UNIVERSIDADE DE LISBOA

FACULDADE DE CIÊNCIAS

DEPARTAMENTO DE INFORMÁTICA

Question 2 - Microarray Data

Lucas da Costa Monteiro

nº 52849

Mestrado em Bioinformática e Biologia Computacional

Projeto em Métodos Estatísticos em Bioninformática

07/2022

Contents

List of Figures

1 Introduction

Background Two Channel Microarrays are cost-effective platforms for comparative analysis of gene expression (Zhonggang Hou et al., 2015). DNA microarrays contain thousands of different nucleotide sequences attached in microscopic spots, representing unique, and known, regions of genes in the genome. The arrays are hybridized with cDNA prepared from the two samples that are to be compared, while unbound material is washed away.

This cDNA is labeled with two different fluorophores, typically the control cDNA is dyed with Cy3, a fluorescent dye with green color, and the experimental cDNA with Cy5 corresponding to a red color so that a scanner can measure the fluorescent signals for each gene region. Background noise is also measured by the scanner. If the fluorescent signal is *more red* in some genes, for example, it means that this gene has a higher expression in the experimental group than in the control.

In this work three *mRNA* samples from saphenous vein tissues collected in an experiment carried out in a Laboratory of Molecular Cardiology of a Brazilian Institute will be analysed. These tissues were maintained in a culture *ex-vivo* and submitted to two experimental conditions: **arterial regimen**, or ***Art***, and **venous regimen**, or ***Ven***.

The samples were processed with Background Two Channel Microarray analysis, where the sample *Art* corresponds to the red channel and the sample *Ven* to the green channel. The goal is to analyze this data and conclude if some genes are differentially expressed in the experimental conditions.

2 Chip1 Data Analysis

The first patient data is registered in the `chip1.txt` file. It contains 2994 gene IDs, each one with the correspondent *Art* and *Ven* fluorescent intensity values and the respective background intensities.

2.1 The data

R's genArise library (Mayén et al., 2005) will be used because it contains specific functions to perform an analysis of microarray obtained data. First, the data is imported as a Spot object since many other functions of *genArise* that carry out transformations on the data require this type of object as an argument.

```
chip1.spot <- read.spot( "chip1.txt", cy3 = "Ven", cy5 = "Art",
                         bg.cy3 = "BgVen", bg.cy5 = "BgArt", ids = "ID",
                         header = T, sep = "\t", is.ifc = F )
```

The *Ven* was set to the green channel and *Art* to the red channel, i.e Cy3 and Cy5 respectively. The library also provides plot functions to visualize the data. The dispersion diagram

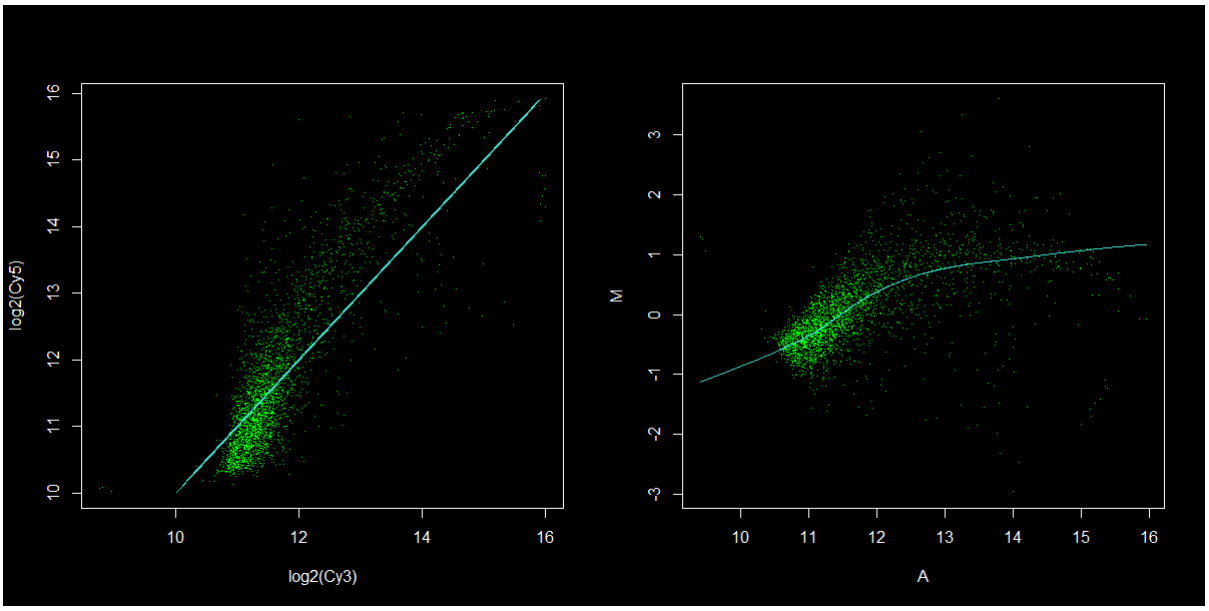

Figure 1: Visualization of *chip1.txt* raw data. **Left** The dispersion diagram of the $\log_2$ of the red vs the green (Cys5 vs Cys3) intensities. The blue line represents $\log_2(\text{Cys5}) = \log_2(\text{Cys3})$ and each green point a gene. **Right** The M vs A plot of the gene intensities. M represents the $\log_2$ of the ratio R, i.e $\log_2(\text{Cys5/Cys3}) = \log_2(\text{Cys5}) - \log_2(\text{Cys3})$ and A the $\log_2$ average of the point intensities, $\frac{1}{2}\Big(\log_2(\text{Cys5}) + \log_2(\text{Cys3})\Big)$. The blue line is calculated by local regression.

and the MA-plot of the data are represented in Figure 1.

In the dispersion diagram, the $\log_2$ of the red vs the green (Cys5 vs Cys3) intensities are plotted. Points above the blue line, where $\log_2(\text{Cys5}) = \log_2(\text{Cys3})$, represent genes with higher expression in the red channel while bellowing in the green.

The M vs A plot represents the $\log_2$ of the ratio R vs the $\log_2$ average of the point intensities. R is given by Cys5/Cys3 so $\text{M} = \log_2(\text{Cys5}) - \log_2(\text{Cys3})$. When $\log_2(\text{Cys5}) = \log_2(\text{Cys3})$, $\text{M} = 0$ so we can make a similar analysis for the M vs A plot, when M is positive the gene is better represented by the red channel and if negative in the green. However, these results can be flawed due to background noise, and a more precise analysis is conducted if this is considered.

2.2 Background Noise Correction

The luminous noise from the environment can disrupt the fluorescent readings of the microarray. So background intensity is a measure so that it can be posteriorly subtracted, during the data cleaning. The *genArise* library provides a function that does this automatically.

```
c1_bgcor <- bg.correct(chip1.spot) # background corrected Spot object
```

No major differences are found after background correction, as in Figure 2. However, the data still requires cleaning for a precise analysis.

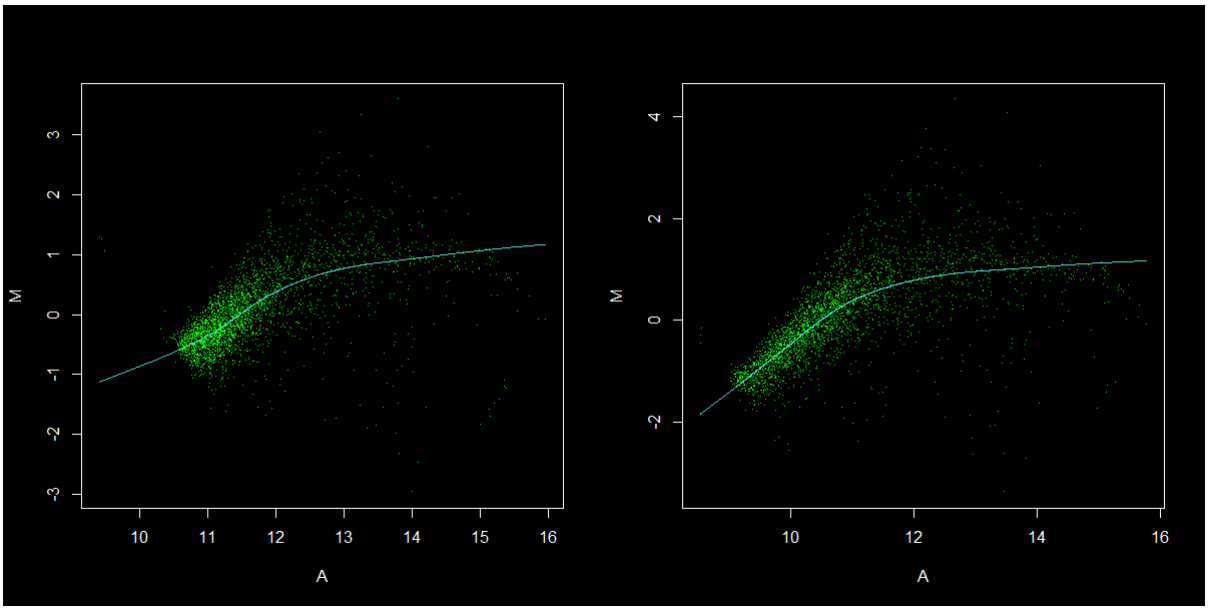

Figure 2: Comparison of the M vs A plots of the *chip1.txt* data before (left) and after (right) background correction.

2.3 Data Normalization

Normalization is a general term for a collection of methods that are directed at resolving the systematic errors and bias introduced by the microarray experimental platform (Sousa, L. (2022). Analysis of Microarray Data [14]). Once again, the library allows for simple normalization with a unique function.

```
c1_cnorm <- global.norm(c1_bgcor) # normalized Spot object
```

The differences are clear and can be visualized in Figure 3. Points orbit around M = 0, varying about two units and concentrating at $\log_2$ averages of about 10.

Now the data is clean and can proceed to analysis. Positive M values indicate higher expression in the red channel and negative M values in the green one. However, for which genes these statements are statistically significant, i.e for which genes do we have evidence to say that they are differentially expressed? For this purpose, one can conduct a statistical test by computing the *Z-score* of all genes.

2.4 *Z-score* test

The *Z-score* is the number of standard deviations that a result is from the mean. With the *Z-scores*, if associated with a standard normal distribution, it is possible to analyze with a defined confidence level which genes are expressed in a given channel. Computing all *Z-scores* using *genArise* library `Zscore()` function, for M vs A analysis.

```
zscore.ds <- Zscore(c1_cnorm, type="ma")
zscore.ds <- zscore.ds@dataSets # gene z-scores
```

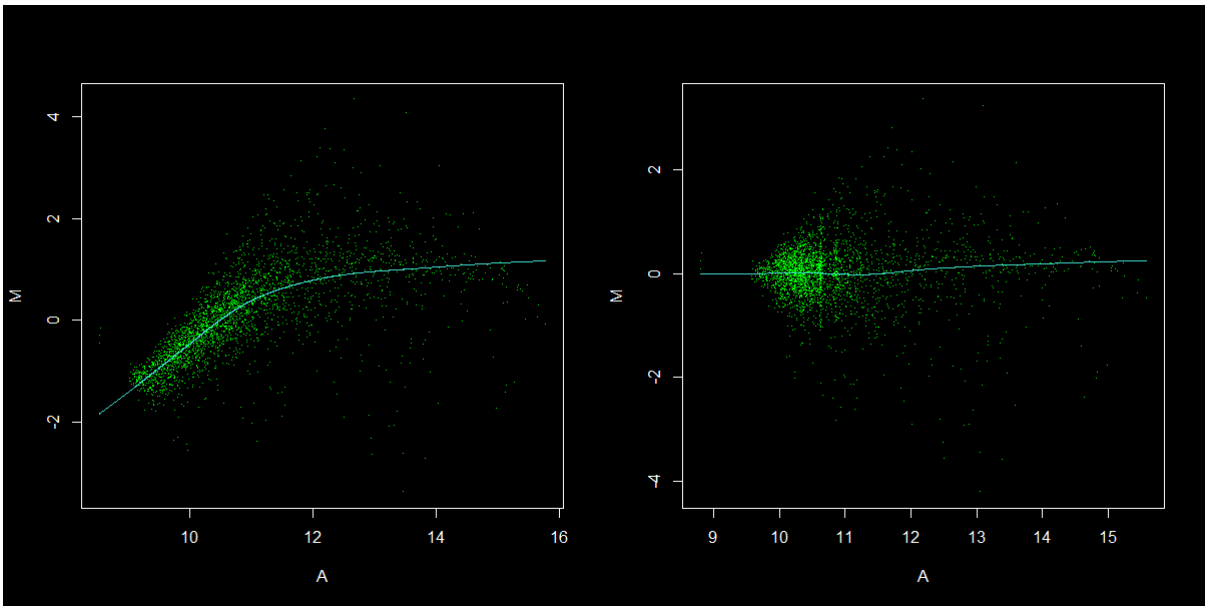

Figure 3: Comparison of the M vs A plots of the *chip1.txt* data before (left) and after (right) normalization. The data was previously background corrected.

Plotting the results into a histogram, it is confirmed that the *Z-scores* follow, in a qualitative way, a normal distribution. Computing its mean and standard deviation, ~ -0.00058 and ~ 0.995 respectively, it's possible to say that the variables are indeed *Z-scores* since the mean approaches zero and the standard deviation the unit. The result can be visualized in Figure 4. This way, for a 95% confidence level, i.e a *p-value* of 0.05, the critical *Z-score* values are -1.96 and +1.96 standard deviations. If the *Z-score* falls outside this range, the observed value is probably too unusual to be the result of random chance. Since the *Z-scores* are analyzed for M vs A, i.e $\log_2$(Cys5/Cys3), positive values indicate expression in the arterial regimen, while negative in the venous regimen. Lists of genes differentially expressed (**deg**) in the different regimens can then be created.

```
cutoff <- 1.96 # 95% confidence level in z-test
ven <- which(zscore.ds@dataSets$Zscore < -cutoff) # ven deg indexes
art <- which(zscore.ds@dataSets$Zscore > cutoff) # art deg indexes
```

```
> zscore.ds@dataSets$Id[ven] # ven deg IDs
  [1] "Id277"  "Id815"  "Id812"  "Id581"  "Id546"  "Id24"   "Id814"  "Id2699"
  [9] "Id2154" "Id286"  "Id2971" "Id53"   "Id498"  "Id1626" "Id2972" "Id1362"
 [17] "Id91"   "Id549"  "Id324"  "Id328"  "Id454"  "Id19"   "Id136"  "Id1100"
 [25] "Id1098" "Id168"  "Id115"  "Id762"  "Id110"  "Id478"  "Id504"  "Id2728"
 [33] "Id97"   "Id69"   "Id2980" "Id78"   "Id96"   "Id235"  "Id1580" "Id1379"
 [41] "Id518"  "Id139"  "Id2657" "Id111"  "Id769"  "Id164"  "Id2981" "Id2533"
 [49] "Id116"  "Id145"  "Id369"  "Id117"  "Id610"  "Id1840" "Id636"  "Id252"
 [57] "Id1830" "Id336"  "Id703"  "Id1674" "Id1446" "Id487"  "Id2986" "Id476"
 [65] "Id1394" "Id1271" "Id2656" "Id1579" "Id491"  "Id1217" "Id80"   "Id2093"
 [73] "Id2359" "Id2168" "Id1819" "Id15"   "Id39"   "Id638"  "Id2360" "Id253"
```

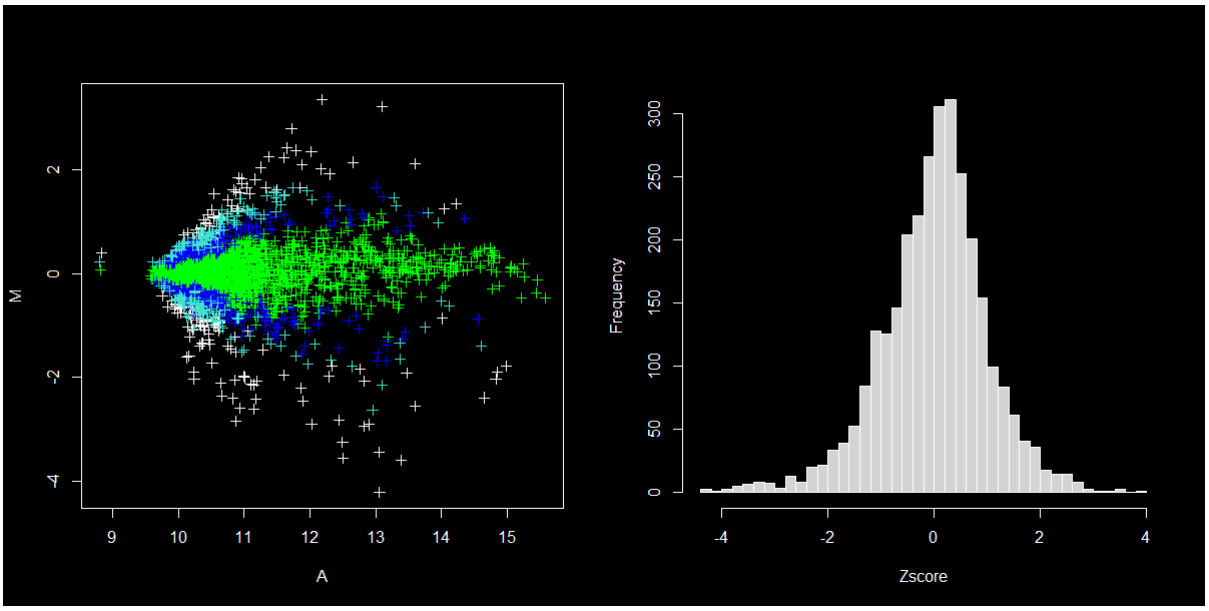

Figure 4: Visualization of the *Z-scores* of *chip1.txt* treated data. **Left** M vs A plot with points with the absolute values of *Z-score* lower than 1, between 1 and 1.5, between 1.5 and 2 and larger than 2 represented as green, blue, cyan and white respectivelly. **Right** Bell-shaped *Z-scores* histogram. The distribution mean approaches zero and a unit standart variation.

```
 [81] "Id2089" "Id605"  "Id2722" "Id76"   "Id2357" "Id2356" "Id2095" "Id2091"
 [89] "Id2595" "Id2618" "Id2642" "Id2748" "Id483"  "Id89"   "Id126"  "Id2749"
 [97] "Id1979" "Id2723" "Id440"  "Id2715"
> zscore.ds@dataSets$Id[art] # art deg IDs
 [1] "Id1892" "Id2684" "Id2852" "Id2565" "Id2625" "Id2584" "Id2593" "Id2906"
 [9] "Id2062" "Id1482" "Id2272" "Id1511" "Id2063" "Id2389" "Id2046" "Id2911"
[17] "Id1548" "Id2570" "Id535"  "Id387"  "Id2830" "Id2306" "Id2647" "Id2330"
[25] "Id354"  "Id2207" "Id393"  "Id2624" "Id420"  "Id1582" "Id1989" "Id2899"
[33] "Id2520" "Id484"  "Id2606" "Id297"  "Id2238" "Id2247" "Id2528" "Id233"
[41] "Id1638" "Id2879" "Id2598" "Id435"  "Id2958" "Id1555" "Id372"  "Id471"
[49] "Id2066" "Id1914" "Id281"  "Id501"  "Id2037" "Id152"  "Id544"  "Id347"
[57] "Id444"  "Id95"   "Id428"  "Id134"  "Id1399" "Id1583" "Id884"  "Id1726"
[65] "Id1458"
```

3 Differential Gene Expression

3.1 Chip Analysis Generalization

Two more patients were operated and two arrays under the same conditions as the first were obtained. The intensities are registered in `chip2.txt` and `chip3.txt`. The analysis performed on the first chip can be replicated with the new data. For this, `chipAnalysis()` was created.

```
M1 = log2(c1_cnorm@spotData$Cy5) - log2(c1_cnorm@spotData$Cy3)
chip1 <- list("M"=M1, "zscore"=zscore.ds, "ven"=ven, "art"=art)

chipAnalysis <- function(filename) {
  chip.spot <- read.spot(filename, cy3 = "Ven", cy5 = "Art",
                         bg.cy3 = "BgVen", bg.cy5 = "BgArt",
                         ids = "ID", header = T, sep = "\t", is.ifc = F)

  c_bgcor <- bg.correct(chip.spot)
  c_cnorm <- global.norm(c_bgcor)
  zscore.ds <- Zscore(c_cnorm, type="ma")

  cutoff <- 1.96 # 95% confidence level in z-test
  ven <- which(zscore.ds@dataSets$Zscore < -cutoff) # ven deg indexes
  art <- which(zscore.ds@dataSets$Zscore > cutoff) # art deg indexes

  M = log2(c_cnorm@spotData$Cy5) - log2(c_cnorm@spotData$Cy3)

  return(list("M"=M, "zscore.ds"=zscore.ds, "ven"=ven, "art"=art))
}

chip2 <- chipAnalysis("chip2.txt")
chip3 <- chipAnalysis("chip3.txt")
```

`chipn` is a list containing treated data from the nth patient. It contains the **M** values of the background-corrected, normalized data; the **zscore**s of the same data, and the lists of differentially expressed genes in Art and Ven regimens, **art** and **ven** respectively. By analyzing the lists of different samples, a set of unique genes with differential expressions can be created.

3.2 List intersection

The first logical approach to create this list is to make the intersection of the `art` and `ven` lists of the different chips. These lists contain a few dozens of genes, being the smallest composed of 30 genes (`ven` of chip2) and the largest of 142 (`art` of chip2).

```
> intersect(c1_art, c2_art)
[1]  871  987 1149 2392 2827
> intersect(c2_art, c3_art)
[1]  841  903 1350 1500 2028
> intersect(c1_art, c3_art)
[1] 1622 1867 2901
> intersect(c1_ven, c2_ven)
[1]  189 1924 2874
> intersect(c2_ven, c3_ven)
integer(0)
```

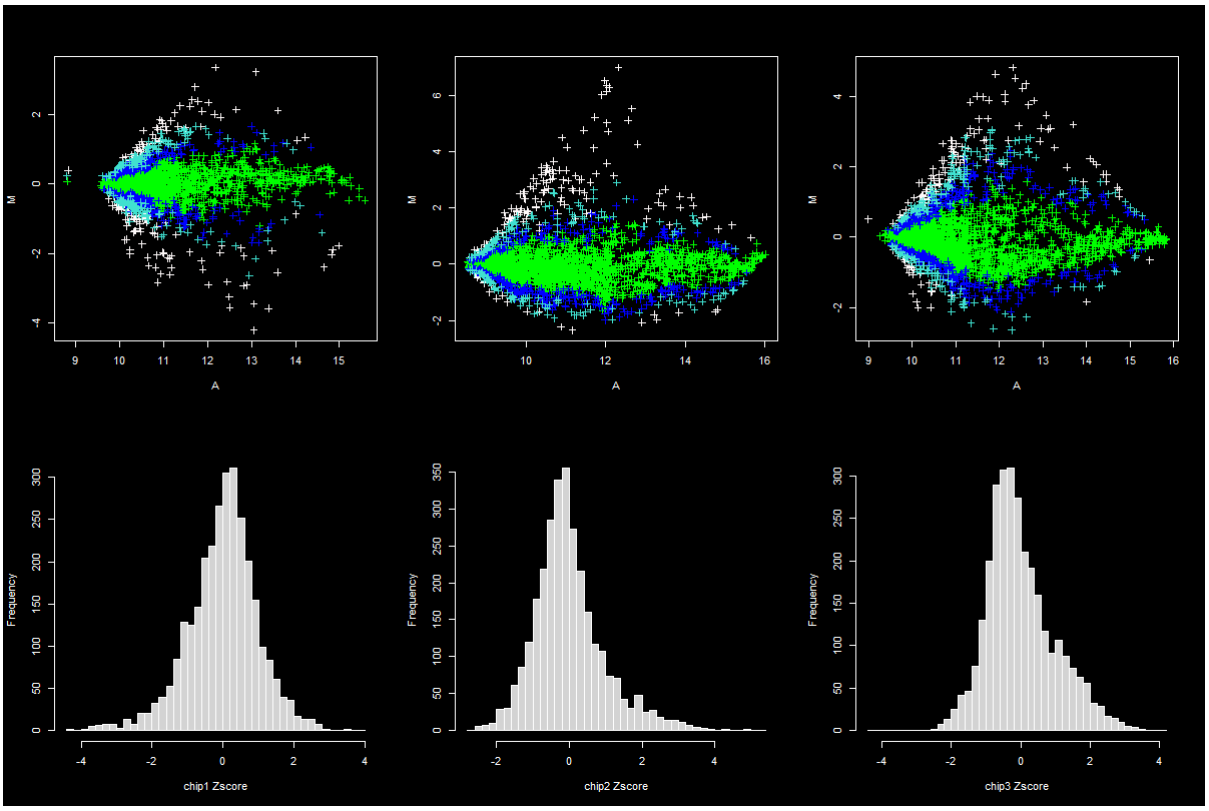

Figure 5: Visualization of the *Z-scores* of *chip1.txt*, *chip2.txt*, and *chip3.txt* treated data. **Top** M vs A plots with points with the absolute values of *Z-score* lower than 1, between 1 and 1.5, between 1.5 and 2 and larger than 2 represented as green, blue, cyan and white respectivelly. **Bottom** Bell-shaped *Z-scores* histograms, with a distribution mean of around zero and a unit standart deviation.

```
> intersect(c1_ven, c3_ven)
integer(0)
> intersect(intersect(c1_art, c2_art), c3_art)
integer(0)
```

The analysis, however, reveals that the intersection of the lists is the null set. Supposing that genes encountered in at least one intersection are considered as differentially expressed, then the following (indexes) lists could be created:

$$DEG(Art) = \{841, 871, 903, 987, 1149, 1350, 1500, 1622, 1867, 2028, 2392, 2827, 2901\}$$

$$DEG(Ven) = \{189, 1924, 2874\}$$

Despite capturing the fact that the number of DEG in the Art regime is larger, this rule isn't of much interest since it doesn't consider the *Z-scores* of DEG discarded genes during the chip analysis.

A rough approach would be to sum the *Z-scores* of the different chips. Since this distribution would not behave as a *Z-score*, the function `scale()` is used. Analysing which genes

have statistical significance to be considered as a DEG, the list of **art_deg** and **ven_deg** are obtained.

```
totzscore <- chip1$zscore.ds@dataSets$Zscore +
             chip2$zscore.ds@dataSets$Zscore +
             chip3$zscore.ds@dataSets$Zscore
totzscore <- scale(totzscore)

cutoff <- 1.96 # 95% confidence level in z-test
art_deg <- which(totzscore > cutoff) # art deg indexes
ven_deg <- which(totzscore < -cutoff) # ven deg indexes
```

```
> art_deg # Art DEG indexes
  [1]    1   87  129  184  268  301  314  385  386  424  583  585  618  622
 [15]  650  714  730  753  771  840  841  864  871  903  941  987 1079 1146
 [29] 1149 1193 1207 1237 1249 1267 1300 1306 1312 1350 1351 1384 1416 1433
 [43] 1480 1485 1500 1532 1597 1620 1622 1642 1683 1708 1714 1729 1763 1772
 [57] 1786 1810 1828 1859 1867 1871 1893 1910 1916 1945 1969 1980 2027 2028
 [71] 2066 2101 2113 2136 2146 2191 2216 2254 2255 2264 2284 2319 2342 2352
 [85] 2354 2380 2392 2413 2423 2437 2450 2470 2494 2553 2554 2602 2686 2740
 [99] 2827 2901 2968
> ven_deg # Ven DEG indexes
 [1]   41  102  117  124  180  197  265  333  343  369  404  447  483  578
[15]  584  616  647  670  701  706  780  806  858  868  894  931  934 1017
[29] 1077 1087 1136 1189 1348 1424 1467 1472 1519 1556 1674 1677 1757 1785
[43] 1792 1924 1971 2002 2117 2153 2167 2240 2268 2418 2592 2620 2723 2770
[57] 2820 2858 2865 2874 2900 2932 2962 2966 2978 2980
```

The rough approach seems to average out the results of the different chips, maintaining the DEG encountered in the list intersections, and about the average number of elements. Note that these last lists are in indexes and the gene ID corresponds to that index.

3.3 Bayesian method of Lonnstedt and Speed

An approach to improving on the t-statistic-based methods is the empirical Bayes method for analyzing replicated two-channel microarray data proposed by Loennstedt and Speed (2002) (Sousa, L. (2022). Analysis of Microarray Data [35]). For each gene, it regards its log ratio, $\mathbf{M_i}$, as a random variable with normal distribution and an indicator, $\mathbf{I_i}$, for whether a gene is differentially expressed or not. This way, the *B-statistic*, or **B**, is measured by Equation 1 representing the logarithm of posterior odds that that gene is differentially expressed.

$$B_i = \frac{\ln \mathrm{P}(\mathrm{I_i} = 1|\mathrm{M_i})}{\ln \mathrm{P}(\mathrm{I_i} = 0|\mathrm{M_i})} \tag{1}$$

So, if $B_i > 0$ the odd that gene i is differentially expressed is larger than one. The larger the B, the larger the odds of the gene being differentially expressed.

`limma+ package` (Phipson et al., 2016) provides functions that calculate the B-statistics for the genes in a series of microarrays. First, the data is fitted into the convenient MArrayLM object using `lmFit` function and the calculated M values. No design was set since the function defaults to the unit vector, i.e no dye swaps.

```
fit <- lmFit(cbind(chip1$M, chip2$M, chip3$M))
```

Then, the `eBayes()` function computes the different statistics of the object. These statistics are visualized with `topTable()`, where the number of top-ranked genes from the fitted model needs to be set. Setting this number to the top 10 genes, the following output is seen.

```
fit <- eBayes(fit)
```

```
> topTable(fit, number=10)
          logFC    AveExpr         t     P.Value adj.P.Val         B
2940 -1.4321107 -1.4321107 -5.221233 0.002878312 0.8787405 -4.391070
478  -1.2321108 -1.2321108 -4.861150 0.003970191 0.8787405 -4.397467
224   1.2826723  1.2826723  4.434195 0.005942133 0.8787405 -4.406566
2595 -1.7985885 -1.7985885 -4.100034 0.008293489 0.8787405 -4.415098
2319  1.8988617  1.8988617  4.052605 0.008706910 0.8787405 -4.416425
1493 -1.0445104 -1.0445104 -4.014341 0.009057633 0.8787405 -4.417519
2370  1.3681663  1.3681663  3.904021 0.010162455 0.8787405 -4.420789
104  -1.0240358 -1.0240358 -3.812493 0.011196432 0.8787405 -4.423642
207  -0.9234513 -0.9234513 -3.677121 0.012952276 0.8787405 -4.428105
2915  1.2285642  1.2285642  3.668245 0.013077884 0.8787405 -4.428409
```

No genes have the B-statistic larger than zero, meaning that using this method in this experiment's conditions, no genes are differentially expressed.

4 Conclusions

In the paper, treatment and analysis of the data of three Background Two Channel Microarrays studies was conducted. The data was background corrected, normalized, and *Z-score* tested, yielding different lists of differentially expressed genes for each chip. How can the list of DEGs of each regime be determined based on our data? In subsection 3.2 list intersection methods are explored and in subsection 3.3 a more sophisticated empirical Bayes method. Both yielded that that are no differentially expressed genes in these experimental conditions, although a rough attempt at creating a DEG list is made in the list intersection chapter.

What do these results say about the experiment? Since no DEG was found to be differentially expressed in the different samples and, besides that, some were being over-expressed in a sample while under-expressed in another, it seems that the regimen that the saphenous vein tissues were submitted to didn't influence the studied genes. The analyses of more samples would be needed to refine the results.

5 References

1. Hou Z, Jiang P, Swanson SA, Elwell AL, Nguyen BK, Bolin JM, Stewart R, Thomson JA. A cost-effective RNA sequencing protocol for large-scale gene expression studies. Sci Rep. 2015 Apr 1;5:9570. doi: 10.1038/srep09570. PMID: 25831155; PMCID: PMC4381617.

2. Mayén APG, Guillé GC, Ruiz LR, Coutiño GC. The genArise Package. (2006, January 6). http://www.ifc.unam.mx/genarise/pdfs/genArise_images.pdf

3. Sousa, L. (2022). Analysis of Microarray Data

4. Phipson, B, Lee, S, Majewski, IJ, Alexander, WS, and Smyth, GK (2016). Robust hyperparameter estimation protects against hypervariable genes and improves power to detect differential expression. Annals of Applied Statistics 10(2), 946–963.