Fadhillah Putri Taha
H071171301

Summary Micro Array Analysis

Bagaimana cara mempelajari *preprocess probe data*, *filter genes*, berbagai program visualisasi, untuk menggunakan *gene ontology identifiers*, untuk memproses *public available gene expression data.*

6.1 Probe Data

Teknik *micro array* menggunakan property hibridasi dari asam nukleat, yang diberi label sehingga dapat memberikan gambaran kasar pada suatu objek molekul. Dimana data mentah dari *scanner affymetrix* disimpan di dalam suatu tempat/*stored* bernama file DAT, dan data yang akan diolah bernama file CEL.

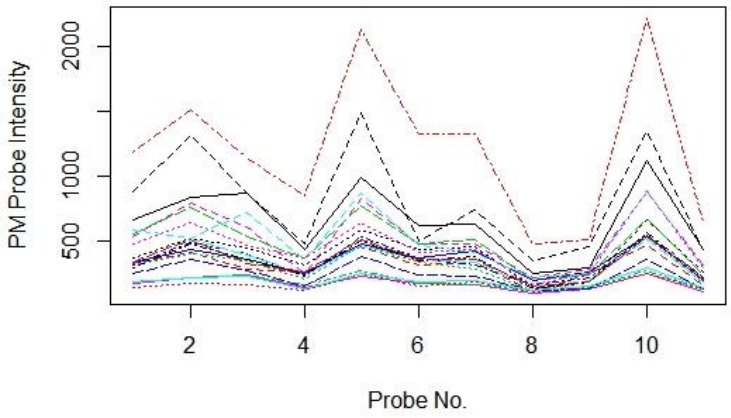
```
> #Chapter 6 - Micro Array Analysis
> #Example1
> #Membangun dataset MLL.B dari package ALLMLL
> library(affy)
> data(MLL.B, package = "ALLMLL")
> MLL.B
AffyBatch object
size of arrays=712x712 features (22 kb)
cdf=HG-U133B (22645 affyids)
number of samples=20
number of genes=22645
annotation=hgu133b
notes=
> #Mencetak struktur objek
> slotNames(MLL.B)
 [1] "cdfName"           "nrow"              "ncol"
 [4] "assayData"         "phenoData"         "featureData"
 [7] "experimentData"    "annotation"        "protocolData"
[10] ".__classVersion__"
> #Mencetak jumlah baris dan kolom nilai ekspresi MLL.B
> dim(exprs(MLL.B))
[1] 506944     20
> #Mengekstraksi anotasi
> annotation(MLL.B)
[1] "hgu133b"
> #mencetak 10 nama pertama dari probe yang digunakan
> probeNames(MLL.B)[1:10]
 [1] "200000_s_at" "200000_s_at" "200000_s_at" "200000_s_at"
 [5] "200000_s_at" "200000_s_at" "200000_s_at" "200000_s_at"
 [9] "200000_s_at" "200000_s_at"
```

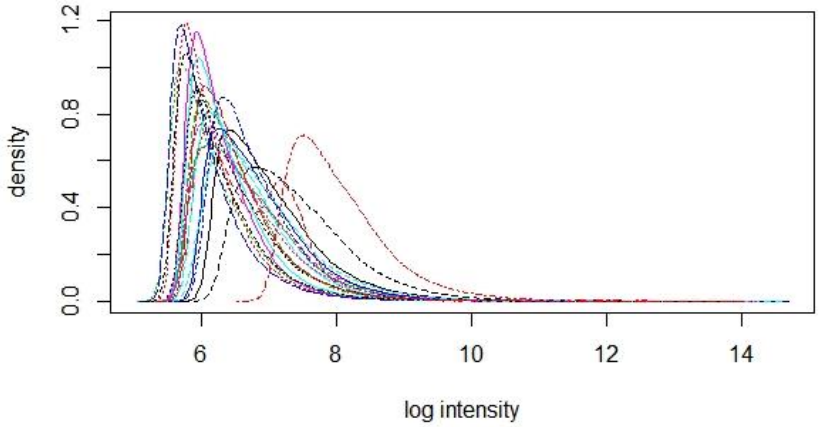
Fadhillah Putri Taha
H071171301

Summary Micro Array Analysis

Terdapat *package* bernama *affy* yang dapat digunakan untuk membaca data vector yang telah ditentukan dari *scanner affymetrix.*

```
> #Mencetak nilai PM
> pm(MLL.B, "200000_s_at")[1:4,1:3]
             JD-ALD009-v5-U133B.CEL JD-ALD051-v5-U133B.CEL
200000_s_at1                  661.5                  321.5
200000_s_at2                  838.8                  409.3
200000_s_at3                  865.3                  275.5
200000_s_at4                  425.8                  253.5
             JD-ALD052-v5-U133B.CEL
200000_s_at1                  312.5
200000_s_at2                  395.3
200000_s_at3                  341.3
200000_s_at4                  196.8
> #Membuat grafik variabilitas data probe yang akan didapat - Plot nil
ai intensitas
> matplot(pm(MLL.B,"200000_s_at"),type="l", xlab = "Probe No.",ylab="P
M Probe Intensity")
> hist(MLL.B)
Hit <Return> to see next plot: matplot(MLL.B, pairs = TRUE, plot.metho
d="smoothScatter") #ndabisa
> image(MLL.B)
```





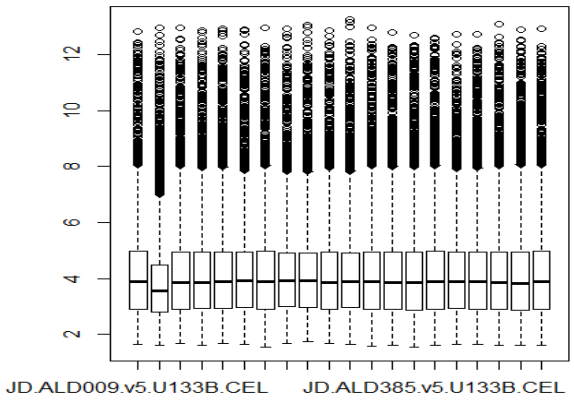
Fadhillah Putri Taha
H071171301

Summary Micro Array Analysis

6.2 Preprocessing Methods

Agar dapat membuat kesimpulan yang relevan secara biologis, metode *preprocessing probe intensitas* sangat perlu dilakukan. Dan metode ini dilakukan dalam tiga langkah, yaitu pengoreksian latar belakang, normalisasi, dan *summarization.*

```
> #Preprocessing Methods> #Mencetak latarbelakang yang tersedia
> bgcorrect.methods()
[1] "bg.correct" "mas"        "none"       "rma"
> #Mengoreksi nilai PM - menghapus nilai MM
> pmcorrect.methods()
[1] "mas"        "methods"    "pmonly"     "subtractmm"
> #Normalisasai Data
> normalize.methods(MLL.B)
[1] "constant"         "contrasts"        "invariantset"
[4] "loess"            "methods"          "qspline"
[7] "quantiles"        "quantiles.robust"
> #Pengumpulan nilai intensitas probe menjadi ekspresi gen
> express.summary.stat.methods()
[1] "avgdiff"      "liwong"       "mas"          "medianpolish"
[5] "playerout"
> #Example 1
> #Menggabungkan koreksi latar belakang RMA dengan normalisasi konstan
> eset <- expresso(MLL.B, bgcorrect.method = "rma", normalize.method = "
constant", pmcorrect.method = "pmonly", summary.method = "avgdiff")
background correction: rma
normalization: constant
PM/MM correction : pmonly
expression values: avgdiff
background correcting...done.
normalizing...done.
22645 ids to be processed
|                    |
|####################|
> #Example 2
> #Menggabungkan koreksi latar belakang konvolusi, normalisasi kuantil,
dan peringkasan berdasarkan model multi-array sesuai dengan cara yang ku
at oleh algoritma polian median
> library(affy)
> data(MLL.B, package = "ALLMLL")
> eset3 <- rma(MLL.B)
Background correcting
Normalizing
Calculating Expression
> boxplot(data.frame(exprs(eset3)))
```



Fadhillah Putri Taha
H071171301

Summary Micro Array Analysis

```
> #Example 3
> #Memperoleh gambaran umum tentang jumlah pasien yang berada dalam fase
penyakit tertentu
> #BiocManager::install("ALL")
> library(ALL)
> data(ALL, package = "ALL")
> slotNames(ALL)
[1] "experimentData"    "assayData"         "phenoData"
[4] "featureData"       "annotation"        "protocolData"
[7] ".__classVersion__"
> row.names(exprs(ALL))[1:10]
 [1] "1000_at"   "1001_at"   "1002_f_at" "1003_s_at" "1004_at"
 [6] "1005_at"   "1006_at"   "1007_s_at" "1008_f_at" "1009_at"
> feno <- pData(ALL)
> #menghitung kolom mad dan median - mengurangi median dari
setiap entri kolom
> #Membagi setiap entri kolom oleh MAD
> ALL1pp <- ALL1 <- ALL[,ALL$mol == "ALL1/AF4"]
> mads <- apply(exprs(ALL1), 2, mad)
> meds <- apply(exprs(ALL1), 2, median)
> dat <- sweep(exprs(ALL1), 2, meds)
> exprs(ALL1pp) <- sweep(dat, 2, mads, FUN="/")
```

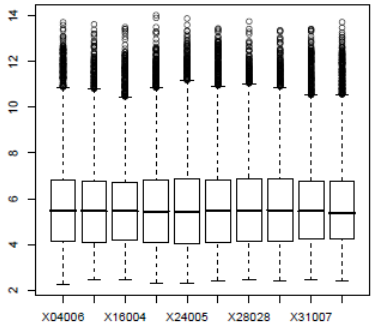


Figure 6.3: Boxplot of the ALL1/AF4 patients.

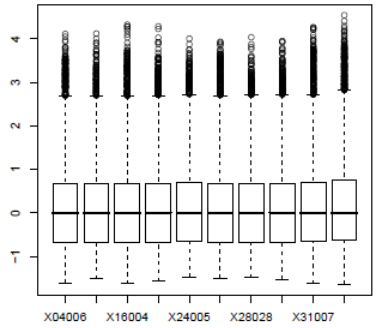


Figure 6.4: Boxplot of the ALL1/AF4 patients after median subtraction and MAD division.

6.3 Gene Filtering

Penyaringan gen.

```
> #Gene Filtering
> #Menghitung koefisien variasi per gen untuk data ALL1pp
> #Example1
> cvval <- apply(exprs(ALL1pp),1,function(x){sd(x)/abs(mean(x))})
> sum(cvval<0.2)
[1] 4751
```

Fadhillah Putri Taha
H071171301

Summary Micro Array Analysis

```
> #Example 2
> #if (!requireNamespace("BiocManager", quietly = TRUE))
> #install.packages("BiocManager")
> #BiocManager::install("genefilter")
> #Penggabungan beberapa filter
> library("genefilter")
> f1 <- function(x)(IQR(x)>0.5)
> f2 <- pOverA(.25, log2(100))
> f3 <- function(x) (median(2^x) > 300)
> f4 <- function(x) (shapiro.test(x)$p.value > 0.05)
> f5 <- function(x) (sd(x)/abs(mean(x))<0.1)
> f6 <- function(x) (sqrt(10)* abs(mean(x))/sd(x) > qt(0.975,9))
> ff <- filterfun(f1,f2,f3,f4,f5,f6)
> library("ALL"); data(ALL)
> selected <- genefilter(exprs(ALL[,ALL$BT=="B"]), ff)
> sum(selected)
[1] 317

> #Example 3
> #Membuat faktor logis patientB
> #menunjukkan pasien dengan B-cell ALL (TRUE) dan T-cell ALL (FALSE
)
> library("genefilter");library("ALL"); data(ALL)
> patientB <- factor(ALL$BT %in% c("B","B1","B2","B3","B4"))
> f1 <- function(x) (shapiro.test(x)$p.value > 0.05)
> f2 <- function(x) (t.test(x ~ patientB)$p.value < 0.05)
> sel1 <- genefilter(exprs(ALL[,patientB==TRUE]), filterfun(f1))
> sel2 <- genefilter(exprs(ALL[,patientB==FALSE]), filterfun(f1))
> sel3 <- genefilter(exprs(ALL), filterfun(f2))
> selected <- sel1 & sel2 & sel3
> ALLs <- ALL[selected,]
> sum(ALLs,)

> #Membuat diagram venn
> #BiocManager::install("limma")
> library(limma)
> x <- matrix(as.integer(c(sel1,sel2,sel3)),ncol = 3,byrow=FALSE)
> colnames(x) <- c("sel1","sel2","sel3")
> vc <- vennCounts(x, include="both")
> vennDiagram(vc)
Hit <Return> to see next plot: 2
Warning message:
In doTryCatch(return(expr), name, parentenv, handler) :
  display list redraw incomplete
```

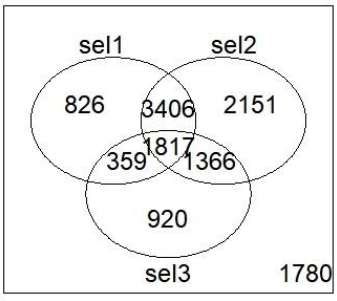


Diagram veen menunjukkan bahwa terdapat 1817 gen yang lulus pada tahapan tiga filter yang dioperasikan, terdapat 1780 gen yang telah gugur pada filter pertama, dan sebanyak 3406 gen yang termasuk *normality-test*, namun tidak lulus pada t-test.

Fadhillah Putri Taha
H071171301

Summary Micro Array Analysis

6.4 Apllications of Linear Models

Package limma sering digunakan untuk menganallisis dara *microarray* model linear, seperti ANOVA.

Menganalisis variansi, dengan tipe analisis ditentukan menggunakan factor yang menentukan model matriks. Kemudian, model linier disesuaikan dengan data dan prosedur *Bayes empiris* digunakan untuk mengadaptasi varian gen tertentu dengan panduga varians global.

```
> #Applications of Linear Models
> #Menganalisis variansi
> library("ALL"); library("limma");
> data(ALL, package = "ALL")
> allB <- ALL[,which(ALL$BT %in% c("B","B1","B2"))]
> design.ma <- model.matrix(~ 0 + factor(allB$BT))
> colnames(design.ma) <- c("B","B1","B2")
> fit <- lmFit(allB, design.ma)
> fit <- eBayes(fit)
> toptab <- topTable(fit, coef=2,5,adjust.method="fdr")
> print(toptab[,1:5],digits=4)
                logFC AveExpr     t   P.Value adj.P.Val
AFFX-hum_alu_at 13.42   13.50 326.0 3.165e-99 3.996e-95
32466_at        12.68   12.70 306.3 1.333e-97 8.413e-94
32748_at        12.08   12.11 296.3 9.771e-97 3.616e-93
35278_at        12.44   12.45 295.5 1.146e-96 3.616e-93
34593_g_at      12.64   12.58 278.0 4.431e-95 1.119e-91


> #Mencetak hipotesis tertenti
> cont.ma <- makeContrasts(B-B1,B1-B2, levels=factor(allB$BT))
> cont.ma
      Contrasts
Levels B - B1 B1 - B2
    B       1       0
    B1     -1       1
    B2      0      -1
> #Membuat kontras matriks
> fit1 <- contrasts.fit(fit, cont.ma)
> fit1 <- eBayes(fit1)
> toptabcon <- topTable(fit, coef=2,5,adjust.method="fdr")
> print(toptabcon[,1:5],digits=4)
                logFC AveExpr     t   P.Value adj.P.Val
AFFX-hum_alu_at 13.42   13.50 326.0 3.165e-99 3.996e-95
32466_at        12.68   12.70 306.3 1.333e-97 8.413e-94
32748_at        12.08   12.11 296.3 9.771e-97 3.616e-93
35278_at        12.44   12.45 295.5 1.146e-96 3.616e-93
34593_g_at      12.64   12.58 278.0 4.431e-95 1.119e-91
> toptabcon <- topTable(fit1, coef=2,5,adjust.method="fdr")
> print(toptabcon[,1:5],digits=4)
           logFC AveExpr      t   P.Value adj.P.Val
33358_at  1.4890   5.260  7.374 5.737e-10 7.242e-06
1389_at  -1.7852   9.262 -7.081 1.816e-09 9.744e-06
1914_at   2.0976   4.939  7.019 2.315e-09 9.744e-06
36873_at  1.8646   4.303  6.426 2.361e-08 7.452e-05
37471_at  0.8701   6.551  6.106 8.161e-08 2.061e-04
```

Method false discovery rate(tdr) digunakan untuk meningkatkan nilai-p dan mengurangi jumlah *false positive.*

Fadhillah Putri Taha
H071171301

Summary Micro Array Analysis

```
> #Example 2
> #Menggabungkan output khas
> #BiocManager::install("annaffy")
> #BiocManager::install("hgu95av2.db")
> library("annaffy");library("hgu95av2.db")
> saveHTML(anntable, "ALLB123.html", title = "B-cell 012 ALL")
Error in saveHTML(anntable, "ALLB123.html", title = "B-cell 012 ALL")
:
  object 'anntable' not found
> aafTableAnn()
```

Fungsi *aafTableAnn* dari data, dikumpulkan dari *output topTable* pada model linear, *annotation package*, dan fungsi *aaf.handler*. Data tersebut mengandung *probe*, symbol, *locus result* dari *anntable* dalam format *HTML* yang tersimpan pada *working directory* atau desktop.

Hgu95av2.db merupakan paket meta data yang menghubungkan permintaan informasi melalui *call to aaf.handler*. Tabel yang dihasilkan disimpan sebagai HTML pada *working directory* (getwd()) atau desktop.

```
> #Example 3
> #Merangkum hasil dalam tabel HTML berdasarkan nilai-p dari mis. ana
lisis varians (ANOVA)
> #BiocManager::install("multtest")
> library("multtest"); library("annaffy"); library("hgu95av2.db")
> library("ALL"); data(ALL, package = "ALL")
> ALLB <- ALL[,which(ALL$BT %in% c("B","B1","B2"))]
> panova <- apply(exprs(ALLB), 1, function(x) anova(lm(x ~ ALLB$BT))$
Pr[1])
> genenames <- featureNames(ALLB)[panova<0.000001]
> atab <- aafTableAnn(genenames, "hgu95av2.db", aaf.handler()[c(1:3,8
:9,11:13)])
Warning messages:
1: In result_fetch(res@ptr, n = n) :
  SQL statements must be issued with dbExecute() or dbSendStatement()
instead of dbGetQuery() or dbSendQuery().
2: In result_fetch(res@ptr, n = n) :
  SQL statements must be issued with dbExecute() or dbSendStatement()
instead of dbGetQuery() or dbSendQuery().
> saveHTML(atab, file="ANOVAonB-cellGroups.html")
```

Data *getGEO* diunduh ke disk dan selanjutnya akan diproses di *Rsystem.*

Fadhillah Putri Taha
H071171301

Summary Micro Array Analysis

```
> #Example 4
> #Menganalisis public available data
> #memilih gen dengan efek interaksi yang signifikan
> #BiocManager::install("GEOquery")
> library(GEOquery); library(limma); library(hgu95av2.db);library(annaffy)
> gds <- getGEO("GDS1365")
Using locally cached version of GDS1365 found here:
C:\Users\asus\AppData\Local\Temp\RtmpKQj22p/GDS1365.soft.gz
Parsed with column specification:
cols(
  .default = col_double(),
  ID_REF = col_character(),
  IDENTIFIER = col_character()
)
See spec(...) for full column specifications.
> eset <- GDS2eSet(gds,do.log2=T)
Using locally cached version of GPL8300 found here:
C:\Users\asus\AppData\Local\Temp\RtmpKQj22p/GPL8300.annot.gz
> prot <- pData(eset)$protocol
> time <- pData(eset)$time
> pvalt<- apply(exprs(eset)[1:12625,], 1,
> pvalt <- data.frame(t(pvalt))
> colnames(pvalt) <- c("meffprot","mefftime","interaction")
> genenames <- featureNames(eset)[pvalt$meffprot< 0.01 &
+   pvalt$mefftime < 0.01 & pvalt$interaction < 0.01]
> atab <- aafTableAnn(genenames,"hgu95av2.db",aaf.handler()[c(1:3,8:9,11:1
3)])
> saveHTML(atab, file="Two-way ANOVA protocol by time.html")
```

Fadhillah Putri Taha
H071171301

Summary Micro Array Analysis

6.5 Searching an annotation Package

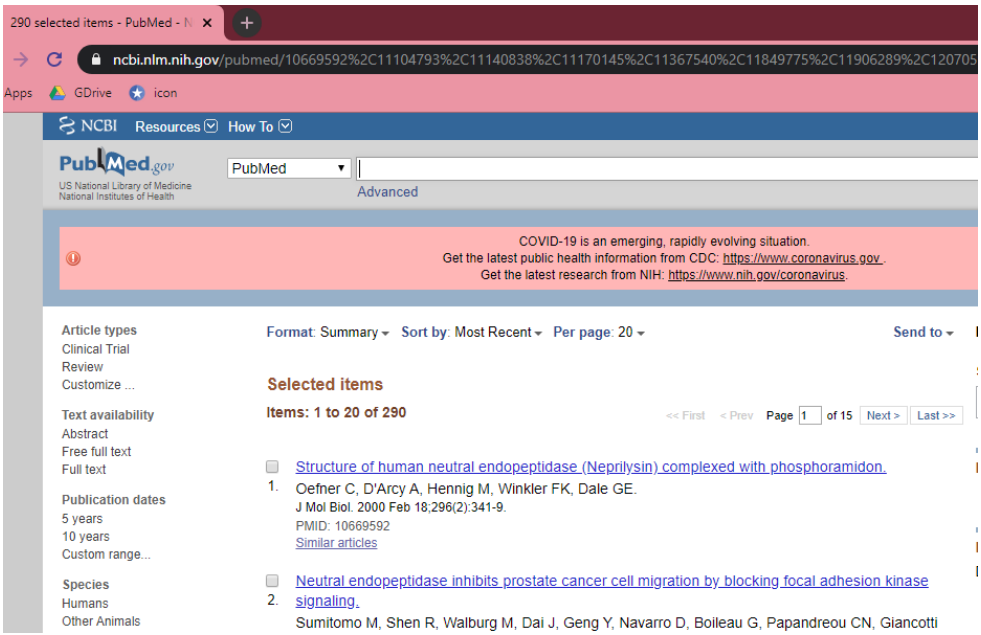
Rincian informasi pada *microarray* disimpan pada *annotation package*.

```
> #Mencari package anotasi
> library("ALL"); data(ALL)
> annotation(ALL)
[1] "hgu95av2"
> # the annotation package we need is hgu95av2.db
> library(hgu95av2.db)
> ls("package:hgu95av2.db")
 [1] "hgu95av2"               "hgu95av2.db"            "hgu95av2_dbconn"
 [4] "hgu95av2_dbfile"        "hgu95av2_dbInfo"        "hgu95av2_dbschema"
 [7] "hgu95av2ACCNUM"         "hgu95av2ALIAS2PROBE"    "hgu95av2CHR"
[10] "hgu95av2CHRLENGTHS"     "hgu95av2CHRLOC"         "hgu95av2CHRLOCEND"
[13] "hgu95av2ENSEMBL"        "hgu95av2ENSEMBL2PROBE" "hgu95av2ENTREZID"
[16] "hgu95av2ENZYME"         "hgu95av2ENZYME2PROBE"   "hgu95av2GENENAME"
[19] "hgu95av2GO"             "hgu95av2GO2ALLPROBES"   "hgu95av2GO2PROBE"
[22] "hgu95av2MAP"            "hgu95av2MAPCOUNTS"      "hgu95av2OMIM"
[25] "hgu95av2ORGANISM"       "hgu95av2ORGPKG"         "hgu95av2PATH"
[28] "hgu95av2PATH2PROBE"     "hgu95av2PFAM"           "hgu95av2PMID"
[31] "hgu95av2PMID2PROBE"     "hgu95av2PROSITE"        "hgu95av2REFSEQ"
[34] "hgu95av2SYMBOL"         "hgu95av2UNIGENE"        "hgu95av2UNIPROT"
> #Membuat konten environment
> ChrNrOfProbe <- as.list(hgu95av2CHR)
> ChrNrOfProbe[1]
$`1000_at`
[1] "16"
> #Mencari environment berdasarkan nama
> get("1389_at", env = hgu95av2ACCNUM)
[1] "J03779"
> get("1389_at", env = hgu95av2ENTREZID)
[1] "4311"
> get("1389_at", env = hgu95av2SYMBOL)
[1] "MME"
> get("1389_at", env = hgu95av2GENENAME)
[1] "membrane metalloendopeptidase"
> get("1389_at", env = hgu95av2SUMFUNC)
Error in get("1389_at", env = hgu95av2SUMFUNC) :
  object 'hgu95av2SUMFUNC' not found
> get("1389_at", env = hgu95av2UNIGENE)
[1] "Hs.307734"
> #Mencari nukleotida database
> library(annotate)
> genbank("J03779",disp="browser")
```

Fadhillah Putri Taha
H071171301

Summary Micro Array Analysis

```
> genbank(1430782,disp="data",type="uid")
Read 3 items
$`doc`
$`file`
[1] "<buffer>"
$version
[1] "1.0"
$children
$children$`Entrezgene-Set`
<Entrezgene-Set>
 <Error>GeneId 1430782 not found.</Error>
</Entrezgene-Set>
attr(,"class")
[1] "XMLDocumentContent"
$dtd
$`external`
NULL
$internal
$`elements`
NULL
$entities
NULL
attr(,"class")
[1] "InternalDTD"
attr(,"class")
[1] "DTDList"
attr(,"class")
[1] "XMLDocument"
[2] "XMLAbstractDocument"
> get("1389_at", env = hgu95av2CHRLOC)
        3         3         3         3
155079646 155079915 155080010 155080010
        3
155024123
> #Lokasi cytoband
> get("1389_at", env = hgu95av2MAP)
[1] "3q25.2"
> ll1<-GOENTREZID2GO[["4121"]]
Error: object 'GOENTREZID2GO' not found
```

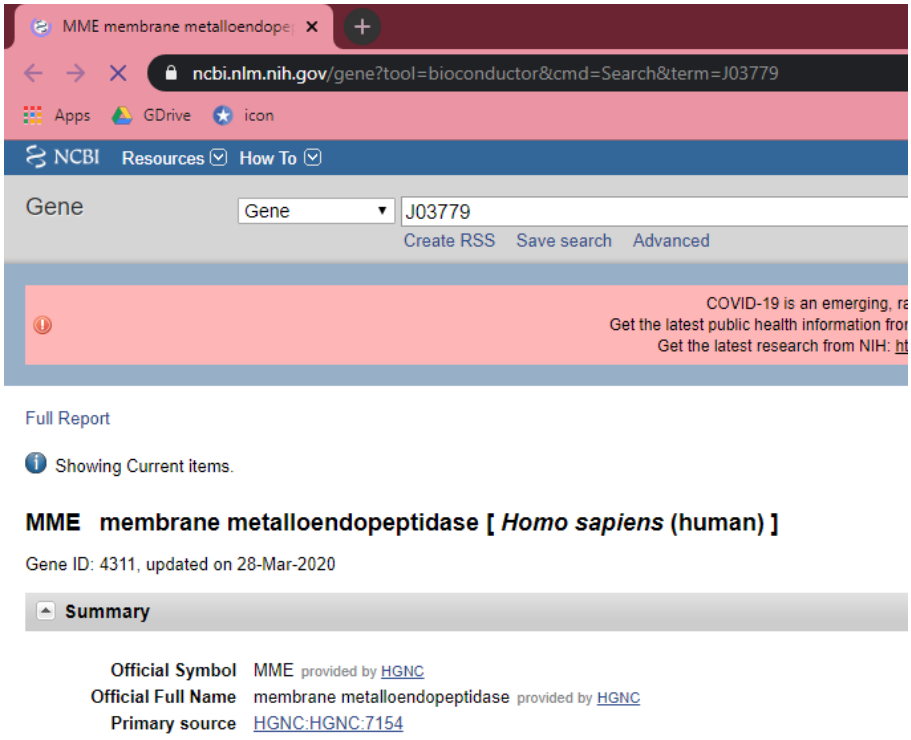


Fadhillah Putri Taha
H071171301

Summary Micro Array Analysis

6.6 Using Annotation to Search Literature

```
> #Pencarian Literatur menggunakan anotasi
> library(hgu95av2.db);library(annotate); library(ALL); data(ALL)
> pmid <- get("1389_at",env=hgu95av2PMID)
> pubmed(pmid,disp="browser")
> absts <- pm.getabst("1389_at", "hgu95av2")
pm.titles(absts)
ne <- pm.abstGrep("neutral endopeptidase",absts[[1]])
```



Fadhillah Putri Taha
H071171301

Summary Micro Array Analysis

6.7 Searching GO Numbers and Evidence

Untuk mencari GO Numbers, file daftar anotasi harus diekstak terlebih dahulu.

```
> #Mencari nomor GO dan evidence - extract a list from the annotation
files hgu95av2GO
> go1389 <- get("1389_at", env = hgu95av2GO)
> idl <- lapply(go1389,function(x) x$GOID)
> idl[[1]]
[1] "GO:0001822"
> #memilih GO numbers yang berelasi dengan proses biologi
> library(annotate)
> getOntology(go1389,"BP")
 [1] "GO:0001822" "GO:0002003" "GO:0006508" "GO:0006518" "GO:0019233"
"GO:0043312" "GO:0046449"
 [8] "GO:0050435" "GO:0071345" "GO:0071492" "GO:0071493" "GO:0090399"
"GO:0097242"
> getEvidence#go1389)
function (inlist)
{
    ans <- sapply(inlist, function(z) {
        if (!.isMissingGOEntry(z))
            z$Evidence
        else z
    })
    ans[!is.na(ans)]
}
<bytecode: 0x00000200db48b058>
<environment: namespace:annotate>
> #Membuat list
> go1389TAS <- subset(go1389,getEvidence(go1389)=="TAS")
> #Mengekstrak informasi dari list
> sapply(go1389TAS,function(x) x$GOID)
  GO:0002003   GO:0043312   GO:0005886   GO:0030667
"GO:0002003" "GO:0043312" "GO:0005886" "GO:0030667"
> sapply(go1389TAS,function(x) x$Evidence)
GO:0002003 GO:0043312 GO:0005886 GO:0030667
     "TAS"      "TAS"      "TAS"      "TAS"
> sapply(go1389TAS,function(x) x$Ontology)
GO:0002003 GO:0043312 GO:0005886 GO:0030667
      "BP"       "BP"       "CC"       "CC"
```

Fadhillah Putri Taha
H071171301

Summary Micro Array Analysis

6.8 GO Parents and Children

```
+ #6.8 Go Parents and children
+ #Example 1
+ #Mengumpulkan Informasi Go
+ GOMFPARENTS$"GO:0003700"
> GOMFCHILDREN$"GO:0003700"
        is_a         is_a         is_a
"GO:0000981" "GO:0001011" "GO:0001130"
        is_a         is_a         is_a
"GO:0001199" "GO:0004874" "GO:0034246"
        is_a
"GO:0098531"
> #Mengumpulkan ontologi, orang tua, dan pengidentifikasi anak dalam
vektor
> go1389 <- get("1389_at", env = hgu95av2GO)
> gonr <- getOntology(go1389, "BP")
> gP <- getGOParents(gonr)
> gC <- getGOChildren(gonr)
> gPC <- c(gonr,gP,gC)
> pa <- sapply(gP,function(x) x$Parents)
> ch <- sapply(gC,function(x) x$Children)
> gonrc <- c(gonr,unlist(pa),unlist(ch))

#Example 2
#Seleksi Probe oleh GO
#BiocManager::install("GO")
library(GO); library(annotate); library("ALL"); data(ALL)
go1389 <- get("1389_at", env = hgu95av2GO)
gonr <- getOntology(go1389, "BP")
gP <- getGOParents(gonr)
pa <- sapply(gP,function(x) x$Parents)
probes <- mget(pa,hgu95av2GO2ALLPROBES)
probeNames <- unlist(probes)
ALLpr <- ALL[probeNames,]
dim(exprs(ALLpr))
```

6.9 Gene Filtering by a Biological Term

Menyaring gen yang merupakan istilah biologis, seperti represi transkripsi.

```
> #6.9
> library("GO"); library("annotate");library("hgu95av2.db")
Error in library("GO") : there is no package called ‘GO’
> GOTerm2Tag <- function(term) {
+   GTL <- eapply(GOTERM, function(x) {grep(term, x@Term, value=TRUE)
})
+   Gl <- sapply(GTL, length)
+   names(GTL[Gl>0])
+ }
>  GOTerm2Tag("transcriptional repressor")
 [1] "GO:0001141" "GO:0001210" "GO:0001214" "GO:0001217" "GO:0001218"
"GO:0001219" "GO:0001220"
 [8] "GO:0017053" "GO:0090568" "GO:0090569"
```

Fadhillah Putri Taha
H071171301

Summary Micro Array Analysis

6.10 Significance per Chromosome

Perkromosom dapat diuji jika rasio peluangnya tidak sama dnegan 1.

```
> #6.10 Signifikasi per kromosom
> library("ALL"); data(ALL); library("hgu95av2.db")
> rawp <- apply(exprs(ALL), 1, function(x) t.test(x ~ ALL$remission
)$p.value)
> xx <- as.list(hgu95av2CHR)
> AffimIDChr <- names(xx[xx=="19"])
> names(rawp) <- featureNames(ALL)
> f <- matrix(NA,2,2)
> f[1,1] <- sum(rawp[AffimIDChr]<0.05); f[1,2] <- sum(rawp[AffimIDC
hr]>0.05)
> f[2,1] <- sum(rawp<0.05) - f[1,1] ; f[2,2] <- sum(rawp>0.05) - f[
1,2]
> print(f)
     [,1]  [,2]
[1,]  106   599
[2,]  832 11088
> fisher.test(f)

        Fisher's Exact Test for Count Data

data:  f
p-value = 1.355e-12
alternative hypothesis: true odds ratio is not equal to 1
95 percent confidence interval:
 1.877281 2.940808
sample estimates:
odds ratio
  2.358134
```

Fadhillah Putri Taha
H071171301

Summary Micro Array Analysis

Micro Array Analysis

- *Source:* https://www.youtube.com/watch?v=0ATUjAxNf6U dan https://www.youtube.com/watch?v=UnFumDj4eeI

***DNA microarray* biasa disebut chip-chip DNA yang menyimpan ribuan gen pada permukaan solid, seperti *microscope slide.* Chip DNA memungkinkan sebanyak sepuluh ribu ekspresi gen di dalamnya yang dapat dianalisis secara bersamaan.** Contohnya, *yeast* digunakan sebagai *model system* untuk menggambarkan salah satu fungsi *micro array.* Salah satu fungsinya lagi adalah untuk menentukan gen aktif dan gen yang ditekan. Kegunaan yang lain adalah untuk membandingkan dua sampel cDNA. Jika dua gen dibandingkan, setiap gen diukur secara bersamaan. Contohnya, ketika dibandingkan apa yang terjadi pada *yeast gemes* ketika berkembang dengan keadaan aerobic dan keadaan non-aerobik. Sel-sel tumbuh dan beradaptasi dengan gen mana yang harus diaktivasi atau ditekan dalam keadaan untuk bertahan. Kemudian **mRNA diisolasi, sel-sel terpental di dalam *centrifuge* dan sel-sel terkumpul dalam pellet, dan membuang cairannya. Selanjutnya mRNA diekstrak dari sel-sel. Selanjutnya RNA dipindahkan ke dalam *tube* baru dan membuat cDNA dari mRNA, dimana setiap mRNA dikonver menjadi merah atau hijau cDNA**, sehingga mRNA menurun. **Selanjutnya mencampur cDNA merah dan hijau.**

Chip DNA mengandung ribuan *spot*, dan setiap *spot* memiliki perbedaan kode rangkaian *yeast* yang berbeda dari gen yang berbeda. Setiap *spot* terbuat dari DNA yang bisa dipasangkan dengan *cDNA complementary*/bebas. Selanjutnya cDNA yang telah dicampurkan, **diinkubasi dengan chip DNA**. Kemudian cDNA merah dan hijau terikat pada *spot*, namun hanya cDNA merah yang terikat dengan *spot* ini dan hanya cDNA hijau pula yang terikat pada *spot* lainnya. **Selanjutnya cDNA yang tidak terikat harus dibersihkan untuk melihat mana yang terikat dengan *micro array*.** Kemudian mendeteksi cDNA terikat sehingga dapat divisualisasikan.

Selanjutnya **menyimpan *microscope slide* yang berisi *micro array* di dalam *scanner.*** Pertama, laser hijau memindai *micro array*. Selanjutnya laser merah memindai *micro array*. Dan hasil gambar *micro array* setelah dipindai laser**, dibuatlah visualisasi dengan menggabungkan gambar**, dan menghasilkan bahwa gen aerobic berlabel hijau dan gen anaerobic berlabel merah, dan gen berlabel kuning yang mengekspresikan keduanya.