Analyze the Expression Profiles of H1-hESC

Ricky Lim

07 March 2013

1 Goals

- Analyze the RNA expression profiles from GEO public datasets

2 Get Public Datasets

Two replicates of RNA expression profiles (Agilent Human Whole-genome array) from GEO (GSM661186 and GSM661187).

```
> library(Biobase)
> library(GEOquery)
> library(plyr)
> # get the exp profiles from GEO
> h1_rep1 <- getGEO('GSM661186', destdir='input/')
> h1_rep2 <- getGEO('GSM661187', destdir='input/')
> # load the exp profiles from GEO
> h1_rep1 <- getGEO(filename='input/GSM661186.soft')
> h1_rep2 <- getGEO(filename='input/GSM661187.soft')
> #check datasets
> dim(Table(h1_rep1))
> dim(Table(h1_rep2))
> Table(h1_rep1)[1:5,]
> Table(h1_rep2)[1:5,]
```

2.1 Annotate the Expression Profiles

The annotation file that was used: GPL4133

```
> #annotations
> Meta(h1_rep1)$platform
> Meta(h1_rep2)$platform
> # using GPL4133
>
> # load GPL4133's annotation files
> gp4133 <- getGEO('GPL4133', destdir='input/')
> gp4133 <- getGEO(filename='input/GPL4133.soft')
> Meta(gp4133)$title
> colnames(Table(gp4133))
> Table(gp4133)[10:15, 1:5]
> # Get the Annotation with only ID and ENSEMBL ID
> gp4133_geneID <- Table(gp4133)[, c("ID", "ENSEMBL_ID")]
```

2.2 Map datasets with the ENSEMBL ID

```
> # get the dataframe of exp profiles
> h1_rep1 <- Table(h1_rep1)
> h1_rep2 <- Table(h1_rep2)
> colnames(h1_rep1) <- c('ID','VALUE')
> colnames(h1_rep2) <- c('ID', 'VALUE')
> head(h1_rep1)
> head(h1_rep2)
> dim(h1_rep1)
> dim(h1_rep2)
> # map dataset with gene id (ensembl_id)
> h1_rep1Gene <- join(h1_rep1, gp4133_geneID, by='ID')
> h1_rep2Gene <- join(h1_rep2, gp4133_geneID, by='ID')
> head(h1_rep1Gene)
> head(h1_rep2Gene)
```

3 Pre-processing

```
> # remove NAs
> h1_rep1Gene <- h1_rep1Gene[complete.cases(h1_rep1Gene),]
> h1_rep2Gene <- h1_rep2Gene[complete.cases(h1_rep2Gene),]
> head(h1_rep1Gene)
> # remove rows without ENSEMBL_ID
> h1_rep1Gene <- h1_rep1Gene[!(h1_rep1Gene$ENSEMBL_ID==""),]
> h1_rep2Gene <- h1_rep2Gene[!(h1_rep2Gene$ENSEMBL_ID==""),]
> head(h1_rep1Gene)
> head(h1_rep2Gene)
> nrow(h1_rep1Gene)
> nrow(h1_rep2Gene)
> # combine replicates
> h1 <- join(h1_rep1Gene, h1_rep2Gene, by="ID")
> head(h1)
> dim(h1)
> # average out the expression values
> # get only the ID, ENSEMBEL_ID, exp values from replicate 1 and replicate 2
> h1 <- h1[,c(1,3,2,4)]
> head(h1)
> colnames(h1) <- c("ID", "ENSEMBL_ID", "REP1", "REP2")
> h1$AVERAGE <- rowMeans(h1[,c("REP1","REP2")])
> head(h1)
> # log the average expression value
> h1$log_average <- log(h1$AVERAGE)
> head(h1)
> dim(h1)
>
>
```

4 Quantile Discrimination

Non-expressed with quantile < 0.25 and expressed with quantile > 0.75

```
> # check the quantile distribution of the average exp value
> quantile(h1$AVERAGE, na.rm=TRUE)
> # get the subset of H1 expression values (q > 75%)
> H1_expressed <- h1[h1$AVERAGE > 3150.706297,c(1,2)]
> H1_nonExpressed <- h1[h1$AVERAGE < 99.974448,c(1,2)]
> dim(H1_expressed)
> dim(H1_nonExpressed)
> head(H1_expressed)
> head(H1_nonExpressed)
```

5 Map Ensembl ID with Genomic Coordinates

The file of Ensembl coordinates was obtained from

```
/software/private/gfilion/Genomes/hg19/genes/ensGene.txt
```

```
> # load the ensembl_Coordinate
> library(plyr)
> Ensembl_coord <- read.table("input/ensGene.txt", header=FALSE)
> head(Ensembl_coord)
> Ensembl_id <- Ensembl_coord[,c("V2", "V3", "V5","V6")]
> head(Ensembl_id)
> colnames(Ensembl_id) <- c("ENSEMBL_ID", "chr","start","end")
> # merge H1_expressed and H1_nonExpressed subsets with Ensembl_id
> H1Expressed_bed <- join(H1_expressed, Ensembl_id, by= "ENSEMBL_ID")
> H1NonExpressed_bed <- join(H1_nonExpressed, Ensembl_id, by="ENSEMBL_ID")
> H1Expressed_bed <- H1Expressed_bed[complete.cases(H1Expressed_bed),]
> H1NonExpressed_bed <- H1NonExpressed_bed[complete.cases(H1NonExpressed_bed),]
> head(H1Expressed_bed)
> head(H1NonExpressed_bed)
> H1Expressed_bed <- H1Expressed_bed[,3:5]
> H1NonExpressed_bed <- H1NonExpressed_bed[,3:5]
> head(H1Expressed_bed)
> head(H1NonExpressed_bed)
> dim(H1Expressed_bed)
> dim(H1NonExpressed_bed)
```

5.1 Output

```
> write.table(H1Expressed_bed, "output/H1ExpressedGene.hg19.bed",
+             quote=FALSE, sep="\t", row.names=FALSE, col.names=FALSE)
> write.table(H1NonExpressed_bed, "output/H1NonExpressedGene.hg19.bed",
+             quote=FALSE, sep="\t", row.names=FALSE, col.names=FALSE)
>
```