Gene Set Enrichment Analysis

R. Gentleman

July 5, 2006

1 Introduction

We begin by loading up the appropriate libraries.

```
> library("Biobase")
> library("annotate")
> library("Category")
> library("hgu95av2")
> library("genefilter")
```

In this case study we will see how to use gene set enrichment analysis (Subramanian et al., 2005; Tian et al., 2005). We will make use of the data from an investigation into acute lymphoblastic leukemia (ALL) reported in Chiaretti et al. (2004) for our examples. We will primarily concentrate on the two sample problem, where the data can be divided into two distinct groups, and we want to understand differences in gene expression between the two groups. For the ALL data we will compare those samples with BCR/ABL to those that have no observed cytogenetic abnormalities (NEG).

The basic idea behind gene set enrichment analysis is that we want to use predefined sets of genes, perhaps based on function, in order to better interpret the observed gene expression data. In some ways the ideas here are quite similar to those that the usual Hypergeomtric testing is based on.

Preprocessing

As for all analyses, we must first load the data and process it. In the code chunk below we load the data and then select the subset we are interested in.

Question 1
How many samples are in our subset? How many are BCR/ABL and how many NEG?

Next, we remove from our data set those probes that have no mapping to EntrezGene, since we will not be able to find any metadata for these probes.

```
> entrezIds <- mget(geneNames(eset), envir = hgu95av2LOCUSID)
> haveEntrezId <- names(entrezIds)[sapply(entrezIds, function(x) !is.na(x))]
> numNoEntrezId <- length(geneNames(eset)) - length(haveEntrezId)
> eset <- eset[haveEntrezId, ]
```

Next we do some basic prefiltering. My preference is to filter genes according to their variability across samples. In the code below we compute the IQR (approximately) and then select for our gene set those genes that have an IQR above the median value.

```
> lowQ = rowQ(eset, floor(0.25 * numBN))
> upQ = rowQ(eset, ceiling(0.75 * numBN))
> iqrs = upQ - lowQ
> selected <- iqrs > 0.5
> nsFiltered <- eset[selected, ]
> numNsWithDups <- length(geneNames(nsFiltered))
```

In the next code chunk, we find all probes that map to a single gene. We want only one probe set to represent each gene (otherwise we have to do a lot of downstream adjustments) and our decision here is to chose the one that shows the most variation, as measured by the IQR, across samples.

```
> nsFilteredIqr <- iqrs[selected]
> uniqGenes <- findLargest(geneNames(nsFiltered), nsFilteredIqr,
+     "hgu95av2")
> nsFiltered <- nsFiltered[uniqGenes, ]
> numSelected <- length(geneNames(nsFiltered))
> numBcrAbl <- sum(nsFiltered$mol.biol == "BCR/ABL")
> numNeg <- sum(nsFiltered$mol.biol == "NEG")
```

Question 2
How many genes have been selected for our analysis?

2 Using KEGG

We now want to use KEGG to assign genes to pathways, and to then use those pathways for our gene sets.

```
> havePATH <- sapply(mget(geneNames(nsFiltered), hgu95av2PATH),
+     function(x) if (length(x) == 1 && is.na(x)) FALSE else TRUE)
> numNoPATH <- sum(!havePATH)
> nsF <- nsFiltered[havePATH, ]
```

Question 3
How many genes are we left with?

Now we must compute the incidence matrix, that is the matrix that maps between probes and the pathways. This matrix has zero's and ones in it. The last two commands rearrange the rows of the A matrix so that they are in the same order as the genes in our *exprSet* object.

```
> Am = PWAmat("hgu95av2")
> egN = unlist(mget(geneNames(nsF), hgu95av2LOCUSID))
> sub1 = match(egN, row.names(Am))
> Am = Am[sub1, ]
> dim(Am)
```

```
[1] 1231  173
```

```
> table(colSums(Am))
```

```
  0   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18  19
  8  14   8   3  11  12   9   9   5   5   7   4   5   4   3   7   3   4   2   3
 20  21  22  23  24  25  26  27  28  34  36  37  38  40  41  42  44  47  50  51
  2   4   1   3   1   1   1   3   2   2   1   1   1   2   1   5   1   1   1   1
 53  60  61  62  65  66  75  84 117
  2   2   1   2   1   1   1   1   1
```

Question 4
How many categories and how many genes are represented by the A matrix? How many categories have fewer than 10 genes in them? What is the largest number of categories a gene is in?

Next we will compute the per gene test statistics using the *rowttests*, there are several other fast test statistic computations that you can do as well (e.g. *rowFtests*).

```
> rtt = rowttests(nsF, "mol.biol")
> rttStat = rtt$statistic
```

Next we further reduce the A matrix, by removing all categories that have fewer than 10 genes in them. When carrying out your own analyses you should select a value you are comfortable with.

```
> Amat = t(Am)
> rs = rowSums(Amat)
> Amat2 = Amat[rs > 10, ]
> rs2 = rs[rs > 10]
> nCats = length(rs2)
```

And now it is fairly easy to compute the per category test statistics and to produce a qq-plot.

```
> tA = as.vector(Amat2 %*% rttStat)
> tAadj = tA/sqrt(rs2)
> names(tA) = names(tAadj) = row.names(Amat2)
```

And now for the qq-plot. We see that there is one pathway that has a remarkably low observed value (less than -5) so we will take a closer look at this pathway.

To find the pathway, we first find the value, and then use

```
> smPW = tAadj[tAadj < -5]
> pwName = KEGGPATHID2NAME[[names(smPW)]]
> pwName
```

```
[1] "Ribosome"
```

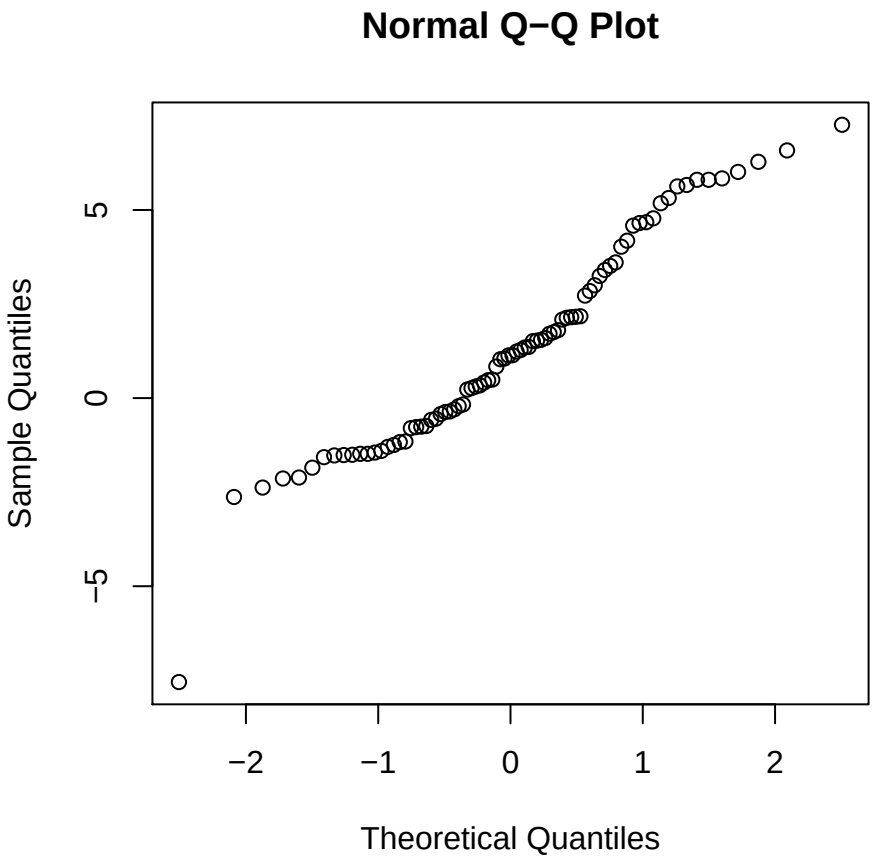

Figure 1: The per category qq-plot.

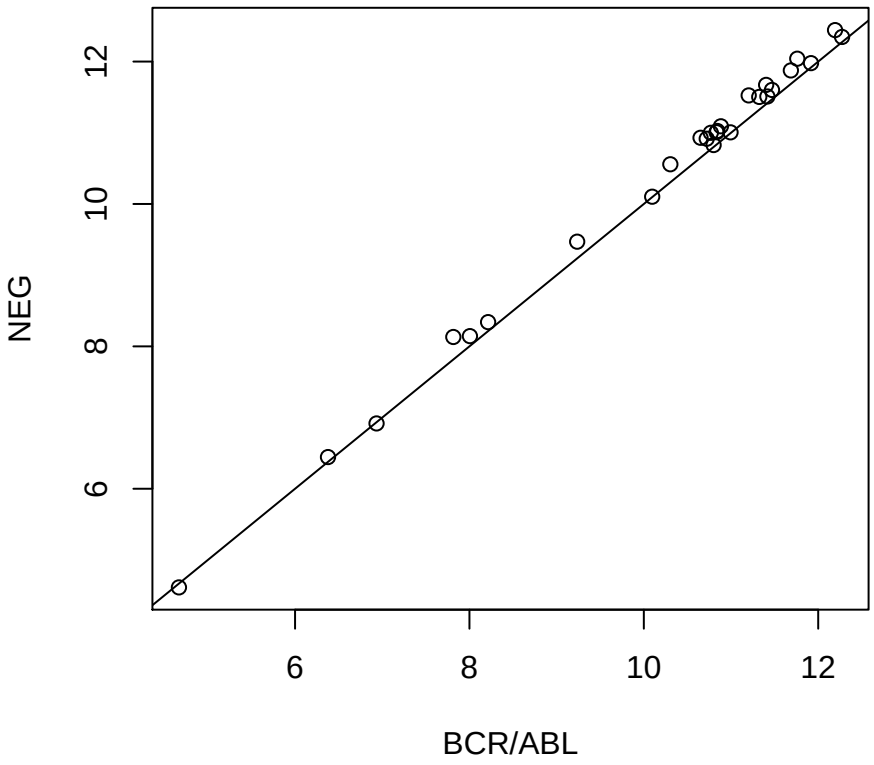

Figure 2: The mean plot for the Ribosome pathway.

Now we can produce some summary plots based on the genes annotated at this pathway. The mean plot presents a comparison of the average expression value for each of our two groups, for each gene in the specified pathway. That is, each point in this plot represents one gene and the value on the x-axis is the mean in the BCR/ABL samples while the value on the y-axis is the mean value in the NEG samples.

Question 5
What do you notice in this plot?

And finally a heatmap.

Question 6
What sorts of things do you notice in the heatmap? The gene labeled `41214_at` *has a very unusual pattern of expression. Can you guess what is happening? Hint: look at which chromosome it is on.*

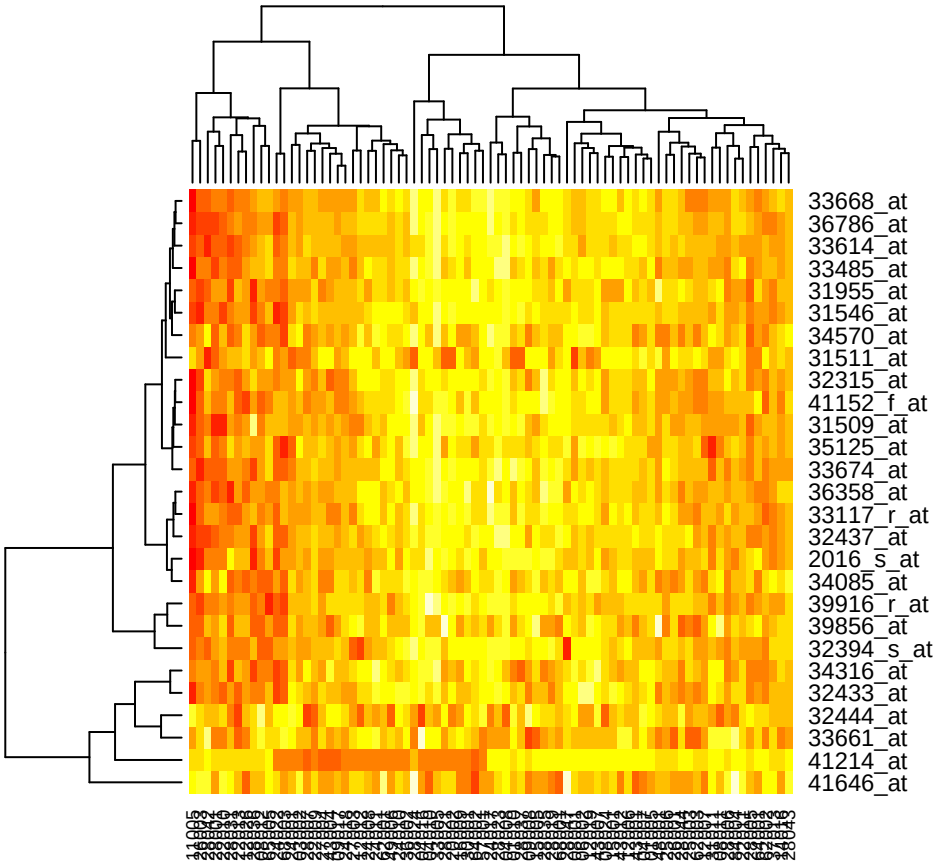

Figure 3: A heatmap for the Ribosome pathway.

Exercise 1
Repeat these plots for the pathway with the largest observed average t-statistic.

Permutation testing

The analysis above was based on a presumption that the data are approximately Normally distributed. However, this is sometimes viewed as a relatively strong assumption and there is some interest in using another approach. For GSEA it is relatively straightforward to compute a permutation t-test. The *ttperm* in the Category package can be used for this purpose.

In the next code chunk we compute the permutation distribution for this same problem. The value returned by *ttperm* is a list, the first entry is the observed t-statistic (the return value of a call to *rowttests*) while the second element is itself a list of length B, the number of permutations.

In the code chunk below we compute the permutation distribution based on 100 permutations, in practice you would typically use a much larger value.

```
> NPERM = 100
> ttp = ttperm(exprs(nsF), nsF$mol.biol, NPERM)
> permDm = do.call("cbind", lapply(ttp$perms, function(x) x$statistic))
> permD = Amat2 %*% permDm
> pvals = matrix(NA, nr = nCats, ncol = 2)
> dimnames(pvals) = list(row.names(Amat2), c("Lower", "Upper"))
> for (i in 1:nCats) {
+     pvals[i, 1] = sum(permD[i, ] < tA[i])/NPERM
+     pvals[i, 2] = sum(permD[i, ] > tA[i])/NPERM
+ }
> ord1 = order(pvals[, 1])
> lowC = (row.names(pvals)[ord1])[pvals[ord1, 1] < 0.05]
> highC = row.names(pvals)[pvals[, 2] < 0.05]
> getPathNames(lowC)

$`03010`
[1] "Ribosome"

> getPathNames(highC)

$`04620`
[1] "Toll-like receptor signaling pathway"

$`04510`
[1] "Focal adhesion"

$`04512`
[1] "ECM-receptor interaction"

$`04514`
```

```
[1] "Cell adhesion molecules (CAMs)"

$`04630`
[1] "Jak-STAT signaling pathway"

$`04520`
[1] "Adherens junction"

$`04640`
[1] "Hematopoietic cell lineage"

$`05120`
[1] "Epithelial cell signaling in Helicobacter pylori infection"

$`04530`
[1] "Tight junction"

$`04650`
[1] "Natural killer cell mediated cytotoxicity"

$`04060`
[1] "Cytokine-cytokine receptor interaction"

$`04660`
[1] "T cell receptor signaling pathway"

$`04210`
[1] "Apoptosis"

$`04810`
[1] "Regulation of actin cytoskeleton"

$`04670`
[1] "Leukocyte transendothelial migration"

$`04330`
[1] "Notch signaling pathway"

$`04080`
[1] "Neuroactive ligand-receptor interaction"

$`04940`
[1] "Type I diabetes mellitus"

$`04350`
```

```
[1] "TGF-beta signaling pathway"

$`04010`
[1] "MAPK signaling pathway"

$`04610`
[1] "Complement and coagulation cascades"

$`04612`
[1] "Antigen processing and presentation"

$`04360`
[1] "Axon guidance"

$`04730`
[1] "Long-term depression"
```

```
> lnhC = length(highC)
```

Exercise 2
How many pathways have low t-statistics? How many have high? How do you interpret these? What p-value is the most extreme? What does the heatmap look like for this gene set?

A more substantial exercise

Exercise 3
Repeat this analysis using the chromosome bands as your categories. You will probably want to use the MAPAmat function.

References

S. Chiaretti, X Li, R Gentleman, A Vitale, M. Vignetti, F. Mandelli, J. Ritz, , and R. Foa. Gene expression profile of adult T-cell acute lymphocytic leukemia identifies distinct subsets of patients with different response to therapy and survival. *Blood*, 103:2771–2778, 2004.

A. Subramanian, P. Tamayo, V. K. Mootha, et al. Gene set enrichment analysis: A knowledge-based approach for interpreting genome-wide expression profiles. *Proc. Natl. Acad. Sci. of the U.S.A.*, 102(43):15545–15550, 2005.

L. Tian, S. A. Greenberg, S. W. Kong, et al. Discovering statistically significant pathways in expression profiling studies. *Proc. Natl. Acad. Sci. of the U.S.A.*, 102(38):13544–13549, 2005.