val < 0.05 & logFC < -5)
results[rownames(b), "threshold"] <- "3"
table(results$threshold)
```

```
##
##     1     2     3
```

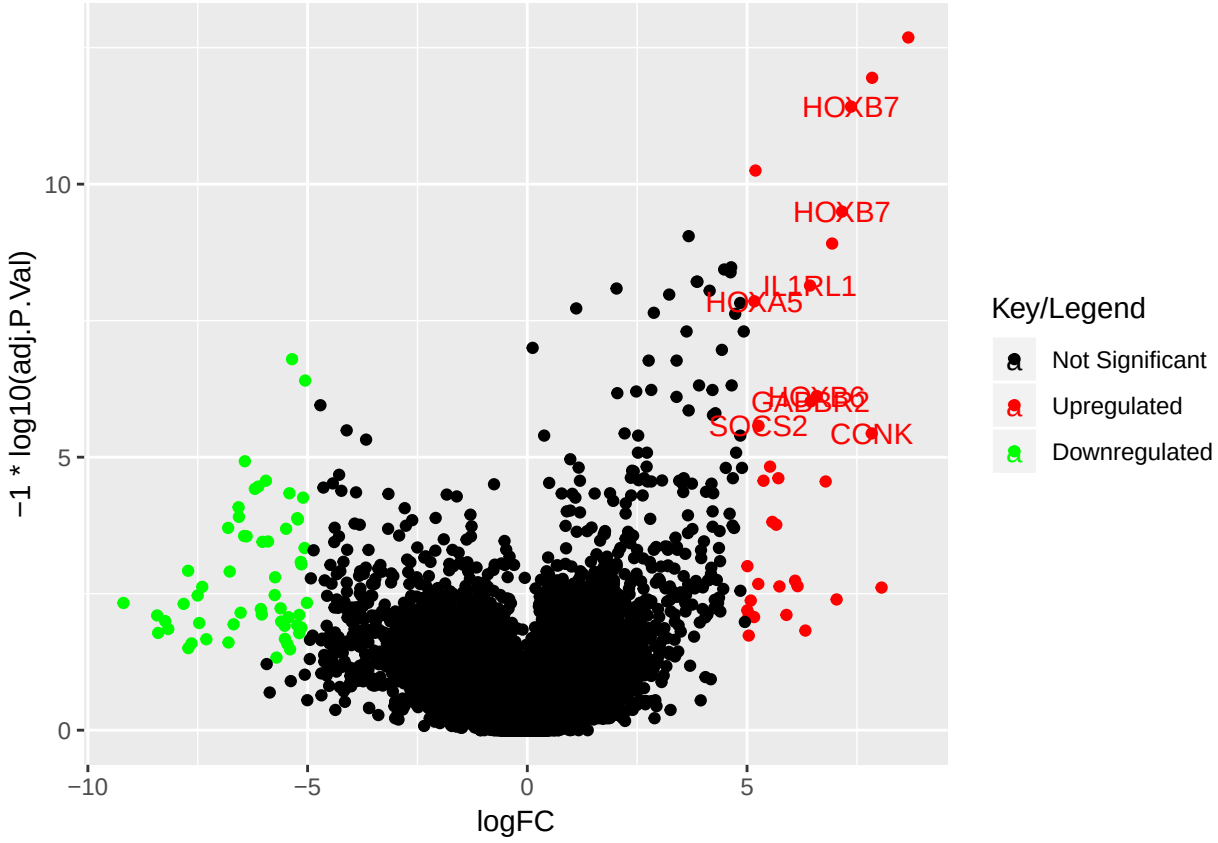
```
## 54587    33    55
```

```
library(ggplot2)
volcano <- ggplot(data = results,
                  aes(x = logFC, y = -1*log10(adj.P.Val),
                      colour = threshold,
                      label = getsymbols))

volcano <- volcano +
  geom_point() +
  scale_color_manual(values = c("black", "red", "green"),
                     labels = c("Not Significant", "Upregulated", "Downregulated"),
                     name = "Key/Legend")

volcano +
  geom_text(data = subset(results, logFC > 5 & -1*log10(adj.P.Val) > 5), aes(x = logFC, y = -1*log10(ad
```

```
## Warning: Removed 4 rows containing missing values (geom_text).
```



Question 2

We take 200 random samples from the 1,354,896 rows in eset

```
x<-sample(1:1354896,200, FALSE)
```

We take one choroid column, one retina column, one iris column and one huvec column. (200 samples in total)

```
colnames(eset)
```

```
##  [1] "iris"    "retina"  "retina"  "iris"    "retina"  "iris"    "choroid"
##  [8] "choroid" "choroid" "huvec"   "huvec"   "huvec"
```

```
sample<-as.data.frame(eset[x,c(1,2,7,10)])
nrow(sample)
```

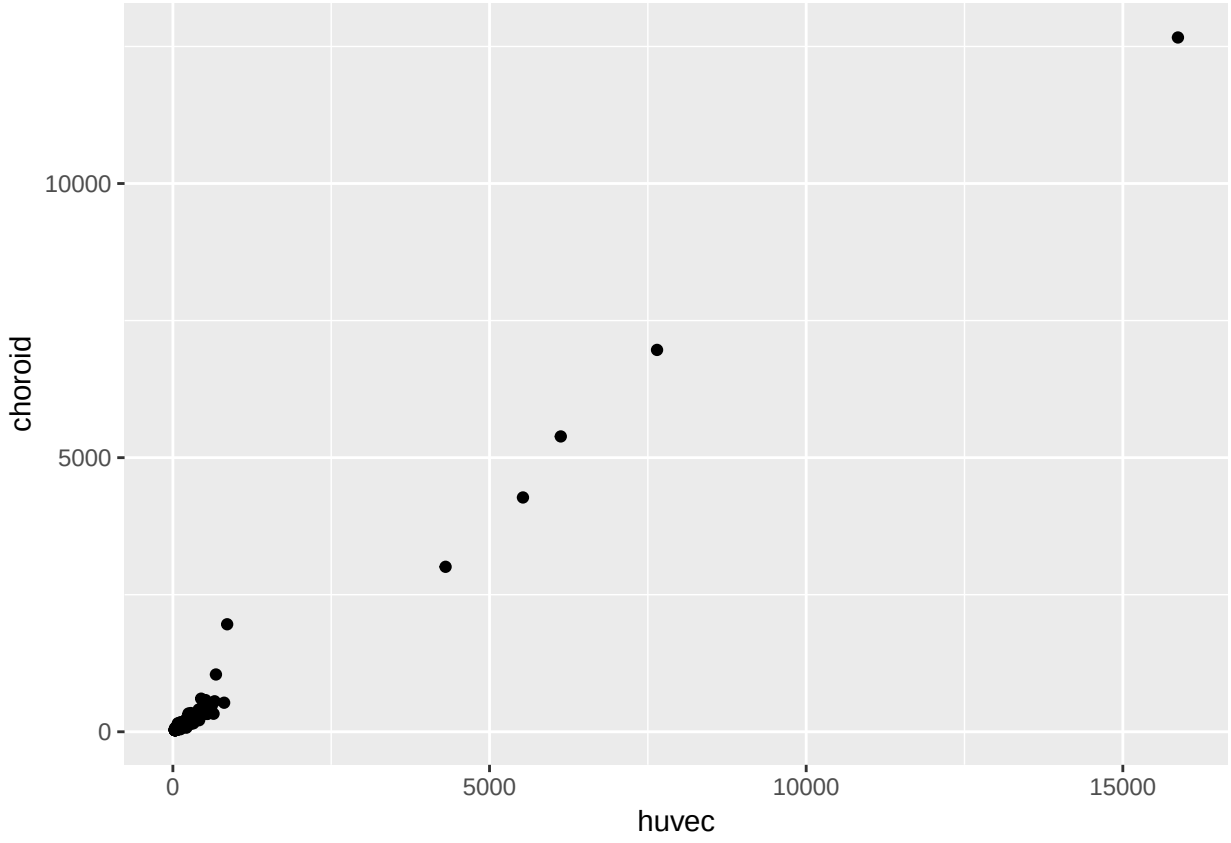
```
## [1] 200
```

sample is the original and sample_log will be the log2 of it.

```
sample_log<-log2(sample)
```

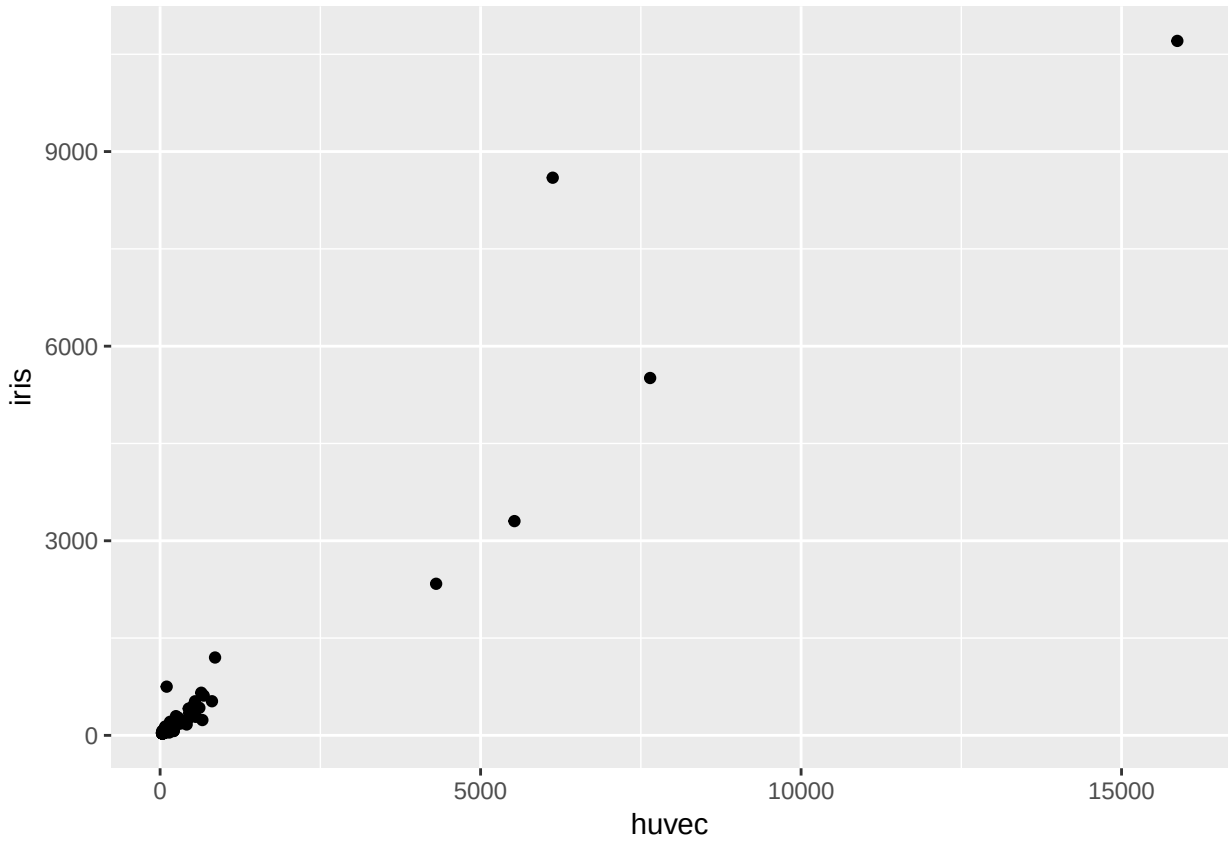
Sample we plot huvec_choroid

```
ggplot(sample, aes (huvec,choroid)) + geom_point()
```



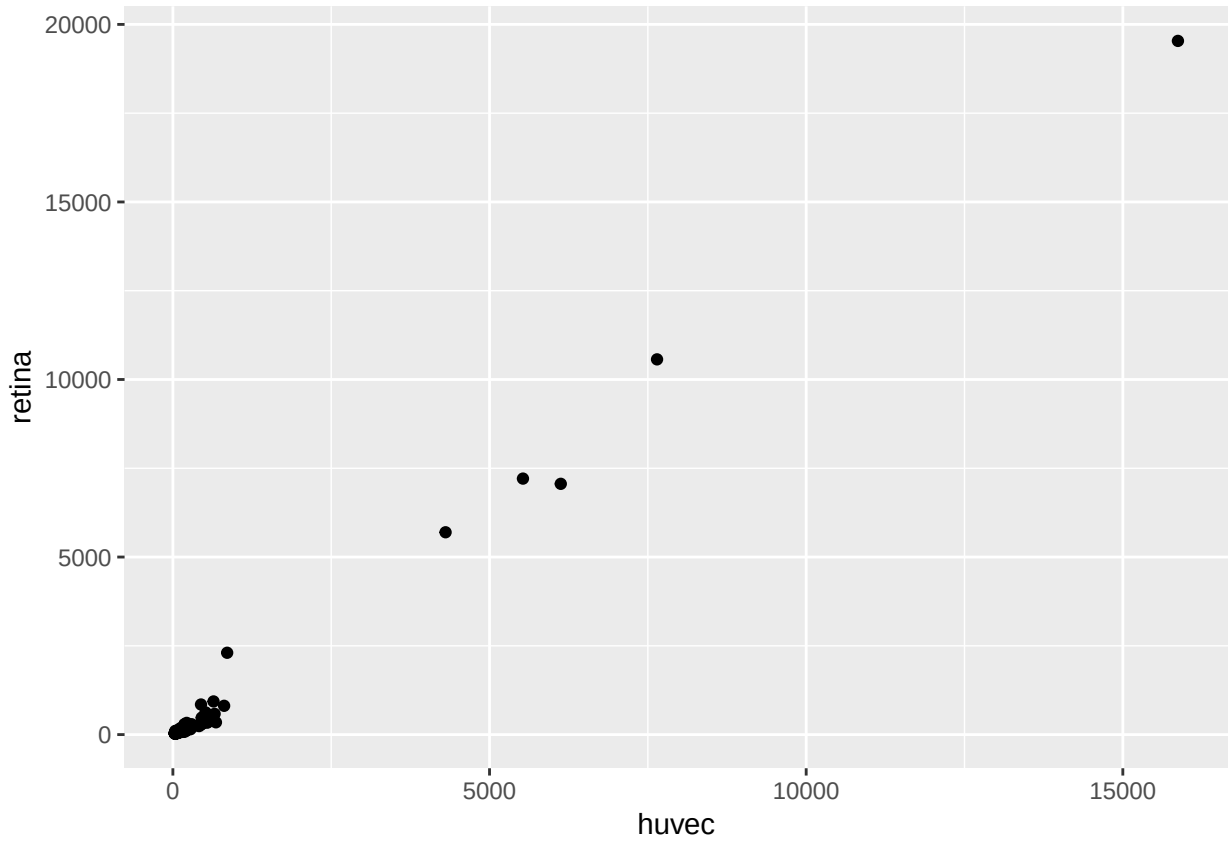
we plot huvec_iris

```
ggplot(sample, aes (huvec,iris)) + geom_point()
```



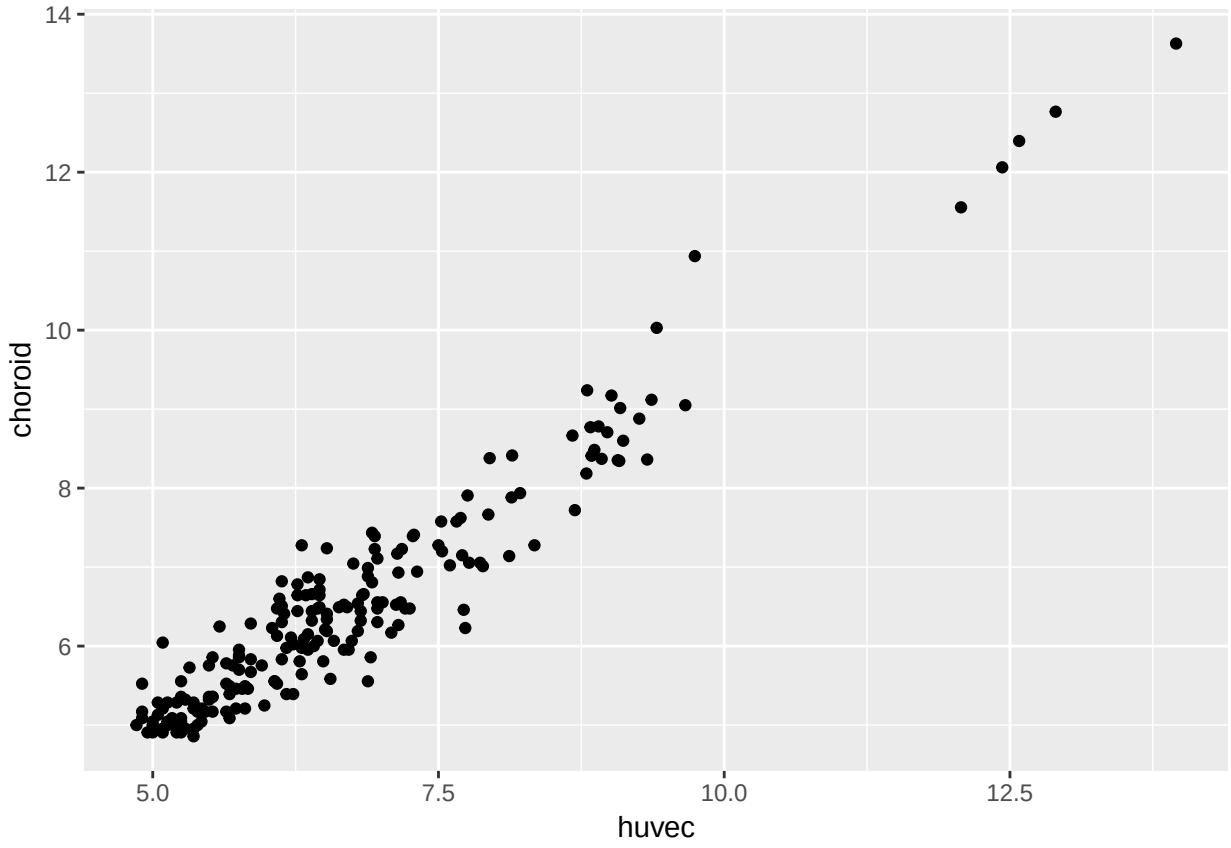
we plot huvec_retina

```
ggplot(sample,aes (huvec,retina)) + geom_point()
```



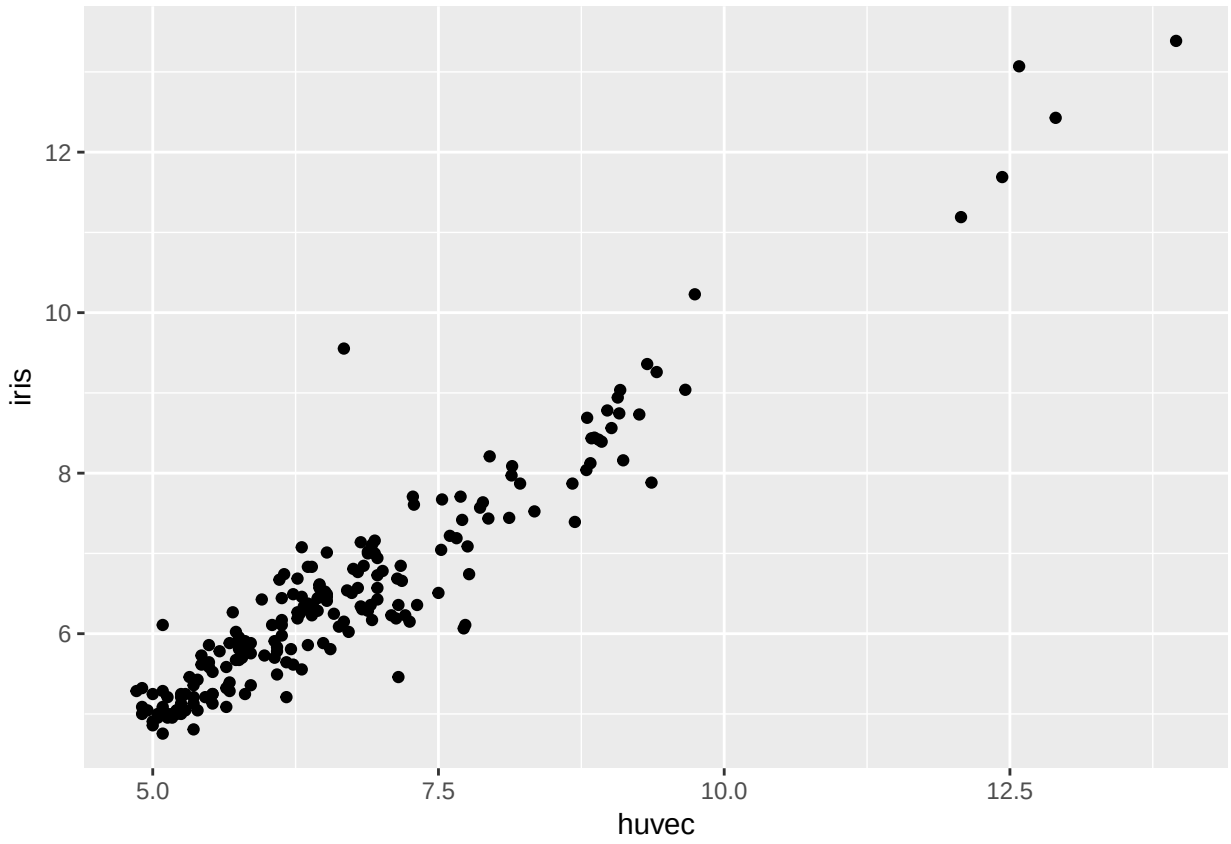
sample_log we plot huvec_choroid

```
ggplot(sample_log, aes (huvec,choroid)) + geom_point()
```



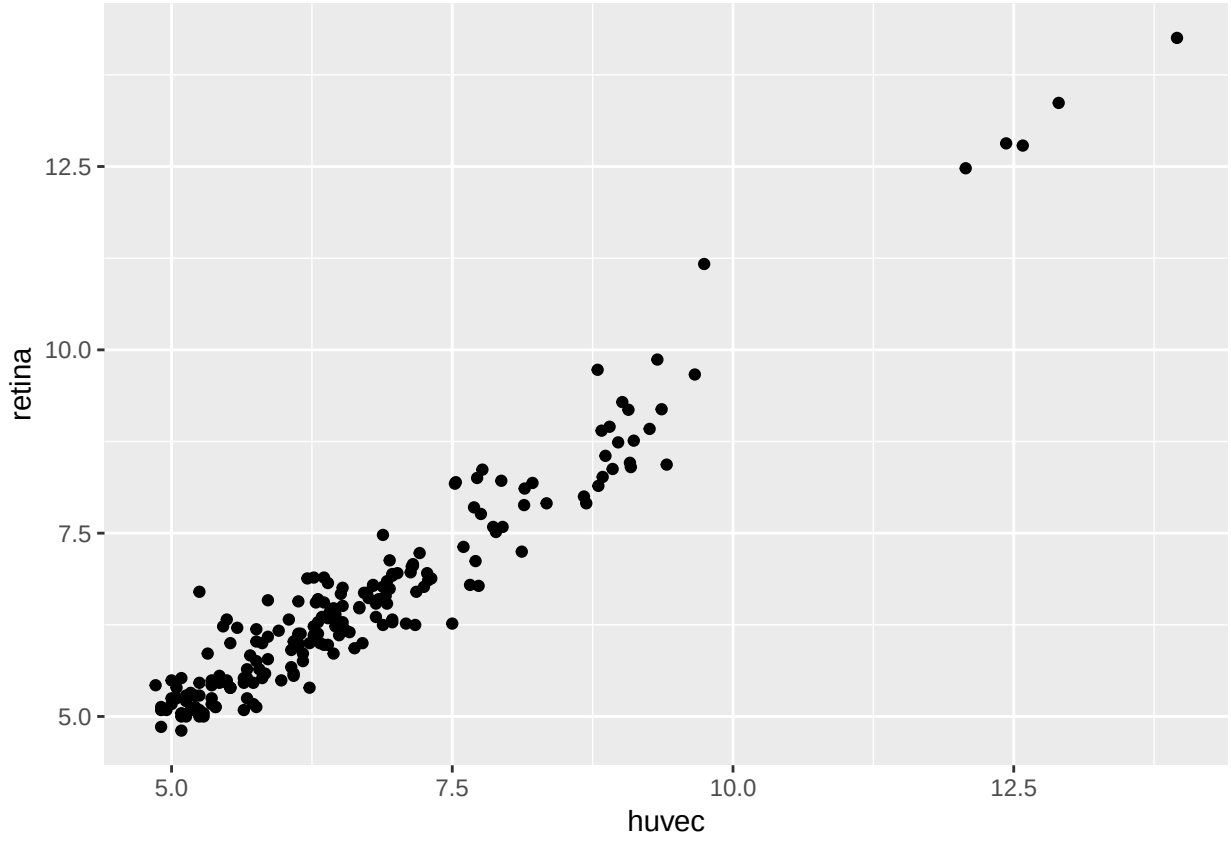
we plot huvec_iris

```
ggplot(sample_log, aes (huvec,iris)) + geom_point()
```



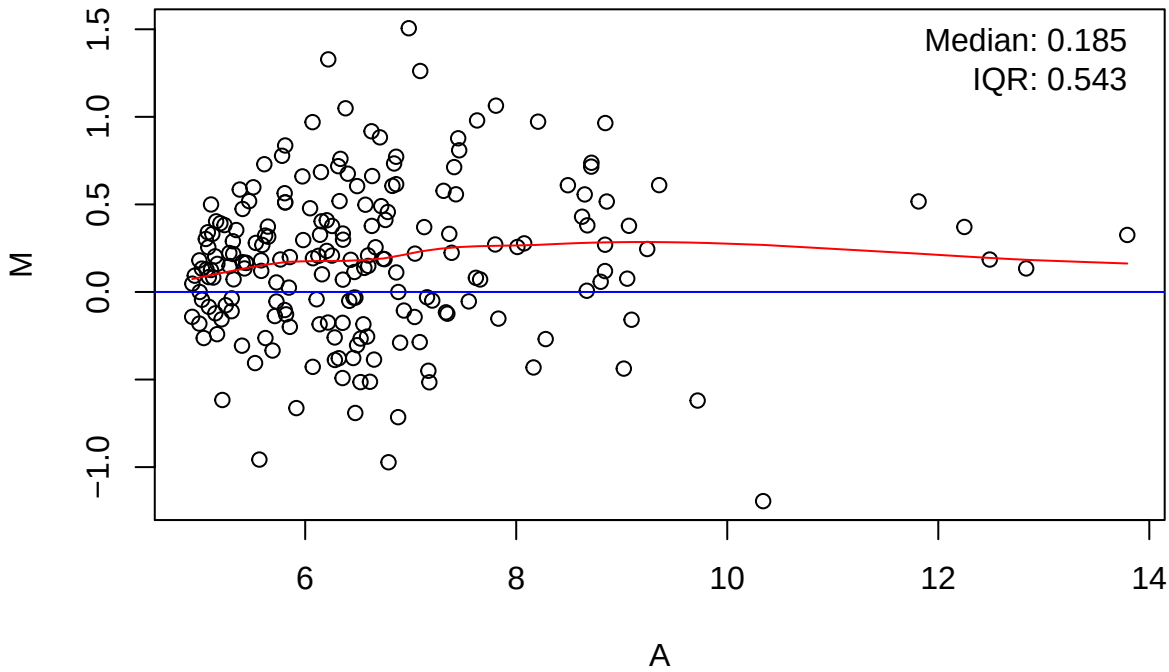
we plot huvec_retina

```
ggplot(sample_log,aes (huvec,retina)) + geom_point()
```



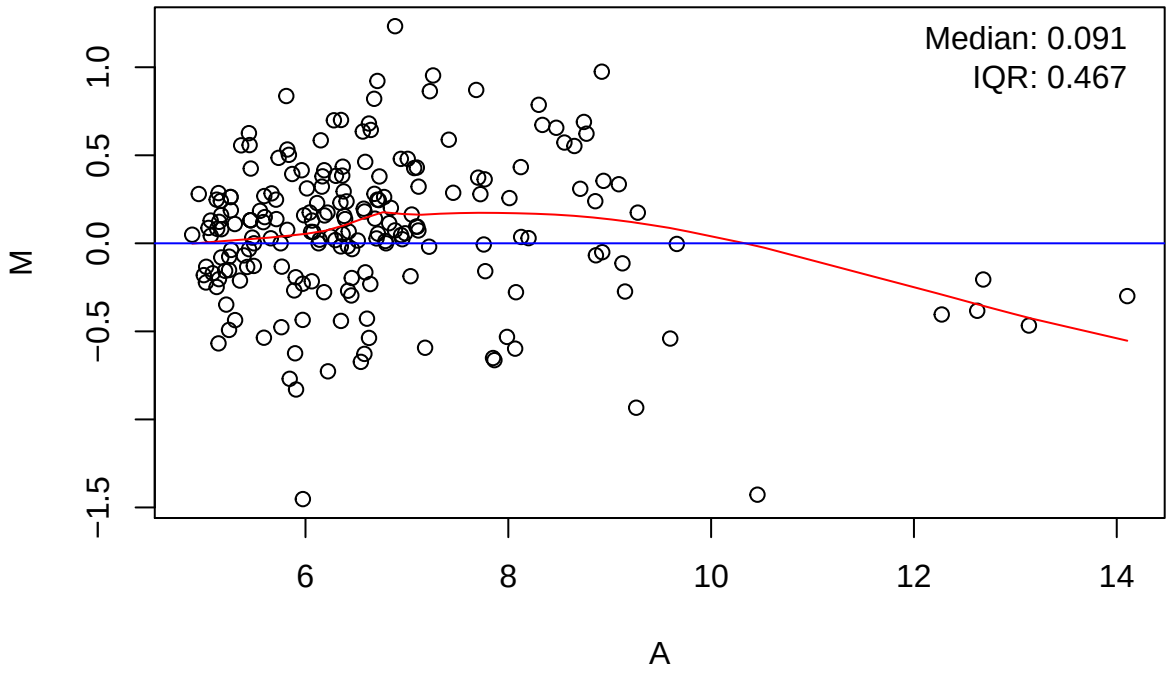
ma_plot

```
y1 <- (sample[, c("huvec", "choroid")])
ma.plot( rowMeans(log2(y1)), log2(y1[, 1])-log2(y1[, 2]), cex=1 )
```

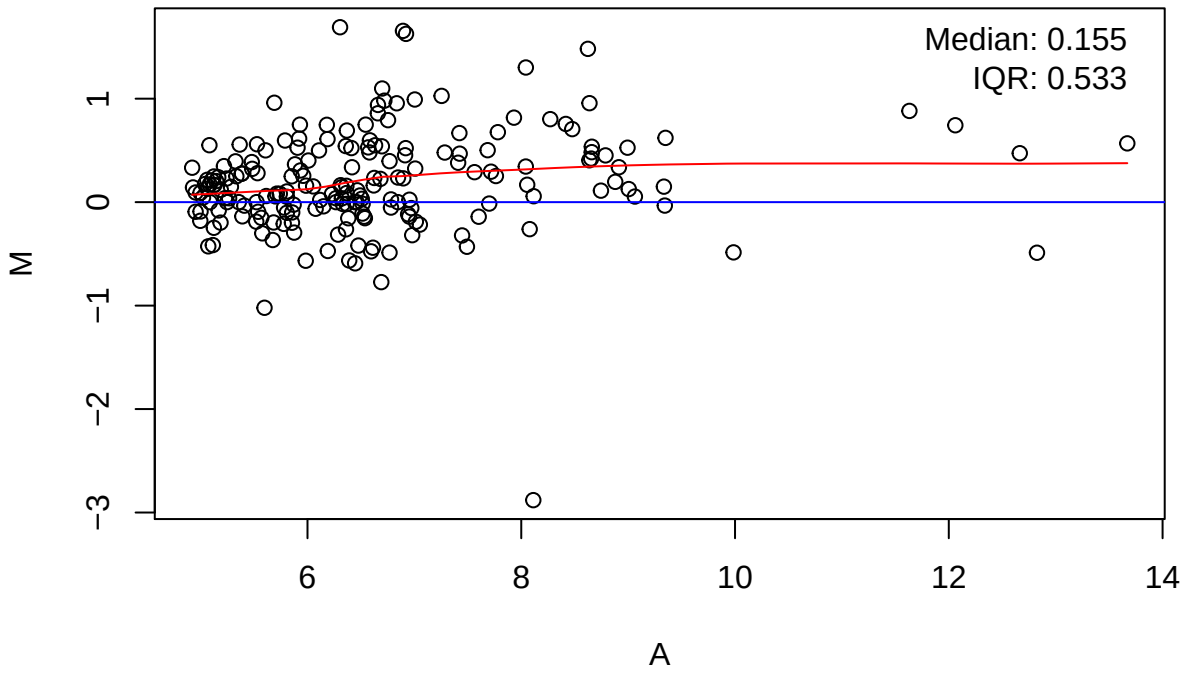



```
y2 <- (sample[, c("huvec", "retina")])
```

```
ma.plot( rowMeans(log2(y2)), log2(y2[, 1])-log2(y2[, 2]), cex=1 )
```

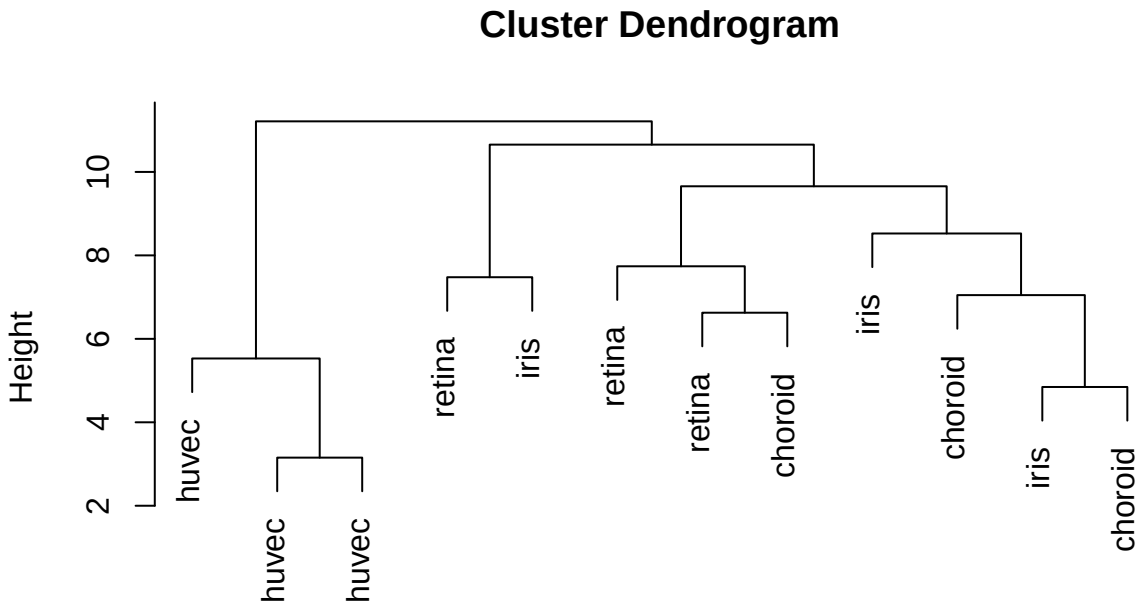



```
y3 <- (sample[, c("huvec", "iris")])
ma.plot( rowMeans(log2(y3)), log2(y3[, 1])-log2(y3[, 2]), cex=1 )
```



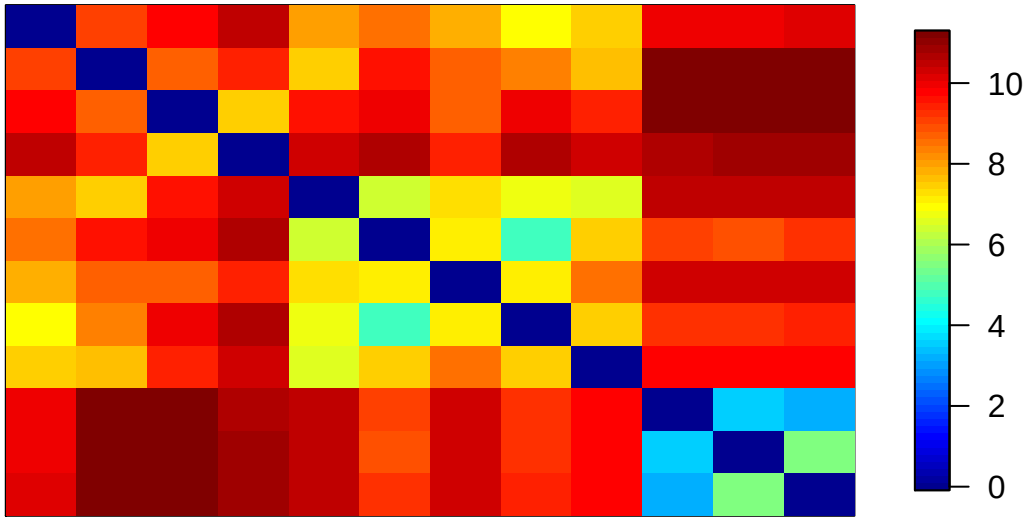
We plot the clusters, that were already analized in task 1.

```
plot(clusters)
```

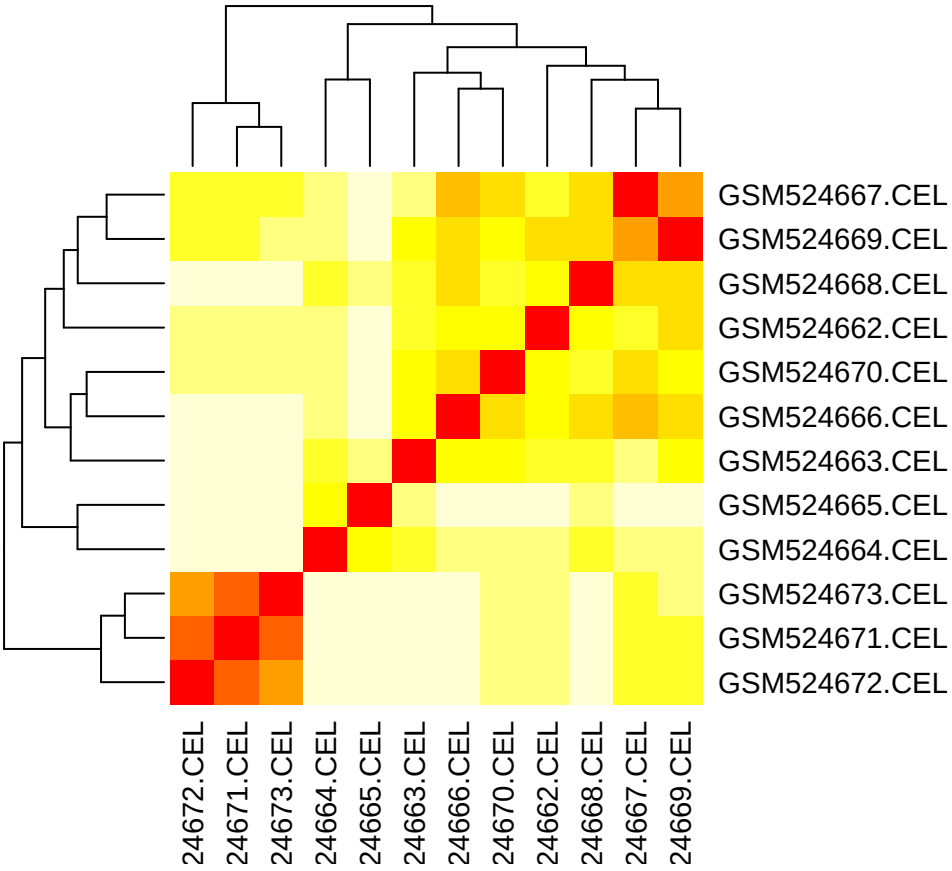


heatmap

```
library(plsgenomics)
distance_matrix = as.matrix(distance)
matrix.heatmap(distance_matrix)
```



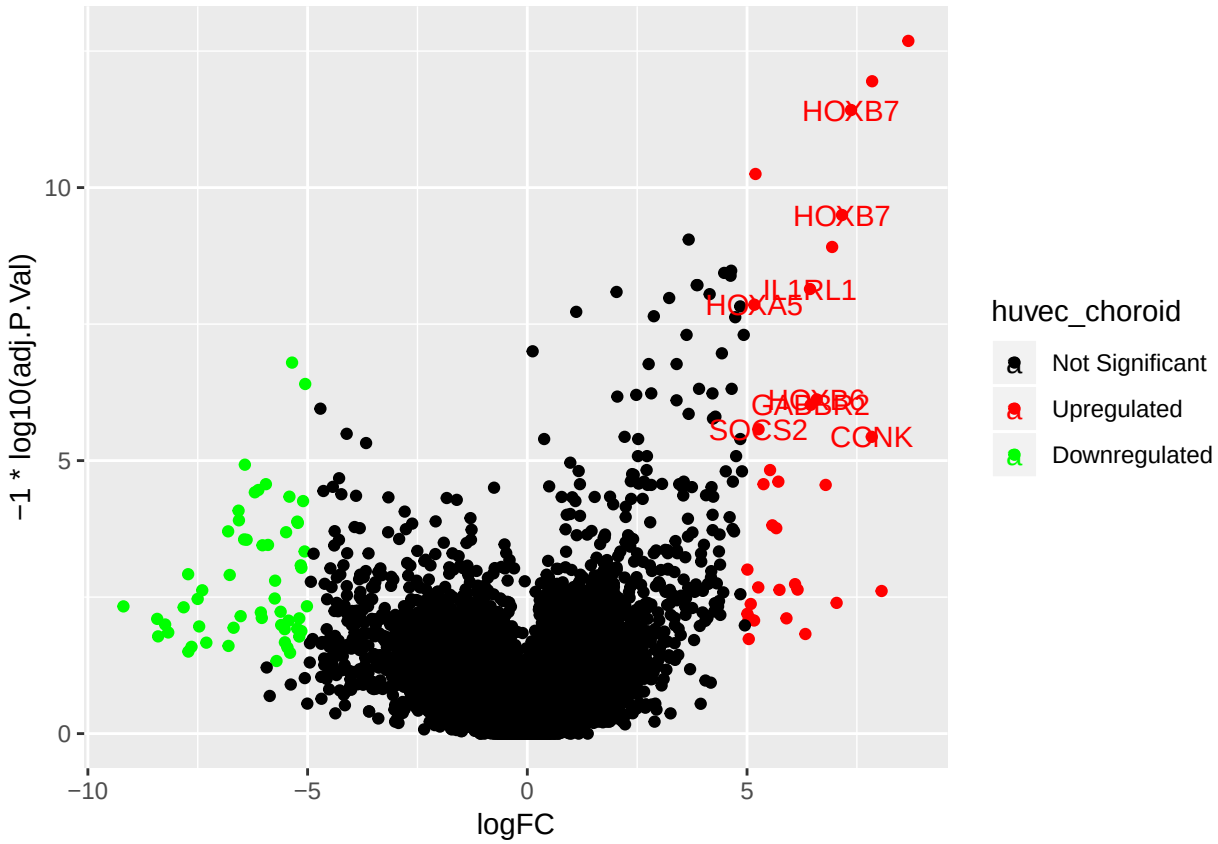

```
stats::heatmap(distance_matrix, Colv=F, scale='none')
```



Question 3

choroid

```
## Warning: Removed 4 rows containing missing values (geom_text).
```



We retrieve the genes:

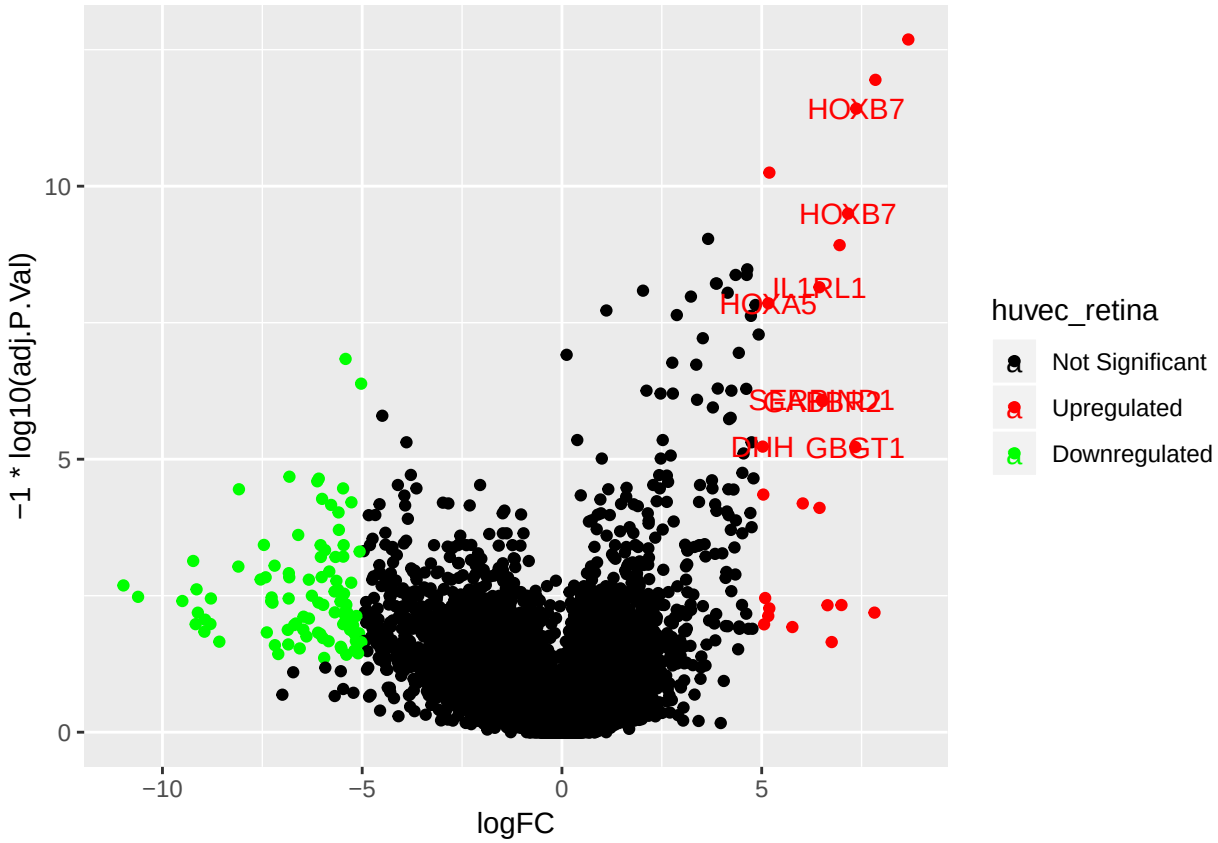
```
a<-subset(results, logFC > 5 & -1*log10(adj.P.Val) > 5)
b<-lapply(a$getsymbols, as.character)
choroid_u<-unique(b)
choroid_u
```

```
## [[1]]
## [1] NA
##
## [[2]]
## [1] "HOXB7"
##
## [[3]]
## [1] "IL1RL1"
##
## [[4]]
## [1] "HOXA5"
##
## [[5]]
## [1] "HOXB6"
##
## [[6]]
## [1] "GABBR2"
##
## [[7]]
## [1] "SOCS2"
##
```

```
## [[8]]
## [1] "CCNK"
```

retina

```
## Warning: Removed 4 rows containing missing values (geom_text).
```



We retrieve the genes:

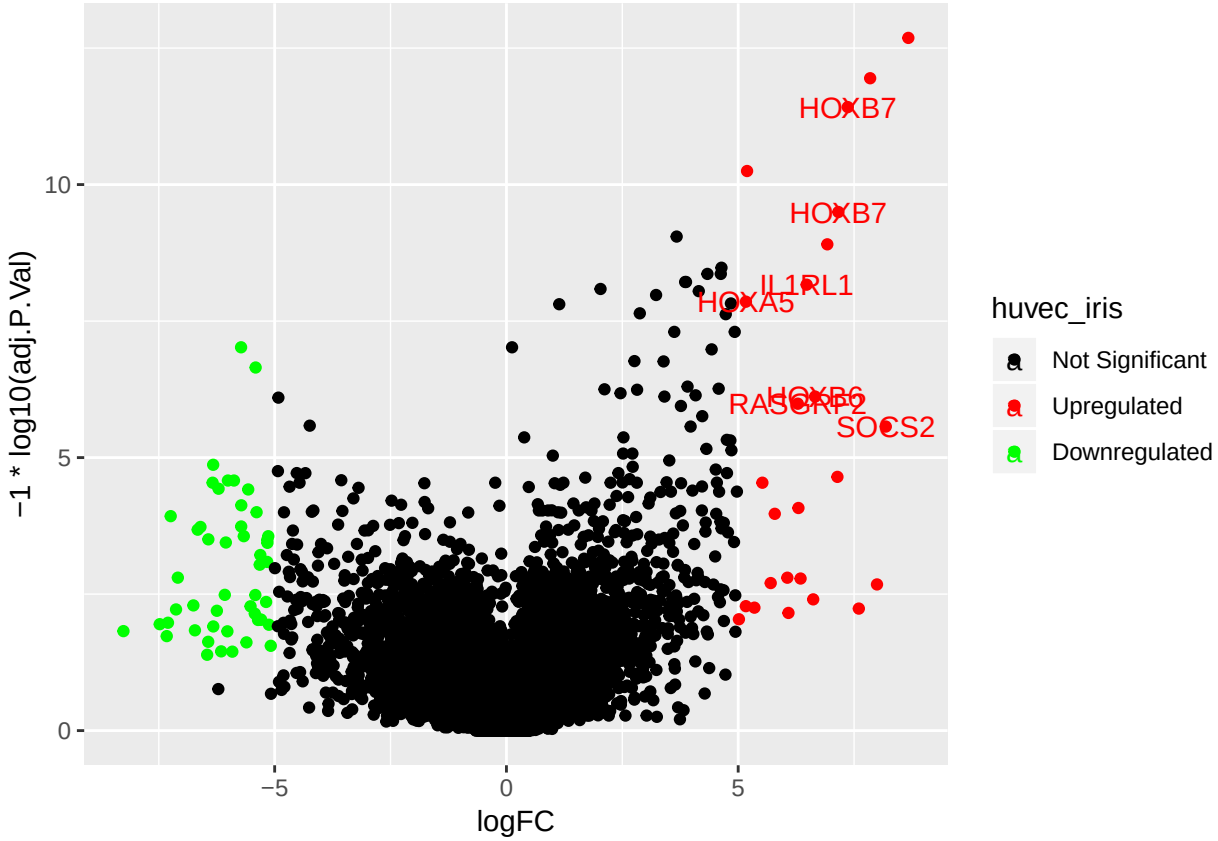
```
a<-subset(results, logFC > 5 & -1*log10(adj.P.Val) > 5)
b<-lapply(a$getsymbols, as.character)
retina_u <-unique(b)
retina_u
```

```
## [[1]]
## [1] NA
##
## [[2]]
## [1] "HOXB7"
##
## [[3]]
## [1] "IL1RL1"
##
## [[4]]
## [1] "HOXA5"
##
## [[5]]
## [1] "SERPIND1"
##
## [[6]]
```

```
## [1] "GABBR2"
##
## [[7]]
## [1] "DHH"
##
## [[8]]
## [1] "GBGT1"
```

iris

```
## Warning: Removed 4 rows containing missing values (geom_text).
```



We retrieve the genes:

```
a<-subset(results, logFC > 5 & -1*log10(adj.P.Val) > 5)
b<-lapply(a$getsymbols, as.character)
iris_u <-unique(b)
iris_u
```

```
## [[1]]
## [1] NA
##
## [[2]]
## [1] "HOXB7"
##
## [[3]]
## [1] "IL1RL1"
##
## [[4]]
## [1] "HOXA5"
```

```
##
## [[5]]
## [1] "HOXB6"
##
## [[6]]
## [1] "RASGRF2"
##
## [[7]]
## [1] "SOCS2"
```

Resulting from three vulcano plots, if certain genes occur more frequently in “Iris”, “Retina” and “Choroid”, these will appear as red dots. In the plot these are mentioned as significantly upregluated. To be able to arrive at this conclusion a model is fit. If signifiantly more genes occur in one sample they will come up as a red dot in the plots.

For each respective pair the following genes are considered to be significantly differentially expressed.

Huvec-Choroid: “HOXB7”, “IL1RL1”, “HOXA5”, “HOXB6”, “GABBR2”, “SOCS2”, “CCNK”

Huvec-Retina: “HOXB7”, “IL1RL1”, “HOXA5”, “SERPIND1”, “GABBR2”, “DHH”, “GBGT1”

Huvec-Iris: “HOXB7”, “IL1RL1”, “HOXA5”, “HOXB6”, “RASGRF2”, “SOCS2”

Question 4

HOXB7

Official name: homeobox B7

This gene is from the Antp homeobox family. It is part of a cluster of homeobox B genes which can be found in chromosome 17. If this gene is frequently present in can indicate presence of melanoma or ovarian carcinoma.

Gene Ontology (GC) terms:

GO_ID	Qualified_GO_Term
GO:0000978	RNA polymerase II proximal promoter sequence-specific DNA binding
GO:0000981	RNA polymerase II transcription factor activity, sequence-specific DNA binding
GO:0001077	transcriptional activator activity, RNA polymerase II proximal promoter sequence-specific DNA binding
GO:0003677	DNA binding
GO:0003700	DNA binding transcription factor activity

IL1RL1

Official name: interleukin 1 receptor like 1

The protein encoded by this gene is a part of the interleukin 1 family. The same gene has been studied in mouses and research suggested that it can be induced by proinflammatory stimuli, and may be important for the function of T cells.

Gene Ontology (GC) terms:

GO_ID	Qualified_GO_Term
GO:0002826	negative regulation of T-helper 1 type immune response
GO:0006955	immune response

GO_ID	Qualified_GO_Term
GO:0007165	signal transduction
GO:0019221	cytokine-mediated signaling pathway
GO:0032689	negative regulation of interferon-gamma production

HOXA5

Official name: homeobox A5

Proteins encoded from this gene are temporally regulated during embryonic development. Methylation of this gene can led to the loss of how frequently it is expressed. Because the encoded protein upregulates the tumor suppressor p53, this protein can fulfill an important role in tumorigenesis.

Gene Ontology (GC) terms:

GO_ID	Qualified_GO_Term
GO:0000978	RNA polymerase II proximal promoter sequence-specific DNA binding IDA 10879542
GO:0000981	RNA polymerase II transcription factor activity, sequence-specific DNA binding NAS 19274049
GO:0001077	transcriptional activator activity, RNA polymerase II proximal promoter sequence-specific DNA binding IDA
GO:0003677	DNA binding IDA 8657138
GO:0003700	DNA binding transcription factor activity

HOXB6

Official name: homeobox B6

The HOXB6 gene e is part of Antp homeobox family. The protein that is encoded from this gene is involved in the development of lungs and skin.

Gene Ontology (GC) terms:

GO_ID	Qualified_GO_Term
GO:0005634	nucleus

GABBR2

Official name: gamma-aminobutyric acid type B receptor subunit 2

The protein encoded from this gene is part of the G-protein coupled receptor 3 family and GABA-B receptor subfamily. The receptors influence the release of neurotransmitters.

Gene Ontology (GC) terms:

GO_ID	Qualified_GO_Term
GO:0004930	G-protein coupled receptor activity
GO:0004965	contributes_to G-protein coupled GABA receptor activity
GO:0005515	protein binding
GO:0046982	protein heterodimerization activity

SOCS2

Official name: suppressor of cytokine signaling 2

The protein that his gene encodes by this gene is involved in the insulin-like growth factor-1 receptor (IGF1R).

Gene Ontology (GC) terms:

GO_ID	Qualified_GO_Term
GO:0004860	protein kinase inhibitor activity
GO:0005070	SH3/SH2 adaptor activity
GO:0005131	growth hormone receptor binding
GO:0005159	insulin-like growth factor receptor binding
GO:0005515	protein binding

CCNK

Offical name: cyclin K

This gene encodes a protein that is part of the transcription cyclin family. This gene fulfills two roles in regulating CDK and RNA polymerase II activities.

Gene Ontology (GC) terms:

GO_ID	Qualified_GO_Term
GO:0004674	protein serine/threonine kinase activity
GO:0004693	cyclin-dependent protein serine/threonine kinase activity
GO:0005515	protein binding
GO:0008353	RNA polymerase II carboxy-terminal domain kinase activity
GO:0016538	cyclin-dependent protein serine/threonine kinase regulator activity

SERPIND1

Official name: serpin family D member 1

This gene is part of the serpin gene superfamily. Serpins are important during inflammation, blood clotting, and cancer metastasis.

Gene Ontology (GC) terms:

GO_ID	Qualified_GO_Term
GO:0004866	endopeptidase inhibitor activity
GO:0004867	serine-type endopeptidase inhibitor activity
GO:0008201	heparin binding
GO:0030414	peptidase inhibitor activity

DHH

Official name: desert hedgehog signaling molecule

This gene belongs to the hedgehog family. These genes are important during morphogenesis.

Gene Ontology (GC) terms:

GO_ID	Qualified_GO_Term
GO:0005113	patched binding
GO:0005509	calcium ion binding
GO:0005515	protein binding
GO:0008233	peptidase activity
GO:0008270	zinc ion binding

GBGT1

Official name: globoside alpha-1,3-N-acetylgalactosaminyltransferase 1

This gene encodes a glycosyltransferase which is important during the synthesis of Forssman glycolipid. Great expressions of this gene can create host tropism to microorganisms.

Gene Ontology (GC) terms:

GO_ID	Qualified_GO_Term
GO:0016740	transferase activity
GO:0016757	transferase activity, transferring glycosyl groups
GO:0016758	transferase activity, transferring hexosyl groups
GO:0046872	metal ion binding
GO:0047277	globoside alpha-N-acetylgalactosaminyltransferase activity

RASGRF2

Official name: Ras protein specific guanine nucleotide releasing factor 2

The RASGRF2 gene encodes a nucleotide which which activates RAS and RAS related proteins.

Gene Ontology (GC) terms:

GO_ID	Qualified_GO_Term
GO:0005085	guanyl-nucleotide exchange factor activity
GO:0005088	Ras guanyl-nucleotide exchange factor activity
GO:0005089	Rho guanyl-nucleotide exchange factor activity
GO:0005515	protein binding
GO:0005516	calmodulin binding