9. More Bioconductor packages

Thomas Lumley
Ken Rice

Universities of Washington and Auckland

São Paulo, January 2014
GWAS analysis

Genome-Wide Association Studies (GWAS) are currently popular – typically, these genotype e.g. 1M SNPs on several thousand subjects in (large) established studies

- Usually on 1000’s of subjects
- ‘Simple’ $t$-tests, regressions, for each SNP (like microarrays)
- 1M anything takes a long time! (up to 72 hours)
- Just loading big datasets is non-trivial – but some tools are available
GWAS analysis

snpStats is a Bioconductor package for GWAS analysis – maintained by David Clayton (analysis lead on Wellcome Trust)

```r
> biocLite("snpStats") #in a new session
> library(snpStats)
> data(for.exercise)
> ls()
[1] "snp.support" "snps.10" "subject.support"
```

A ‘little’ case-control dataset (Chr 10) based on HapMap – three objects; snp.support, subject.support and snps.10
GWAS analysis

> summary(snp.support)
  chromosome position     A1     A2
  Min. :10       Min. : 101955 A:14019   C: 2349
  1st Qu.:10       1st Qu.:28981867 C:12166   G:12254
  Median :10       Median : 67409719 G: 2316   T:13898
  Mean :10         Mean : 66874497
  3rd Qu.:10       3rd Qu.:101966491
  Max. :10         Max. :135323432

> summary(subject.support)
  cc       stratum
  Min. :0.0   CEU : 494
  1st Qu.:0.0 JPT+CHB:506
  Median :0.5
  Mean :0.5
  3rd Qu.:1.0
  Max. :1.0
GWAS analysis

> show(snps.10) # show() is generic
A SnpMatrix with 1000 rows and 28501 columns
Row names: jpt.869 ... ceu.464
Col names: rs7909677 ... rs12218790
> summary(snps.10)
$rows
   Call.rate Heterozygosity
  Min.  :0.9879 Min.  :0.0000
  Median:0.9900 Median:0.3078
  Mean  :0.9900 Mean  :0.3074
  Max.  :0.9919 Max.  :0.3386
$cols
  Calls Call.rate  MAF      P.AA
  Min. : 975 Min. :0.975 Min. :0.0000 Min. :0.00000
  Median: 990 Median:0.990 Median:0.2315 Median:0.26876
  Mean : 990 Mean :0.990 Mean :0.2424 Mean :0.34617
  Max. :1000 Max. :1.000 Max. :0.5000 Max. :1.00000
     P.AB P.BB       z.HWE
  Min. :0.0000 Min. :0.00000 Min. : -21.9725
  Median:0.3198 Median:0.27492 Median: -1.1910
  Mean :0.3074 Mean :0.34647 Mean : -1.8610
  Max. :0.5504 Max. :1.00000 Max. :  3.7085
  NA’s :     4.0000
GWAS analysis

- 28501 SNPs, all with Allele 1, Allele 2

- 1000 subjects, 500 controls (cc=0) and 500 cases (cc=1)

- Far too much data for a regular `summary()` of `snps.10` — even in this small example
GWAS analysis

We’ll use just the column summaries, and a (mildly) ‘clean’ subset;

```r
> snpsum <- col.summary(snps.10)
> use <- with(snpsum, MAF > 0.01 & z.HWE^2 < 200)

> table(use)
use
FALSE  TRUE
  317 28184
```
GWAS analysis

Now do single-SNP tests for each SNP, and extract the $p$-value for each SNP, along with its location;

```r
tests <- single.snp.tests(cc, data = subject.support, + snp.data = snps.10)
pos.use <- snp.support$position[use]
p.use <- p.value(tests, df=1)[use]
```

We’d usually give a table of ‘top hits,’ but...
GWAS analysis

`plot(hexbin(pos.use, -log10(p.use), xbin = 50))`
GWAS analysis

qq.chisq(chi.squared(tests, df=1)[use], df=1)

QQ plot

Expected distribution: chi-squared (1 df)
GWAS analysis

tests2 <- single.snp.tests(cc, stratum, data = subject.support, + snp.data = snps.10)
qq.chisq(chi.squared(tests2, 1)[use], 1)
Handling short-read sequences

The `ShortRead` package handles short-read (aka ‘next-generation’) sequencing reads, mostly for QC and preprocessing.

It comes with a (small) subset of a Solexa sequencing run: we will look at this example.

The data are in a structured set of folders, so the first step is to specify where they can be found.

```
exptPath <- "/Users/tlumley/Library/R/2.15/library/ShortRead/extdata"
```

and then use the `SolexaPath()` function to create a simple summary of the structure, which we will pass to other functions instead of a filename.
Handling short-read sequences

> sp <- SolexaPath(exptPath)
> sp
class: SolexaPath
experimentPath: /Users/tlumley/Library/R/2.15/library/ShortRead/extdata
dataPath: Data
scanPath: NA
imageAnalysisPath: C1-36Firecrest
baseCallPath: Bustard
analysisPath: GERALD
> imageAnalysisPath(sp)
[1] "/Users/tlumley/Library/R/2.15/library/ShortRead/extdata/
   Data/C1-36Firecrest"
> analysisPath(sp)
[1] "/Users/tlumley/Library/R/2.15/library/ShortRead/extdata/
   Data/ C1-36Firecrest/Bustard/GERALD"
The function `readAligned()` reads in the aligned sequence fragments,

```r
> aln <- readAligned(sp, "s_2_export.txt")
> aln
class: AlignedRead
length: 1000 reads; width: 35 cycles
chromosome: NM NM ... chr5.fa 29:255:255
position: NA NA ... 71805980 NA
strand: NA NA ... + NA
alignQuality: NumericQuality
alignData varLabels: run lane ... filtering contig
```
Examining data

The `sread()` function returns the bases read

```r
> sread(aln)
A DNAStringSet instance of length 1000
   width  seq
[1] 35 CCAGAGCCCCCCGCTCACTCCTGAACCAGTCTCTC
[2] 35 AGCCTCCCTCTTTCTGAATATACGGCAGAGCTGTT
[3] 35 ACCAAAAACACCACATACACGAGCAACACGTAC
[4] 35 AATCGGAAGAGCTCGTATGCCGGCTTCTGCTTGGA
[5] 35 AAAGATAAAACTCTAGCCACCTCCTCCTTTCTCTA
...
```

and the `quality()` function gives the read quality, coded with Z as the best
Examining data

> quality(aln)
class: SFastqQuality

quality:
A BStringSet instance of length 1000

        width   seq
[1]     35  YQMIMIMMLMMIGIGMFICMFFFIMMHIIHAAGAH
[2]     35  ZZSUYYXYZQYYXUYZYYZZZZZZZIMFHXQSUPPO
[3]     35  LGDHLILLLLLLLIGFLLALDFDILHFIAECAE
[4]     35  JJJYYIYVSYYYYYYYYSDYYWVUYNNVSVQQELQ
[5]     35  LLLIILIIDLLHLLLLLLLLLLLLLLLLLL
[6]     35  YYYYYYYYWWVGGMGUHQHMUFMICDMCDHQHEDDD
[7]     35  ZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZ
[8]     35  ZZZZZZZZZZZZUZZZZZZZZZZZZZZZYZZZZZUHU
[9]     35  ZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZUH

Note how the quality goes down later in the read.
Examining data

We can read other information, such as the fluorescence intensities for each base

```r
> int <- readIntensities(sp)
> int
class: SolexaIntensity
dim: 256 4 36
readInfo: SolexaIntensityInfo
intensity: ArrayIntensity
measurementError: ArrayIntensity
> splom(intensity(int)[[,,2]], pch=".", cex=3)
> splom(intensity(int)[[,,35]], pch=".", cex=3)
```

and then create a scatterplot matrix of the intensities for the 2nd and 35th bases. The plots show the correlation between A-T and C-G channels, and the deterioration towards the end of the read.
Intensities: 2nd position

Scatter Plot Matrix
Intensities: 35th position

Scatter Plot Matrix

A 3000
4000
5000
3000
5000
0
1000
2000
0 1000
C 4000
6000
8000
4000
8000
0
2000
4000
0 2000
G
2000
3000
2000
3000
0
1000
0 1000
T
10000
15000
10000
0
5000
0 5000
DIfferential expression by RNAseq

The edgeR package (among others) estimates differential RNA expression from RNAseq experiments. It’s a sequel to the limma package for microarray gene expression data.

RNAseq produces a count for each transcript, rather than the continuous measure produced by microarray experiments, and the statistical analysis relies on models for variation in counts.

We will look at an RNAseq experiment comparing gene expression in prostate cancer cells with and without treatment with an androgen (a testosterone analogue).

The experiment had three treated samples and four control samples, sequenced in seven of the lanes of a single flow cell on an Illumina 1G sequencer.

The data have been mapped to the human genome and turned into counts of transcripts in a simple text file.
RNAseq of prostate cancer

\> x <- read.delim("pnas_expression.txt", row.names=1, 
        stringsAsFactors=FALSE)

\> head(x)

<table>
<thead>
<tr>
<th></th>
<th>lane1</th>
<th>lane2</th>
<th>lane3</th>
<th>lane4</th>
<th>lane5</th>
<th>lane6</th>
<th>lane8</th>
<th>len</th>
</tr>
</thead>
<tbody>
<tr>
<td>ENSG00000215696</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>330</td>
</tr>
<tr>
<td>ENSG00000215700</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2370</td>
</tr>
<tr>
<td>ENSG00000215699</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1842</td>
</tr>
<tr>
<td>ENSG00000215784</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2393</td>
</tr>
<tr>
<td>ENSG00000212914</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>384</td>
</tr>
<tr>
<td>ENSG00000212042</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>92</td>
</tr>
</tbody>
</table>

The `stringsAsFactors` argument is needed because we want to keep the transcript names as strings, not turn them into factors.
**RNAseq of prostate cancer**

We also need to specify which 'lanes' are treated and which are control.

```
> targets

<table>
<thead>
<tr>
<th>Lane</th>
<th>Treatment</th>
<th>Label</th>
</tr>
</thead>
<tbody>
<tr>
<td>Con1</td>
<td>1</td>
<td>Control Con1</td>
</tr>
<tr>
<td>Con2</td>
<td>2</td>
<td>Control Con2</td>
</tr>
<tr>
<td>Con3</td>
<td>2</td>
<td>Control Con3</td>
</tr>
<tr>
<td>Con4</td>
<td>4</td>
<td>Control Con4</td>
</tr>
<tr>
<td>DHT1</td>
<td>5</td>
<td>DHT DHT1</td>
</tr>
<tr>
<td>DHT2</td>
<td>6</td>
<td>DHT DHT2</td>
</tr>
<tr>
<td>DHT3</td>
<td>8</td>
<td>DHT DHT3</td>
</tr>
</tbody>
</table>
```
**RNAseq of prostate cancer**

Putting the two together and filtering out low-expression transcripts gives data for analysis

```r
y <- DGEList(counts=x[,1:7], group=targets$Treatment, genes=data.frame(Length=x[,8]))
colnames(y) <- targets$Label
keep <- rowSums(cpm(y)>1) >= 3
y <- y[keep,]
y$samples$lib.size <- colSums(y$counts)
```

Here, `cpm()` means counts per million
Overall differences?

`plotMDS(y)` does multidimensional scaling, projecting the data into two dimensions to see how well separated the samples are. In this case there are big treated/control differences and smaller between-replicate differences.
Differential expression (at last)

The statistical analysis has two steps. First, the variability is estimated, then the treatment and control groups are compared for each gene, using a test related to Fisher’s exact test.

\[
y <- \text{estimateCommonDisp}(y, \text{verbose=TRUE})
y <- \text{estimateTagwiseDisp}(y)
\]

\[
et <- \text{exactTest}(y)
top <- \text{topTags}(et)
\]
Differential expression (at last)

The output gives the gene name the log fold-change, the overall abundance, the \( p \)-value and FDR.

<table>
<thead>
<tr>
<th>Comparison of groups: DHT-Control</th>
<th>Length</th>
<th>logFC</th>
<th>logCPM</th>
<th>PValue</th>
<th>FDR</th>
</tr>
</thead>
<tbody>
<tr>
<td>ENSG00000151503</td>
<td>5605</td>
<td>5.704455</td>
<td>9.651026</td>
<td>0.000000e+00</td>
<td>0.000000e+00</td>
</tr>
<tr>
<td>ENSG00000096060</td>
<td>4093</td>
<td>4.881110</td>
<td>9.886097</td>
<td>1.065531e-315</td>
<td>8.787436e-312</td>
</tr>
<tr>
<td>ENSG00000166451</td>
<td>1556</td>
<td>4.539306</td>
<td>8.781549</td>
<td>1.168385e-217</td>
<td>6.423783e-214</td>
</tr>
<tr>
<td>ENSG00000127954</td>
<td>3919</td>
<td>8.054706</td>
<td>7.152971</td>
<td>3.838688e-200</td>
<td>1.582883e-196</td>
</tr>
<tr>
<td>ENSG00000162772</td>
<td>1377</td>
<td>3.204621</td>
<td>9.689874</td>
<td>8.819449e-170</td>
<td>2.909360e-166</td>
</tr>
<tr>
<td>ENSG00000113594</td>
<td>10078</td>
<td>3.960785</td>
<td>7.986306</td>
<td>1.108187e-144</td>
<td>3.046406e-141</td>
</tr>
<tr>
<td>ENSG00000116133</td>
<td>4286</td>
<td>3.143633</td>
<td>8.740383</td>
<td>1.291088e-138</td>
<td>3.042172e-135</td>
</tr>
<tr>
<td>ENSG00000115648</td>
<td>2920</td>
<td>2.513583</td>
<td>11.429414</td>
<td>6.775660e-130</td>
<td>1.396972e-126</td>
</tr>
<tr>
<td>ENSG00000123983</td>
<td>4305</td>
<td>3.473311</td>
<td>8.534717</td>
<td>9.250437e-129</td>
<td>1.695297e-125</td>
</tr>
<tr>
<td>ENSG00000116285</td>
<td>3076</td>
<td>4.104133</td>
<td>7.306976</td>
<td>1.758623e-128</td>
<td>2.900673e-125</td>
</tr>
</tbody>
</table>
Differential expression (at last)

We could use biomaRt or one of the other annotation packages to convert to other gene names

```r
> getBM(attributes="hgnc_symbol",filters="ensembl_gene_id",
       values=rownames(top$table),mart=human)

  hgnc_symbol
1     FKBP5
2      LIFR
3     MLPH
4    DHCR24
5   ERRFI1
6    ACSL3
7    STEAP4
8  NCAPD3
9     ATF3
10   CENPN
```