Differential Expression and Annotation

Chao-Jen Wong

Fred Hutchinson Cancer Research Center

November 23, 2009

1. Differential Expression
2. Moderated t-statistics and Linear Models
3. Using the limma Package
4. Annotation

Outline

1. Differential Expression
2. Moderated t-statistics and Linear Models
3. Using the limma Package
4. Annotation

- Identify differentially expressed genes associated with biological or experimental conditions.
- Many different gene-by-gene approaches: t-statistics, empirical Bayesian, moderate t-statistics, ROC, etc.
- Primarily concerned with two-class problems.
- Data with n samples and p probes ($p >> n$).

A	A	A	A	A	B	B	B	B	B
$x_{1,1}$	$x_{1,2}$	$x_{1,3}$	$x_{1,4}$	$x_{1,5}$	$x_{1,6}$	$x_{1,7}$	$x_{1,8}$	$x_{1,9}$	$x_{1,10}$
$x_{2,1}$	$x_{2,2}$	$x_{2,3}$	$x_{2,4}$	$x_{2,5}$	$x_{2,6}$	$x_{2,7}$	$x_{2,8}$	$x_{2,9}$	$x_{2,10}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$x_{p,1}$	$x_{p,2}$	$x_{p,3}$	$x_{p,4}$	$x_{p,5}$	$x_{p,6}$	$x_{p,7}$	$x_{p,8}$	$x_{p,9}$	$x_{p,10}$

Outline

1. Differential Expression
2. **Moderated t-statistics and Linear Models**
3. Using the limma Package
4. Annotation

Getting Dataset and Nonspecific Filtering

Get ALL dataset.

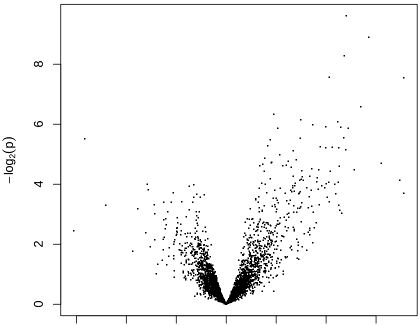
Data preperation – code from BioC intro

```
> library(ALL)
> library(hgu95av2.db)
> data(ALL)
> bcell <- grep("^B", as.character(ALL$BT))
> types <- c("NEG", "BCR/ABL")
> moltyp <- which(as.character(ALL$mol.biol) %in% types)
> # subsetting
> ALL_bcrneg <- ALL[, intersect(bcell, moltyp)]
> ALL_bcrneg$BT <- factor(ALL_bcrneg$BT)
> ALL_bcrneg$mol.biol <- factor(ALL_bcrneg$mol.biol)
> # nonspecific filter
> library(genefilter)
> filt_bcrneg <- nsFilter(ALL_bcrneg,
+                    require.entrez=TRUE,
+                    require.GOBP=TRUE,
+                    remove.dupEntrez=TRUE,
+                    feature.exclude="^AFFX",
+                    var.cutoff=0.5)
> ALLfilt_bcrneg <- filt_bcrneg$eset
```

Fold-change versus *t*-test

code: *t*-test

```
> tt <- rowttests(ALLfilt_bcrneg, "mol.biol")
> plot(tt$dm,  -log10(tt$p.value), pch=".",
+     xlab=expression(mean~log[2]~fold~change),
+     ylab=expression(-log[2](p)))
```



Fold-change and *t*-test

t-statistics:

$$t_g = \frac{\mu_x - \mu_y}{\sqrt{\sigma_x^2 - \sigma_y^2}}$$

Drawback:

- The variance in small samples might be noisy.
- Genes with small fold-change might be significant from statistical, not biological point of view.

Moderate t-statisitcs

- An overall estimate variation s_0^2 is computed.
- Per-gene deviation variation s_g^2 is computed.
- Shrinkage variation:

$$\tilde{s}_g^2 = \frac{d_0 s_0^2 + d_g s_g^2}{d_0 + d_g},$$

  where $\frac{d_0}{d_0+d_g}$ is weight coefficient associated with all probes and $\frac{d_g}{d_0+d_g}$ is associated with gene g.
- The difference in means between two classes, $\hat{\beta}_g$, is computed using empirical Bayes approach.
- Moderate t-statistics:

$$\tilde{t_g} = \frac{\hat{\beta}_g}{\tilde{s}_g \sqrt{\nu_g}}$$

Define parameters in linear models

$$y_i = \beta_1 a_{ij} + \beta_2 b_{ij} + \varepsilon_i$$

```
> model.matrix(~mol.biol + 0,
+                ALLfilt_bcrneg)

      mol.biolBCR/ABL mol.biolNEG
01005               1           0
01010               0           1
03002               1           0
04007               0           1
04008               0           1
04010               0           1
04016               0           1
06002               0           1
08001               1           0
08011               1           0
08012               0           1
08024               0           1
09008               1           0
09017               0           1
11005               1           0
12006               1           0
12007               1           0
12012               1           0
```

Define parameters in linear models

$$y_i = \beta_1 a_{ij} + \beta_2 b_{ij} + \varepsilon_i$$

```
> model.matrix(~mol.biol + 0,
+              ALLfilt_bcrneg)

      mol.biolBCR/ABL mol.biolNEG
01005               1           0
01010               0           1
03002               1           0
04007               0           1
04008               0           1
04010               0           1
04016               0           1
06002               0           1
08001               1           0
08011               1           0
08012               0           1
08024               0           1
09008               1           0
09017               0           1
11005               1           0
12006               1           0
12007               1           0
12012               1           0
```

$$y_i = \mu + \beta a_{ij} + \varepsilon_i$$

```
> model.matrix(~ mol.biol,
+              ALLfilt_bcrneg)

      (Intercept) mol.biolNEG
01005           1           0
01010           1           1
03002           1           0
04007           1           1
04008           1           1
04010           1           1
04016           1           1
06002           1           1
08001           1           0
08011           1           0
08012           1           1
08024           1           1
09008           1           0
09017           1           1
11005           1           0
12006           1           0
12007           1           0
12012           1           0
```

Outline

1. Differential Expression

2. Moderated t-statistics and Linear Models

3. Using the limma Package

4. Annotation

Using limma

1. Use design matrix to establish parameters of the model.
2. Define contrast model if needed (i.e., $contr = c(1, -1)$).
3. Use linear model to fit contrast parameters: lmFit().
4. Use function eBayes to get moderate t-statistics and relevant statistics.

code: design matrix

```
> library(limma)
> #cl = as.numeric(ALLfilt_bcrneg$mol.biol=="BCR/ABL")
> #design <- cbind(mean=1, diff=cl)
> design <- model.matrix( ~mol.biol + 0, ALLfilt_bcrneg)
> colnames(design) <- c("BCR_ABL", "NEG")
> # contr <- makeContrasts(BCR_ABL-NEG, levels=design)
> contr <- c(1, -1)
```

Using limma

Code: linear models and eBayes

```
> fit <- lmFit(exprs(ALLfilt_bcrneg), design)
> fit1 <- contrasts.fit(fit, contr)
> fit2 <- eBayes(fit1)
> #syms <- unlist(mget(featureNames(ALLfilt_bcrneg), hgu95av2SYMBOL))
> topTable(fit2, adjust.method="BH",
+          number=5)

          ID    logFC  AveExpr
1117  1635_at 1.202675 7.897095
3050  1674_at 1.427212 5.001771
2171 40504_at 1.181029 4.244478
2816 40202_at 1.779378 8.621443
799  37015_at 1.032702 4.330511
            t      P.Value    adj.P.Val
1117 7.408878 1.017739e-10 3.910154e-07
3050 7.059429 4.898793e-10 9.410581e-07
2171 6.705277 2.368917e-09 3.033793e-06
2816 6.354009 1.107794e-08 1.064036e-05
799  6.299154 1.406498e-08 1.080753e-05
             B
1117 13.998069
```

Reference

- G.K. Smyth, Linear models and empirical Bayes methods for assessing differential expression in microarray experiments, *Statistical Applications in Genetics and Molecular Biology*, 3(1), 20004.
- G. K. Smyth, *limma: Linear Models for Microarray Data*, Bioconductor package vignette, 2005.
- Y. Benjamini and Y. Hochbert, Controlling the false discovery rate: a practical and powerful approach to multiple testing, *Journal of the Royal Statistical Society, Series B*, 57(1): 289-300, 1995.

Exercise

1. Go through the example.
2. Try to get a list of genes whose adjusted *p*-value is less than 0.005 and get the genes' names and symbols of these genes.

Outline

1. Differential Expression
2. Moderated t-statistics and Linear Models
3. Using the limma Package
4. **Annotation**

Annotation and metadata

Further investigation to understand genes that have been identified.

- HTML table for a list of genes: htmlpage or saveHTML.

```
> library(annotate)
> top20Gene <- topTable(fit2, adjust.method="BH",
+                       number=20, genelist=syms)
> htmlpage(genelist=as.data.frame(top20Gene$ID),
+          othernames=top20Gene,
+          filename="top20gene.html",
+          table.head=c("probe ID", names(top20Gene)))
> broweURL("top20Gene.html")
```

- Visualization, i.e., heatmap of the top 40 significant genes.
- Categories such GO and KEGG.
- Annotation packages.

Bioconductor annotation packages

Main areas of annotation in Bioconductor AnnotationDbi packages:

- Organism level: org.Mm.eg.db.
- Platform level: hgu133plus2.db.
- System-biology level: GO.db or KEGG.db.

biomaRt:

- Query web-based ‘biomart’ resource for genes, sequence, SNPs, and etc.

Other packages:

- GenomeGraphs – visualization.
- rtracklayer – export to UCSF web browsers.

Organism-level annotation

There are a number of organism annotation packages with names starting with org, e.g., org.Hs.eg.db – genome-wide annotation for human.

```
> library(org.Hs.eg.db)
> org.Hs.eg()
> org.Hs.eg_dbInfo()
> org.Hs.egGENENAME
```

Basic structure

Bi-maps, from `ENTREZ` identifier to `GENENAME`, with `Lkeys` and `Rkeys`.

- Lkeys: probes id or pathway id
- reversible

```
> map <- org.Hs.egGENENAME
> map

GENENAME map for Human (object of class "AnnDbBimap")

> head(Lkeys(map)) ## probeset id

[1] "1"         "10"        "100"
[4] "1000"      "10000"     "100008586"

> map[["1000"]]

[1] "cadherin 2, type 1, N-cadherin (neuronal)"

> revmap(map)[["adenosine deaminase"]] ## reversible

[1] "100"
```

Working with GO.db

- Encodes the hierarchical structure of GO terms.
- Includes information of the mapping between GO terms and Entrez ID.

```
> library(GO.db)
> ls("package:GO.db")
```

```
 [1] "GO"             "GOBPANCESTOR"
 [3] "GOBPCHILDREN"   "GOBPOFFSPRING"
 [5] "GOBPPARENTS"    "GOCCANCESTOR"
 [7] "GOCCCHILDREN"   "GOCCOFFSPRING"
 [9] "GOCCPARENTS"    "GO_dbconn"
[11] "GO_dbfile"      "GO_dbInfo"
[13] "GO_dbschema"    "GOMAPCOUNTS"
[15] "GOMFANCESTOR"   "GOMFCHILDREN"
[17] "GOMFOFFSPRING"  "GOMFPARENTS"
[19] "GOOBSOLETE"     "GOSYNONYM"
[21] "GOTERM"
```