mda14jx Final Project - Effects of Hypoxia on human Neutrophils

December 4, 2017

```
In [1]: options(jupyter.plot_mimetypes ='image/png')
```

1 ** THE EFFECT OF HYPOXIA ON HUMAN NEUTROPHILS **

In this study, the effects of Hypoxia on human Neutrophils were investigated in order to identify the possible involvement of inflammatory response in adverse prognosis of hypoxia-related disease, such as pulmonary hypertension and myocardial infarction. Primary cultures of human neutrophils were studied in both normal and hypoxia conditions. A gene expression profile of the neutrophils in both conditions were done after centain amounts of time in culture, and quantified using Affymetrix GeneChip HGU133 PLUS 2. The study was conducted on two separate samples.

This report aims to estimate gene expression levels, and analyse the results to identify the genes that are changing between the two conditions, defining the potential pathways that hypoxia may have altered in neutrophils.

1.1 ** Data Analysis **

Workflow Step 1: Load packages with data from Bioconductor, library(affy) - mas5, rma, library(puma)

Step 2: Load and read data, create affybatch. Annotate with pData.

Step 3: Analysis of gene expression data with different methods and normalisation techniques. - Create eset - Extract gene expression - First diagnostic using density() and boxplot() - Normalisation by log2 if required

Step 4: Diagnostics of the data with plotting techniques - MAPlot - boxplot

Step 5: Differential Expression Analysis - For `puma`, combine the data using an bayesian Hierarchical model - Check the dimension and the `pData()` for the eset of the combined values. Calculate the FC and plot the data with a MA plot using the command ma.plot()
- MAPlot - use of `limma` for DE analysis. Remember the three core steps of `limma` * **Step 1**: build the design contrast matrix * **Step 2**: fit the linear model * **Step 3**: calculate the p-values and FDRs with a empirical Bayes test

Step 6: Visualisation of Data with PCA - perform PCA in R using the command `prcomp()` - It needs the traspose command `t()` since the input for the `prcomp()` wants the genes in the columns - For probabilistic PCA you can use `pumaPCA()`

Step 7: Hierarchical clustering of DE (Differentially Expressed) genes - To perform this we need to activate a library called `gplots`. We will use the command `heatmap.2()`. - We do clustering a the selected genes from our DE analysis this is to search for patterns in of differentially regulatend pathways.

Step 8: Functional/Pathway analysis of DE targets using PANTHER or DAVID

** FEEDBACK: well organised workflow. You need to add more details for the throsholds used and teh parameters of the analysis. This would make the analysis reproducible **

1.1.1 Step 1:

```
In [2]: library(affy)
```

```
Loading required package: BiocGenerics
Loading required package: parallel

Attaching package: BiocGenerics

The following objects are masked from package:parallel:

    clusterApply, clusterApplyLB, clusterCall, clusterEvalQ,
    clusterExport, clusterMap, parApply, parCapply, parLapply,
    parLapplyLB, parRapply, parSapply, parSapplyLB

The following objects are masked from package:stats:

    IQR, mad, xtabs

The following objects are masked from package:base:

    anyDuplicated, append, as.data.frame, as.vector, cbind, colnames,
    do.call, duplicated, eval, evalq, Filter, Find, get, grep, grepl,
    intersect, is.unsorted, lapply, lengths, Map, mapply, match, mget,
    order, paste, pmax, pmax.int, pmin, pmin.int, Position, rank,
    rbind, Reduce, rownames, sapply, setdiff, sort, table, tapply,
    union, unique, unlist, unsplit

Loading required package: Biobase
Welcome to Bioconductor

    Vignettes contain introductory material; view with
    'browseVignettes()'. To cite Bioconductor, see
    'citation("Biobase")', and for packages 'citation("pkgname")'.
```

This step loads the *affy* package, which is part of the BioConductor project, allowing for data analysis and exploration of Affymetrix oligonucleotide array probe level data. It summarises the probe set intensities, forming one expression measure (data available for analysis) for each gene. The package includes plotting functions for the probe level data useful for quality control, making it useful in the initial analysis of the data, it includes plotting functions for the data that can be useful for quality control of data, RNA degradation assessments, normaliasation and background correction procedures. It also allows for probe level data to be converted to expression measures. In this project, MAS 5.0 and RMA are used for perform the analysis.

1.1.2 Step 2:

Set working directory

```
In [3]: setwd("~/Autumn2016/ProjectC/data_projectC")
```

```
In [4]: getwd()
```

In order to load the data that is required, a working directory must be set, leading to where the data is saved.

```
In [5]: hypoxia_filenames <- c("LPGMa.CEL","LPGMb.CEL","LPHa.CEL","LPHb.CEL")
        affybatch.hypoxia <- ReadAffy(filenames=hypoxia_filenames)
```

The files that contain the data are saved in the .CEL format, indicating the files contain measured intensities and locations for an array that has been hybridised.

```
In [6]: show(affybatch.hypoxia)
```

```
Warning message:
replacing previous import AnnotationDbi::tail by utils::tail when loading hgu133plus2cdfWarning
replacing previous import AnnotationDbi::head by utils::head when loading hgu133plus2cdf


AffyBatch object
size of arrays=1164x1164 features (18 kb)
cdf=HG-U133_Plus_2 (54675 affyids)
number of samples=4
number of genes=54675
annotation=hgu133plus2
notes=
```

The data shows the size of the array is 1154x1164 (18kb), cdf maps each gene that is in the array (54675 genes), and there are 4 samples.

```
In [7]: phenoData(affybatch.hypoxia)
        pData(affybatch.hypoxia)
```

```
Out[7]: An object of class 'AnnotatedDataFrame'
          sampleNames: LPGMa.CEL LPGMb.CEL LPHa.CEL LPHb.CEL
          varLabels: sample
          varMetadata: labelDescription
```

pData retrieves information on experimental phenotypes that are recorded.

```
In [8]: pData(affybatch.hypoxia)<- data.frame(
            "Condition"=c("Normal", "Normal", "Hypoxia", "Hypoxia"),
            "Sample"=c("1", "2", "1", "2"),
            row.names=rownames(pData(affybatch.hypoxia)))
        pData(affybatch.hypoxia)
```

1.1.3 Step 3:

This step involves the analysis of gene expression data with different methods and normalisation techniques. The methods convert the probe level data to expression values, which is achieved through: * Reading in probe level data * Background correction * Normalization * Probe specific background correction * Summarising the probe set values into one expression measure

RMA and MAS 5.0 creates two different types of ExpressionSets, from which the gene expression values will be extracted.

```
In [9]: eset_rma<-rma(affybatch.hypoxia)
        show(eset_rma)
```

```
Background correcting
Normalizing
Calculating Expression
ExpressionSet (storageMode: lockedEnvironment)
assayData: 54675 features, 4 samples
  element names: exprs
protocolData
  sampleNames: LPGMa.CEL LPGMb.CEL LPHa.CEL LPHb.CEL
  varLabels: ScanDate
  varMetadata: labelDescription
phenoData
  sampleNames: LPGMa.CEL LPGMb.CEL LPHa.CEL LPHb.CEL
  varLabels: Condition Sample
  varMetadata: labelDescription
featureData: none
experimentData: use 'experimentData(object)'
Annotation: hgu133plus2
```

```
In [10]: eset_mas5<-mas5(affybatch.hypoxia)
         show(eset_mas5)
```

```
background correction: mas
PM/MM correction : mas
expression values: mas
background correcting...done.
54675 ids to be processed
|                    |
|####################|
ExpressionSet (storageMode: lockedEnvironment)
assayData: 54675 features, 4 samples
  element names: exprs, se.exprs
protocolData
  sampleNames: LPGMa.CEL LPGMb.CEL LPHa.CEL LPHb.CEL
  varLabels: ScanDate
  varMetadata: labelDescription
phenoData
```

```
  sampleNames: LPGMa.CEL LPGMb.CEL LPHa.CEL LPHb.CEL
  varLabels: Condition Sample
  varMetadata: labelDescription
featureData: none
experimentData: use 'experimentData(object)'
Annotation: hgu133plus2
```

```
In [11]: e_rma<-exprs(eset_rma)
         head(e_rma)
```

```
In [12]: e_mas5<-exprs(eset_mas5)
         head(e_mas5)
```

** FEEDBACK: this is very clear well done.**

```
In [13]: density(e_rma)
         density(e_mas5)
```

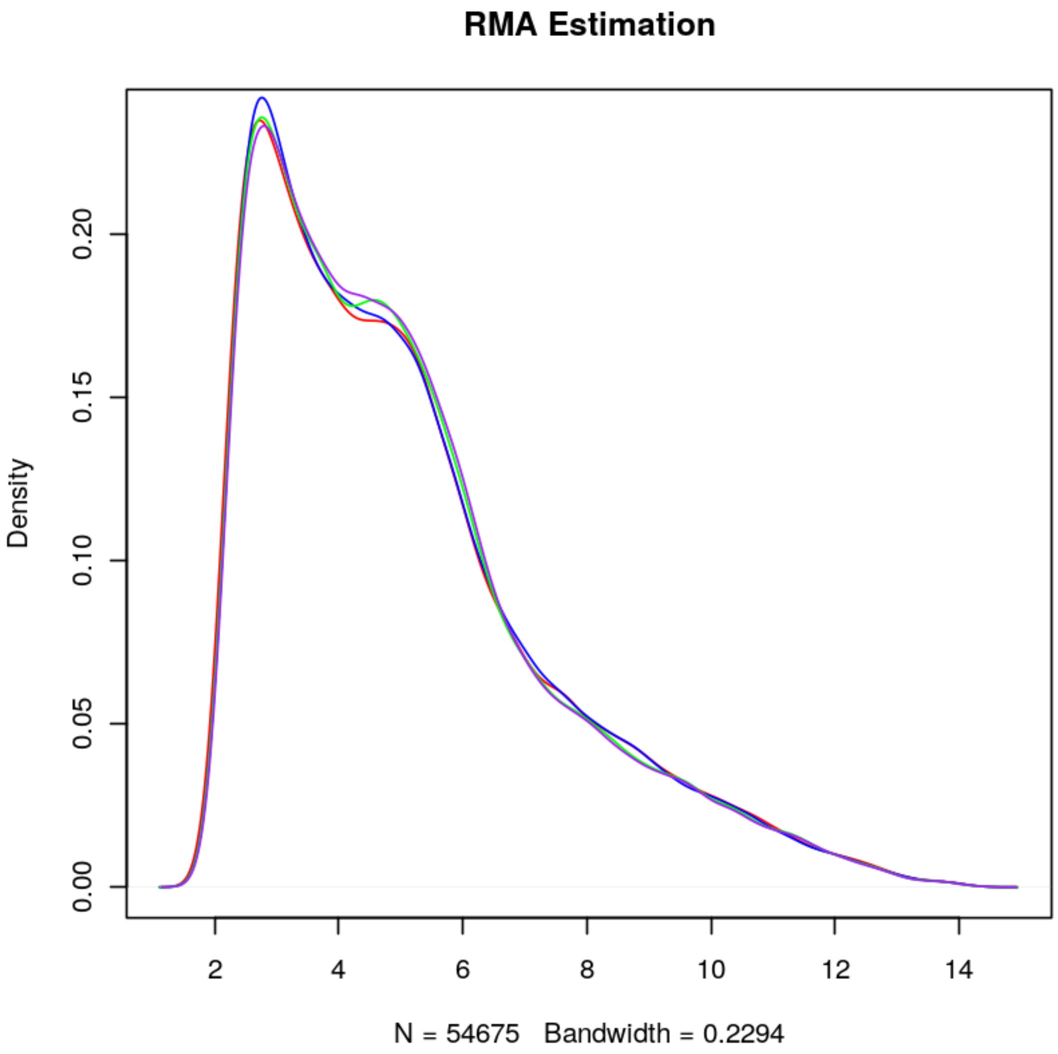
```
Out[13]:
         Call:
                 density.default(x = e_rma)

         Data: e_rma (218700 obs.);        Bandwidth 'bw' = 0.1711

                x                 y
          Min.   : 1.276   Min.   :8.300e-07
          1st Qu.: 4.654   1st Qu.:1.038e-02
          Median : 8.031   Median :4.186e-02
          Mean   : 8.031   Mean   :7.396e-02
          3rd Qu.:11.408   3rd Qu.:1.444e-01
          Max.   :14.785   Max.   :2.408e-01
```

```
Out[13]:
         Call:
                 density.default(x = e_mas5)

         Data: e_mas5 (218700 obs.);        Bandwidth 'bw' = 14.33

                x                   y
          Min.   :  -42.87   Min.   :0.000e+00
          1st Qu.:16716.38   1st Qu.:2.380e-07
          Median :33475.62   Median :7.530e-07
          Mean   :33475.62   Mean   :5.338e-05
          3rd Qu.:50234.86   3rd Qu.:3.687e-06
          Max.   :66994.10   Max.   :1.014e-02
```

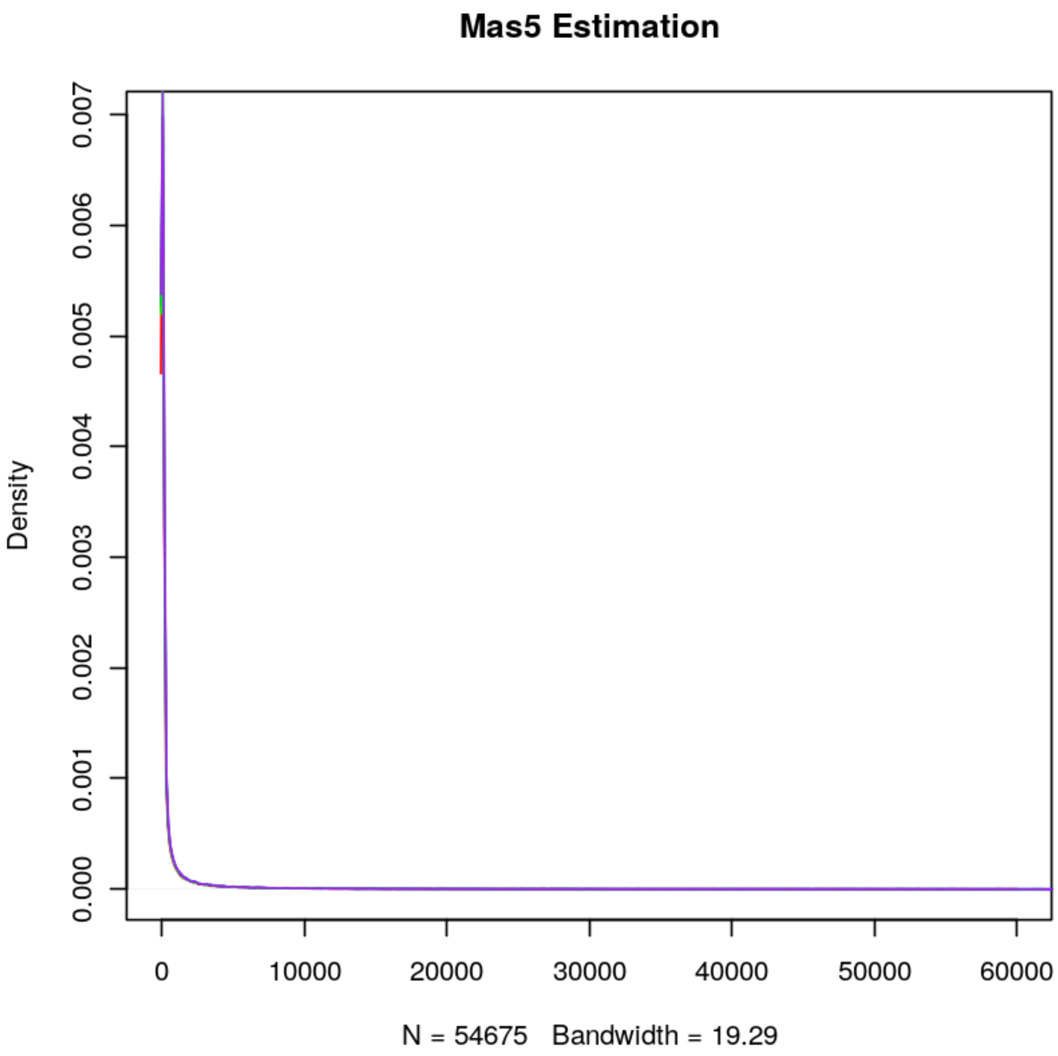
```
In [14]: par(mfrow=c(1,1))
         plot(density(e_rma[,1]),col="red", main="RMA Estimation")
```

```
lines(density(e_rma[,2]),col="blue")
lines(density(e_rma[,3]),col="green")
lines(density(e_rma[,4]),col="purple")
plot(density(e_mas5[,1]),col="red",main="Mas5 Estimation")
lines(density(e_mas5[,2]),col="blue")
lines(density(e_mas5[,3]),col="green")
lines(density(e_mas5[,4]),col="purple")
```

Out[14]:

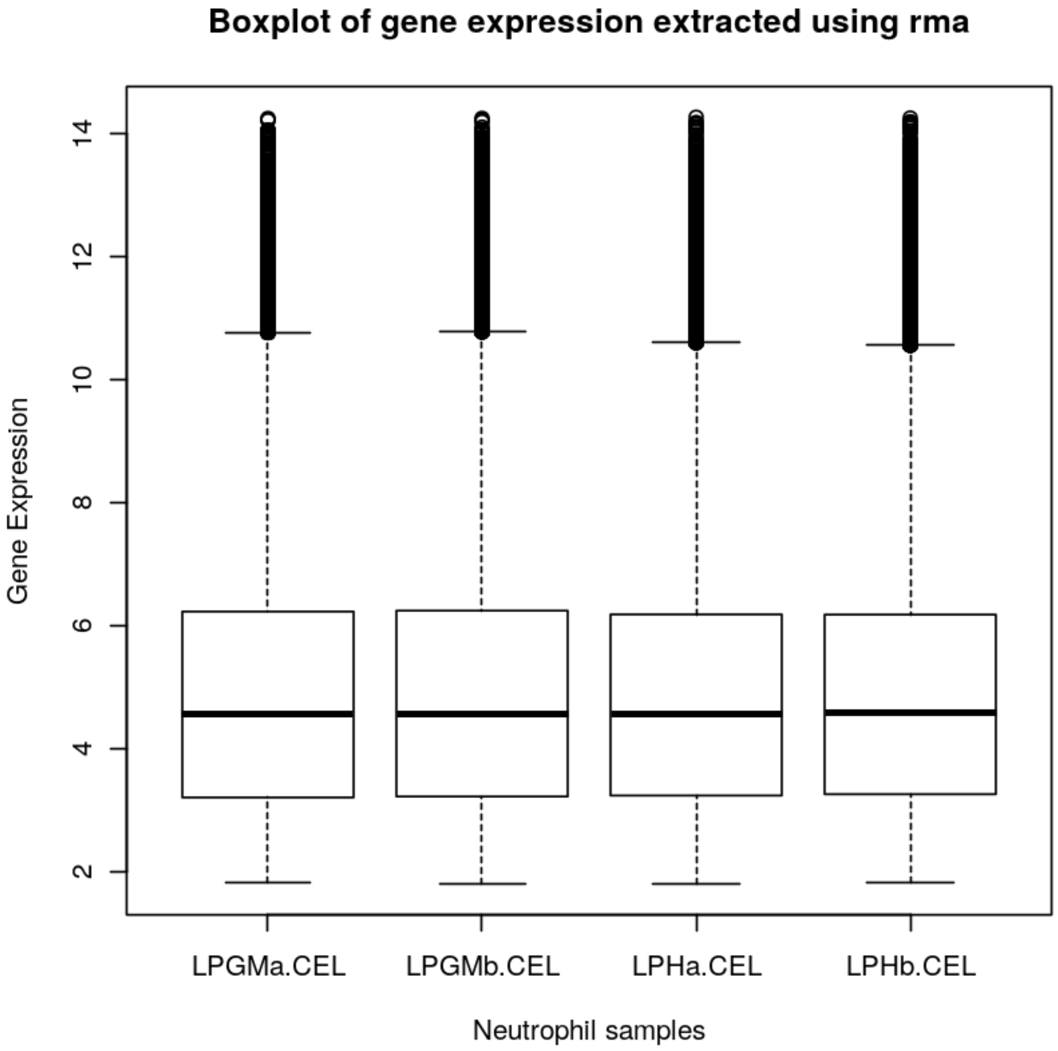


Out[14]:

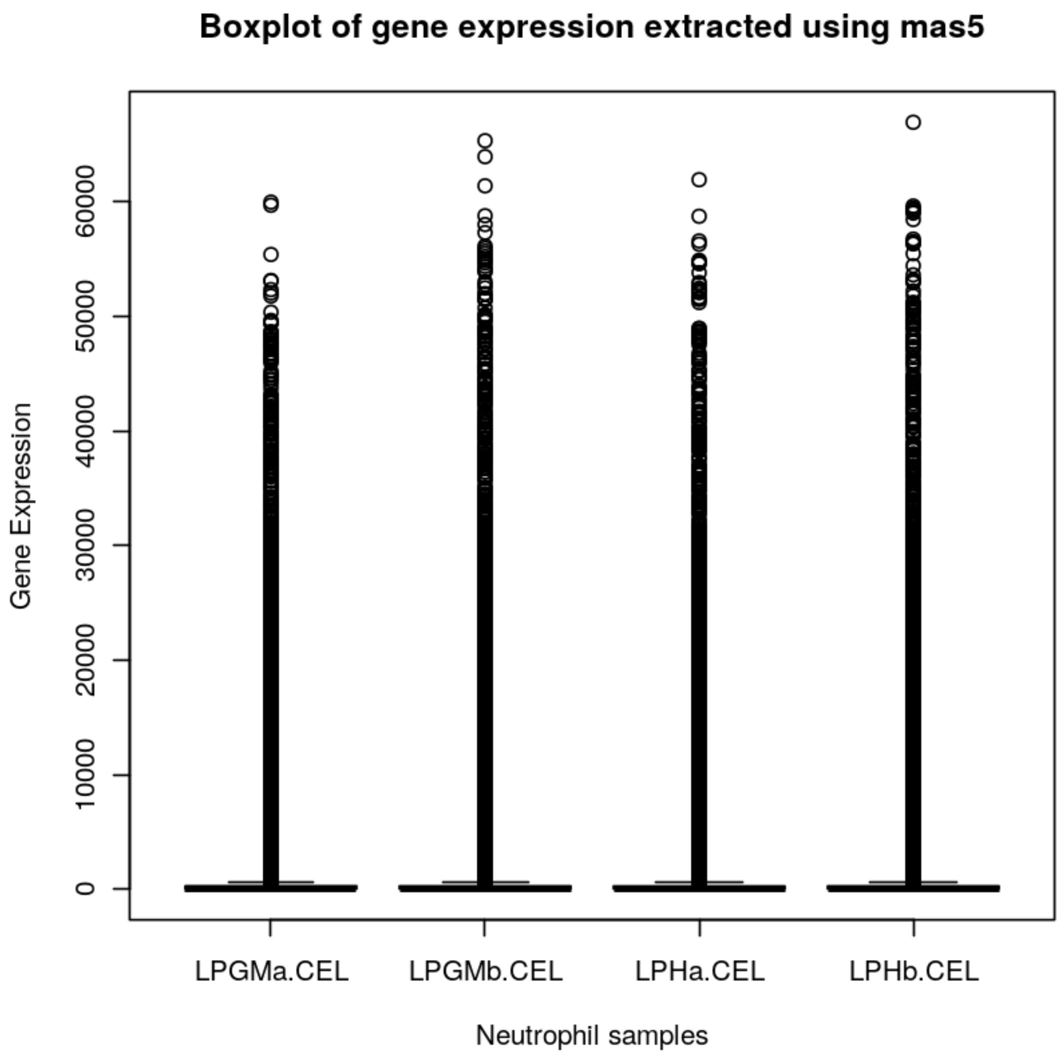


** FEEDBACK: the problem you are having with the above boxplot is beacuse you have not logged the data. **

```
In [15]: par(mfrow=c(1,1))
         boxplot((e_rma), xlab="Neutrophil samples", ylab="Gene Expression", main="Boxplot of ge
         boxplot((e_mas5), xlab="Neutrophil samples", ylab="Gene Expression", main="Boxplot of g

   Out[15]:
```


Boxplot of gene expression extracted using rma
Gene Expression
2
4
6
8
10
12
14
LPGMa.CEL
LPGMb.CEL
LPHa.CEL
LPHb.CEL
Neutrophil samples

```
Out[15]:
```


Boxplot of gene expression extracted using mas5

```
In [16]: log2e_mas5<-log2(e_mas5)
         head(log2e_mas5)
```

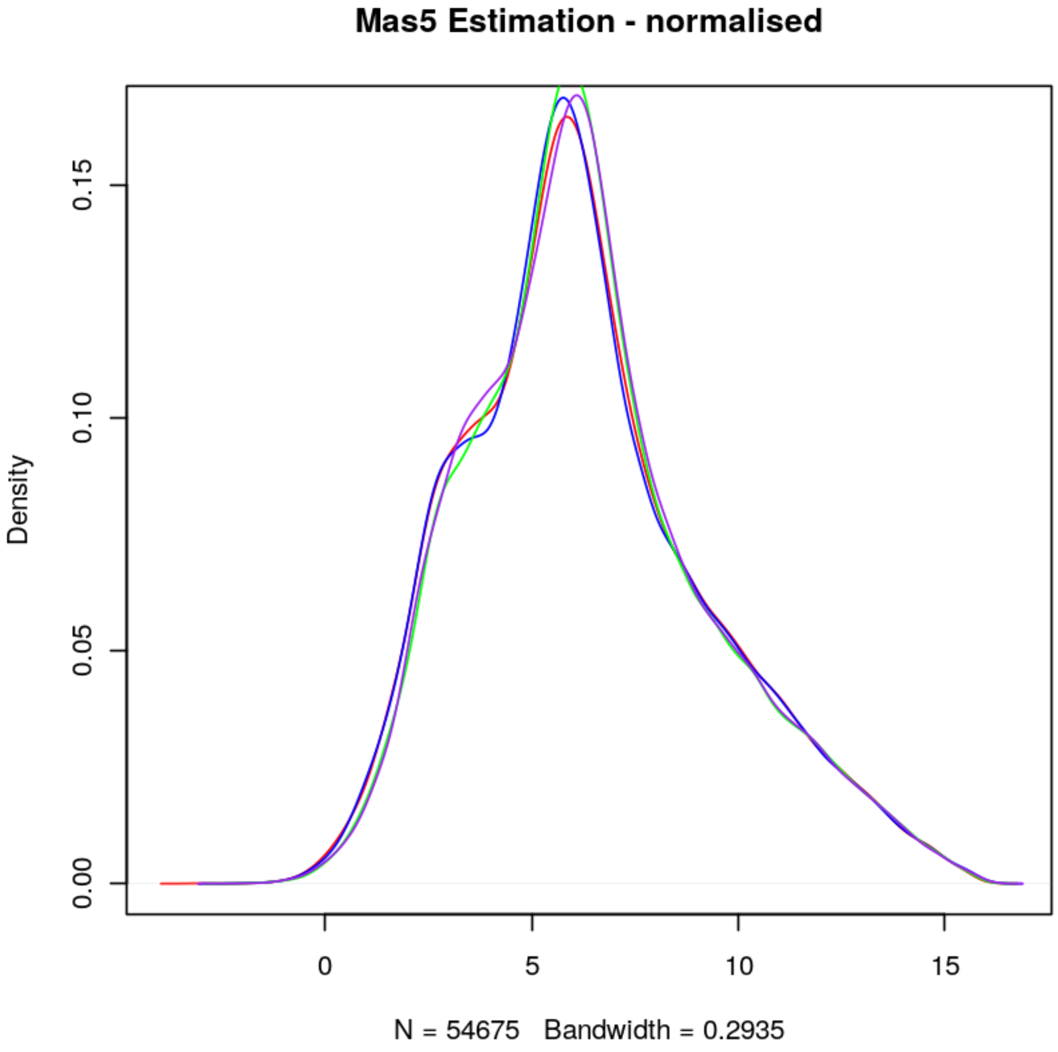
```
In [17]: density(log2e_mas5)
```

```
Out[17]:
         Call:
                 density.default(x = log2e_mas5)

         Data: log2e_mas5 (218700 obs.);        Bandwidth 'bw' = 0.2168


                x                 y
          Min.   :-3.749   Min.   :1.800e-07
```

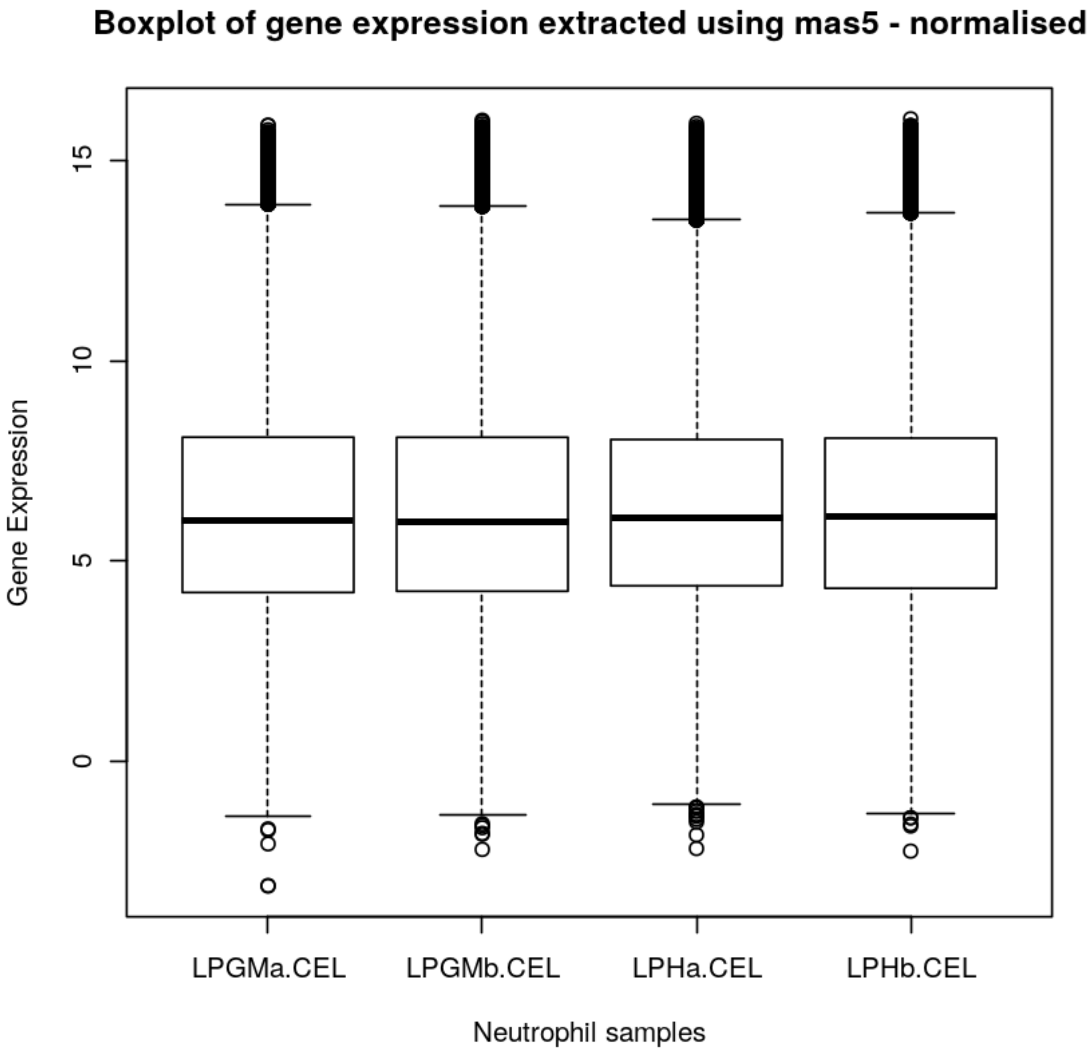
```
 1st Qu.: 1.358   1st Qu.:4.223e-03
 Median : 6.466   Median :3.180e-02
 Mean   : 6.466   Mean   :4.890e-02
 3rd Qu.:11.574   3rd Qu.:8.519e-02
 Max.   :16.681   Max.   :1.705e-01
```

```
In [18]: par(mfrow=c(1,1))
         plot(density(log2e_mas5[,1]),col="red",main="Mas5 Estimation - normalised")
         lines(density(log2e_mas5[,2]),col="blue")
         lines(density(log2e_mas5[,3]),col="green")
         lines(density(log2e_mas5[,4]),col="purple")
         boxplot((log2e_mas5), xlab="Neutrophil samples", ylab="Gene Expression", main="Boxplot
```

```
Out[18]:
```



Out[18]:



Expression values for mas5 and rma are extracted and the first diagnostics performed on the data, using density() and boxplot(). Initial mas5 estimation showed the data was difficult to read due to the large values of the outliers, therefore a log2 transformation was performed, to change the scale and make the plots more readable. The transformation also eliminated much of the negative values. No transformation or normalisation was required however, as the medians are aligned, with no negative outliers. Therefore, further analysis is continued with the use of rma extracted expressions.

```
In [19]: require(puma)
```

```
Loading required package: puma
Loading required package: oligo
Loading required package: oligoClasses
Welcome to oligoClasses version 1.32.0

Attaching package: oligoClasses

The following object is masked from package:affy:

    list.celfiles

Loading required package: Biostrings
Loading required package: S4Vectors
Loading required package: stats4
Loading required package: IRanges
Loading required package: XVector
================================================================================
Welcome to oligo version 1.34.2
================================================================================

Attaching package: oligo

The following objects are masked from package:affy:

    intensity, MAplot, mm, mm<-, mmindex, pm, pm<-, pmindex,
    probeNames, rma

Loading required package: mclust
Package 'mclust' version 5.2
Type 'citation("mclust")' for citing this R package in publications.
```

puma (Propagating Uncertainty in Microarray Analysis) is another bioconductor package. Microarrays measure the expression level of thousands of genes simultaneously, therefore there are many significant soutces of uncertainties associated with it; these uncertainties must be considered to accurately infer from the data. Earlier methods used (mas5 and rma) only provide single point estimates that summarises the target concentration. By using probabilistic models such as *puma* for probe-level analysis, it is possible to associate gene expression levels with credibility intervals that quantify the measurement uncentainty associated with the estimate of target concentration with a sample. *puma* performs analysis through: * Calculation of expression levels and confidence measures for those levels from raw .CEL data * Combine uncertainty information from replicated arrays * Determine differential expression between conditions, or between more complex contrasts such as interaction terms * Cluster data taking the expression level uncertainty into account * Perform a noise-propagation version of principal compinent analysis (PCA)

```
In [20]: eset_puma<-mmgmos(affybatch.hypoxia)
         show(eset_puma)
```

```
Model optimising ...
Expression values calculating ...
```

```
Done.
Expression Set (exprReslt) with
        54675 genes
        4 samples
        An object of class 'AnnotatedDataFrame'
  sampleNames: LPGMa.CEL LPGMb.CEL LPHa.CEL LPHb.CEL
  varLabels: Condition Sample
  varMetadata: labelDescription
```

```
In [21]: eset_puma_normd <-pumaNormalize(eset_puma)
```

```
In [22]: e_puma<-exprs(eset_puma)
         head(e_puma)
```

```
In [23]: density(e_puma)
```

```
Out[23]:
         Call:
                 density.default(x = e_puma)

         Data: e_puma (218700 obs.);        Bandwidth 'bw' = 0.2729

                x                 y
          Min.   :-35.061   Min.   :0.000e+00
          1st Qu.:-22.699   1st Qu.:2.170e-06
          Median :-10.336   Median :2.474e-05
          Mean   :-10.336   Mean   :2.020e-02
          3rd Qu.:  2.026   3rd Qu.:2.640e-02
          Max.   : 14.388   Max.   :1.145e-01
```

```
In [24]: e_puma_normd<-exprs(eset_puma_normd)
         head(e_puma_normd)
```

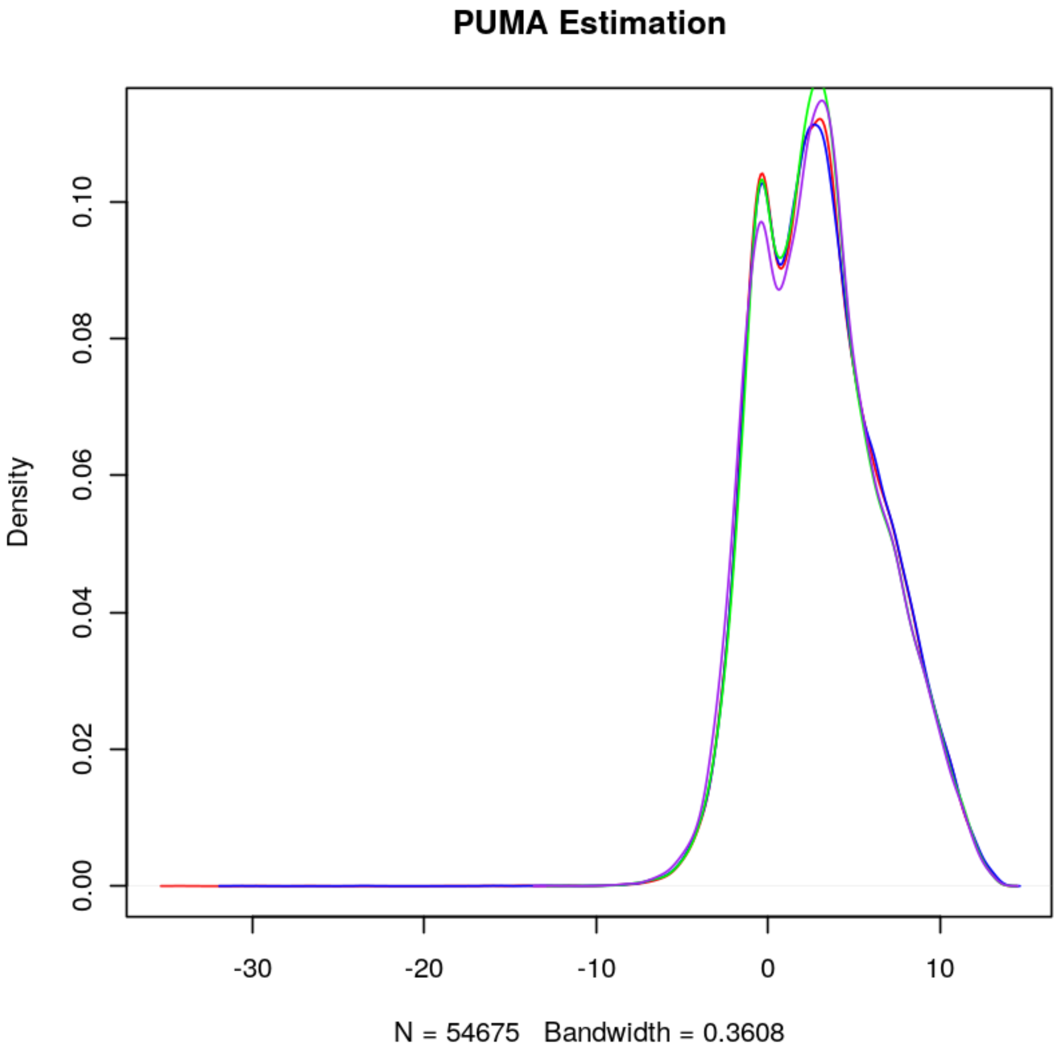
```
In [25]: density(e_puma_normd)
```

```
Out[25]:
         Call:
                 density.default(x = e_puma_normd)

         Data: e_puma_normd (218700 obs.);        Bandwidth 'bw' = 0.2729

                x                 y
          Min.   :-35.061   Min.   :0.000e+00
          1st Qu.:-22.699   1st Qu.:2.170e-06
          Median :-10.336   Median :2.474e-05
          Mean   :-10.336   Mean   :2.020e-02
          3rd Qu.:  2.026   3rd Qu.:2.640e-02
          Max.   : 14.388   Max.   :1.145e-01
```

After performing pumaNormalize() on the data, the first diagnostic tests showed that there is no difference to the data prior to normalisation, therefore indicating that the pumadata is already normalised.

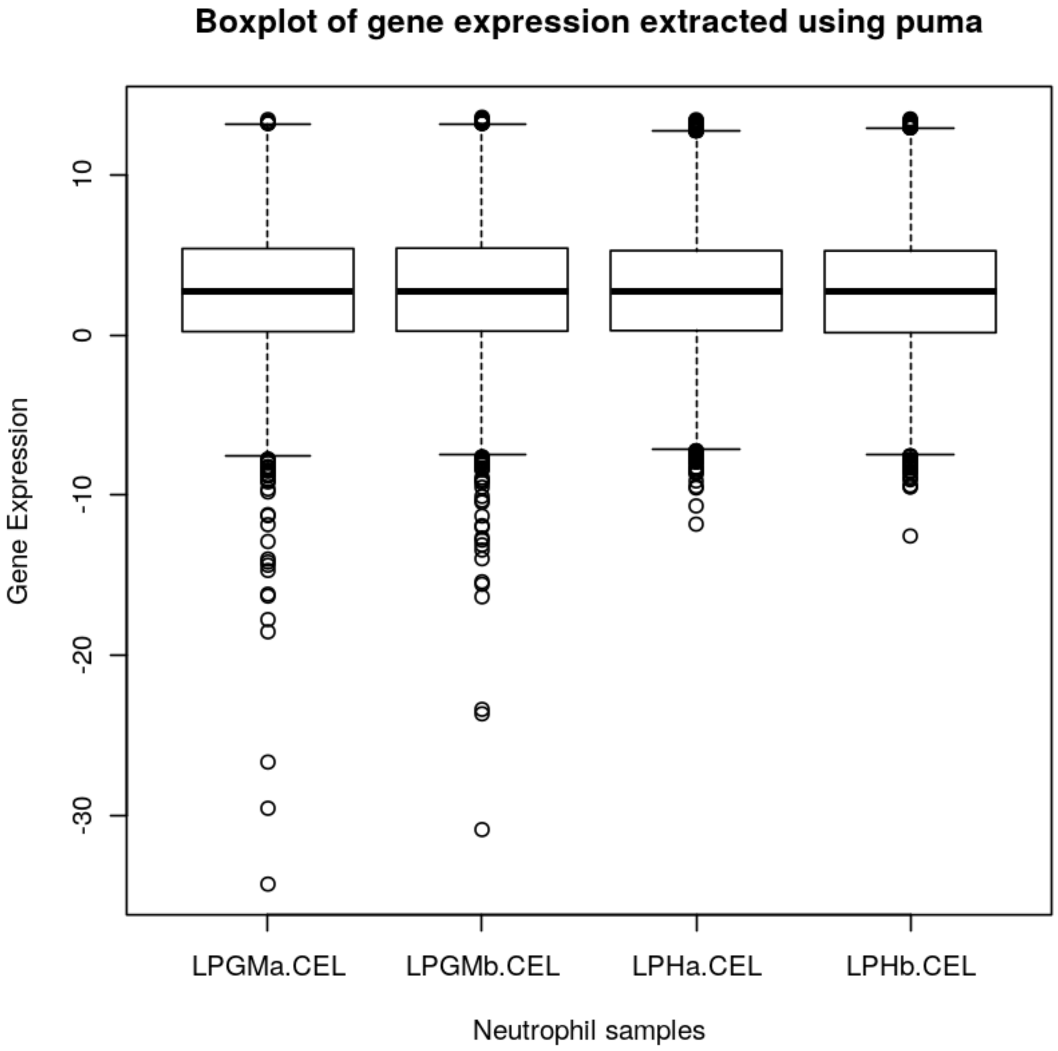
```
In [26]: plot(density(e_puma[,1]),col="red", main="PUMA Estimation")
         lines(density(e_puma[,2]),col="blue")
         lines(density(e_puma[,3]),col="green")
         lines(density(e_puma[,4]),col="purple")
```

```
Out[26]:
```



```
In [27]: boxplot((e_puma), xlab="Neutrophil samples", ylab="Gene Expression", main="Boxplot of g
```

```
Out[27]:
```



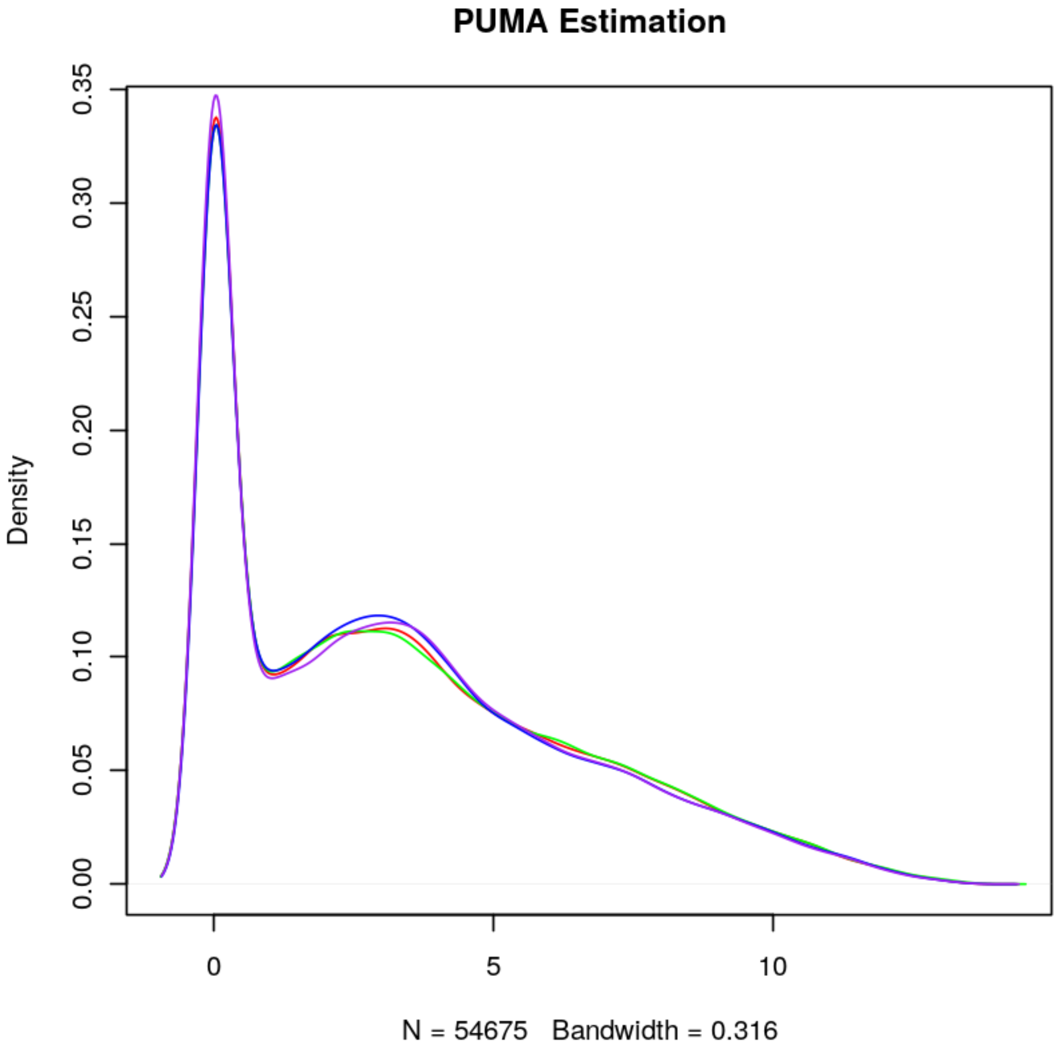
Although the data is shown to be normalized, and the medians are aligned, it can also be seem from the boxplot that there is a large number of negative outliers, therefore the negative gene expression values are set to zero, to further normalise the data.

```
In [28]: for (i in 1:4) {
             y<-e_puma[,i]
             y[y<0] <-0
             e_puma[,i] <- y
         }
         head(e_puma)

In [29]: density(e_puma)

Out[29]:
         Call:
```

```
        density.default(x = e_puma)

Data: e_puma (218700 obs.);        Bandwidth 'bw' = 0.2384

       x                 y
 Min.   :-0.7153   Min.   :0.0000002
 1st Qu.: 3.0348   1st Qu.:0.0172600
 Median : 6.7849   Median :0.0533883
 Mean   : 6.7849   Mean   :0.0665816
 3rd Qu.:10.5351   3rd Qu.:0.0965064
 Max.   :14.2852   Max.   :0.4313578
```

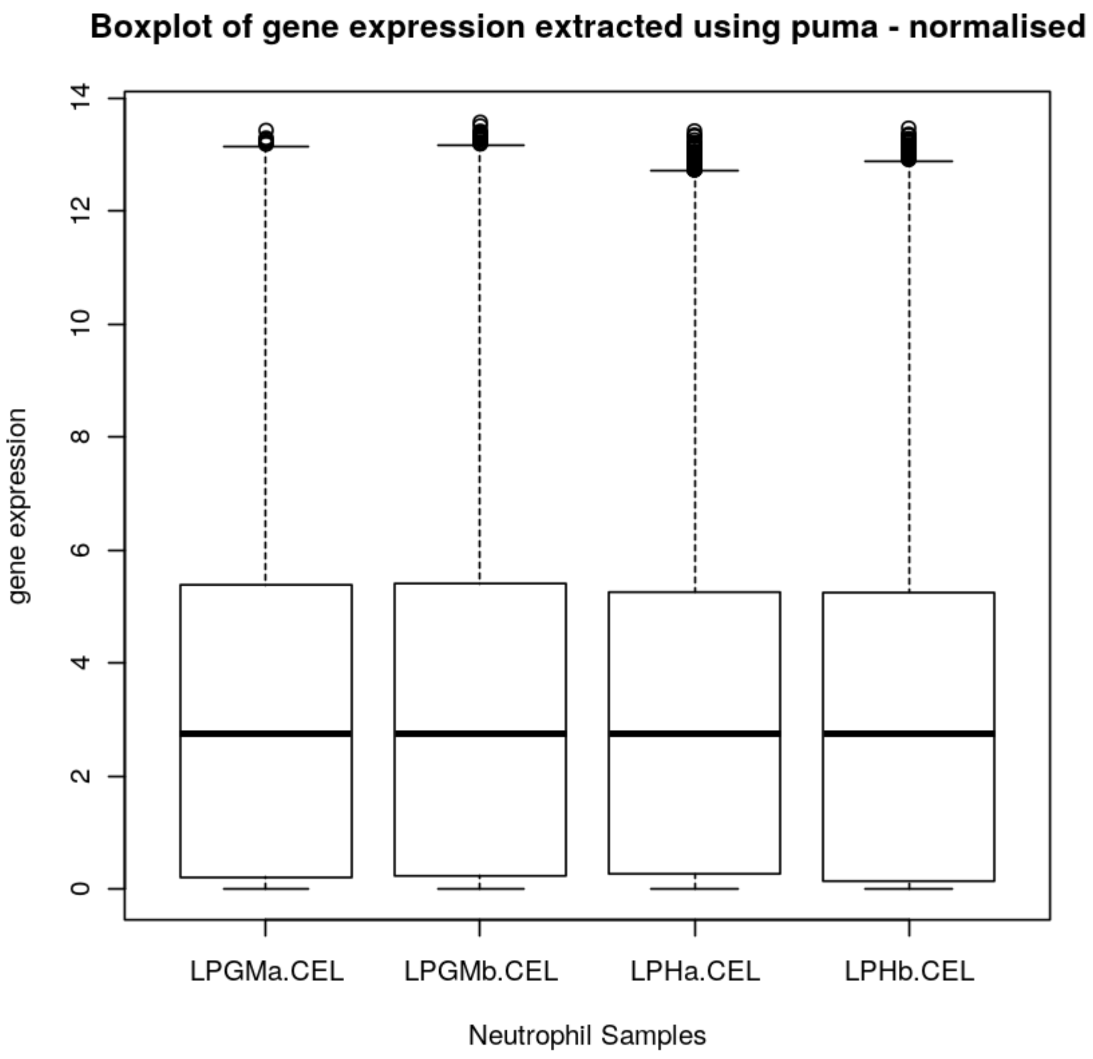
```
In [30]: plot(density(e_puma[,1]), col="red", main="PUMA Estimation")
         lines(density(e_puma[,2]),col="green")
         lines(density(e_puma[,3]),col="blue")
         lines(density(e_puma[,4]),col="purple")
```

```
Out[30]:
```



```
In [31]: boxplot(e_puma,main="Boxplot of gene expression extracted using puma - normalised", xla
```

```
Out[31]:
```



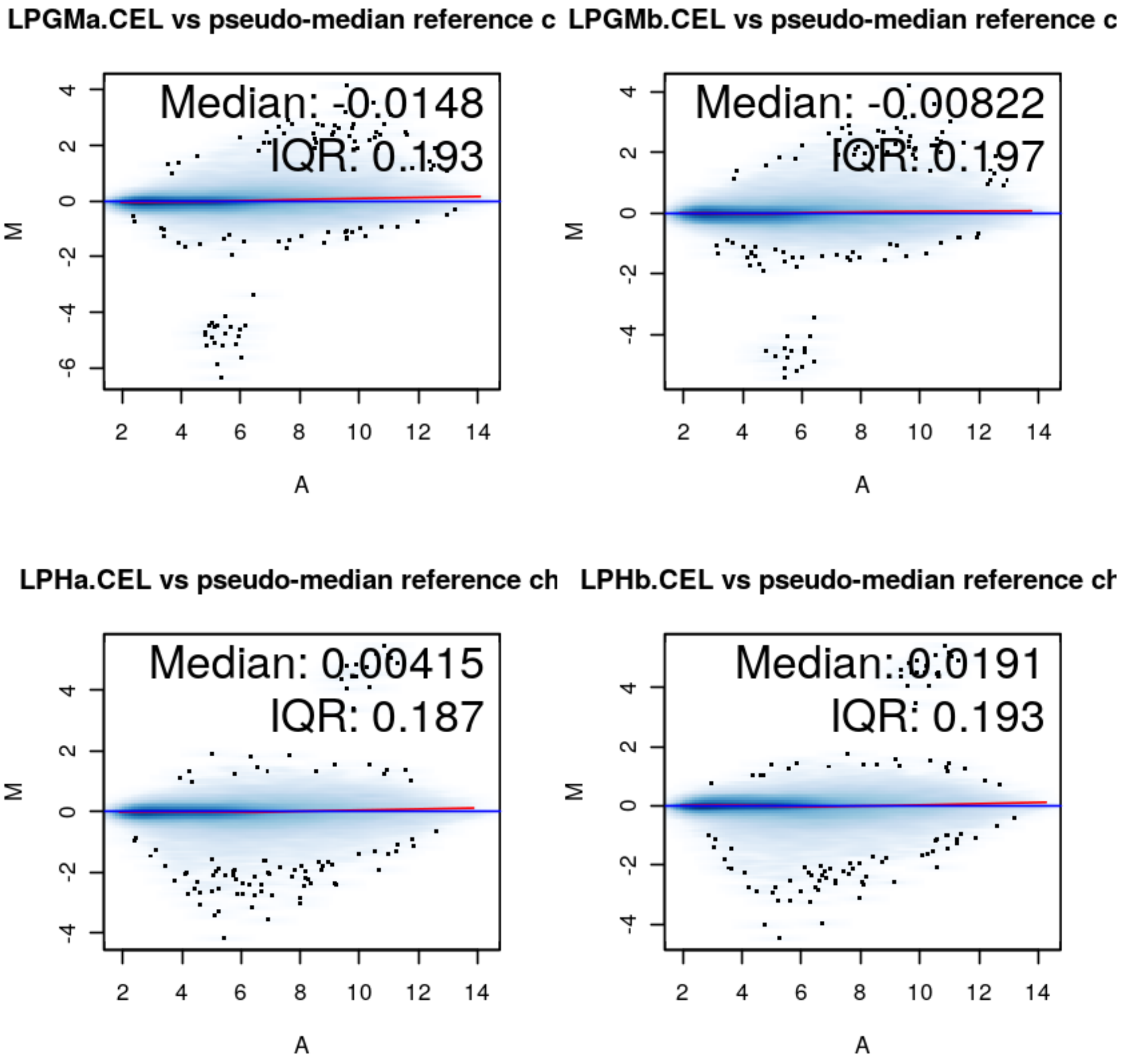
Boxplots show the differences in probe intensity behaviour between arrays. Boxplots are useful in the visualisation of data for first diagnostics, ensuring all the samples are comparable. Box plots show are able to illustrate: * Median * Upper Quartile * Lower Quartile * Range * Individual extreme values (Outliers)

The boxplots above show that gene expression extracted using rma does not need to be normalised as the medians are aligned, and no negative outliers. The mas5 boxplot showed the data must be log2 transformed in order for comparison to be possible. For puma, the results needed to be normalised due to the high number of negative outliers present, although the medians are aligned.

All three analysis techniques showed a similar range of values following normalisation.

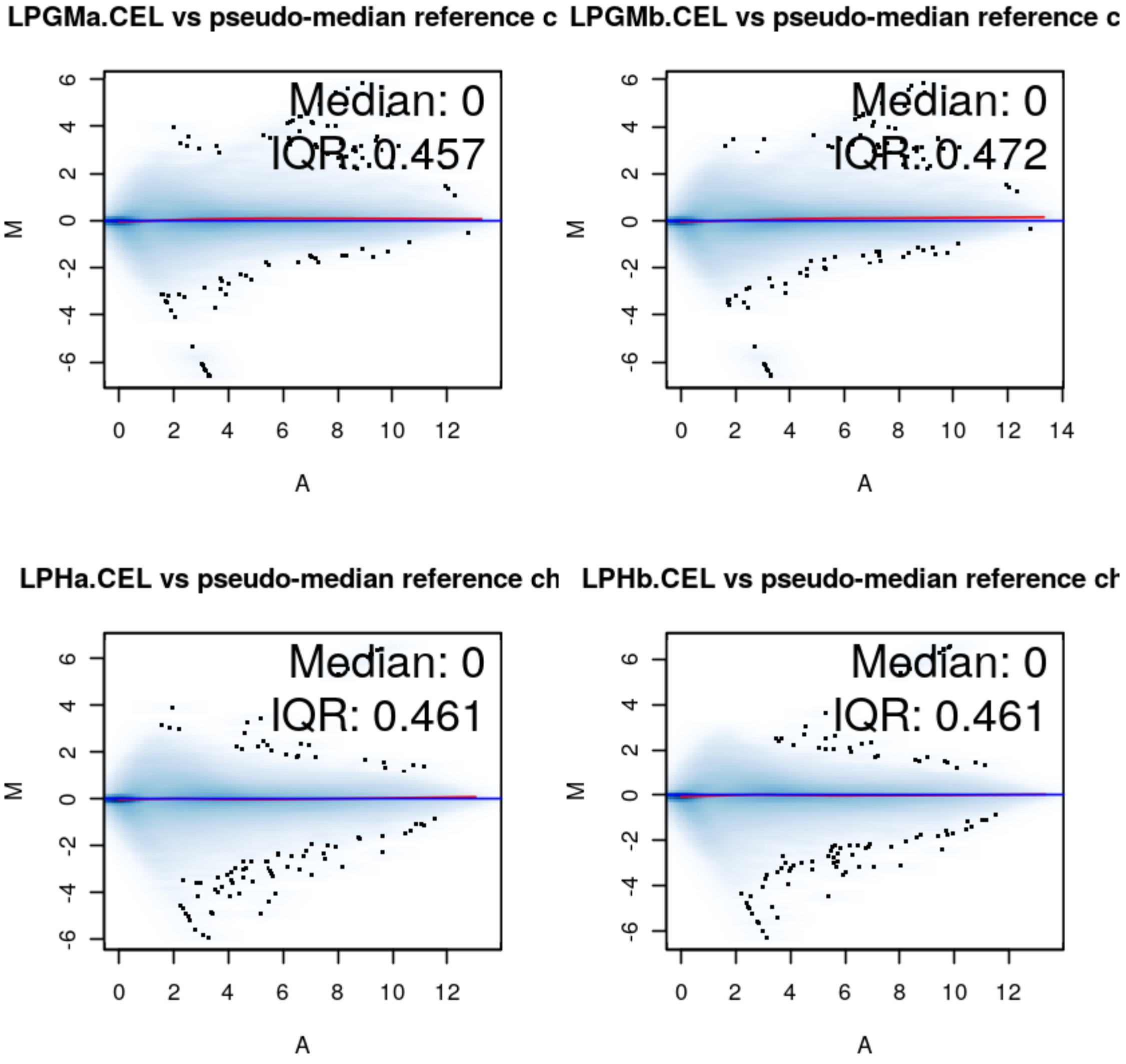
```
In [32]: par(mfrow=c(2,2))
         MAplot(e_rma)
```

```
Out[32]:
```



```
In [33]: par(mfrow=c(2,2))
         MAplot(e_puma)
```

Out[33]:



In MA plots, each Affymetrix marray is compared to a pseudo-array, which consist of the median intensity of each probe over all arrays, the plot shows to what extent the variability in expression depends on the expression level. M is the difference between the intensity of a probe on the array and the median intensity of that probe over all arrays A is the average intensity of a probe on that array and the median intensity of that probe over all arrays.

The cloud of data points in the MA plot is centered around M=0, based on the assumption that the majority of the genes are not differentially expressed, an the number of upregulated genes is similar to the number of downregulated genes.

From the MA plots above, it can be deduced that there appears to be a greater number of downregulated genes in neutrophils under hypoxia conditions than in normal conditions.

** FEEDBACK: well done, this is clear and well annotated.**

1.1.4 Step 5:

```
In [36]: eset_puma_comb<- pumaCombImproved(eset_puma_normd)
```

```
pumaComb expected completion time is 3 hours
...20%...40%...60%...80%...100%
...
```

```
In [65]: save(eset_puma_comb, file="eset_pumacomb.RDA")
```

```
In [34]: load("eset_pumacomb.RDA")
         ls()
```

```
In [35]: show(eset_puma_comb)
```

```
ExpressionSet (storageMode: lockedEnvironment)
assayData: 54675 features, 4 samples
  element names: exprs, se.exprs
protocolData: none
phenoData
  sampleNames: Hypoxia.1 Normal.1 Hypoxia.2 Normal.2
  varLabels: Condition Sample
  varMetadata: labelDescription
featureData: none
experimentData: use 'experimentData(object)'
Annotation:
```

```
In [36]: pData(eset_puma_comb)
```

** FEEDBACK: the proble you having here is that the eset_puma_comb should only have two samples, one for each condition you have. This is due to teh fact that you did not use pData() on the eset_puma. **

```
In [37]: dim(eset_puma_comb)
```

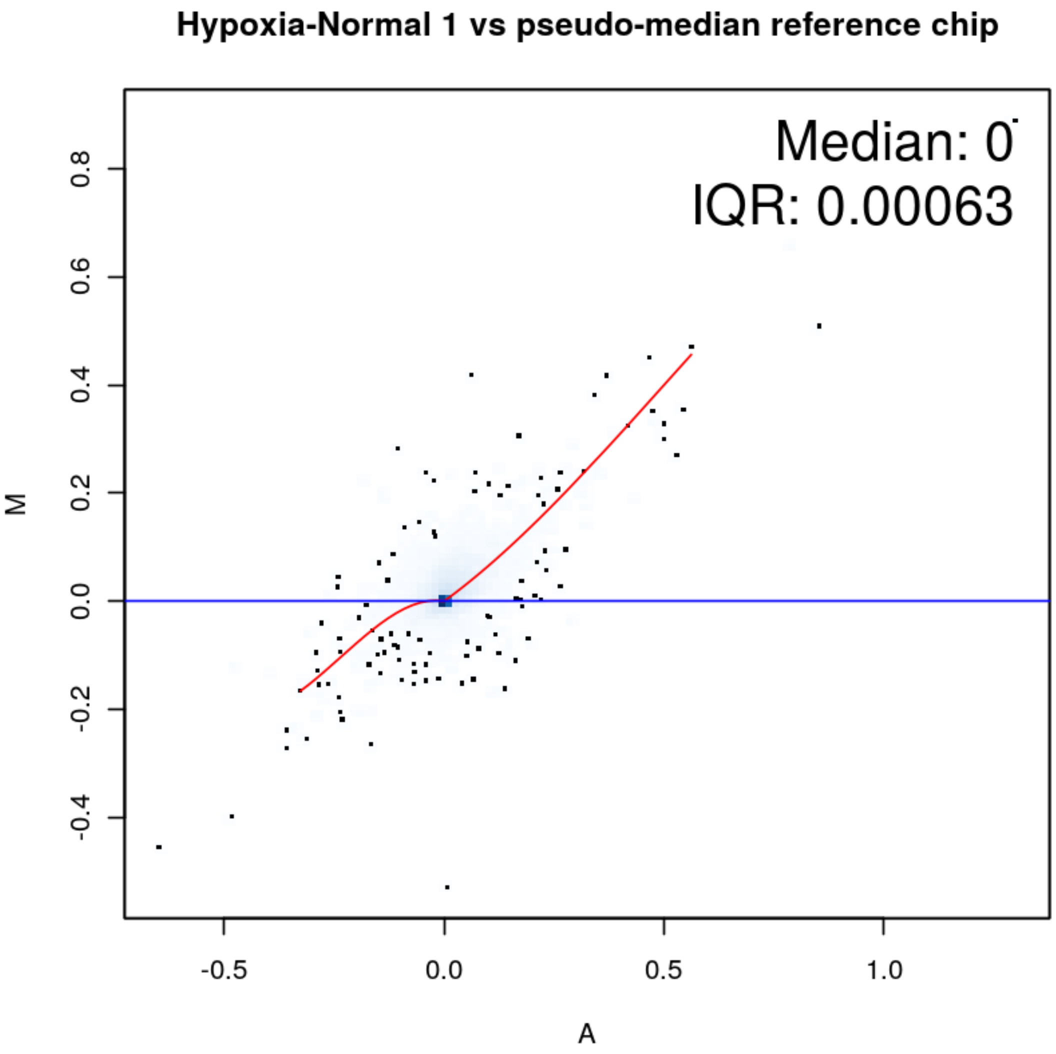
```
In [38]: hypoxia_comb_puma<-exprs(eset_puma_comb)
         for(i in 1:4) {
             temp<-hypoxia_comb_puma[,i]
             temp[temp<0] <-0
             hypoxia_comb_puma[,i]<- temp
         }
```

```
In [39]: FC_puma<- hypoxia_comb_puma[,1:2] - hypoxia_comb_puma[,3:4]
         colnames(FC_puma) <- c("Hypoxia-Normal 1","Hypoxia-Normal 2")
         head(FC_puma)
```

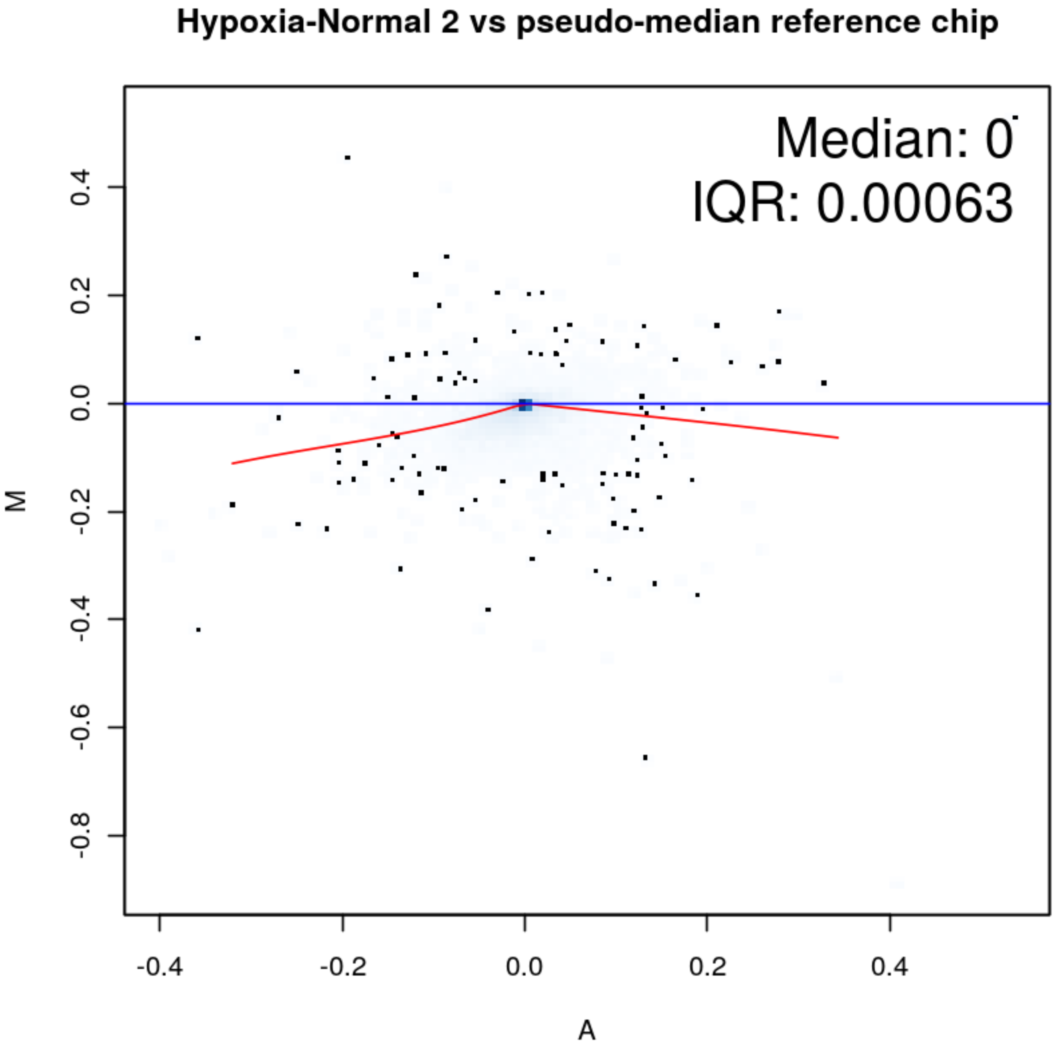
```
In [40]: MAplot(FC_puma)
```

```
Warning message in KernSmooth::bkde2D(x, bandwidth = bandwidth, gridsize = nbin, :
Binning grid too coarse for current (small) bandwidth: consider increasing 'gridsize'Warning mes
Binning grid too coarse for current (small) bandwidth: consider increasing 'gridsize'
```

Out[40]:



Out[40]:

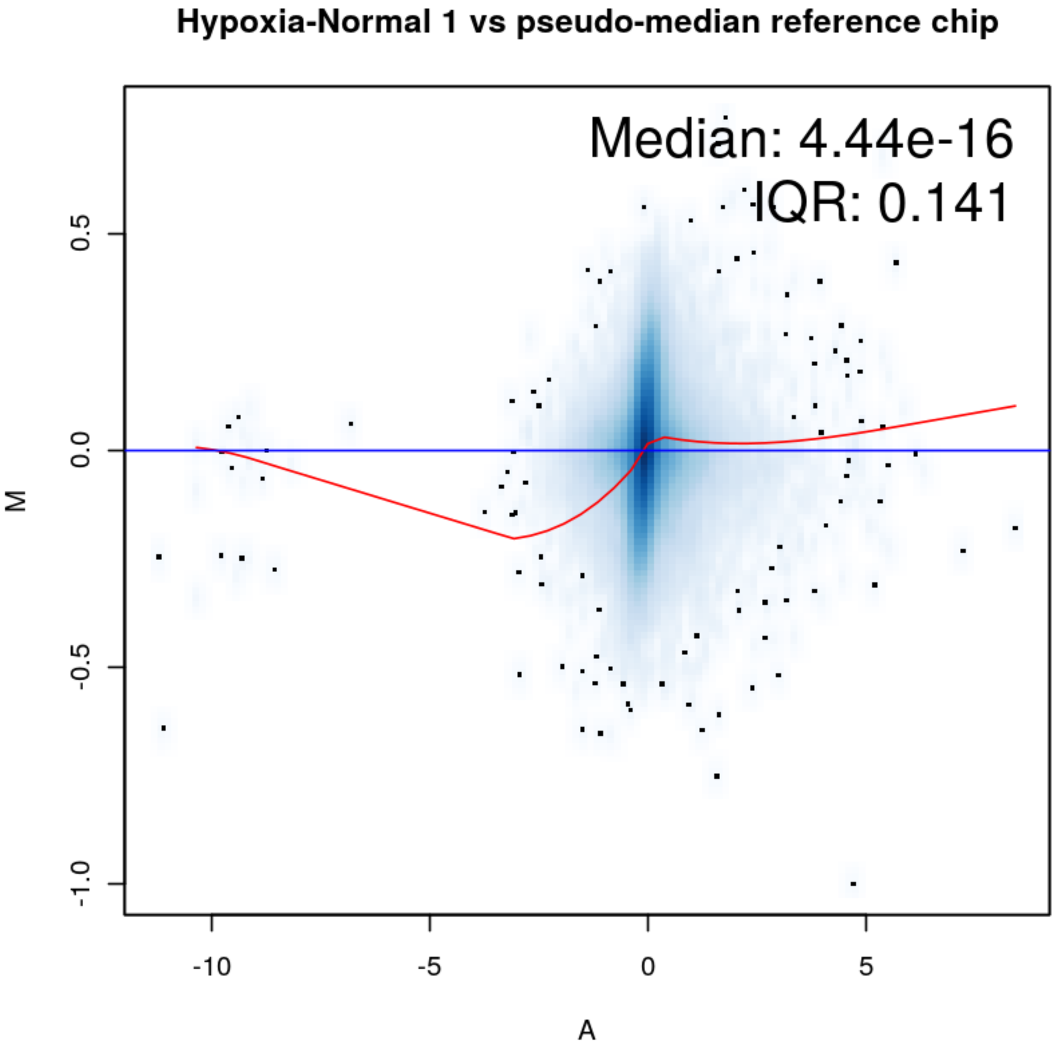


** FEEDBACK: wrong annotation is also why the MA plot do not look right. **

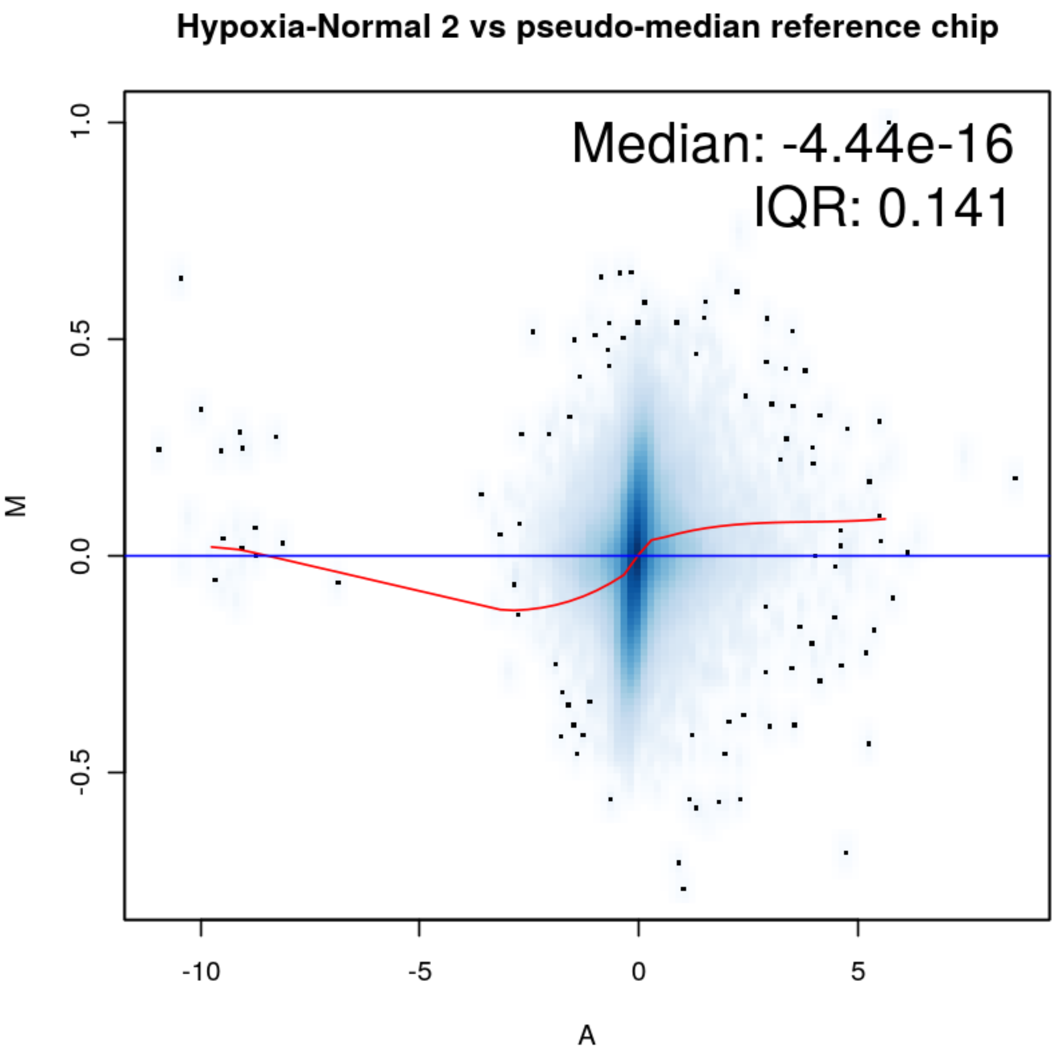
```
In [41]: FC_rma<- e_rma[,1:2] - e_rma[,3:4]
         colnames(FC_rma) <- c("Hypoxia-Normal 1","Hypoxia-Normal 2")
         head(FC_rma)
```

```
In [42]: MAplot(FC_rma)
```

```
Out[42]:
```


Hypoxia-Normal 1 vs pseudo-median reference chip
Median: 4.44e-16
IQR: 0.141
M
0.5
0.0
-0.5
-1.0
-10
-5
0
5
A

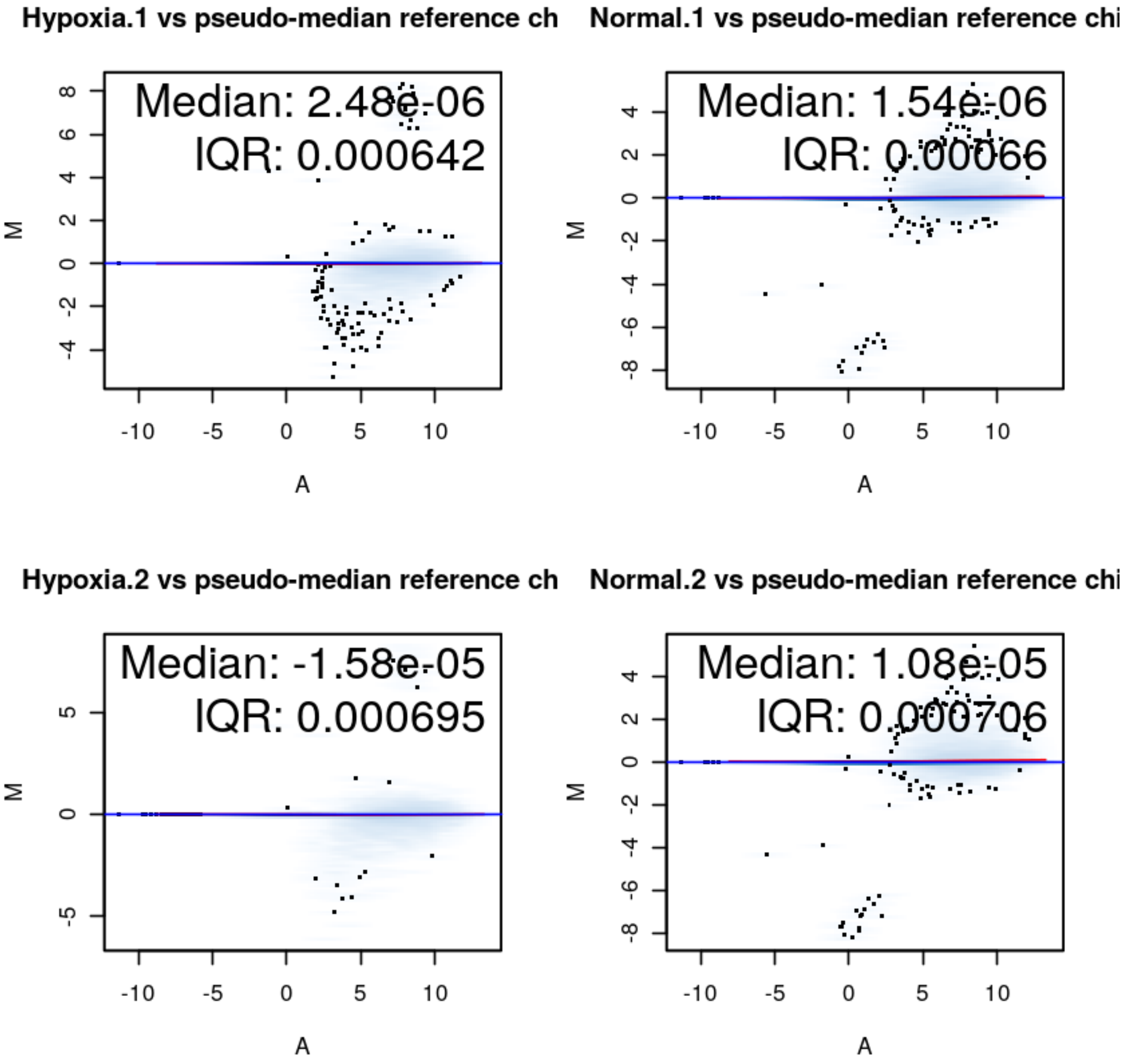
```
Out[42]:
```



```
In [43]: groups<-c("H1","N1","H2","N2")
         hypoxia_table<-data.frame(sampleNames(eset_puma_comb),groups)
         group1<-factor(groups[1:2])
         group2<-factor(groups[3:4])

         group1
         group2

In [44]: hypoxia_table

In [45]: par(mfrow=c(2,2))
         MAplot(eset_puma_comb)
```

```
Warning message in KernSmooth::bkde2D(x, bandwidth = bandwidth, gridsize = nbin, :
Binning grid too coarse for current (small) bandwidth: consider increasing 'gridsize'Warning mes
Binning grid too coarse for current (small) bandwidth: consider increasing 'gridsize'Warning mes
Binning grid too coarse for current (small) bandwidth: consider increasing 'gridsize'Warning mes
Binning grid too coarse for current (small) bandwidth: consider increasing 'gridsize'
```

Out[45]:



```
In [46]: library(limma)
         group<-factor(c("Normal","Normal","Hypoxia","Hypoxia"))
         design<-model.matrix(~0+group)
         colnames(design)<-c("Normal","Hypoxia")
         contrast.matrix_puma<- makeContrasts(Normal,Hypoxia,levels=design)
```

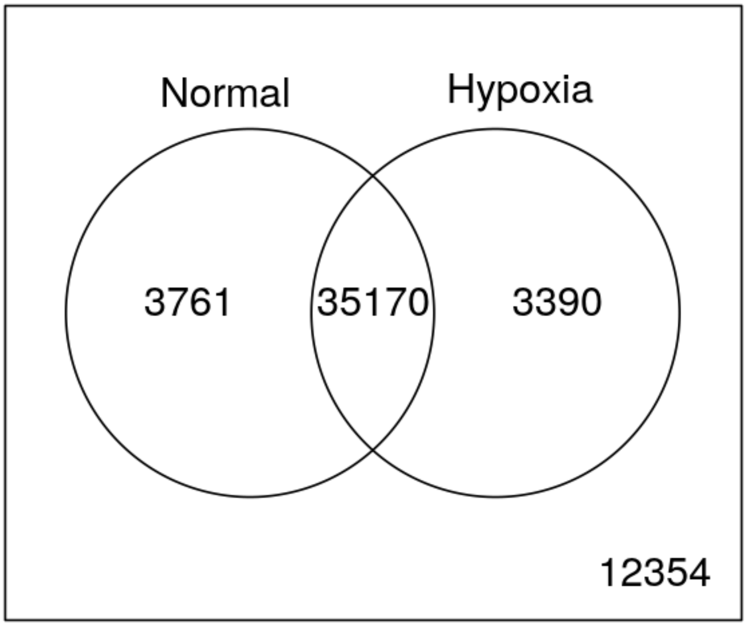
```
        design
        contrast.matrix_puma

        fit<-lmFit(eset_puma,design)
        fit2<-contrasts.fit(fit,contrast.matrix_puma)
        fit3<-eBayes(fit2)

        topDEGenes<-topTable(fit3, coef=1, adjust="BH", n=100, lfc=1)
        topDEGenes
```

```
Attaching package: limma

The following object is masked from package:oligo:

    backgroundCorrect

The following object is masked from package:BiocGenerics:

    plotMA
```

Limma is a package for differential expression analysis of data arising from microarray experiments. A linear model is fit to the expression data for each gene. Empirical Beyes (a shrinkage method) is used to borrow information across genes making the analyses stable. Linear models are used to analyse designed microarray experiments, allowing for very general experiments to be analysed easily. Two matrices need to be specified. The design matrix provides a representation of the different RNA targets which have been hybridized to the arrays. The contrast matrix allows the coefficients designed by the design matrix to be combined into contrasts of interest. Each contrast corresponds to a comparison of interest between the RNA targets.

```
In [47]: results_puma<-decideTests(fit3, method="global",lfc=1)
         vennDiagram(results_puma)
```

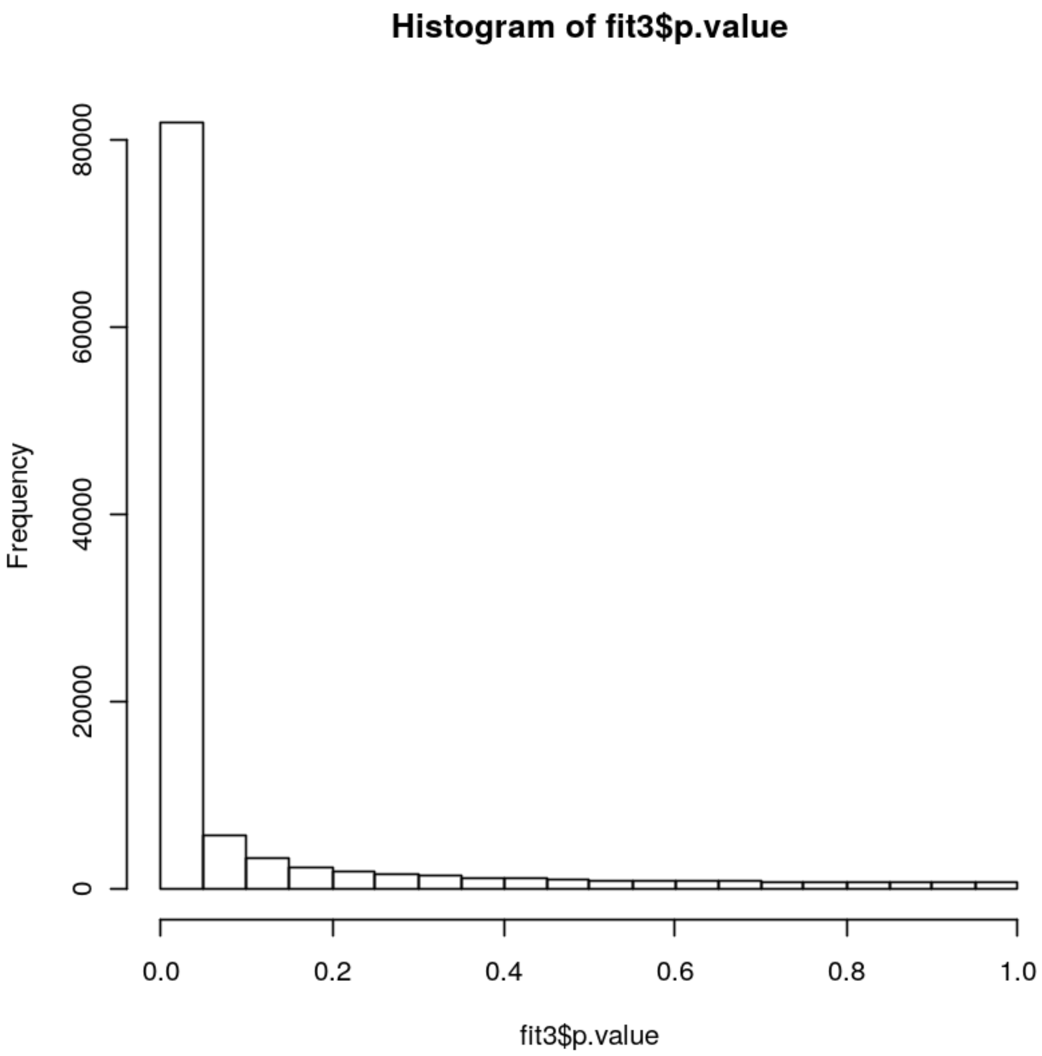
```
   Out[47]:
```



The Venn Diagram shows that 3761 genes and 3390 genes were expressed in only normal and only hypoxia conditions, respectively. 35170 genes were expressed in both normal and hypoxia conditions.

```
In [48]: hist(fit3$p.value)
```

```
Out[48]:
```



```
In [49]: dim(topDEGenes)

In [50]: rownames(topDEGenes)

In [51]: write.table(rownames(topDEGenes),"/projects/ddda6a8e-2bca-47f5-b1d6-79b2c48d0e30/Autumn

In [52]: pumaDERes<-pumaDE(eset_puma_comb)
         pumaDERes

Out[52]: DEResult object:
           DEMethod = pumaDE
           statisticDescription = Probability of Positive Log Ratio (PPLR)
           statistic = 54675 probesets x 7 contrasts
```

```
In [53]: getwd()
```

```
In [54]: write.reslts(pumaDERes, file="pumaDERes")
```

```
In [55]: library(hgu133plus2.db)
         library(annotate)

         geneProbes<-as.character(rownames(topDEGenes))
         annotated_list<-select(hgu133plus2.db, geneProbes,c("SYMBOL","GENENAME"))
         annotated_list
```

```
Loading required package: AnnotationDbi
Loading required package: org.Hs.eg.db
Loading required package: DBI


Loading required package: XML
'select()' returned 1:many mapping between keys and columns
```

```
In [56]: annotated_list[,2]
```

```
In [57]: write.table(annotated_list[,2],"/projects/ddda6a8e-2bca-47f5-b1d6-79b2c48d0e30/Autumn20
```

```
In [58]: dir()
```

```
In [59]: pumaDE_stat<-read.csv("pumaDERes_statistics.csv")
         pumaDE_FC<-read.csv("pumaDERes_FCs.csv")
```

```
In [60]: head(pumaDE_stat)
```

```
In [61]: probeid<-pumaDE_stat[,1]
         PPLR_N1vsH1<-pumaDE_stat[,2]
         PPLR_N2vsH2<-pumaDE_stat[,5]


         pumaRes<-data.frame(probeid,PPLR_N1vsH1,PPLR_N2vsH2)
         pumaRes
```

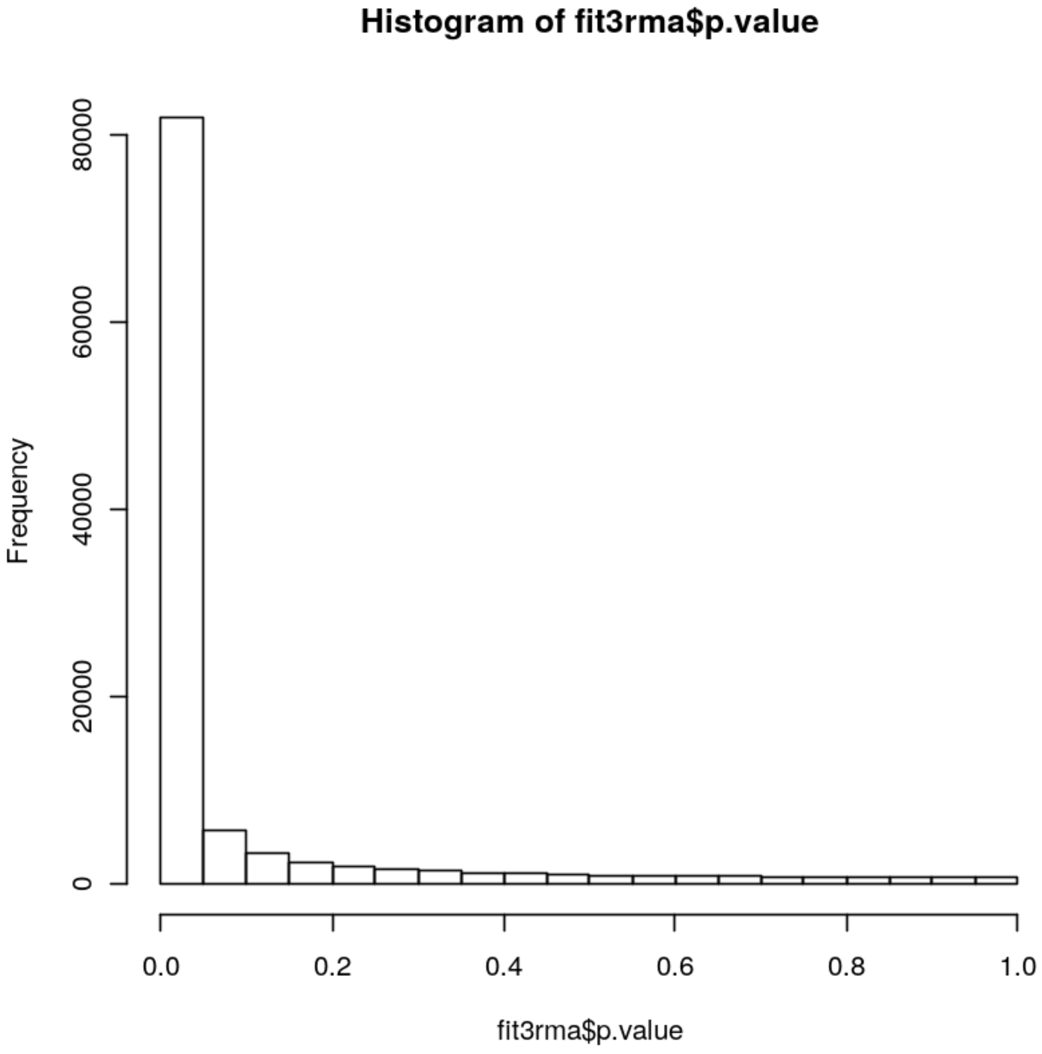
```
In [62]: down_N1vsH1<-pumaRes[pumaRes$PPLR_N1vsH1<=0.2,1]
         up_N1vsH1<-pumaRes[pumaRes$PPLR_N1vsH1>=0.8,1]

         down_N2vsH2<-pumaRes[pumaRes$PPLR_N2vsH2<=0.2,1]
         up_N2vsH2<-pumaRes[pumaRes$PPLR_N2vsH2>=0.8,1]



         downDE<-data.frame(match(down_N1vsH1,down_N2vsH2))
         downDE<-downDE[!is.na(downDE)]
         upDE<-data.frame(match(up_N1vsH1,up_N2vsH2))
         upDE<-upDE[!is.na(upDE)]
```

```
In [63]: DE<-data.frame(match(downDE,upDE))
         DE<-DE[!is.na(DE)]
         length(DE)

In [64]: head(pumaDE_FC)

In [65]: geneProbes<-as.character(pumaDE_FC$X)
         annotated_list<-select(hgu133plus2.db,geneProbes,c("SYMBOL","GENENAME"))
         DEGenes=annotated_list[pumaRes[DE,1],]
         DEGenes
         dim(DEGenes)
```

```
'select()' returned 1:many mapping between keys and columns
```

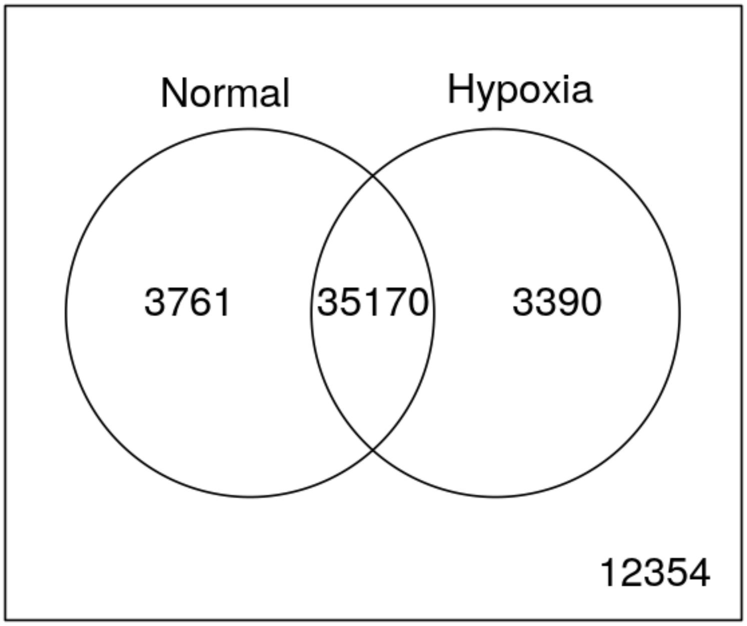
```
In [66]: group<-factor(c("Normal","Normal","Hypoxia","Hypoxia"))
         design<-model.matrix(~0+group)
         colnames(design)<-c("Normal","Hypoxia")
         contrast.matrix_rma<- makeContrasts(Normal,Hypoxia,levels=design)
         design
         contrast.matrix_rma

         fitrma<-lmFit(eset_rma,design)
         fit2rma<-contrasts.fit(fit,contrasts=contrast.matrix_rma)
         fit3rma<-eBayes(fit2rma)

         topDEGenes_rma<-topTable(fit3, coef=1, adjust="BH", n=100, lfc=1)
         topDEGenes_rma

         dim(topDEGenes_rma)

In [67]: hist(fit3rma$p.value)
```

Out[67]:



In [68]:
```
results_rma<-decideTests(fit3rma, method="global",lfc=1)
vennDiagram(results_rma)
```

Out[68]:



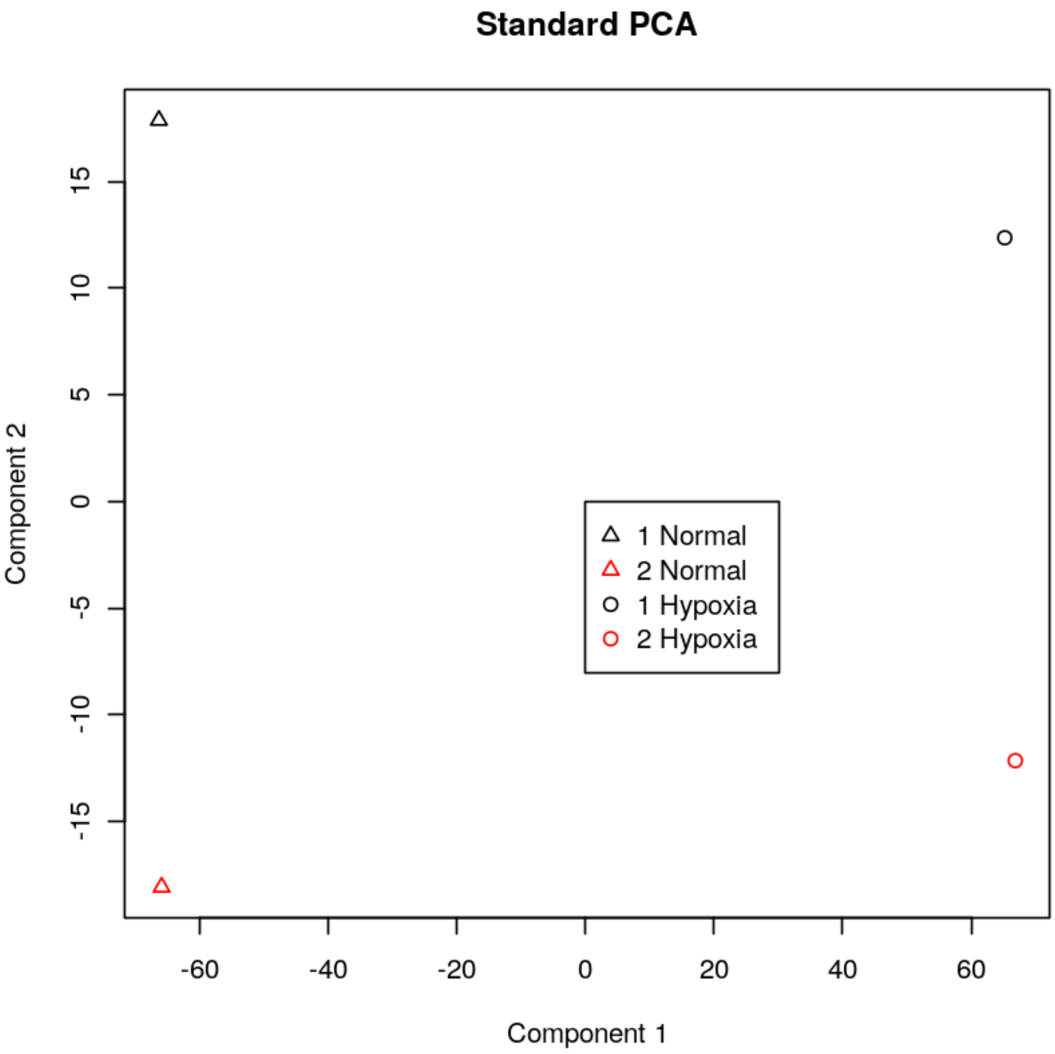
1.1.5 Step 6:

PCA is a mathematical algorithm that reduces the dimensionality of the data while retaining most of the variation in the data set. It does so by identifying directions, called principal components, along which the variation in the data is maximal. PCA plots check whether the overall variability of the samples reflect their groupings.

```
In [69]: pca_hypoxia <- prcomp(t(e_rma))


         plot(pca_hypoxia$x, xlab="Component 1", ylab="Component 2",
              pch=unclass(as.factor(pData(eset_rma)[,1])),
              col=unclass(as.factor(pData(eset_rma)[,2])), main="Standard PCA")
```

```
groups<-paste(eset_rma$Sample, eset_rma$Condition, sep =" ")

legend(0,0,groups,pch=unclass(as.factor(pData(eset_rma)[,1]))
, col=unclass(as.factor(pData(eset_rma)[,2])))
```

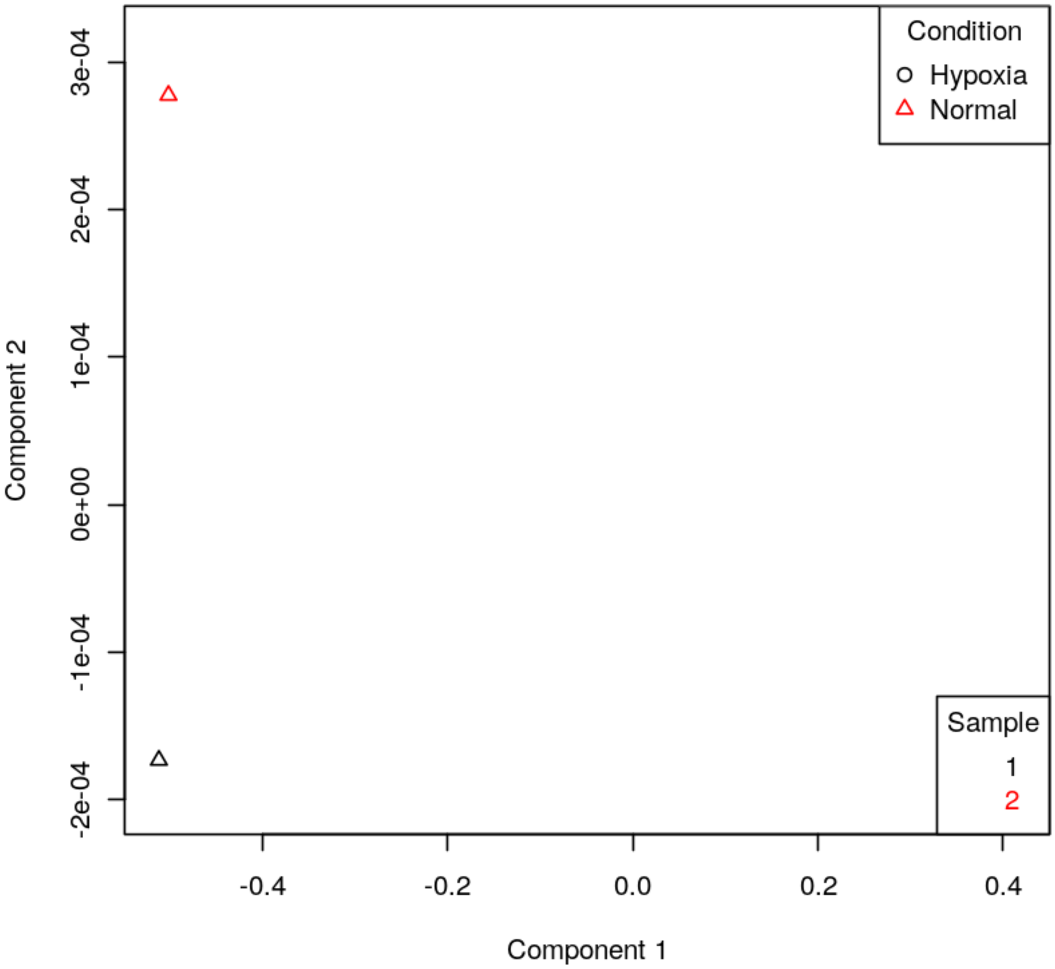
Out[69]:



```
In [70]: pumapca_hypoxia=pumaPCA(eset_puma_normd)
         plot(pumapca_hypoxia)
```

```
Iteration number: 1
Iteration number: 2
Iteration number: 3
Iteration number: 4
Iteration number: 5
```

Out[70]:



The two PCA plots show that the gene expressions of the neutrophils under the two conditions do vary. This is clear on the plots as the points for the two hypoxia samples are on right side of the plot, whereas the two normal samples are found on the left of the plot. It is unclear whether there is a clear difference between the gene expressions of the two sample groups themselves; in order to observe a clear difference between samples, more conditions are required, such as different levels of hypoxia.

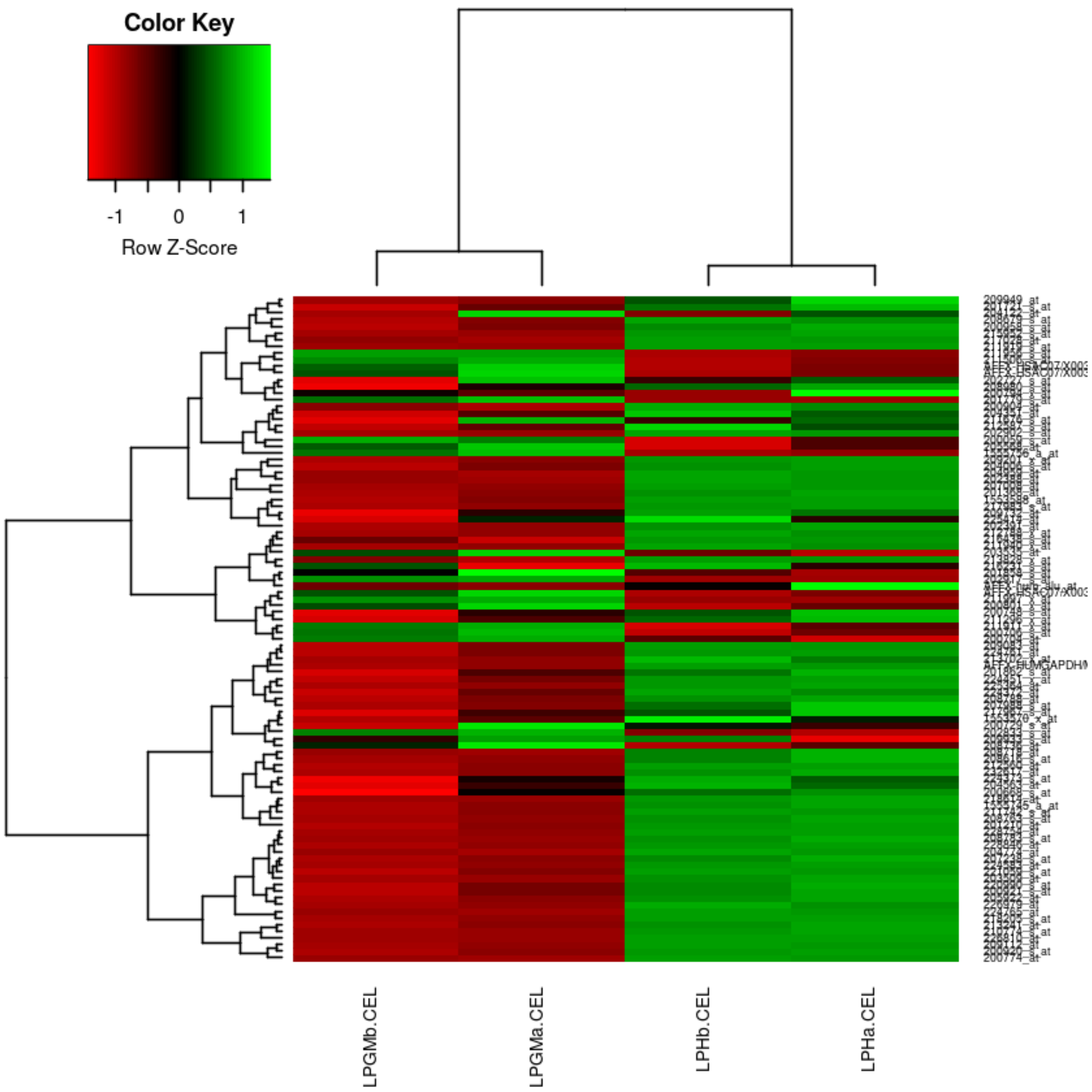
1.1.6 Step 7:

Heat maps and clustering are often used in gene expression analysis studies to visualise the data and for quality control. It is a graphical representation of the data where the individual values in the matrix are represented as colours. They compares the level of gene expression of a number of samples, allowing for immediate visualisation of the data by assigning different colours to each gene, and it is possible to see clusters of genes with similar or hugely different expression values.

```
In [71]: library(gplots)

         tID<-rownames(topDEGenes)
         ind<-1
         j<-1
         for (i in 1: length(tID)) {
                 ind[j]<-which(rownames(eset_rma)==tID[i],arr.ind=TRUE)
                 j<-j+1
         }


         topExpr<-e_rma[ind,]
         heatmap.2(topExpr, col=redgreen(75), scale="row",
         key=TRUE, symkey=FALSE, density.info="none", trace="none", cexRow=0.5, cexCol=0.8)
```

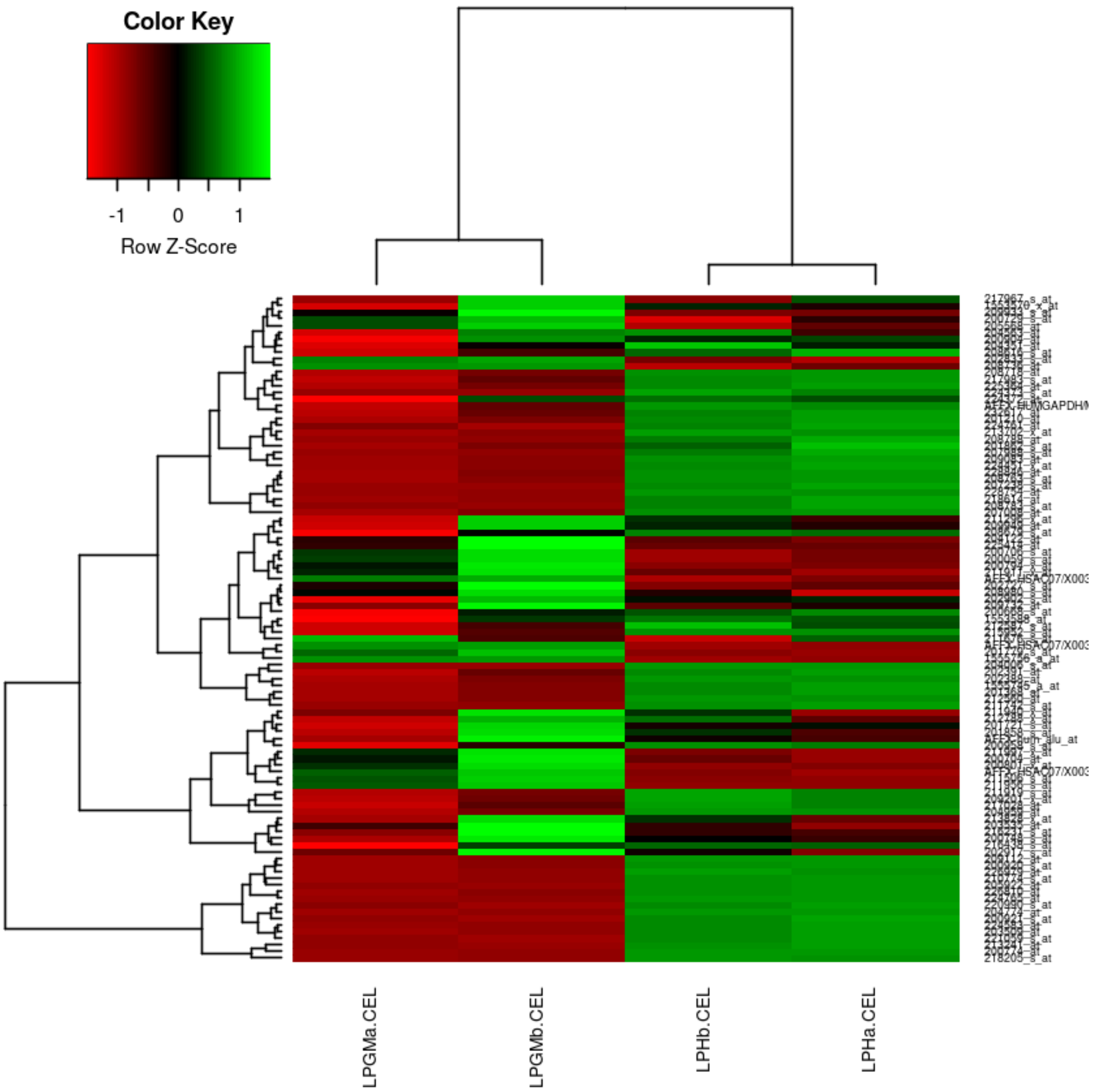
```
Attaching package: gplots

The following object is masked from package:IRanges:

    space

The following object is masked from package:stats:

    lowess
```

Out[71]:



```
In [72]: library(gplots)

         tID<-rownames(topDEGenes)
         ind<-1
         j<-1
         for (i in 1: length(tID)) {
                 ind[j]<-which(rownames(eset_puma)==tID[i],arr.ind=TRUE)
                 j<-j+1
         }


         topExpr<-e_puma[ind,]
         heatmap.2(topExpr, col=redgreen(75), scale="row",
        key=TRUE, symkey=FALSE, density.info="none", trace="none", cexRow=0.5, cexCol=0.8)
```

```
Out[72]:
```



The heatmaps above are generated from eset_rma and eset_puma data, both showing similar patterns of gene expression, as indicated by the colours. From both methods, it can be deduced that a lot of genes that are expressed in samples order normal conditions are not expressed in samples under hypoxia, confirming that hypoxia has an effect on neutrophil gene expression.

1.1.7 Step 8:

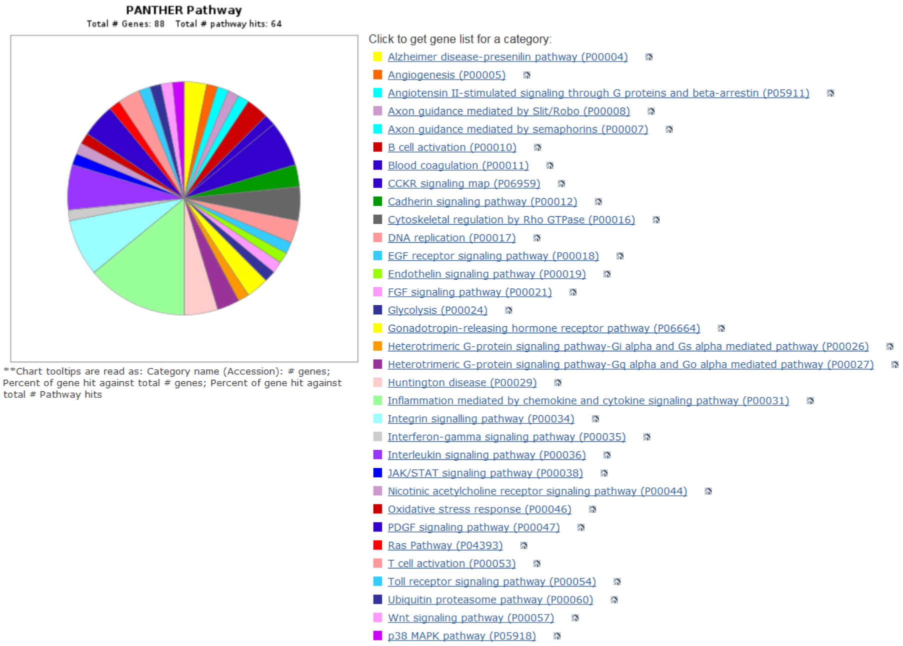
This step involves the functional/ pathway analysis of differentially expressed targets using PANTHER or DAVID. DAVID is the online Database for Annotation, Visualization and Integrated Discovery, which can be used to convert a list of gene IDs. PANTHER (Protein ANalysis THrough

Evolutionary Relationships) can be used to classify proteins and identify the key pathways involved in the difference in gene expression observed. PANTHER is used in this project to identify the key pathways in regulating gene expression in neutrophils under hypoxia and normal conditions.

```
In [73]: setwd("~/Autumn2016/ProjectC/data_projectC")
```

```
In [82]: GeneList<-read.table("pantherGeneList.txt", fill=TRUE)
         GeneList
```

```
In [83]: pantherChart<-read.table("pantherChart.txt", fill=TRUE)
         pantherChart
```



PANTHER Pathway

Total # Genes: 88 Total # pathway hits: 64

**Chart tooltips are read as: Category name (Accession): # genes; Percent of gene hit against total # genes; Percent of gene hit against total # Pathway hits

Click to get gene list for a category:

- Alzheimer disease-presenilin pathway (P00004)
- Angiogenesis (P00005)
- Angiotensin II-stimulated signaling through G proteins and beta-arrestin (P05911)
- Axon guidance mediated by Slit/Robo (P00008)
- Axon guidance mediated by semaphorins (P00007)
- B cell activation (P00010)
- Blood coagulation (P00011)
- CCKR signaling map (P06959)
- Cadherin signaling pathway (P00012)
- Cytoskeletal regulation by Rho GTPase (P00016)
- DNA replication (P00017)
- EGF receptor signaling pathway (P00018)
- Endothelin signaling pathway (P00019)
- FGF signaling pathway (P00021)
- Glycolysis (P00024)
- Gonadotropin-releasing hormone receptor pathway (P06664)
- Heterotrimeric G-protein signaling pathway-Gi alpha and Gs alpha mediated pathway (P00026)
- Heterotrimeric G-protein signaling pathway-Gq alpha and Go alpha mediated pathway (P00027)
- Huntington disease (P00029)
- Inflammation mediated by chemokine and cytokine signaling pathway (P00031)
- Integrin signalling pathway (P00034)
- Interferon-gamma signaling pathway (P00035)
- Interleukin signaling pathway (P00036)
- JAK/STAT signaling pathway (P00038)
- Nicotinic acetylcholine receptor signaling pathway (P00044)
- Oxidative stress response (P00046)
- PDGF signaling pathway (P00047)
- Ras Pathway (P04393)
- T cell activation (P00053)
- Toll receptor signaling pathway (P00054)
- Ubiquitin proteasome pathway (P00060)
- Wnt signaling pathway (P00057)
- p38 MAPK pathway (P05918)

Pathway analysis using PANTHER

From PANTHER, the gene list and the pathways they work in have been identified, with the piechart showing the percentage of genes that are present in each pathway. The most prominent pathway in the effects of hypoxia on human neutrophils is identified as the Inflammation mediated by chemokine and cytokine signaling pathway.

1.1.8 Discussion

In this project, the aim was to stimate gene expression levels, and analyse the results to identify the genes that are changing between the two conditions of normal and hypoxia, defining the potential pathways that hypoxia may have altered in neutrophils. The methods of RMA and MAS5 were used, and first diagnostics performed in order to identify the suitable method to continue with. RMA was chosen as no further normalisation was required. The PUMA package was also used, and the data combined using a Bayesian Hierarchical model, further analysis was done in order to obtain the fold change in gene expression. Limma was used for Differential Expression Analysis, and the p-value calculated. The data was visualised using PCA, indicating that there is a clear difference between the gene expression of neutrophils under normal, or hypoxia conditions. This was further supported by the heatmaps generated.

Through the use of PANTHER, it was possible to identify the key pathways that are regulating the effects of hypoxia on human neutrophils - The Inflammation mediated by chemikine and cytokine signaling pathway.

1.1.9 References

https://www.bioconductor.org/packages/devel/bioc/vignettes/puma/inst/doc/puma.pdf
https://www.bioconductor.org/packages/devel/bioc/manuals/puma/man/puma.pdf
https://www.bioconductor.org/packages/release/bioc/vignettes/affy/inst/doc/affy.pdf
http://svitsrv25.epfl.ch/R-doc/library/Biobase/html/00Index.html#P
http://bioinfo.cipf.es/babelomicstutorial/maplot http://www.bioinformatics.babraham.ac.uk/projects/seqmo
https://www.biostars.org/p/101727/ http://www.nature.com/ng/journal/v32/n4s/pdf/ng1032.pdf
http://bmcgenomics.biomedcentral.com/articles/10.1186/1471-2164-15-
103 http://www.nature.com/nbt/journal/v26/n3/pdf/nbt0308-303.pdf
http://wiki.bits.vib.be/index.php/Analyze_your_own_microarray_data_in_R/Bioconductor#MA_plots
http://arrayanalysis.org/main.html