Integrating BioConductor Packages in the Analysis of Affymetrix Data

James W. MacDonald

UMCCC Affymetrix and cDNA

Microarray Core Facility

Goals

- Step though analysis of Affy data
  - – QA – Final report
  - – Simplify analysis by using wrapper functions
  - – Primarily use *affycoretools*
  - – Some discussion of writing wrapper functions

Analysis of Affy data

- CEL files → Finished output
  - CEL files contain raw Affymetrix data
  - Finished output
    - Some sort of data presentation (HTML/text tables)
    - Description of analysis

Wrapper functions

- Write functions that ‘wrap’ existing functions to perform common tasks.
  - – Analyses use multiple packages
    - *affy, limma, annaffy, GOstats, biomaRt, annotate*, etc.
    - Data structures may be similar, but packages are not explicitly designed to work together.
  - – Relatively similar analyses result in lots of replicated R code.

An extended example

- Getting started
- Model data/make comparisons
- Create output/documentation

Getting started

- Read data into R
- Check quality of raw data
- Compute expression values
- Check quality of expression values

Read data into R

- ReadAffy() – *affy* package
- Read in Cel files
  - R_HOME/library/affycoretools/examples
- Twelve samples, three replicates, four sample types (A, B, C, D)

Get Code

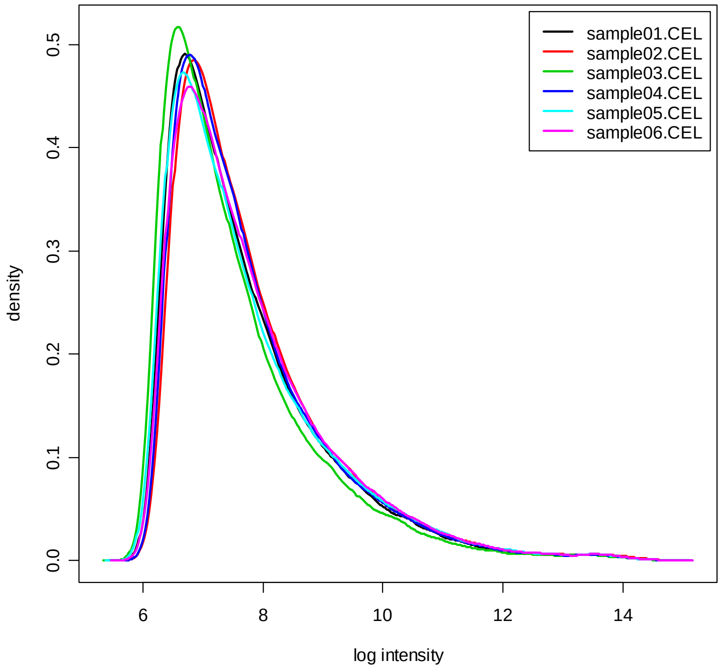
- Code for this lab can either be downloaded, or installed by USB drive
  - source(http://www.umich.edu/~jmacdon/getR.R)
  - Drag the BioC2007.R file to your current working directory (use getwd() to see what that is).

Code chunk 1

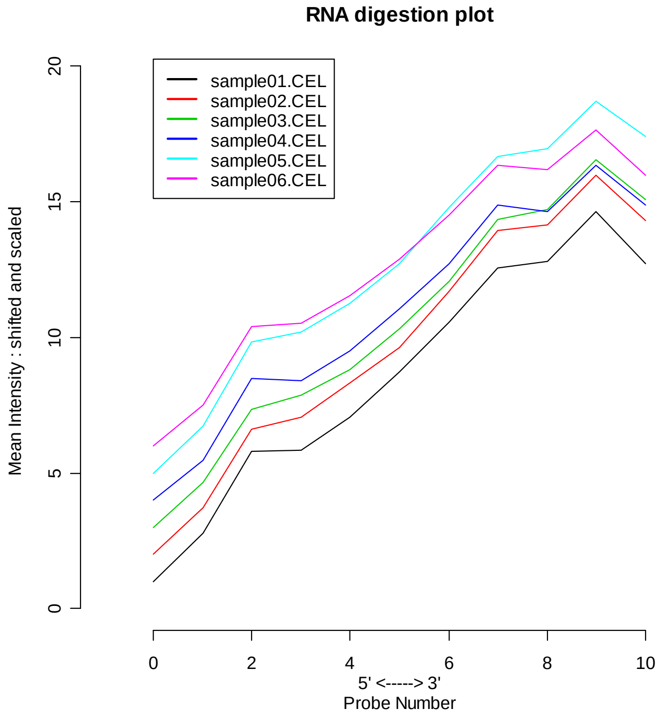
```
library(affycoretools)
library(KEGG)
library(xtable)
## make AnnotatedDataFrame
pd <- read.AnnotatedDataFrame(paste(system.file("examples",package="affycoretools"),
                                    "/pdata.txt", sep=""), header = TRUE, row.names = 1)

## no celfiles in package any more, fake this step
#dat <- ReadAffy()
#eset <- rma(dat)

load(paste(system.file("examples", package="affycoretools"),
            "/abatch.Rdata", sep=""))
load(paste(system.file("examples", package="affycoretools"),
            "/exprSet.Rdata", sep=""))

## load annotation package
options(show.error.messages = FALSE)
a <- try(do.call("library", list(annotation(eset))))
options(show.error.messages = TRUE)
if(inherits(a, "try-error")){
source("http://www.bioconductor.org/biocLite.R")
biocLite(annotation(eset))
do.call("library", list(annotation(eset)))
}
```

Check quality of raw data

plotHist(dat[,1:6])



plotDeg(dat[,1:6])



Compute expression values

- Various methods
  - rma() – *affy* package
  - gcrma() – *gcrma* package
  - mas5() – *affy* package
  - affystart() – *affycoretools* package

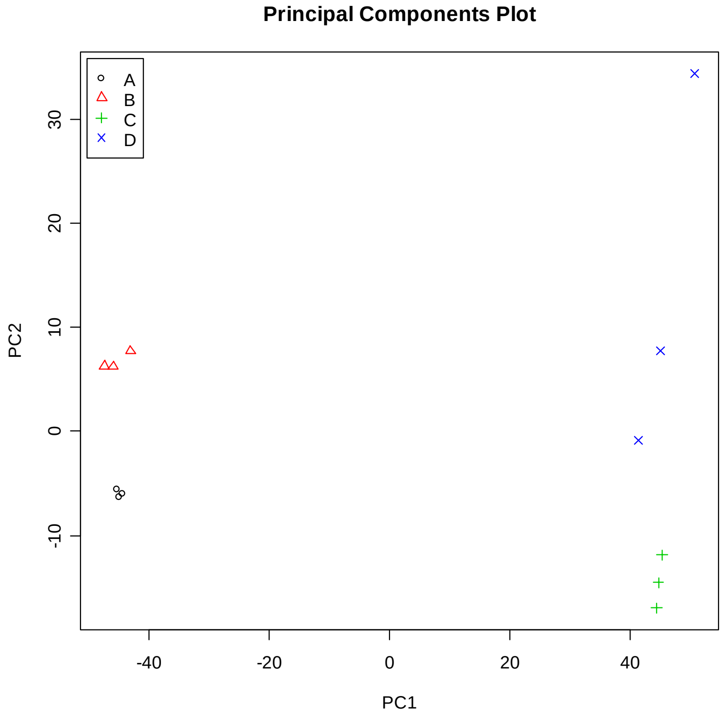
Code chunk 3-6

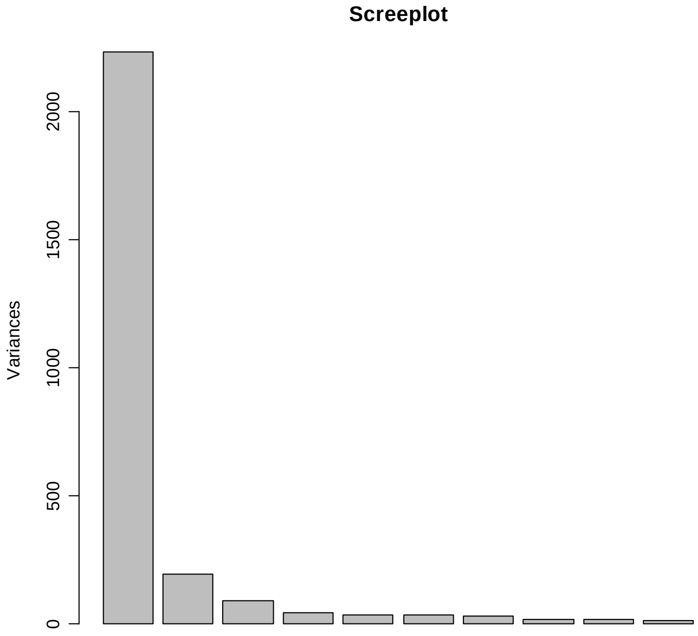
```
###################################################
### chunk number 3:
###################################################
plotHist(dat, sampleNames(eset))
plotHist(dat[,1:6])
plotHist(dat[,7:12])

###################################################
### chunk number 4:
###################################################
plotDeg(dat, sampleNames(eset))
plotDeg(dat[,1:6])
plotDeg(dat[,7:12])
```

Check quality of expression values

- plotPCA() – *affycoretools* package
- image() – *affyPLM* package
- boxplot() – *affyPLM* package
- Mbox() – *affyPLM* package

plotPCA()


Principal Components Plot

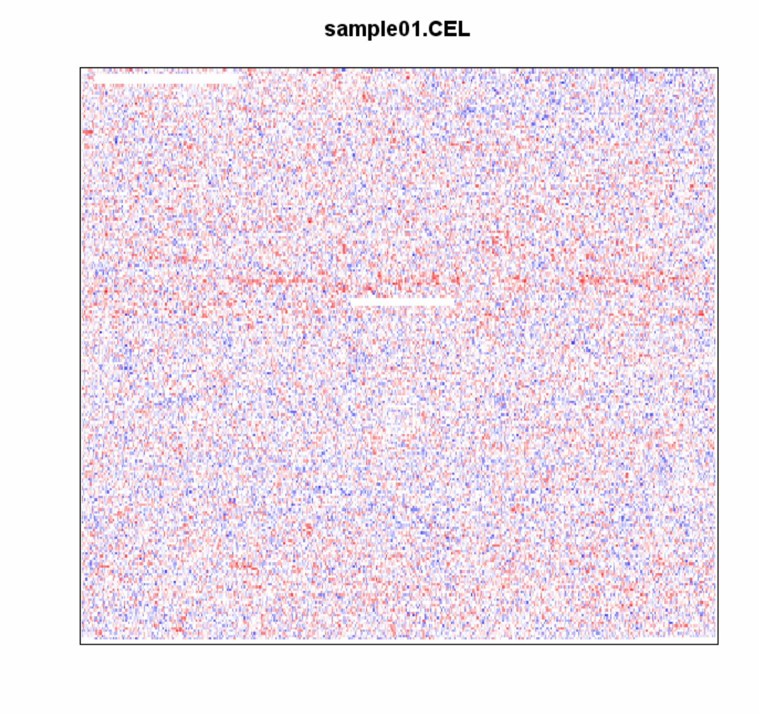

Screeplot

Code chunk 6

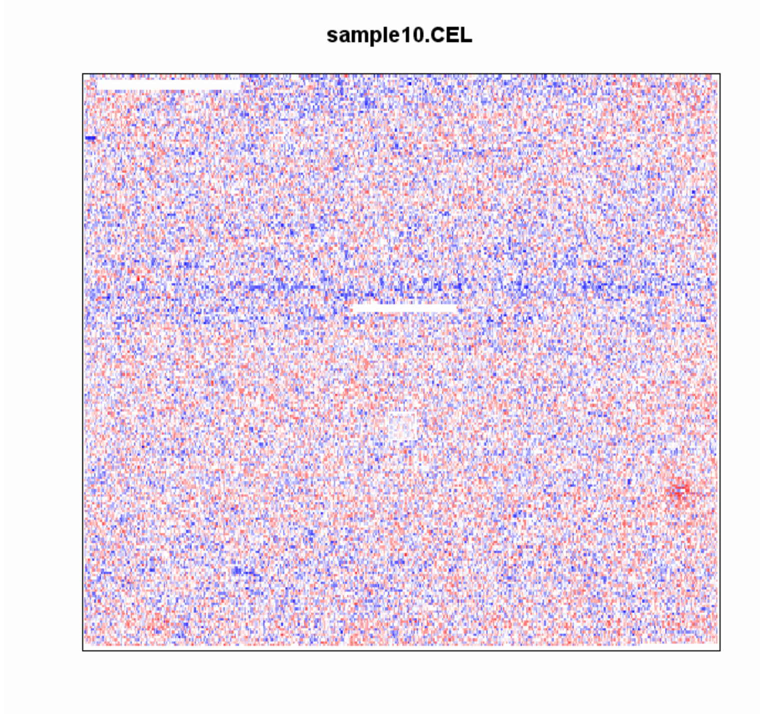
```
####################################################
### chunk number 6:
####################################################
plotPCA(eset, groups = rep(1:4, each = 3),
        groupnames = unique(paste(pData(pd)[,1], pData(pd)[,2], sep = "-")))
plotPCA(eset[,1:6], groups=rep(1:2, each=3),
        groupnames=unique(paste(pData(pd)[1:4,1], pData(pd)[1:4,2], sep="-")))
plotPCA(eset[,7:12], groups=rep(1:2, each=3),
        groupnames=unique(paste(pData(pd)[7:10,1], pData(pd)[7:10,2], sep="-")))
```

image()

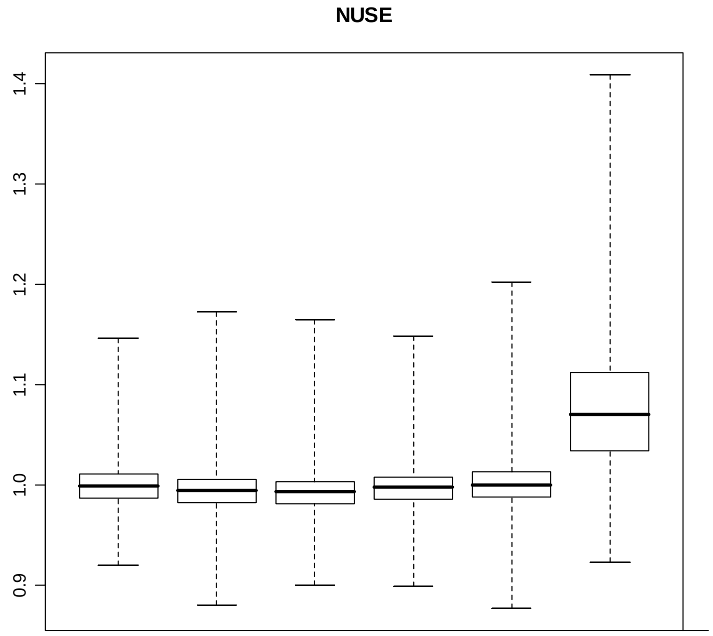
image(pset, type = "resid", which = 1)

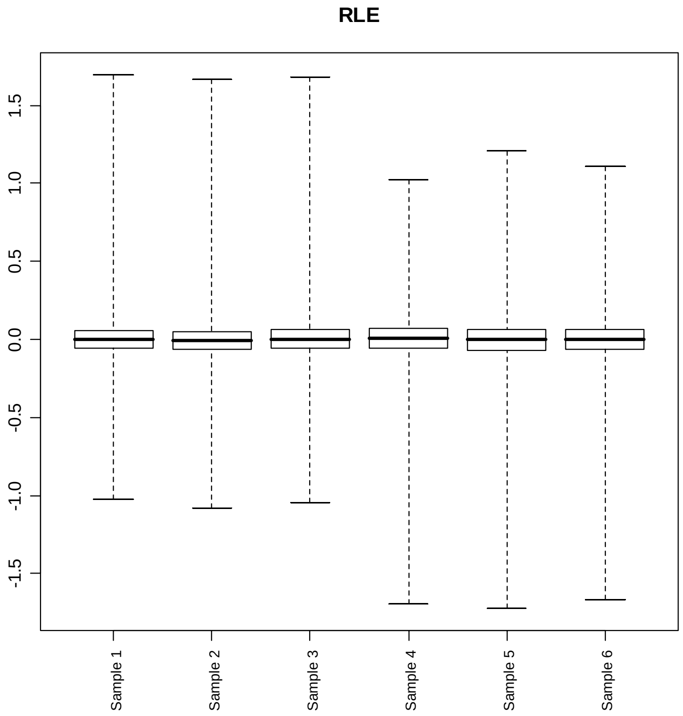


image(pset, type = "resid", which = 10)



boxplot()/Mbox()





Model data/make comparisons

- *limma* package
  - Why *limma*?
- Three step process
  - Design matrix
  - Contrasts matrix
  - Empirical Bayes adjustment

Design matrix

- Matrix of (usually) 0, 1 used to specify model
- Usually easiest to use model.matrix()
- Two models
  - Factor effects
  - Cell means

Cell means model

$$y_{ij} = \mu_i x_i + \varepsilon_{ij}$$

i = 1, 2, 3, 4 (Samples)
j = 1, 2, 3 (Replicates)

In this parameterization:

μ represents the sample mean (hence cell means model)

$\boldsymbol{\varepsilon}$ represents the *error*

$$\begin{pmatrix} y_{11} \\ y_{12} \\ y_{13} \\ y_{21} \\ y_{22} \\ y_{23} \\ y_{31} \\ y_{32} \\ y_{33} \\ y_{41} \\ y_{42} \\ y_{43} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix} \text{x} \begin{pmatrix} \mu_1 \\ \mu_2 \\ \mu_3 \\ \mu_4 \end{pmatrix} + \begin{pmatrix} \varepsilon_{11} \\ \varepsilon_{12} \\ \varepsilon_{13} \\ \varepsilon_{21} \\ \varepsilon_{22} \\ \varepsilon_{23} \\ \varepsilon_{31} \\ \varepsilon_{32} \\ \varepsilon_{33} \\ \varepsilon_{41} \\ \varepsilon_{42} \\ \varepsilon_{43} \end{pmatrix}$$

Cell means model

$$y_{11} = \mu_1 \cdot 1 + \mu_2 \cdot 0 + \mu_3 \cdot 0 + \mu_4 \cdot 0 + \varepsilon_{11}$$

$$\Downarrow$$

$$y_{11} = \mu_1 + \varepsilon_{11} \longleftarrow$$

Here μ_1 estimates the mean expression for A samples.

$$y_{21} = \mu_1 \cdot 0 + \mu_2 \cdot 1 + \mu_3 \cdot 0 + \mu_4 \cdot 0 + \varepsilon_{21}$$

$$\Downarrow$$

$$y_{21} = \mu_2 + \varepsilon_{21} \longleftarrow$$

Here μ_2 estimates the mean expression for B samples.

Cell means design matrix

```
> design <- model.matrix(~ 0 +  factor(rep(1:4, each = 3)))
> colnames(design) <- LETTERS[1:4]
> design

        A       B       C       D
1       1       0       0       0
2       1       0       0       0
3       1       0       0       0
4       0       1       0       0
5       0       1       0       0
6       0       1       0       0
7       0       0       1       0
8       0       0       1       0
9       0       0       1       0
10      0       0       0       1
11      0       0       0       1
12      0       0       0       1
```

Code chunk 9

```
###################################################
### chunk number 9:
###################################################
## filter data to remove probesets that aren't changing
index <- apply(exprs(eset)[,1:6], 1, var) > 0.01
eset1 <- eset[index,]
## create design matrix and give reasonable column names
design <- model.matrix(~ 0 + factor(rep(1:4,each=3)))
colnames(design) <- LETTERS[1:4]
```

Contrasts matrix

- A contrast is a comparison between parameter estimates
- *limma* requires a matrix that specifies the requested comparisons (contrasts matrix)

What is a contrasts matrix?

- Matrix of (usually) 0, 1, -1 used to make comparisons
  - Can use decimal values to compare means of groups
- Best visualized with example

Parameter Estimates

Contrasts Matrix

$$\left(\begin{array}{cccc} A & B & C & D \\ 7.11 & 10.94 & 3.16 & 12.93 \\ 7.19 & 15.05 & 16.71 & 4.55 \\ 3.4 & 16.71 & 13.2 & 13.09 \\ 11.21 & 2.97 & 7.33 & 10.45 \\ 9.72 & 13.05 & 15.41 & 3.42 \\ 5.38 & 9.55 & 3.43 & 10.62 \\ 3.36 & 10.73 & 15.49 & 10.67 \\ 13.51 & 9.15 & 3.01 & 5.37 \\ 5.71 & 9.16 & 5.28 & 8.08 \\ 6.26 & 1.94 & 2.27 & 9.1 \\ 1.96 & 6.69 & 4.11 & 4.46 \\ 4.49 & 1.6 & 6.63 & 6.45 \\ 10.17 & 5 & 16.43 & 14.19 \\ 12.81 & 14.77 & 13.77 & 12.18 \\ 8.32 & 14.45 & 11.97 & 7.55 \\ 5.07 & 13.2 & 3.77 & 7.19 \end{array}\right) \text{x} \begin{pmatrix} 1 & 0 \\ -1 & 0 \\ 0 & 1 \\ 0 & -1 \end{pmatrix} = \begin{pmatrix} -3.83 & -9.77 \\ -7.86 & 12.16 \\ -13.31 & 0.11 \\ 8.24 & -3.12 \\ -3.33 & 11.99 \\ -4.17 & -7.19 \\ -7.37 & 4.82 \\ 4.36 & -2.36 \\ -3.45 & -2.8 \\ 4.32 & -6.83 \\ -4.73 & -0.35 \\ 2.89 & 0.18 \\ 5.17 & 2.24 \\ -1.96 & 1.59 \\ -6.13 & 4.42 \\ -8.13 & -3.42 \end{pmatrix}$$

Simplification

Parameter estimates:

$$A \text{ x } 1 \quad B \text{ x } -1 \quad C \text{ x } 0 \quad D \text{ x } 0 \longrightarrow A - B \longrightarrow \begin{pmatrix} 1 \\ -1 \\ 0 \\ 0 \end{pmatrix}$$

Code chunk 10

```
###################################################
### chunk number 10:
###################################################
## set up a contrasts matrix using makeContrasts()
contrast <- makeContrasts(A - B, C - D, levels = design)

## now do the same using matrix()
contrast <- matrix(c(1,-1,0,0,0,0,1,-1), ncol=2,
              dimnames=list(unique(type), paste(type[c(1,7)],type[c(4,10)],
              sep=" vs ")))
```

Empirical Bayes Adjustment

- Why do we need this?

$$statistic = \frac{\text{difference of means}}{\text{some measure of intra-group variability}}$$

- Mean is efficient
- Variance is not
  - Borrow strength

Code chunk 11

```
########################################################
### chunk number 11:
########################################################
## fit model
fit <- lmFit(eset1, design)
fit2 <- contrasts.fit(fit, contrast)
## empirical Bayes step
fit2 <- eBayes(fit2)
```

Create output/documentation

- Output
  - – HTML tables
  - – text tables
  - – graphics
- Documentation
  - – Written record of the analysis
  - – graphics

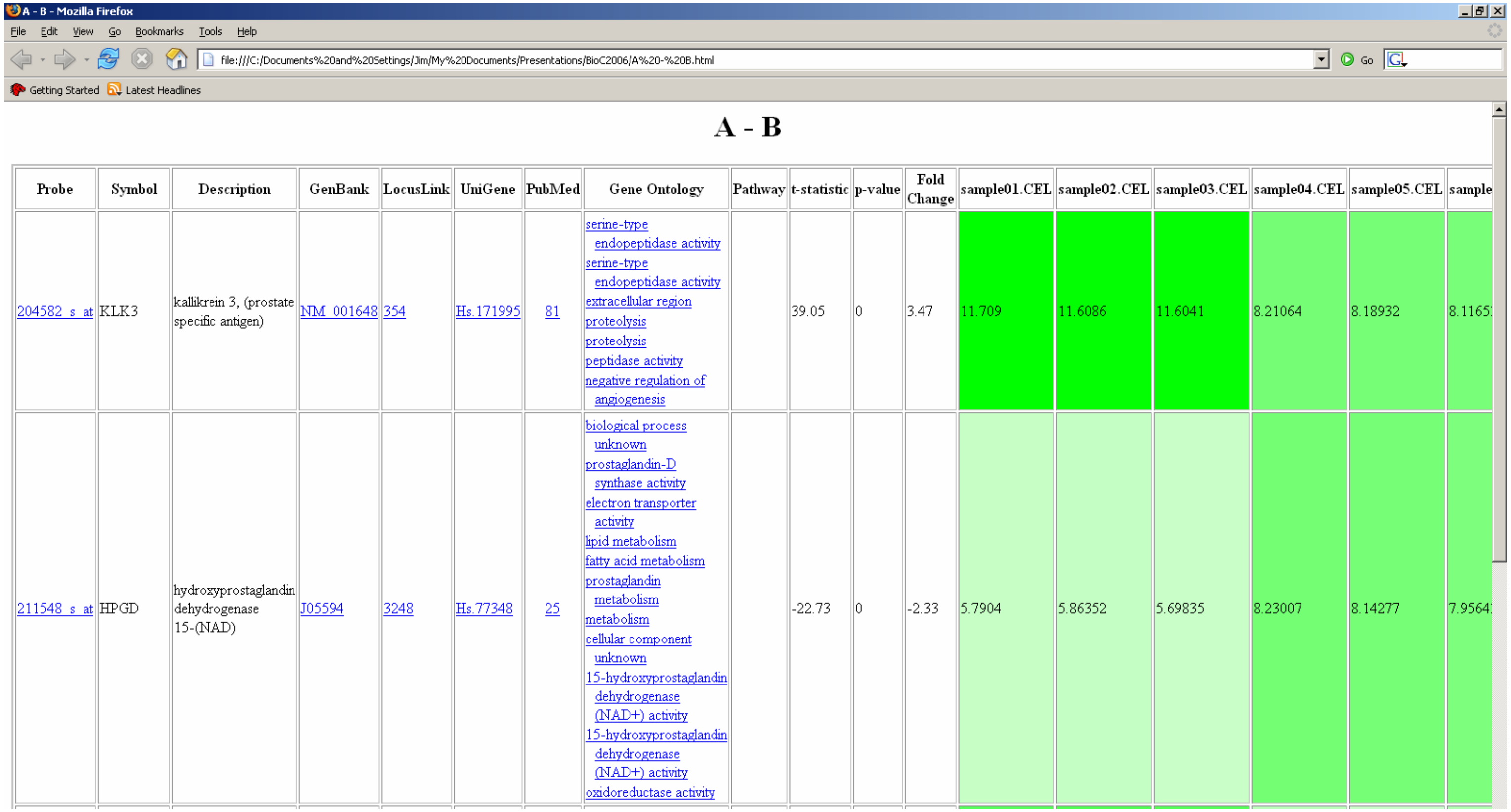
HTML/text tables

- HTML tables
  - – interactive exploration of results
  - – links to databases
- Text tables
  - – easier to manipulate

HTML/text tables

- *annaffy* package
  - *limma2annaffy()*
- *annotate* package/*biomaRt* package
  - limma2biomaRt()

HTML tables


A - B - Mozilla Firefox
File Edit View Go Bookmarks Tools Help
file:///C:/Documents%20and%20Settings/Jim/My%20Documents/Presentations/BioC2006/A%20-%20B.html
Go
Getting Started
Latest Headlines

A - B

Probe	Symbol	Description	GenBank	LocusLink	UniGene	PubMed	Gene Ontology	Pathway	t-statistic	p-value	Fold Change	sample01.CEL	sample02.CEL	sample03.CEL	sample04.CEL	sample05.CEL	sample
204582_s_at	KLK3	kallikrein 3, (prostate specific antigen)	NM_001648	354	Hs.171995	81	serine-type endopeptidase activity serine-type endopeptidase activity extracellular region proteolysis proteolysis peptidase activity negative regulation of angiogenesis		39.05	0	3.47	11.709	11.6086	11.6041	8.21064	8.18932	8.1165
211548_s_at	HPGD	hydroxyprostaglandin dehydrogenase 15-(NAD)	J05594	3248	Hs.77348	25	biological process unknown prostaglandin-D synthase activity electron transporter activity lipid metabolism fatty acid metabolism prostaglandin metabolism metabolism cellular component unknown 15-hydroxyprostaglandin dehydrogenase (NAD+) activity 15-hydroxyprostaglandin dehydrogenase (NAD+) activity oxidoreductase activity		-22.73	0	-2.33	5.7904	5.86352	5.69835	8.23007	8.14277	7.9564

Building HTML tables (*annaffy*)

- Select probesets (genes) for a comparison
- Create a table containing annotation links
- Create a table containing the statistics
- Merge these two tables
- Create a table containing the expression values
- Merge these two tables
- Output the table as HTML
- Output the table as text
- Select next set of probesets and repeat above steps

Code chunk 12

```
####################################################
### chunk number 12:
####################################################
## output text and HTML tables using limma2annaffy()
out <- limma2annaffy(eset1, fit2, design, contrast, annotation(eset),
                   pfilt = 0.05, fldfilt = 1, save = TRUE, text = TRUE,
                   interactive = FALSE)
```

annotate/biomaRt

- Useful when no annotation package exists
  - Newer/less common chips
  - MBNI re-mapped chips
- limma2biomaRt()
  - Very similar to limma2annaffy()
  - Uses *biomaRt* package to annotate
  - Uses htmlpage() from *annotate* package for HTML table
  - ENSEMBL

Graphical output

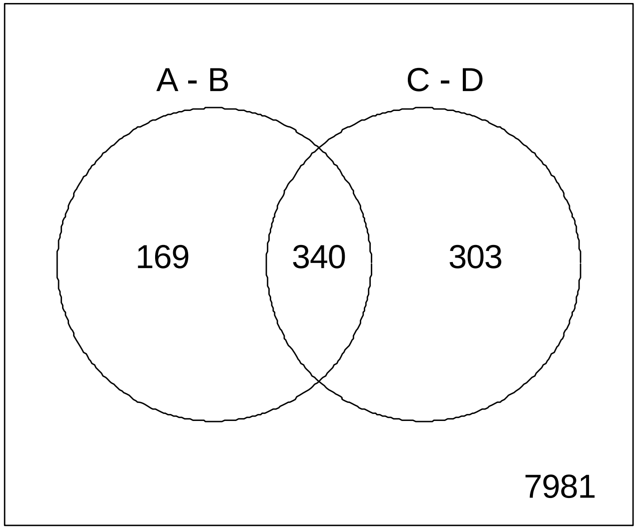
- Quality control plots
- Venn Diagrams

Venn Diagrams

- Common/unique to different comparisons
- decideTests() – *limma* package
- vennCounts2() – *affycoretools* package
  - Select common genes going in same direction
- vennDiagram() – *limma* package

Venn Diagrams

- Nice visual representation
- Great for reports
- But which genes?
- vennSelect() – *affycoretools*
- vennSelectBM() - *affycoretools*



Documentation

- Really two ways to do this
  - Write up something in Word
    - Simple, fast
    - Easiest short term solution
    - Requires boss/client to have Word too
    - Separate analysis/documentation
  - Put analysis/documentation in .Rnw file and use Sweave()
    - Less simple
    - Not a short term solution
    - Requires boss/client to have Acrobat/pdf reader
    - Single analysis/documentation file
    - This is literate programming

What is an .Rnw file?

- Mixture of LaTeX and R code
  - Examples are BioC vignettes
  - Another example in /examples directory of affycoretools package (Statistical_analysis.Rnw)
- Sweave() processes R code and outputs remainder as LaTeX

Why bother?

- Faster in long term
- Consistency in analysis/documentation
- Nicer/more professional looking documentation

Practice

- Run Sweave() on Statistical_analysis.Rnw file
  - Can get updated version we used like this:

  source(http://www.umich.edu/~jmacdon/getRnw.R)
- Run Sweave()

  Sweave("BioC2007.Rnw")
- Convert to pdf

  texi2dvi("BioC2007.tex", pdf=TRUE)

  For help see http://www.ci.tuwien.ac.at/~leisch/Sweave/FAQ.html