

# Chromosomal Transcriptional Patterns

Andreas Bunes

May 29, 2007

This example explains how to apply the method detailed in the manuscript 'Identification of aberrant chromosomal regions from gene expression microarray studies applied to human breast cancer' to a microarray gene expression data set. The method aims to identify spatial transcriptional patterns on the genome. The R source codes which implements this unsupervised approach is loaded (files for download are available at <http://www.dkfz.de/mga2/people/bunes/CTP/> including this document `ctp.pdf`).

```
> source("calculateStatistics.R")
> source("calculatePValues.R")
```

Installation of the packages required for running this example:

```
> source("http://www.bioconductor.org/biocLite.R")
> biocLite("Biobase")
> biocLite("macat")
> biocLite("hgu95av2")
```

Loading of the packages:

```
> library(Biobase)
> library(macat)
> library(hgu95av2)
> loaddatapkg("stjudem")
```

The example as used by the MACAT approach is chosen to demonstrate the usage of the codes. MACAT identifies genes on chromosome 6 in a supervised analysis while comparing class 43 'T' samples versus all 284 other samples. You may run the MACAT example by executing the following command or use a web browser like `konqueror` to visualize the result which predominantly points to the cytoband 6p21.3, cf. file 'Results6\_T.html'.

```
> demo(macatdemo)
> konqueror("Results6_T.html")
```

In this example, the preprocessing of the data accounts for duplicated 'gene-Name's and removes duplicated 'gene symbols'. As in the MACAT example only genes on chromosome 6 are considered.

```

> expr <- stjude$expr[stjude$geneName, ]
> selSix <- stjude$chromosome == "6"
> ids <- rownames(expr[selSix, ])
> chip <- stjude$chip
> thisEnv = paste(chip, "SYMBOL", sep = "")
> symbol = mget(ids, env = eval(as.symbol(thisEnv)))
> selSixFinal <- which(selSix)[!duplicated(symbol)]

```

All parameters and settings are given in the following. The implementation requires the expression matrix to be of dimensions samples times genes. The *paketSize* indirectly controls the memory usage. Here, about 1GB is required for 327 samples and a *paketSize* of 10000. There is a trade off between computation time and memory consumption. *nullLength* sets the number of draws with replacement. It should translate to multiples of *paketSize*-1. The *windowSize* defines the size of the sliding window, i.e. the number of genes. We restrict the analysis on sample subsets of size  $k = 1, \dots, 43$  to save computing time.

```

> exprSix <- t(expr[selSixFinal, ])
> paketSize <- 10000
> nullLength <- 5 * paketSize - 1
> windowSize <- 20
> maxK <- 43

```

The calculation of  $p_{k,G'}$  for up- and downregulated subsets takes approximately one day.

```

> p <- calculatePValues(expr = exprSix, windowSize = windowSize,
+   nullLength = nullLength, paketSize = paketSize, maxCluster = maxK,
+   verbose = TRUE)
> save(p, file = "p_cached.RData", compress = TRUE)

```

Here, the cached result is loaded.

```

> load("p_cached.RData")

```

The focus of the analysis is set to subsets of size  $k = 43$  while aiming to directly compare with the result of the supervised approach. MACAT's analysis is based on the contrast between all 43 samples of class 'T' versus the rest. In this analysis the ten most significant windows are selected.

```

> p43up <- p[43, , 1]
> topSel <- order(p43up)[1:10]
> tx <- sort(unique(as.vector(sapply(topSel, function(x) seq(x,
+   x + windowSize - 1))))))
> numberOfRegions <- sum((tx[-1] - tx[-length(tx)]) > 1) + 1
> print(numberOfRegions)

```

```

[1] 2

```

The ten most significant windows cover 2 regions on chromosome 6. The result overlaps to a large extent with the one of the supervised approach MACAT, especially on cytoband 6p21.3, cf. file 'Results6\_T.html'. However, this is merely an example to demonstrate the usage of the R source codes. For example, issues like batch effects, correlated gene expression of gene families, etc. have not been investigated, but are discussed in the manuscript.

```

> genes <- names(p43up[tx])
> thisEnv <- paste(chip, "MAP", sep = "")
> cytoband <- mget(genes, env = eval(as.symbol(thisEnv)))
> thisEnv <- paste(chip, "SYMBOL", sep = "")
> symbol <- mget(genes, env = eval(as.symbol(thisEnv)))
> print(cbind(GDash = tx, symbol = unlist(symbol), cytoband = unlist(cytoband)))

```

	GDash	symbol	cytoband
35341_at	"61"	"TRIM38"	"6p21.3"
33067_at	"62"	"HIST1H1A"	"6p21.3"
33049_at	"63"	"HIST1H4B"	"6p21.3"
35598_at	"64"	"HIST1H3E"	"6p21.3"
37018_at	"65"	"HIST1H1C"	"6p21.3"
38913_at	"66"	"HFE"	"6p21.3"
31730_at	"67"	"HIST1H1T"	"6p21.3"
32980_f_at	"68"	"HIST1H2BG"	"6p21.3"
34308_at	"69"	"HIST1H2AC"	"6p21.3"
580_at	"70"	"HIST1H1E"	"6p21.3"
38576_at	"71"	"HIST1H2BD"	"6p21.3"
31523_f_at	"72"	"HIST1H2BE"	"6p21.3"
31693_f_at	"73"	"HIST1H2AD"	"6p21.3"
31522_f_at	"74"	"HIST1H2BF"	"6p21.3"
35127_at	"75"	"HIST1H2AE"	"6p22.2-p21.1"
33030_at	"76"	"HIST1H1D"	"6p21.3"
39969_at	"77"	"HIST1H4C"	"6p21.3"
31524_f_at	"78"	"HIST1H2BI"	"6p21.3"
38759_at	"79"	"BTN3A2"	"6p22.1"
32629_f_at	"80"	"BTN3A1"	"6p22.1"
38241_at	"81"	"BTN3A3"	"6p21.3"
32673_at	"82"	"BTN2A1"	"6p22.1"
36335_at	"83"	"BTN1A1"	"6p22.1"
35737_at	"84"	"HMGN4"	"6p21.3"
153_f_at	"85"	"HIST1H2BJ"	"6p22.1"
284_at	"86"	"HIST1H2AG"	"6p22.1"
32819_at	"87"	"HIST1H2BK"	"6p21.33"
762_f_at	"88"	"HIST1H4I"	"6p21.33"
37039_at	"187"	"HLA-DRA"	"6p21.3"
36108_at	"188"	"HLA-DQB1"	"6p21.3"
38570_at	"189"	"HLA-DOB"	"6p21.3"
39988_at	"190"	"TAP2"	"6p21.3"
40153_at	"191"	"TAP1"	"6p21.3"
38287_at	"192"	"PSMB9"	"6p21.3"
41609_at	"193"	"HLA-DMB"	"6p21.3"
37344_at	"194"	"HLA-DMA"	"6p21.3"
36208_at	"195"	"BRD2"	"6p21.3"
31467_at	"196"	"HLA-DOA"	"6p21.3"
38833_at	"197"	"HLA-DPA1"	"6p21.3"
38095_i_at	"198"	"HLA-DPB1"	"6p21.3"
1026_s_at	"199"	"COL11A2"	"6p21.3"
1362_s_at	"200"	"RXRB"	"6p21.3"

41440_at	"201"	"HSD17B8"	"6p21.3"
35685_at	"202"	"RING1"	"6p21.3"
35278_at	"203"	"RPS29"	"14q"
35963_at	"204"	"PFDN6"	"6p21.3"
40521_at	"205"	"RGL2"	"6p21.3"
41168_at	"206"	"TAPBP"	"6p21.3"
34178_at	"207"	"ZBTB22"	"6p21.3"