Analysis of Microarray Data

Keith Baggerly and Kevin Coombes
Section of Bioinformatics
Department of Biostatistics and Applied Mathematics
UT M. D. Anderson Cancer Center

kabagg@mdanderson.org
kcoombes@mdanderson.org

16 October 2007
Lecture 14: Reporting Differential Expression

- Differential Expression in BioC
- Annotation and Reporting of Data
Putting Some Pieces Together

Let’s examine some aspects of differential expression in R, using some of the datasets from BioConductor.

```r
> library(affy)
> library(ALL);
> data("ALL");
```

This is an ExpressionSet derived from 128 U95Av2 arrays, quantified using RMA. The phenoData has 21 variables, including “mol.biol”. This specifies cytogenetic abnormalities, such as “BCR/ABL” or “NEG”.

Skimming the Data

> class(ALL)
[1] "ExpressionSet"
attr(,"package")
[1] "Biobase"
> slotNames(ALL)
[1] "assayData" "phenoData" "featureData"
[4] "experimentData" "annotation"
[6] ".__classVersion__"
> phenoData(ALL)
  sampleNames: 01005, 01010, ..., LAL4 (128 total)
  varLabels and varMetadata:
    cod: Patient ID
    diagnosis: Date of diagnosis
    ...: ...
date last seen: date patient was last seen (21 total)
Learning about the Experiment

> experimentData(ALL)
Experiment data
    Experimenter name: Chiaretti et al.
    Laboratory: Department of Medical Oncology, Dana
    Contact information:
    Title: Gene expression profile of adult T-cell an
    URL:
    PMID: 14684422 16243790

Abstract: A 187 word abstract is available. Use
> abstract(ALL)
[1] "Gene expression profiles were examined in 33
Picking Something to Focus On

> varLabels(phenoData(ALL))
[1] "cod" "diagnosis" "sex" "age" "BT"
[6] "remission" "CR" "date.cr" "t(4;11)"
[10] "t(9;22)" "cyto.normal" "citog" "mol.biol"
[14] "fusion protein" "mdr" "kinet" "ccr"
[18] "relapse" "transplant" "f.u" "date last seen"

> table(phenoData(ALL)$mol.biol)

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>ALL1/AF4</td>
<td>10</td>
</tr>
<tr>
<td>BCR/ABL</td>
<td>37</td>
</tr>
<tr>
<td>E2A/PBX1</td>
<td>5</td>
</tr>
<tr>
<td>NEG</td>
<td>74</td>
</tr>
<tr>
<td>NUP-98 p15/p16</td>
<td>1</td>
</tr>
</tbody>
</table>
Subsetting the Group

```r
> mySubset <- ALL$mol.biol %in% c("BCR/ABL", "NEG");
> ALLs <- ALL[, mySubset];
```

There are 37 samples with the BCR/ABL fusion, and 74 samples that are negative for this.

Let’s contrast these 2 groups.
Looking for Differences

> library("genefilter");
> g <- ALLs$mol.biol; # choose a factor
> ALLs.t <- rowttests(ALLs, g);

The rowttests function is written in C and is pretty fast. For each row, it returns "statistic" "dm" "df" "p.value" (dm is the difference in means.) We tend to use MultiTtest from the ClassComparison package available on our website, but that’s only because we wrote it.

Unfortunately...
Looking for Differences

> ALLs.t <- rowttests(ALLs, g);
Error in rowttests(ALLs, g) : Number of groups must be <= 2 for 'rowttests'.
> levels(g)
[1] "ALL1/AF4" "BCR/ABL" "E2A/PBX1"
[4] "NEG" "NUP-98" "p15/p16"

Subsetted factors remember where they came from...
Looking for Differences

> ALLs.t <- rowttests(ALLs, g);
Error in rowttests(ALLs, g) : Number of groups must be <= 2 for 'rowttests'.
> levels(g)
[1] "ALL1/AF4"  "BCR/ABL"  "E2A/PBX1"
[4] "NEG"       "NUP-98"   "p15/p16"

Subsetted factors remember where they came from...

> ALLs$mol.biol <- factor(ALLs$mol.biol);
> g <- ALLs$mol.biol;
> ALLs.t <- rowttests(ALLs, g); # works
> hist(ALLs.t$p.value, breaks=100);
Ok, Can We See Them?

> heatmap(exprs(ALLs)); # BAD.

Why?
Ok, Can We See Them?

> heatmap(exprs(ALLs)); # BAD.

Why?

We’re considering too many genes at present. (Quick quiz: how many?) Clustering will hang your computer.

We need to filter our list down.
Some Filtering

```r
> meanThresh <- 100;
> filt1 <- rowMeans(exprs(ALLs)[, g ==
+     levels(g)[1]]) > meanThresh;
> filt2 <- rowMeans(exprs(ALLs)[, g ==
+     levels(g)[2]]) > meanThresh;
> selProbes <- (filt1 | filt2);
> ALLfilt <- ALLs[selProbes, ];
> dim(exprs(ALLfilt));
dim(exprs(ALLfilt));
[1] 0 111
Some Filtering

```r
> meanThresh <- 100;
> filt1 <- rowMeans(exprs(ALLs)[, g ==
+   levels(g)[1]]) > meanThresh;
> filt2 <- rowMeans(exprs(ALLs)[, g ==
+   levels(g)[2]]) > meanThresh;
> selProbes <- (filt1 | filt2);
> ALLfilt <- ALLs[selProbes, ];
> dim(exprs(ALLfilt));
dim(exprs(ALLfilt));
[1] 0 111

> rowMeans(exprs(ALLs))[1:3]
1000_at  1001_at  1002_f_at
```

© Copyright 2004-2007, Kevin R. Coombes and Keith A. Baggerly

GS01 0163: Analysis of Microarray Data
| 7.565085 | 5.019850 | 3.884797 |
Some Filtering (Take Logs!)

```r
> meanThresh <- log2(100);
> filt1 <- rowMeans(exprs(ALLs)[, g ==
+    levels(g)[1]]) > meanThresh;
> filt2 <- rowMeans(exprs(ALLs)[, g ==
+    levels(g)[2]]) > meanThresh;
> selProbes <- (filt1 | filt2);
> ALLfilt <- ALLs[selProbes, ];
> dim(exprs(ALLfilt)); # 3660 by 111, a bit big

> meanThresh <- log2(200);
... 
> dim(exprs(ALLfilt)); # 1771 by 111, better
```
Focus on the Interesting Ones

> filt3 <- ALLs$t$p.value < 0.0001;
> selProbes <- (filt1 | filt2) & filt3;
> ALLfilt <- ALLs[selProbes, ];
> dim(exprs(ALLfilt));  # 104 by 111, ok

Try picturing this...

> spcol <- ifelse(ALLfilt$mol.biol == "NEG",
+ "goldenrod", "skyblue")
> heatmap(exprs(ALLfilt), ColSideColors=spcol);
Huzzah! (Right?)
That Was Odd...

> plot(exprs(ALLfilt)[2,order(ALLfilt$mol.biol)],
  xaxt='n', xlab='Group'); # row 1 was boring.
That Was Odd... Right?

```r
> lines(10 - 3.5*(ALLfilt$mol.biol[order(ALLfilt$mol.biol)] == 'NEG'), col='red');
> title(main = 'BCR/ABL, followed by NEG');
```
What's Going On?

> names(pData(ALLfilt))

[1] "cod"       "diagnosis"  "sex"     "age"
[5] "BT"        "remission"  "CR"      "date.cr"
[9] "t(4;11)"   "t(9;22)"    "cyto.norm" "citog"
[13] "mol.biol" "fus prot"  "mdr"      "kinet"
[17] "ccr"      "relapse"    "transplant" "f.u"
[21] "date last seen"

Are there other variables that may dominate the one I chose?
## What Cells?

```r
> ALLfilt$BT

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>12</td>
<td>B2</td>
<td>B2</td>
<td>B3</td>
<td>B3</td>
<td>B3</td>
<td>B4</td>
<td>B3</td>
<td>B3</td>
<td>B3</td>
<td>B3</td>
</tr>
<tr>
<td>51</td>
<td>B1</td>
<td>B3</td>
<td>B3</td>
<td>B3</td>
<td>B3</td>
<td>B3</td>
<td>B3</td>
<td>B3</td>
<td>B3</td>
<td>B3</td>
</tr>
<tr>
<td>81</td>
<td>T2</td>
<td>T2</td>
<td>T3</td>
<td>T2</td>
<td>T</td>
<td>T4</td>
<td>T2</td>
<td>T3</td>
<td>T3</td>
<td>T</td>
</tr>
<tr>
<td>91</td>
<td>T2</td>
<td>T3</td>
<td>T2</td>
<td>T2</td>
<td>T2</td>
<td>T1</td>
<td>T4</td>
<td>T</td>
<td>T2</td>
<td>T3</td>
</tr>
<tr>
<td>101</td>
<td>T2</td>
<td>T2</td>
<td>T2</td>
<td>T2</td>
<td>T3</td>
<td>T3</td>
<td>T3</td>
<td>T2</td>
<td>T3</td>
<td>T2</td>
</tr>
<tr>
<td>111</td>
<td>T</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```

**Levels:** B B1 B2 B3 B4 T T1 T2 T3 T4
Another View

> table(pData(ALLfilt)$BT, pData(ALLfilt)$mol.biol)

<table>
<thead>
<tr>
<th></th>
<th>BCR/ABL NEG</th>
</tr>
</thead>
<tbody>
<tr>
<td>B</td>
<td>2</td>
</tr>
<tr>
<td>B1</td>
<td>1</td>
</tr>
<tr>
<td>B2</td>
<td>19</td>
</tr>
<tr>
<td>B3</td>
<td>8</td>
</tr>
<tr>
<td>B4</td>
<td>7</td>
</tr>
<tr>
<td>T</td>
<td>0</td>
</tr>
<tr>
<td>T1</td>
<td>0</td>
</tr>
<tr>
<td>T2</td>
<td>0</td>
</tr>
<tr>
<td>T3</td>
<td>0</td>
</tr>
<tr>
<td>T4</td>
<td>0</td>
</tr>
</tbody>
</table>
Matching Patterns

We want entries that begin with B. This is a “regular expression”, and one of the tools for extracting these is “grep”.

```r
> BT <- as.character(ALLfilt$BT);
> grep("B", BT); # returns 1..79
> grep("^B", BT); # same
> grep("^T", BT); # 80..111
> grep("B*", BT); # 1..111 everything!
> grep("B.*", BT); # 1..79
> grep("B$", BT); # 14,26,27,79
> grep("^B$", BT); # same
> grep("^b", BT); # null
> grep("^b", BT, ignore.case=TRUE);
```
Once More Unto the Breach!

```r
> plot(exprs(ALLfilt)[2,], xlab='Sample');
> y1 <- rep(0,111);
> y1[grep("^T",BT)] = 1;
> lines(10 - 3.5*y1, col='red')
> title(main="B Cells, then T Cells");
```
Finally!

![Graph showing B Cells, then T Cells](image-url)
Analysis Redux 1

> mySubset1 <- grep("^B", ALL$BT);
> ALLs1 <- ALL[, mySubset1];
> dim(exprs(ALLs1))
[1] 12625  95
> mySubset2 <- ALLs1$mol.biol %in% c("BCR/ABL", "NEG");
> ALLs2 <- ALLs1[, mySubset2];
> dim(exprs(ALLs2))
[1] 12625  79
> ALLs2$mol.biol <- factor(ALLs2$mol.biol);
> g <- ALLs2$mol.biol;
> ALLs2.t <- rowttests(exprs(ALLs2), g);
> hist(ALLs2.t$p.value, breaks=100);
Analysis Redux 3

```r
> meanThresh <- log2(100);
> filt1 <- rowMeans(exprs(ALLs2)[, g == levels(g)[1]]) > meanThresh;
> filt2 <- rowMeans(exprs(ALLs2)[, g == levels(g)[2]]) > meanThresh;
> filt3 <- ALLs2$t$p.value < 0.0001;
> selProbes <- (filt1 | filt2) & filt3;
> ALLs2Filt <- ALLs2[selProbes,];
> dim(exprs(ALLs2Filt))
[1] 36 79
```
A Better Figure

> spcol <- ifelse(ALLs2Filt$mol.biol == "NEG", "goldenrod", "skyblue");
> heatmap(exprs(ALLs2Filt), ColSideColors=spcol);
So, What About These Genes?

Through all this processing, the gene identities have been preserved, so we can access them easily.

```
> featureNames(ALLs2Filt)[1:3]
[1] "106_at"      "1134_at"    "1635_at"

> ALLs2Filt.t <- rowttests(exprs(ALLs2Filt), g);
> plot(ALLs2Filt.t$statistic)
> index <- order(abs(ALLs2Filt.t$statistic), decreasing = TRUE);
> probeids <- featureNames(ALLs2Filt)[index]
> probeids[1:3]
[1] "1636_g_at"    "39730_at"    "1635_at"
```
Let’s Add to the Report

Grab some of the annotation from the environment

```r
> library("annaffy")
Loading required package: GO
Loading required package: KEGG
> library("hgu95av2")
> syms <- unlist(mget(probeids,
   hgu95av2SYMBOL))
> locuslinks <- unlist(mget(  
   probeids, hgu95av2ENTREZID))
> library("annotate")
```
Make a Web Page

```r
> geneList <- list(probeids);
> repository <- list("affy");
> otherNames <- list(syms, locuslinks);
> head <- c("Probe ID", "Symbol", "LocusLink");
> fileName <- "out1.html";
> htmlpage(genelist = geneList, filename = fileName, title="ALL Interesting",
          othernames = otherNames,
          table.head = head,
          repository = repository)
```

Note: the elements in geneList will be live links!
The Output

Differentially Expressed Genes

ALL Interesting

<table>
<thead>
<tr>
<th>Probe Set</th>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1636_g_at</td>
<td>ABL1</td>
<td>25</td>
</tr>
<tr>
<td>39730_at</td>
<td>ABL1</td>
<td>25</td>
</tr>
<tr>
<td>1635_at</td>
<td>ABL1</td>
<td>25</td>
</tr>
<tr>
<td>40202_at</td>
<td>KLF9</td>
<td>687</td>
</tr>
<tr>
<td>37027_at</td>
<td>AHNAK</td>
<td>79026</td>
</tr>
<tr>
<td>39837_s_at</td>
<td>ZNF467</td>
<td>168544</td>
</tr>
<tr>
<td>40480_s_at</td>
<td>FYN</td>
<td>2534</td>
</tr>
<tr>
<td>33774_at</td>
<td>CASP8</td>
<td>841</td>
</tr>
<tr>
<td>36591_at</td>
<td>TUBA1</td>
<td>7277</td>
</tr>
<tr>
<td>37014_at</td>
<td>MX1</td>
<td>4599</td>
</tr>
<tr>
<td>32338</td>
<td>LSTN1</td>
<td>87</td>
</tr>
</tbody>
</table>

© Copyright 2004-2007, Kevin R. Coombes and Keith A. Baggerly

GS01 0163: Analysis of Microarray Data
More Details

The Affy links take you to NetAffx; you must log in there to see the details.

For htmlpage, values to be linked must come first.

Calling this function without a list of values to be linked (ie, all columns are “othernames”) will break.

More than one set of links can be included in a single table; every such set requires us to specify a repository.

The documentation for htmlpage doesn’t directly state what the set of repositories is...
Checking Code

> htmlpage

... for (i in seq(along = repository)) {
  rows <- paste(rows, getTDRows(genelist[[i]], repository[[i]]))
}

... checking the help for “getTDRows”:

repository: A character string for the name of a public repository. Valid values include ”ll”, ”ug”, ”gb”, ”sp”, ”omim”, ”affy”, ”en”, and ”fb”.

What are These?

- LL: LocusLink
- UG: UniGene
- GB: GenBank
- SP: SwissProt
- OMIM: Online Mendelian Inheritance in Man
- AFFY: NetAffx
- EN: EntrezGene (replaces LocusLink)
- FB: FlyBase
Getting More Sophisticated

The annaffy package contains quite a few more accessors for different types of databases and information. These begin with “aaf” (Annotation for AFfy), and return lists of information.

library(annaffy)
help(package=annaffy)

The list (all prefixed by aaf): ChromLoc, Chromosome, Cytoband, Description, Function, GenBank, GO, LocusLink, Pathway, Probe, PubMed, Symbol, UniGene.
Mapping

Most queries share a common syntax:

```r
> ALLbands <- aafCytoband(probeids, "hgu95av2");
> ALLbandLinks <- getURL(ALLbands);
> ALLbandLinks[[1]]
> ALLbands[[1]]
An object of class "aafCytoband"
Slot "band":
[1] "9q34.1"
Slot "genbank":
[1] "U07563"
```
Assembling a Table, Take 2

> aaf.handler()

[1] "Probe"  "Symbol"     "Description"
[4] "Function" "Chromosome" "Chromosome Location"
[7] "GenBank"  "LocusLink"  "Cytoband"
[10] "UniGene"  "PubMed"     "Gene Ontology"
[13] "Pathway"

> ALLTable <- aafTableAnn(probeids, "hgu95av2");

the argument “colnames” can be used to produce subsets of this output, or to rearrange the order.
Assembling a Table, Take 2

> ALLCols <- aaf.handler()[c(1:3, 9, 5:6)];
> ALLTable <- aafTableAnn(probeids, "hgu95av2", colnames=ALLCols);

These approaches produce tables which are lists of lists:

> ALLTable[[1]][[1]]
[1] "1636_g_at"
attr("class")
[1] "aafProbe"

> ALLTable[[2]][[1]]
[1] "ABL1"
attr("class")
[1] "aafSymbol"
Making a Page, Take 2

> saveHTML(ALLTable, "out2.html")
> saveHTML(ALLTable2, "out3.html")

Why do it twice?
# The Sparse Page

## Bioconductor Affymetrix Probe Listing

<table>
<thead>
<tr>
<th>Probe</th>
<th>Symbol</th>
<th>Description</th>
<th>Function</th>
<th>Chromosome</th>
<th>Chromosome Location</th>
<th>GenBank</th>
<th>LocusLink</th>
<th>Cytoband</th>
</tr>
</thead>
<tbody>
<tr>
<td>1636</td>
<td>ABL1</td>
<td><em>v</em>-abl Abelson murine leukemia viral oncogene homolog 1</td>
<td>9</td>
<td>9</td>
<td>130740384, 130618821</td>
<td>U07563</td>
<td>25</td>
<td>9q34.1</td>
</tr>
</tbody>
</table>
## The Dense Page

### Bioconductor Affymetrix Probe Listing

<table>
<thead>
<tr>
<th>Probe</th>
<th>Symbol</th>
<th>Description</th>
<th>Cytoband</th>
<th>Chromosome</th>
<th>Chromosome Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>1636</td>
<td><em>g at</em></td>
<td>ABL1 v-abl Abelson murine leukemia viral oncogene homolog 1</td>
<td>9q34.1</td>
<td>9</td>
<td>130740384, 130618821</td>
</tr>
<tr>
<td>39730</td>
<td><em>at</em></td>
<td>ABL1 v-abl Abelson murine leukemia viral oncogene homolog 1</td>
<td>9q34.1</td>
<td>9</td>
<td>130740384, 130618821</td>
</tr>
<tr>
<td>1635</td>
<td><em>at</em></td>
<td>ABL1 v-abl Abelson murine leukemia viral oncogene homolog 1</td>
<td>9q34.1</td>
<td>9</td>
<td>130740384, 130618821</td>
</tr>
<tr>
<td>40202</td>
<td><em>at</em></td>
<td>KLF9 Kruppel-like factor 9</td>
<td>9q13</td>
<td>9</td>
<td>-70229068</td>
</tr>
<tr>
<td>37027</td>
<td><em>at</em></td>
<td>AHNAC AHNAK nucleoprotein (desmoyokin)</td>
<td>11q12.2</td>
<td>11</td>
<td>-62039950, -61957591</td>
</tr>
<tr>
<td>39837</td>
<td><em>s at</em></td>
<td>ZNF467 zinc finger protein 467</td>
<td>7q36.1</td>
<td>7</td>
<td>-148899099</td>
</tr>
<tr>
<td>40480</td>
<td><em>s at</em></td>
<td>FYN FYN oncogene related to SRC, FGR, YES</td>
<td>6q21</td>
<td>6</td>
<td>-112089179, -112089186</td>
</tr>
<tr>
<td>33774</td>
<td><em>at</em></td>
<td>CASP8 caspase 8, apoptosis-related cysteine peptidase</td>
<td>2q33-q34</td>
<td>2</td>
<td>201923686, 201948284, 201950747, 201923693</td>
</tr>
<tr>
<td>36591</td>
<td><em>at</em></td>
<td>TUBA1 tubulin, alpha 1 (testis specific)</td>
<td>2q35</td>
<td>2</td>
<td>-219940505</td>
</tr>
<tr>
<td>37014</td>
<td><em>at</em></td>
<td>MX1 myxovirus (influenza virus) resistance 1, interferon-inducible protein p78 (mouse)</td>
<td>21q22.3</td>
<td>21</td>
<td>41720023</td>
</tr>
<tr>
<td>39329</td>
<td><em>at</em></td>
<td>ACTN1 actinin, alpha 1</td>
<td>14q24.1-q24.2</td>
<td>14q24.1</td>
<td>14q22-q24</td>
</tr>
<tr>
<td>32542</td>
<td><em>at</em></td>
<td>FHL1 four and a half LIM domains 1</td>
<td>Xq25</td>
<td>X</td>
<td>134955199</td>
</tr>
<tr>
<td>40051</td>
<td><em>at</em></td>
<td>TRAM2 translocation associated membrane protein 2</td>
<td>6p21.1-p12</td>
<td>6</td>
<td>-52470160</td>
</tr>
<tr>
<td></td>
<td></td>
<td>FYN oncogene related to SRC, FGR</td>
<td></td>
<td></td>
<td>112680179</td>
</tr>
</tbody>
</table>