Analysis of RNA-Seq Data with R/Bioconductor

... 

Thomas Girke

December 8, 2012
Overview

RNA-Seq Analysis
  Aligning Short Reads

Viewing Results in IGV Genome Browser
Outline

Overview

RNA-Seq Analysis
  Aligning Short Reads

Viewing Results in IGV Genome Browser
Packages for RNA-Seq Analysis in R

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

**RNA-seq**

- mRNA
- Fragment Library
- Sequencing
  1. Alternative splicing
  2. Limited expression profiling
  3. SNP detection
  4. Many other applications

**DGE**

- TAG Library
- Sequencing
  1. Expression profiling
    - more appropriate for many biosamples
Identification of Differentially Expressed Genes

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
  Aligning Short Reads

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 in the directory of your current R session.

It contains four simplified alignment files from RNA-Seq experiment SRA023501 and a shortened GFF 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

```r
> targets <- read.delim("./data/targets.txt")
> targets

          Samples Factor Fastq
1        AP3_f14   AP3  SRR064154.fastq
2        AP3_f14   AP3  SRR064155.fastq
3        T1_f14   TRL  SRR064166.fastq
4        T1_f14   TRL  SRR064167.fastq
```
Outline

Overview

RNA-Seq Analysis
  Aligning Short Reads

Viewing Results in IGV Genome Browser
Align Reads and Output Indexed Bam Files

Note: this steps requires the command-line tool bowtie2. If it is not available on a system then one can skip this mapping step and download the pre-generated Bam files from here:

```r
> library(Rsamtools)
> dir.create("results") # Note: all output data will be written to directory 'results'
> system("bowtie2-build ./data/tair10chr.fasta ./data/tair10chr.fasta") # Build indexed reference genome
> targets <- read.delim("./data/targets.txt") # Import experiment design information
> targets
> input <- paste("./data/", targets[,3], sep="")
> output <- paste("./results/", targets[,3], ".sam", sep="")
> reference <- "./results/tair10chr.fasta"
> for(i in seq(along=targets[,3])) {
+ command <- paste("bowtie2 -x ./data/tair10chr.fasta -U", input[i], "-S", output[i])
+ system(command)
+ asBam(file=output[i], destination=gsub(".sam", ",", output[i]), overwrite=TRUE, indexDestination=TRUE)
+ unlink(output[i])
+ }
```
Import Annotation Data from GFF

Annotation data from GFF

```r
> 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"] == "exon")
> gffsub <- gff[subgene_index,] # Returns only gene ranges
> gffsub[1:4, c(2,5)]

GRanges with 4 ranges and 2 metadata columns:

<table>
<thead>
<tr>
<th>seqlines</th>
<th>ranges</th>
<th>strand</th>
<th>type</th>
<th>group</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;Rle&gt;</td>
<td>&lt;IRanges&gt;</td>
<td>&lt;Rle&gt;</td>
<td>&lt;factor&gt;</td>
<td>&lt;factor&gt;</td>
</tr>
<tr>
<td>[1]</td>
<td>Chr1 [3631, 3913]</td>
<td>+</td>
<td>exon</td>
<td>Parent=AT1G01010.1</td>
</tr>
<tr>
<td>[2]</td>
<td>Chr1 [3996, 4276]</td>
<td>+</td>
<td>exon</td>
<td>Parent=AT1G01010.1</td>
</tr>
<tr>
<td>[3]</td>
<td>Chr1 [4486, 4605]</td>
<td>+</td>
<td>exon</td>
<td>Parent=AT1G01010.1</td>
</tr>
<tr>
<td>[4]</td>
<td>Chr1 [4706, 5095]</td>
<td>+</td>
<td>exon</td>
<td>Parent=AT1G01010.1</td>
</tr>
</tbody>
</table>

---

seqlengths:

<table>
<thead>
<tr>
<th>Chr1</th>
<th>Chr2</th>
<th>Chr3</th>
<th>Chr4</th>
<th>Chr5</th>
<th>ChrC</th>
<th>ChrM</th>
</tr>
</thead>
<tbody>
<tr>
<td>100000</td>
<td>100000</td>
<td>100000</td>
<td>100000</td>
<td>100000</td>
<td>100000</td>
<td>100000</td>
</tr>
</tbody>
</table>

> ids <- gsub("Parent=|\..*", ", elementMetadata(gffsub)$group)
> gffsub <- split(gffsub, ids) # Coerce to GRangesList
```
Number of reads overlapping gene ranges

```r
> samples <- as.character(targets$Fastq)
> samplespath <- paste("./results/", samples, ".bam", sep="")
> names(samplespath) <- samples
> countDF <- data.frame(row.names=names(gffsub))
> for(i in samplespath) {
+     aligns <- readBamGappedAlignments(i) # Substitute next two lines with this one.
+     counts <- countOverlaps(gffsub, aligns, ignore.strand=TRUE)
+     countDF <- cbind(countDF, counts)
+ }
> colnames(countDF) <- samples
> countDF[1:4,]
```

```
SRR064154.fastq  SRR064155.fastq  SRR064166.fastq  SRR064167.fastq
AT1G01010    52       26       60       75
AT1G01020    146      77       82       64
AT1G01030     5       13       13       14
AT1G01040    483      347      302      358
```

```r
> write.table(countDF, "./results/countDF", quote=FALSE, sep="\t", col.names = NA)
> countDF <- read.table("./results/countDF")
```
The `summarizeOverlaps` function from the GenomicRanges is easier to use and provides more options. See here [Link](#) for details.

```r
> bfl <- BamFileList(samplespath, index=character())
> countDF2 <- summarizeOverlaps(gffsub, bfl, mode="Union", ignore.strand=TRUE)
> countDF2[1:4,]
```

<table>
<thead>
<tr>
<th>SRR064154.fastq</th>
<th>SRR064155.fastq</th>
<th>SRR064166.fastq</th>
<th>SRR064167.fastq</th>
</tr>
</thead>
<tbody>
<tr>
<td>AT1G01010</td>
<td>52</td>
<td>26</td>
<td>60</td>
</tr>
<tr>
<td>AT1G01020</td>
<td>146</td>
<td>77</td>
<td>82</td>
</tr>
<tr>
<td>AT1G01030</td>
<td>5</td>
<td>1</td>
<td>13</td>
</tr>
<tr>
<td>AT1G01040</td>
<td>480</td>
<td>346</td>
<td>282</td>
</tr>
</tbody>
</table>
Simple RPKM Normalization

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

```r
> returnRPKM <- function(counts, gffsub) {
+   geneLengthsInKB <- sum(width(reduce(gffsub)))/1000 # Length of exon union per gene 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,]
```

<table>
<thead>
<tr>
<th>SRR064154.fastq</th>
<th>SRR064155.fastq</th>
<th>SRR064166.fastq</th>
<th>SRR064167.fastq</th>
</tr>
</thead>
<tbody>
<tr>
<td>AT1G01010</td>
<td>52.478177</td>
<td>24.641790</td>
<td>387.88096</td>
</tr>
<tr>
<td>AT1G01020</td>
<td>140.199699</td>
<td>69.439799</td>
<td>504.40560</td>
</tr>
<tr>
<td>AT1G01030</td>
<td>4.471187</td>
<td>0.839801</td>
<td>74.46772</td>
</tr>
<tr>
<td>AT1G01040</td>
<td>131.564008</td>
<td>88.765242</td>
<td>526.94917</td>
</tr>
</tbody>
</table>
QC Check

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

```r
> library(ape)
> d <- cor(countDF, method="spearman")
> hc <- hclust(dist(1-d))
> plot.phylo(as.phylo(hc), type="p", edge.col=4, edge.width=2, show.node.label=TRUE, no.margin=TRUE)
```
Identify DEGs with Simple Fold Change Method

Compute mean values for replicates

```r
> 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,]

<table>
<thead>
<tr>
<th></th>
<th>SRR064154.fastq</th>
<th>SRR064155.fastq</th>
<th>SRR064166.fastq</th>
<th>SRR064167.fastq</th>
</tr>
</thead>
<tbody>
<tr>
<td>AT1G01010</td>
<td>38.559984</td>
<td>376.61987</td>
<td>376.61987</td>
<td>376.61987</td>
</tr>
<tr>
<td>AT1G01020</td>
<td>104.819749</td>
<td>400.53214</td>
<td>400.53214</td>
<td>400.53214</td>
</tr>
<tr>
<td>AT1G01030</td>
<td>2.655494</td>
<td>67.44964</td>
<td>67.44964</td>
<td>67.44964</td>
</tr>
<tr>
<td>AT1G01040</td>
<td>110.164625</td>
<td>498.83091</td>
<td>498.83091</td>
<td>498.83091</td>
</tr>
</tbody>
</table>
```

Log2 fold changes

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

<table>
<thead>
<tr>
<th></th>
<th>SRR064154.fastq</th>
<th>SRR064155.fastq</th>
<th>SRR064166.fastq</th>
<th>SRR064167.fastq</th>
<th>log2ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>AT1G01010</td>
<td>38.559984</td>
<td>376.61987</td>
<td>376.61987</td>
<td>376.61987</td>
<td>3.287933</td>
</tr>
<tr>
<td>AT1G01020</td>
<td>104.819749</td>
<td>400.53214</td>
<td>400.53214</td>
<td>400.53214</td>
<td>1.934007</td>
</tr>
<tr>
<td>AT1G01030</td>
<td>2.655494</td>
<td>67.44964</td>
<td>67.44964</td>
<td>67.44964</td>
<td>4.666758</td>
</tr>
<tr>
<td>AT1G01040</td>
<td>110.164625</td>
<td>498.83091</td>
<td>498.83091</td>
<td>498.83091</td>
<td>2.178890</td>
</tr>
</tbody>
</table>
```

```r
> write.table(degs2fold, ".//results/degs2fold", quote=FALSE, sep="\t", col.names = NA)
> degs2fold <- read.table("./results/degs2fold")
```
Identify DEGs with DESeq Library

Raw count data are expected here!

```r
> library(DESeq)
> countDF <- read.table("./results/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.
```

<table>
<thead>
<tr>
<th>SRR064154.fastq</th>
<th>SRR064155.fastq</th>
<th>SRR064166.fastq</th>
<th>SRR064167.fastq</th>
</tr>
</thead>
<tbody>
<tr>
<td>AT1G01010</td>
<td>52</td>
<td>26</td>
<td>60</td>
</tr>
<tr>
<td>AT1G01020</td>
<td>146</td>
<td>77</td>
<td>82</td>
</tr>
<tr>
<td>AT1G01030</td>
<td>5</td>
<td>1</td>
<td>13</td>
</tr>
<tr>
<td>AT1G01040</td>
<td>483</td>
<td>347</td>
<td>302</td>
</tr>
</tbody>
</table>

```r
> cds <- estimateSizeFactors(cds) # Estimates library size factors from count data. Alternatively, one can
> cds <- estimateDispersions(cds) # Estimates the variance within replicates
> res <- nbinomTest(cds, "AP3", "TRL") # Calls DEGs with nbinomTest
> res <- na.omit(res)
> res2fold <- res[res$log2FoldChange >= 1 | res$log2FoldChange <= -1,]
> res2foldpadj <- res2fold[res2fold$padj <= 0.05, ]
> res2foldpadj[1:4,1:8]
```

<table>
<thead>
<tr>
<th>id</th>
<th>baseMean</th>
<th>baseMeanA</th>
<th>baseMeanB</th>
<th>foldChange</th>
<th>log2FoldChange</th>
<th>pval</th>
<th>padj</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>AT1G01050</td>
<td>600.91989</td>
<td>273.11390</td>
<td>928.725867</td>
<td>3.40050744</td>
<td>1.765750</td>
<td>5.341934e-12</td>
</tr>
<tr>
<td>7</td>
<td>AT1G01060</td>
<td>302.03514</td>
<td>169.61662</td>
<td>434.453652</td>
<td>2.56138611</td>
<td>1.356925</td>
<td>4.032316e-06</td>
</tr>
<tr>
<td>8</td>
<td>AT1G01070</td>
<td>29.86593</td>
<td>5.66738</td>
<td>54.064485</td>
<td>9.53959078</td>
<td>3.253927</td>
<td>2.421810e-05</td>
</tr>
<tr>
<td>15</td>
<td>AT2G01008</td>
<td>18.59928</td>
<td>34.71273</td>
<td>2.485829</td>
<td>0.07161145</td>
<td>-3.803666</td>
<td>2.417262e-04</td>
</tr>
</tbody>
</table>
Identify DEGs with edgeR Library

Raw count data are expected here!

```r
> library(edgeR)
> countDF <- read.table("./results/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("AP3", "TRL")) # Computes exact test for the negative binomial distribution.
> topTags(et, n=4)

Comparison of groups: TRL-AP3

<table>
<thead>
<tr>
<th>logFC</th>
<th>logCPM</th>
<th>PValue</th>
<th>FDR</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.568721</td>
<td>10.34789</td>
<td>5.473636e-54</td>
<td>8.265190e-52</td>
</tr>
<tr>
<td>4.203665</td>
<td>12.09034</td>
<td>4.331645e-45</td>
<td>3.270392e-43</td>
</tr>
<tr>
<td>3.865628</td>
<td>14.07787</td>
<td>1.674890e-41</td>
<td>8.430278e-40</td>
</tr>
<tr>
<td>3.757064</td>
<td>10.98503</td>
<td>3.360358e-34</td>
<td>1.268535e-32</td>
</tr>
</tbody>
</table>
```

```r
> edge <- as.data.frame(topTags(et, n=50000))
> edge2fold <- edge[abs(edge$logFC) >= 1]  # edge$logFC <= -1,
> edge2foldpadj <- edge2fold[edge2fold$FDR <= 0.01,]
```
Merge Results and Compute Overlaps Among Methods

```r
> 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] 0.9985348

> 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(degs2fold)
> OLlist <- overLapper(setlist=setlist, sep="_", type="vennsets")
> counts <- sapply(OLlist$Venn_List, length)
> vennPlot(counts=counts)
```

Venn Diagram

Unique objects: All = 84; S1 = 61; S2 = 35; S3 = 77
Enrichment of GO Terms in DEG Sets

**GO Term Enrichment Analysis**

```r
> 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,]

<table>
<thead>
<tr>
<th>GOMFID</th>
<th>Pvalue</th>
<th>OddsRatio</th>
<th>ExpCount</th>
<th>Count</th>
<th>Size</th>
<th>Term</th>
</tr>
</thead>
<tbody>
<tr>
<td>GO:0008324</td>
<td>0.002673178</td>
<td>18</td>
<td>2.126582</td>
<td>6</td>
<td>7</td>
<td>cation transmembrane transporter activity</td>
</tr>
<tr>
<td>GO:0015075</td>
<td>0.002673178</td>
<td>18</td>
<td>2.126582</td>
<td>6</td>
<td>7</td>
<td>ion transmembrane transporter activity</td>
</tr>
<tr>
<td>GO:0015077</td>
<td>0.002673178</td>
<td>18</td>
<td>2.126582</td>
<td>6</td>
<td>7</td>
<td>monovalent inorganic cation transmembrane transport activity</td>
</tr>
<tr>
<td>GO:0015078</td>
<td>0.002673178</td>
<td>18</td>
<td>2.126582</td>
<td>6</td>
<td>7</td>
<td>hydrogen ion transmembrane transporter activity</td>
</tr>
</tbody>
</table>
```

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

Analysis of RNA-Seq Data with R/Bioconductor

RNA-Seq Analysis

Aligning Short Reads
Outline

Overview

RNA-Seq Analysis
  Aligning Short Reads

Viewing Results in IGV Genome Browser
Inspect Results in IGV

View results in IGV

- Download and open IGV
- 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/Workshop_Dec_6_10_2012/Rrnaseq/results/SRR064154.fastq.bam
  - http://faculty.ucr.edu/~tgirke/HTML_Presentations/Manuals/Workshop_Dec_6_10_2012/Rrnaseq/results/SRR064155.fastq.bam
  - http://faculty.ucr.edu/~tgirke/HTML_Presentations/Manuals/Workshop_Dec_6_10_2012/Rrnaseq/results/SRR064166.fastq.bam
  - http://faculty.ucr.edu/~tgirke/HTML_Presentations/Manuals/Workshop_Dec_6_10_2012/Rrnaseq/results/SRR064167.fastq.bam

- To view area of interest, enter its coordinates Chr1:49,457-51,457 in position menu on top.
Number of reads overlapping gene ranges

```r
> source("data/Fct/gffexonDEXSeq.R")
> gffexonDEXSeq <- exons2DEXSeq(gff=gff)
> ids <- as.character(elementMetadata(gffexonDEXSeq)[, "ids"])
> countDFdex <- data.frame(row.names=ids)
> for(i in samplespath) {
+     aligns <- readBamGappedAlignments(i) # Substitute next two lines with this one.
+     counts <- countOverlaps(gffexonDEXSeq, aligns)
+     countDFdex <- cbind(countDFdex, counts)
+ } 
> colnames(countDFdex) <- samples
> countDFdex[1:4,1:2]
```

<table>
<thead>
<tr>
<th>Parent=AT1G01010:E001__Chr1_3631_3913_+</th>
<th>Parent=AT1G01010.1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parent=AT1G01010:E002__Chr1_3996_4276_+</td>
<td>Parent=AT1G01010.1</td>
</tr>
<tr>
<td>Parent=AT1G01010:E003__Chr1_4486_4605_+</td>
<td>Parent=AT1G01010.1</td>
</tr>
<tr>
<td>Parent=AT1G01010:E004__Chr1_4706_5095_+</td>
<td>Parent=AT1G01010.1</td>
</tr>
</tbody>
</table>

```r
> write.table(countDFdex, ".results/countDFdex", quote=FALSE, sep="\t", col.names = NA)
> countDFdex <- read.table("./results/countDFdex")
```
Identify genes with differential exon usage

```r
library(DEXSeq)
samples <- as.character(targets$Factor); names(samples) <- targets$Fastq
countDFdex[is.na(countDFdex)] <- 0
## Construct ExonCountSet from scratch
exset <- newExonCountSet2(countDF=countDFdex) # fData(exset)[1:4,]
## Performs normalization
exset <- estimateSizeFactors(exset)
## Evaluate variance of the data by estimating dispersion using Cox-Reid (CR) likelihood estimation
exset <- estimateDispersions(exset)
## Fits dispersion-mean relation to the individual CR dispersion values
exset <- fitDispersionFunction(exset)
## Performs Chi-squared test on each exon and Benjmini-Hochberg p-value adjustment for multiple testing
exset <- testForDEU(exset)
## Estimates fold changes of exons
exset <- estimateLog2FoldChanges(exset)
## Obtain results in data frame
deuDF <- DEUresultTable(exset)
## Count number of genes with differential exon usage
table(tapply(deuDF$padjust < 0.01, geneIDs(exset), any))
```

<table>
<thead>
<tr>
<th>FALSE</th>
<th>TRUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>1</td>
</tr>
</tbody>
</table>
Sample plot showing fitted expression of exons

```r
plotDEXSeq(exset, "Parent=AT1G01100", displayTranscripts=TRUE, expression=TRUE, legend=TRUE)
```

```r
## Generate many plots and write them to results directory
mygeneIDs <- unique(as.character(na.omit(deuDF[deuDF$geneID %in% unique(deuDF$geneID)],)[,"geneID"]))
DEXSeqHTML(exset, geneIDs=mygeneIDs, path="results", file="DEU.html")
```

Parent=AT1G01100 –
> sessionInfo()

R version 2.15.1 (2012-06-22)
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] DEXSeq_1.4.0 xtable_1.7-0 ath1121501.db_2.8.0 org.At.tair.db_2.8.0 GO.db_2.8.0
[11] AnnotationDbi_1.20.1 edgeR_3.0.0 limma_3.14.1 DESeq_1.10.1 lattice_0.20-10
[21] rtracklayer_1.18.0 GenomicRanges_1.10.2 IRanges_1.16.2 BiocGenerics_0.4.0

loaded via a namespace (and not attached):
[1] AnnotationForge_1.0.2 BSgenome_1.26.1 GSEABase_1.20.0 RBGL_1.34.0 RColorBrewer
[10] bitops_1.0-4.1 gee_4.13-18 genefilter_1.40.0 geneplotter_1.36.0 grid_2.15.1
[19] splines_2.15.1 statmod_1.4.16 stats4_2.15.1 stringr_0.6.1 survival_2.36