

Visualization of microarray data

Statistical analysis of gene expression data with R and Bioconductor
University of Copenhagen

Copenhagen Biocenter

Laurent Gautier^{1,2}

August 17-21 2009

Contents

1	Basic graphics	1
2	Plots	1
3	Heatmaps	5
4	Spatial Information	7
5	Networks	8

1 Basic graphics

2 Plots

- Load the data package `gashYHS`. What does the data package contain ?
- Make a subset of the *ExpressionSet* that contains only the samples treated with "diamide_1.5mM"
- Filter out probe signal having the smallest log-ratios
- Plot the probe signal over time for those samples. How many groups of probe/gene profiles would you make from a visual inspection ?

```

> library(gaschYHS)

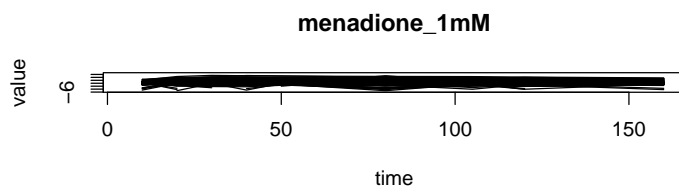
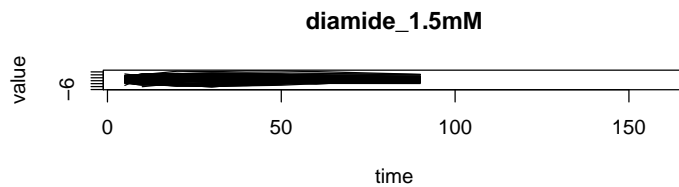
> library(help=gaschYHS)

> data(gaschYHS)

> trt <- c("diamide_1.5mM", "menadione_1mM")
> eset <- gaschYHS[, pData(gaschYHS)$trt %in% trt]
> myfilter <- function(x, threshold=3) {
+   any( abs(x) > threshold )
+ }
> eset <- eset[which(esApply(eset, 1, myfilter)), ]

> # old school
>
> par(mfrow = c(2,1))
> for (trt_i in seq(along=trt)) {
+   plot(0, 0, xlim = range(pData(eset)$time),
+        ylim = range(exprs(eset), na.rm=TRUE),
+        xlab = "time", ylab = "value",
+        type="n", main = trt[trt_i])
+   for (row_i in 1:nrow(exprs(eset))) {
+     eset_subset <- eset[, pData(eset)$trt == trt[trt_i]]
+     points(pData(eset_subset)$time,
+            exprs(eset_subset)[row_i, ],
+            type="l")
+   }
+ }

```



```

> library(reshape)

> library(reshape)

> # new school
>
> dataf <- melt(exprs(eset),
+              varnames = c("symbol", "sample"))

```

```

> dataf <- merge(dataf, pData(eset),
+               by.x = "sample", by.y = "samp")
> dataf$trt <- factor(dataf$trt)

```

```

> library(ggplot2)

```

```

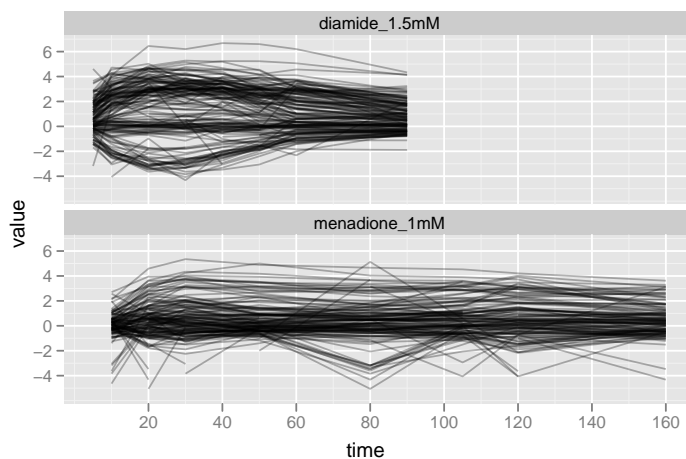
> library(ggplot2)

```

```

> p <- ggplot(dataf) +
+   aes(x = time, y=value, group=symbol) +
+   geom_line(alpha=0.3) +
+   facet_wrap(~ trt, ncol=1)
> print(p)

```



```

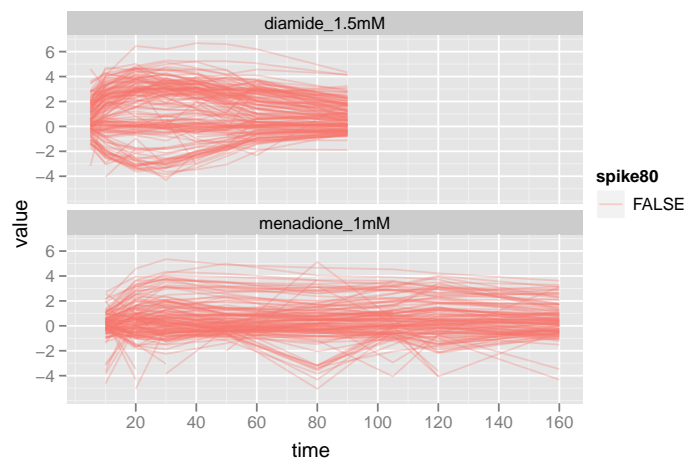
> col_i <-
+   with(pData(eset),
+       time == 80 & trt == "menadione_1mM")
> probe_i <- which(exprs(eset)[,col_i] < -2)

```

```

> # new school
>
> dataf <- cbind(dataf, spike80 = FALSE)
> dataf$selected[ dataf$symbol %in% names(probe_i)] <- TRUE
> p <- ggplot(dataf) +
+   aes(x = time, y=value, group=symbol, col=spike80) +
+   geom_line(alpha=0.3) +
+   facet_wrap(~ trt, ncol=1)
> print(p)

```



3 Heatmaps

1. load the dataset *GolubEsets*
2. Plot a heatmap for the 300 genes with the highest variance, using a correlation-based distance

```
> library(golubEsets)
> data(Golub_Merge)
> library(bioDist)
> library(RColorBrewer)

> library(golubEsets)
> data(Golub_Merge)
> library(bioDist)
> library(RColorBrewer)

> eset ← Golub_Merge
> gene_var ← esApply(eset, 1, var)
> o ← order(gene_var, decreasing = TRUE)
> heatmap(exprs(eset)[o[1:300], ], distfun = cor.dist)
```

4 Spatial Information

- Select an ‘interesting gene’ from the *gaschYHS* dataset used earlier
- Plot genes in the region using biomaRt
- What information is missing for having an accurate plot of expression signal from microarrays ?

```
> library(biomaRt)
> library(GenomeGraphs)

> library(biomaRt)
> library(GenomeGraphs)

> mart ← useMart("ensembl", "scerevisiae_gene_ensembl")
> startpos ←
+   getBM(c("ensembl_gene_id",
+         "start_position",
+         "chromosome_name",
+         "strand"),
+       filters = "ensembl_gene_id",
+       values = c("YER065C", "YER188W", "YFR032C",
+                 "YGL138C", "YHL031C", "YHR033W"),
+       mart=mart)
> minbase ← 270000
> maxbase ← 300000
> genesplus ← makeGeneRegion(start = minbase,
+                             end = maxbase,
+                             strand = "+",
+                             chromosome = "V",
+                             biomart = mart,
+                             DisplayPars(plotId = TRUE,
+                                       idRotation = 90,
+                                       idColor = "gray")
+                             )
> genesmin ← makeGeneRegion(start = minbase,
+                            end = maxbase,
+                            strand = "-",
+                            chromosome = "V",
+                            biomart = mart,
+                            DisplayPars(plotId = TRUE,
+                                       idRotation = -90,
+                                       idColor = "gray")
+                            )
> axis_trk ← makeGenomeAxis(add53 = TRUE,
+                           add35=TRUE,
+                           littleTicks = TRUE)
> p ← gdPlot(list(genesplus, axis_trk, genesmin),
+            minBase = minbase, maxBase = maxbase)
> print(p)
```

NULL

5 Networks

1. Create a mart with `biomaRt::useMart("ensembl", "hsapiens_gene_ensembl")`
- 2.

```
> library(altcdfenvs)
> library(biomaRt)
> library(hgu133plus2probe)
> library(RColorBrewer)
> library(Rgraphviz)

> library(altcdfenvs)
> library(biomaRt)
> library(hgu133plus2probe)
> library(RColorBrewer)
> library(Rgraphviz)
```

sessionInfo()

- R version 2.9.1 Patched (2009-08-09 r49124), i686-pc-linux-gnu
- Locale: LC_CTYPE=en_US.UTF-8;LC_NUMERIC=C;LC_TIME=en_US.UTF-8;LC_COLLATE=en_US.UTF-8;LC_MONETARY=C;LC_MESSAGES=en_US.UTF-8;LC_PAPER=en_US.UTF-8;LC_NAME=C;LC_ADDRESS=C;LC_TELEPHONE=C;LC_IDENTIFICATION=C
- Base packages: base, datasets, graphics, grDevices, grid, methods, stats, utils
- Other packages: affy 1.22.0, affyio 1.12.0, altcdfenvs 2.6.0, AnnotationDbi 1.6.1, Biobase 2.4.1, bioDist 1.16.0, biomaRt 2.0.0, Biostrings 2.12.8, gaschYHS 1.0.0, GenomeGraphs 1.4.1, ggplot2 0.8.3, golubEsets 1.4.5, graph 1.22.2, hgu133plus2probe 2.4.0, hypergraph 1.16.0, IRanges 1.2.3, KernSmooth 2.23-2, makecdfenv 1.22.0, matchprobes 1.16.0, plyr 0.1.9, proto 0.3-8, RColorBrewer 1.0-2, reshape 0.8.3, Rgraphviz 1.22.1, startupmsg 0.6, SweaveListingUtils 0.3.3
- Loaded via a namespace (and not attached): DBI 0.2-4, preprocessCore 1.6.0, RCurl 0.98-1, RSQLite 0.7-1, tools 2.9.1, XML 2.6-0