Bioinformatics Workshop - NM-AIST

Day 3

Analysis of RNA-Seq Experiments with R and Bioconductor

Thomas Girke

July 25, 2012

Overview

RNA-Seq Analysis

Viewing Results in IGV Genome Browser

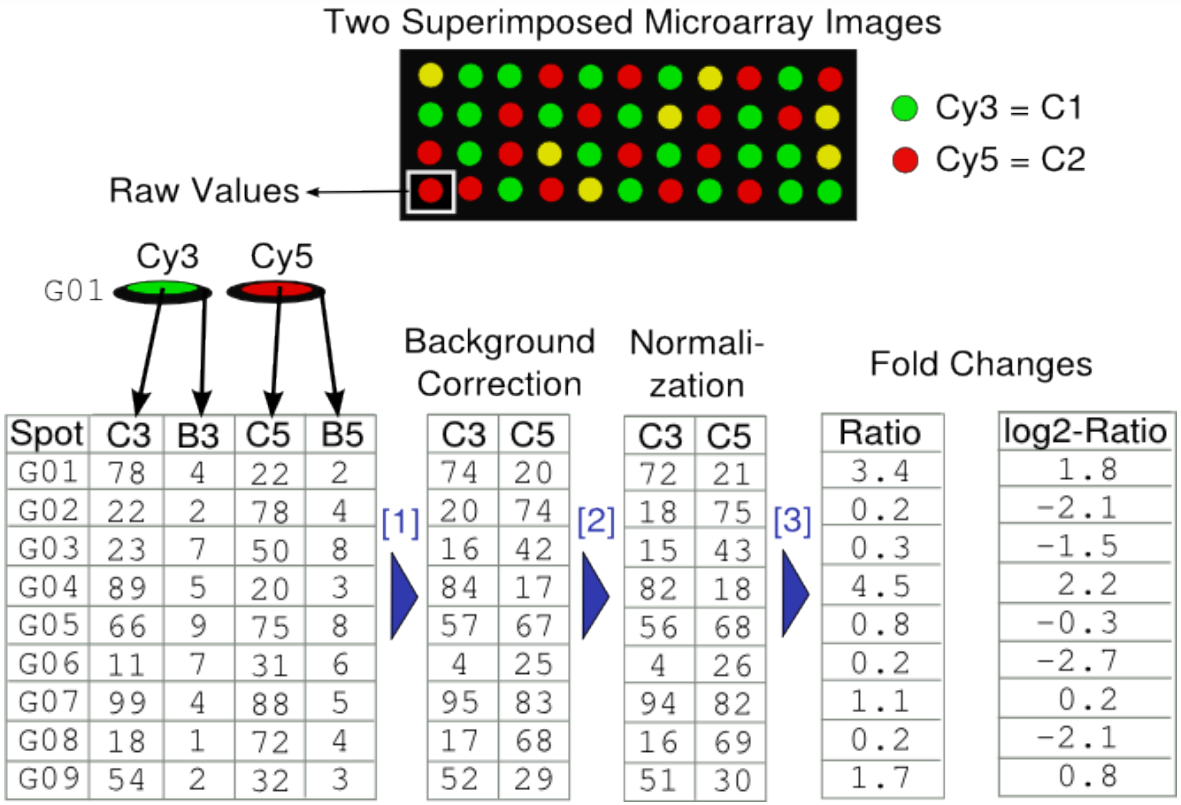
Outline

Overview

RNA-Seq Analysis

Viewing Results in IGV Genome Browser

Comparison with Microarrays (here Dual Color)



Spot	C3	B3	C5	B5
G01	78	4	22	2
G02	22	2	78	4
G03	23	7	50	8
G04	89	5	20	3
G05	66	9	75	8
G06	11	7	31	6
G07	99	4	88	5
G08	18	1	72	4
G09	54	2	32	3

C3	C5
74	20
20	74
16	42
84	17
57	67
4	25
95	83
17	68
52	29

C3	C5
72	21
18	75
15	43
82	18
56	68
4	26
94	82
16	69
51	30

Ratio	log2-Ratio
3.4	1.8
0.2	-2.1
0.3	-1.5
4.5	2.2
0.8	-0.3
0.2	-2.7
1.1	0.2
0.2	-2.1
1.7	0.8

(1) Background correction. Many approaches are available. Most commonly the intensity of the local background is subtracted from the signal intensity: $C1 - B1$ and $C2 - B2$

(2) Normalization step to adjust for global differences between channels. Many methods are available, *e.g.* Loess.

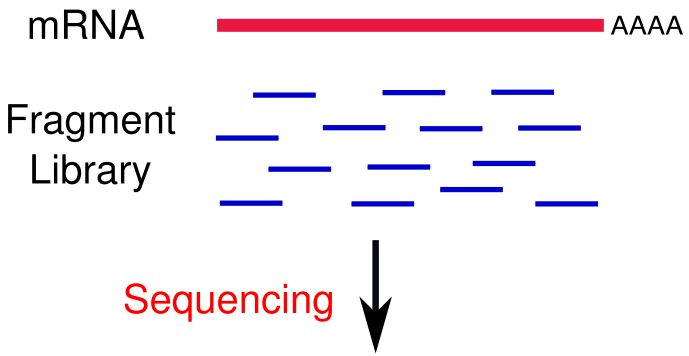
(3) Fold changes to obtain relative fold-changes between samples. Often this is performed in $\log_2$ scale: $\log_2(C3/C5)$ or $\log_2(C3) - \log_2(C5)$

Packages for RNA-Seq Analysis in R

- GenomicRanges Link: high-level infrastructure for range data
- Rsamtools Link: BAM support
- rtracklayer Link: Annotation imports, interface to online genome browsers
- DESeq Link: RNA-Seq DEG analysis
- edgeR Link: RNA-Seq DEG analysis
- DEXSeq Link: RNA-Seq Exon analysis

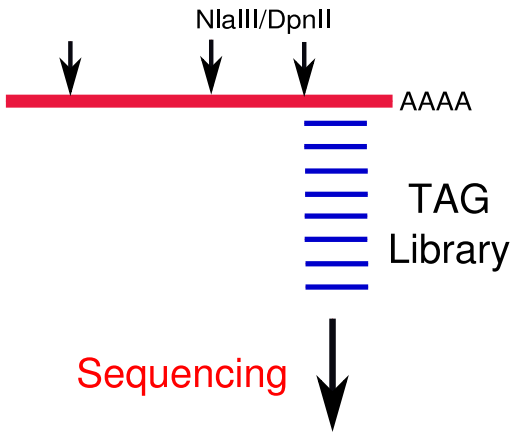
RNA-Seq versus DGE

RNA-seq



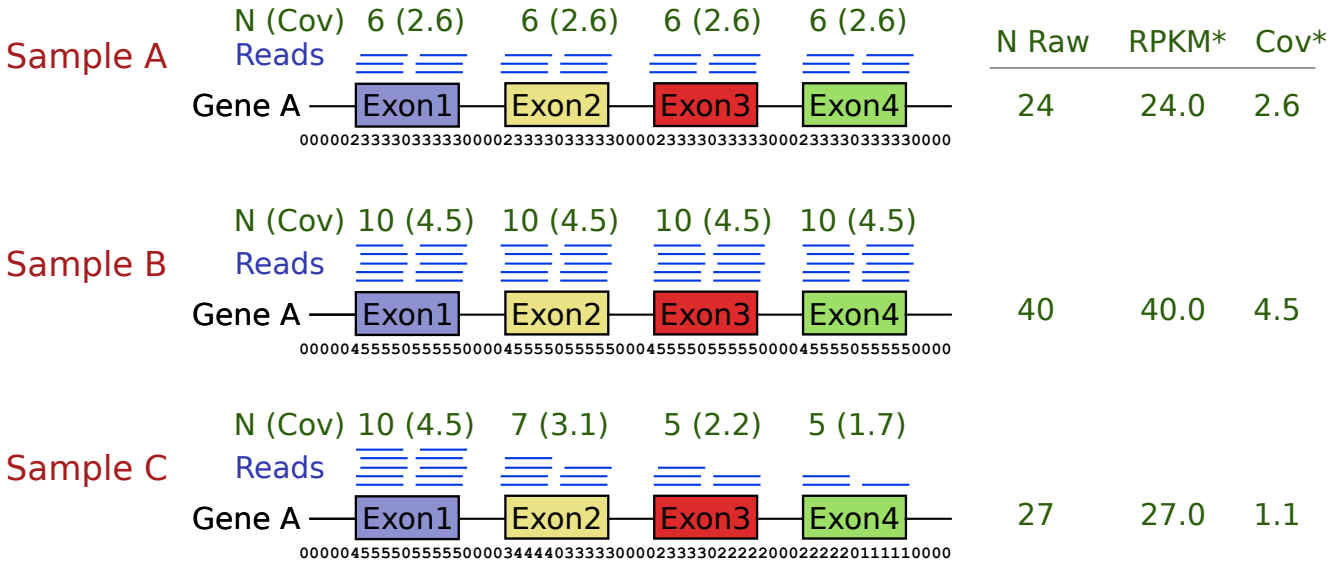
1. Alternative splicing
2. Limited expression profiling
3. SNP detection
4. Many other applications

DGE



1. Expression profiling
→ more appropriate for many biosamples

Identification of Differentially Expressed Genes



	N Raw	RPKM*	Cov*
Sample A	24	24.0	2.6
Sample B	40	40.0	4.5
Sample C	27	27.0	1.1

RPKM*: for 1kb gene and library with 1Mio reads; Cov*: average coverage

Normalization often by library size.

RNA-Seq Analysis Workflow

- Read mapping
- Counting reads overlapping with genes
- Analysis of differentially expressed genes (DEGs)
- Clustering of co-expressed genes
- Gene set/GO term enrichment analysis

Outline

Overview

RNA-Seq Analysis

Viewing Results in IGV Genome Browser

Data Sets and Experimental Variables

- To make the following sample code work, please download and unpack the sample data Link in the directory of your current R session.
- It contains four simplified alignment files from RNA-Seq experiment SRA023501 Link and a shortened GFF Link to allow fast analysis on a laptop.
- The alignments were created by aligning the reads with Bowtie against the Arabidopsis reference genome.
- Note: usually, the aligned reads would be stored in BAM format and then imported into R with the `readBamGappedAlignments` function (see below)!

This information could be imported from an external targets file

```
> targets <- data.frame(Samples=c("AP3 domain, flower stage 4", "AP3 domain, flower stage 4",
+                       "Translatome, flower stage 4", "Translatome, flower stage 4"),
+                       Factor=c("AP3", "AP3", "TRL", "TRL"),
+                       Fastq=c("SRR064154", "SRR064155", "SRR064166", "SRR064167"))
> targets

                      Samples Factor     Fastq
1  AP3 domain, flower stage 4    AP3 SRR064154
2  AP3 domain, flower stage 4    AP3 SRR064155
3 Translatome, flower stage 4    TRL SRR064166
4 Translatome, flower stage 4    TRL SRR064167
```

Import Annotation Data from GFF

Annotation data from GFF

```
> library(rtracklayer); library(GenomicRanges); library(Rsamtools)
> gff <- import.gff("./data/TAIR10_GFF3_trunc.gff", asRangedData=FALSE)
> seqlengths(gff) <- end(ranges(gff[which(elementMetadata(gff)[,"type"]=="chromosome"),]))
> subgene_index <- which(elementMetadata(gff)[,"type"] == "gene")
> gffsub <- gff[subgene_index,] # Returns only gene ranges
> strand(gffsub) <- "*" # For strand insensitive analysis
> gffsub[1:4,1:2]

GRanges with 4 ranges and 2 elementMetadata cols:
      seqnames         ranges strand |   source     type
         <Rle>      <IRanges>  <Rle> | <factor> <factor>
  [1]        1 [ 3631,  5899]      * |   TAIR10     gene
  [2]        1 [ 5928,  8737]      * |   TAIR10     gene
  [3]        1 [11649, 13714]      * |   TAIR10     gene
  [4]        1 [23146, 31227]      * |   TAIR10     gene
  ---
  seqlengths:
             1           2           3           4           5  chloroplast mitochondria
      30427671    19698289    23459830    18585056    26975502      154478      366924

> ids <- as.character(elementMetadata(gffsub)[, "group"])
> gffsub <- split(gffsub, ids) # Coerce to GRangesList
```

Read Counting per Annotation Range

Number of reads overlapping gene ranges

```
> samples <- as.character(targets$Fastq)
> samplespath <- paste("./data/", samples, sep="")
> countDF <- data.frame(row.names=ids)
> for(i in samplespath) {
+         # aligns <- readBamGappedAlignments(i) # Substitute next two lines with this one.
+         aligns <- read.table(i, header=TRUE)
+         aligns <- GRanges(seqnames=aligns$seqnames, IRanges(aligns$start, aligns$end), strand=aligns$stra
+         counts <- countOverlaps(gffsub, aligns)
+         countDF <- cbind(countDF, counts)
+ }
> colnames(countDF) <- samples
> rownames(countDF) <- gsub(".*=", "", rownames(countDF))
> countDF[1:4,]
```

	SRR064154	SRR064155	SRR064166	SRR064167
AT1G01010	50	24	64	76
AT1G01020	132	79	89	59
AT1G01030	5	0	16	15
AT1G01040	491	347	330	374

```
> write.table(countDF, "./data/countDF", quote=FALSE, sep="\t", col.names = NA)
> countDF <- read.table("./data/countDF")
```

Simple RPKM Normalization

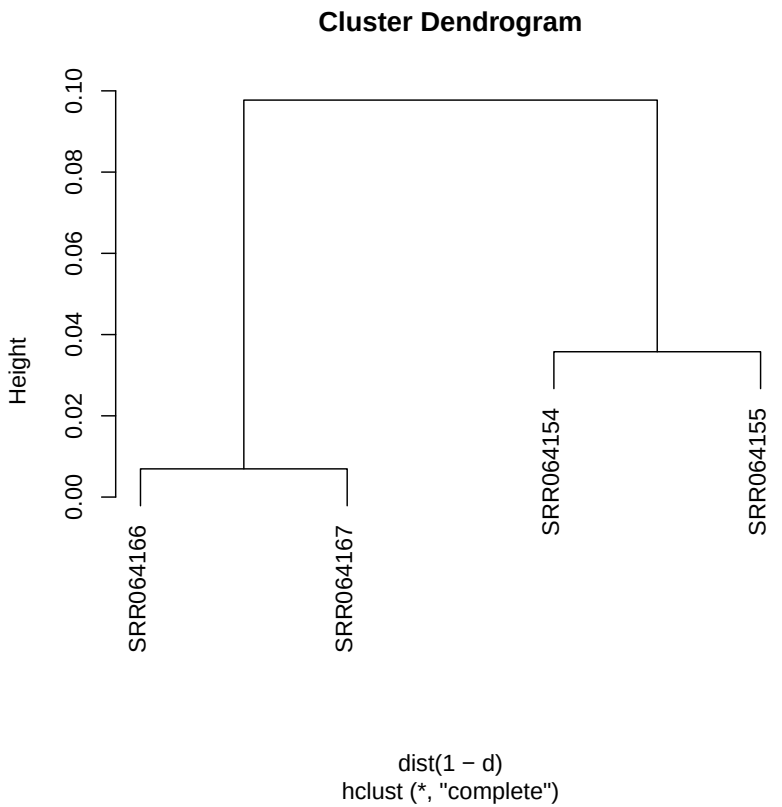
RPKM: reads per kilobase of exon model per million mapped reads

```
> returnRPKM <- function(counts, gffsub) {
+         geneLengthsInKB <- sum(width(gffsub))/1000 # Number of bases per exonRanges element in kbp
+         millionsMapped <- sum(counts)/1e+06 # Factor for converting to million of mapped reads.
+         rpm <- counts/millionsMapped # RPK: reads per kilobase of exon model.
+         rpkm <- rpm/geneLengthsInKB # RPKM: reads per kilobase of exon model per million mapped reads.
+         return(rpkm)
+ }
> countDFrpkm <- apply(countDF, 2, function(x) returnRPKM(counts=x, gffsub=gffsub))
> countDFrpkm[1:4,]

           SRR064154 SRR064155 SRR064166 SRR064167
AT1G01010  41.094413  18.76055 345.02646 347.28848
AT1G01020  87.602174  49.86428 387.42765 217.69927
AT1G01030   4.513225   0.00000  94.73198  75.27872
AT1G01040 113.294785  76.15166 499.46150 479.80419
```

QC Check

QC check by computing a sample correlating matrix and plotting it as a tree

```
> d <- cor(countDF, method="pearson")
> plot(hclust(dist(1-d))) # Sample tree
```



Identify DEGs with Simple Fold Change Method

Compute mean values for replicates

```
> source("http://faculty.ucr.edu/~tgirke/Documents/R_BioCond/My_R_Scripts/colAg.R")
> countDFrpkm_mean <- colAg(myMA=countDFrpkm, group=c(1,1,2,2), myfct=mean)
> countDFrpkm_mean[1:4,]
```

```
          SRR064154_SRR064155 SRR064166_SRR064167
AT1G01010           29.927480           346.15747
AT1G01020           68.733226           302.56346
AT1G01030            2.256612            85.00535
AT1G01040           94.723224           489.63284
```

Log2 fold changes

```
> countDFrpkm_mean <- cbind(countDFrpkm_mean, log2ratio=log2(countDFrpkm_mean[,1]/countDFrpkm_mean[,2]))
> countDFrpkm_mean <- countDFrpkm_mean[is.finite(countDFrpkm_mean[,3]), ]
> degs2fold <- countDFrpkm_mean[countDFrpkm_mean[,3] >= 1 | countDFrpkm_mean[,3] <= -1,]
> degs2fold[1:4,]
```

```
          SRR064154_SRR064155 SRR064166_SRR064167 log2ratio
AT1G01010           29.927480           346.15747 -3.531886
AT1G01020           68.733226           302.56346 -2.138158
AT1G01030            2.256612            85.00535 -5.235323
AT1G01040           94.723224           489.63284 -2.369910
```

```
> write.table(degs2fold, "./data/degs2fold", quote=FALSE, sep="\t", col.names = NA)
> degs2fold <- read.table("./data/degs2fold")
```

Identify DEGs with DESeq Library

Raw count data are expected here!

```
> library(DESeq)
> countDF <- read.table("./data/countDF")
> conds <- targets$Factor
> cds <- newCountDataSet(countDF, conds) # Creates object of class CountDataSet derived from eSet class
> counts(cds)[1:4, ] # CountDataSet has similar accessor methods as eSet class.

          SRR064154 SRR064155 SRR064166 SRR064167
AT1G01010        50        24        64        76
AT1G01020       132        79        89        59
AT1G01030         5         0        16        15
AT1G01040       491       347       330       374

> cds <- estimateSizeFactors(cds) # Estimates library size factors from count data. Alternatively, one can
> cds <- estimateDispersions(cds) # Estimates the variance within replicates
> res <- nbinomTest(cds, "TRL", "AP3") # Calls DEGs with nbinomTest
> res <- na.omit(res)
> res2fold <- res[res$log2FoldChange >= 1 | res$log2FoldChange <= -1,]
> res2foldpadj <- res2fold[res2fold$padj <= 0.01, ]
> res2foldpadj[1:4,1:8]

          id     baseMean   baseMeanA    baseMeanB foldChange log2FoldChange         pval         padj
6  AT1G01050    565.38535   858.46503 2.723057e+02 0.31720066      -1.656532 1.106761e-09 1.291221e-08
7  AT1G01060    299.47779   423.36877 1.755868e+02 0.41473725      -1.269730 4.613776e-05 2.202029e-04
8  AT1G01070     29.30875    53.97968 4.637819e+00 0.08591786      -3.540898 2.499200e-05 1.353315e-04
16 AT2G01010 223420.41841 59874.64881 3.869662e+05 6.46293875       2.692190 4.142756e-06 2.718684e-05
```

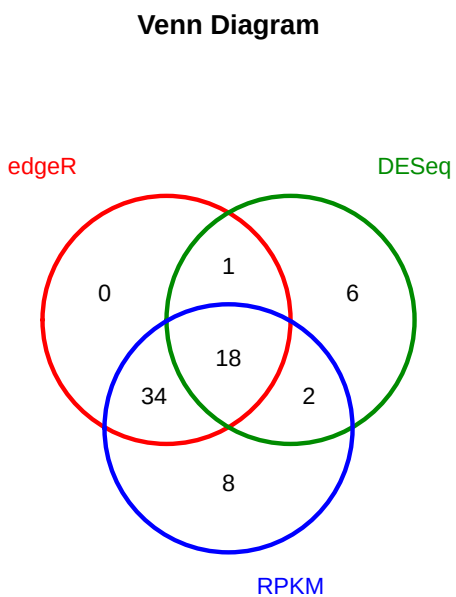
Identify DEGs with edgeR Library

Raw count data are expected here!

```
> library(edgeR)
> countDF <- read.table("./data/countDF")
> y <- DGEList(counts=countDF, group=conds) # Constructs DGEList object
> y <- estimateCommonDisp(y) # Estimates common dispersion
> y <- estimateTagwiseDisp(y) # Estimates tagwise dispersion
> et <- exactTest(y, pair=c("TRL", "AP3")) # Computes exact test for the negative binomial distribution.
> topTags(et, n=4)

Comparison of groups:  AP3-TRL
              logFC   logCPM       PValue          FDR
AT4G00050 -5.720181 10.69408 1.076523e-60 1.550193e-58
AT1G01050 -4.320216 12.29352 5.992852e-55 4.314853e-53
AT3G01120 -4.071353 14.32720 2.290595e-51 1.099485e-49
AT1G01060 -3.893107 11.26311 1.443146e-39 5.195326e-38

> edge <- as.data.frame(topTags(et, n=50000))
> edge2fold <- edge[edge$logFC >= 1 | edge$logFC <= -1,]
> edge2foldpadj <- edge2fold[edge2fold$FDR <= 0.01, ]
```

Merge Results and Compute Overlaps Among Methods

```
> bothDF <- merge(res, countDFrpkm_mean, by.x=1, by.y=0, all=TRUE); bothDF <- na.omit(bothDF)
> cor(bothDF[,"log2FoldChange"], bothDF[,"log2ratio"], method="spearman")

[1] 0.9989686

> source("http://faculty.ucr.edu/~tgirke/Documents/R_BioCond/My_R_Scripts/overLapper.R")
> setlist <- list(edgeR=rownames(edge2foldpadj), DESeq=as.character(res2foldpadj[,1]), RPKM=rownames(degs2f
> OLlist <- overLapper(setlist=setlist, sep="_", type="vennsets")
> counts <- sapply(OLlist$Venn_List, length)
> vennPlot(counts=counts)
```



Enrichment of GO Terms in DEG Sets

GO Term Enrichment Analysis

```
> library(GOstats); library(GO.db); library(ath1121501.db)
> geneUniverse <- rownames(countDF)
> geneSample <- res2foldpadj[,1]
> params <- new("GOHyperGParams", geneIds = geneSample, universeGeneIds = geneUniverse,
+         annotation="ath1121501", ontology = "MF", pvalueCutoff = 0.5,
+         conditional = FALSE, testDirection = "over")
> hgOver <- hyperGTest(params)
> summary(hgOver)[1:4,]

      GOMFID      Pvalue OddsRatio ExpCount Count Size
1 GO:0016168 0.009102704 16.857143 1.153846     4    5                                                  chloroph
2 GO:0046906 0.009102704 16.857143 1.153846     4    5                                                 tetrapyr
3 GO:0015077 0.023160627  8.285714 1.384615     4    6 monovalent inorganic cation transmembrane transport
4 GO:0015078 0.023160627  8.285714 1.384615     4    6                 hydrogen ion transmembrane transport

> htmlReport(hgOver, file = "data/MyhyperGresult.html")
```

Outline

Overview

RNA-Seq Analysis

Viewing Results in IGV Genome Browser

Inspect Results in IGV

View results in IGV

- Download and open IGV Link
- Select in menu in top left corner `A. thaliana (TAIR10)`
- Upload the following indexed/sorted Bam files with `File -> Load from URL...`

http://faculty.ucr.edu/~tgirke/HTML_Presentations/Manuals/Rngsapps/chipseqBioc2012/results/SRR038845.fastq.bam
http://faculty.ucr.edu/~tgirke/HTML_Presentations/Manuals/Rngsapps/chipseqBioc2012/results/SRR038846.fastq.bam
http://faculty.ucr.edu/~tgirke/HTML_Presentations/Manuals/Rngsapps/chipseqBioc2012/results/SRR038848.fastq.bam
http://faculty.ucr.edu/~tgirke/HTML_Presentations/Manuals/Rngsapps/chipseqBioc2012/results/SRR038850.fastq.bam

- To view area of interest, enter its coordinates `Chr1:16656-16956` in position menu on top.

Session Information

```
> sessionInfo()

R version 2.15.0 (2012-03-30)
Platform: x86_64-unknown-linux-gnu (64-bit)
locale:
[1] C

attached base packages:
[1] stats     graphics  utils     datasets  grDevices methods   base

other attached packages:
 [1] xtable_1.7-0         ath1121501.db_2.7.1  org.At.tair.db_2.7.1 GO.db_2.7.1          GOstats_2.22.0
[11] edgeR_2.6.12         limma_3.12.3         DESeq_1.8.3          locfit_1.5-8         Biobase_2.16.0
[21] BiocGenerics_0.2.0

loaded via a namespace (and not attached):
 [1] BSgenome_1.24.0    GSEABase_1.18.0    RBGL_1.32.1        RColorBrewer_1.0-5 RCurl_1.95-1.1     XML_3.
[12] lattice_0.20-10    splines_2.15.0     stats4_2.15.0      survival_2.36-14   tools_2.15.0       zlibbi
```