RNA-seq Analysis 2:

Annotations and Alignments

Pietà Schofield

Contents

Annotations

There are many packages in bioconductor that contain curated annotation data for numerous and diverse organisms. See the list on the BiocViews AnnotationData page

Annotation Packages

There are annotation datasets that are available as Bioconductor packages many of them use the `RSQLite` package to interface to `sqlite` database files that hold the information locally. Bioconductor provides some standard packages for a range of annotation databases. Examples of these are:

- BSGenome packages store the full genomic sequence in Biostrings format for various organisms.
- `AnnotationDbi` packages that store several type of information
  - OrgDb packages store genome wide annotation data in a format for many organisms.
  - ChipD and `probe` annotations that hold information on probe-sets on various micro array platforms

  - InparanoidDb packages that hold homology information on proteins

- Txdb uses the GenomicFeatures package (Lawrence, Huber, Pagès, et al., 2013) to manage database that hold transcript model annotations.
- MeSHDb provide links between genes and the Medical Subject Headings curated database for many organisms

Online Resources

Bioconductor also provides some library packages that are designed to ease searching various on-line databases for annotation data

- BiomaRt Interface to BioMart databases (e.g. Ensembl, COSMIC ,Wormbase and Gramene)
- AnnotationHub The home page for the AnnotationHub package states "The AnnotationHub web resource provides a central location where genomic files (e.g., VCF, bed, wig) and other resources from standard locations (e.g., UCSC, Ensembl) can be discovered"
- GEOQuery This package provides access to the NCBI Gene Expression Omnibus (GEO) is a public repository of micro array data
- SRAdb provides similar access to GEOQuery except to the NCBI Small Reads Archive, the largest public repository of sequencing data from the next generation of sequencing platforms.
- STRINGdb provides an interface to the STRING protein-protein interactions database. It also gives access to KEGG and GO annotations.
- pathview is a tool set for pathway based data integration and visualization and will download pathway information from KEGG using the KEGGgraph package

AnnotationHub Example

We know there is some *Saccharomyces cerevisiae* data coming our way so lets look to see what is available through AnnotationHub and lets look at the data provider ENSEMBL.

```
require(AnnotationHub)
ah <- AnnotationHub()
sc <- query(ah,c("ENSEMBL","Saccharomyces cerevisiae"))
sc
```

```
AnnotationHub with 106 records
# snapshotDate(): 2016-01-25
# $dataprovider: Ensembl, UCSC, ftp://ftp.ncbi.nlm.nih.gov/gene/DATA/
# $species: Saccharomyces cerevisiae
# $rdataclass: FaFile, GRanges, TwoBitFile, OrgDb
# additional mcols(): taxonomyid, genome, description, tags,
#   sourceurl, sourcetype
# retrieve records with, e.g., 'object[["AH289"]]'

            title
  AH289   | Saccharomyces_cerevisiae.EF4.69.cdna.all.fa
  AH290   | Saccharomyces_cerevisiae.EF4.69.dna.toplevel.fa
  AH291   | Saccharomyces_cerevisiae.EF4.69.dna_rm.toplevel.fa
  AH292   | Saccharomyces_cerevisiae.EF4.69.dna_sm.toplevel.fa
  AH293   | Saccharomyces_cerevisiae.EF4.69.ncrna.fa
  ...       ...
  AH50219 | Saccharomyces_cerevisiae.R64-1-1.dna_sm.toplevel.2bit
```

```
  AH50220 | Saccharomyces_cerevisiae.R64-1-1.dna.toplevel.2bit
  AH50221 | Saccharomyces_cerevisiae.R64-1-1.ncrna.2bit
  AH50338 | Saccharomyces_cerevisiae.R64-1-1.82.gtf
  AH50407 | Saccharomyces_cerevisiae.R64-1-1.83.gtf
```

This displays a list of the data items available. A fuller list in the form of a data frame is available with the `mocls()` function, this displays more information including the URL of the on-line data source.

```
mcols(sc)
```

Item `AH49366` is the full genomic sequence for *Saccharomyces cerevisiae* release 81

```
scFaFile <- sc[["AH49366"]]
scFaFile
```

```
class: FaFile
path: /Users/pschofield/.AnnotationHub/56011
index: /Users/pschofield/.AnnotationHub/56012
isOpen: FALSE
yieldSize: NA
```

The sequences in the FaFile object can be retrieved with the `getSeq()` command

```
getSeq(scFaFile)
```

```
  A DNAStringSet instance of length 17
      width seq                                          names
 [1]  230218 CCACACCACACCCACACAC...TGGGTGTGGTGTGTGTGGG I dna:chromosome ...
 [2]  813184 AAATAGCCCTCATGTACGT...GTGTGGTGTGTGGGTGTGT II dna:chromosome...
 [3]  316620 CCCACACACCACACCCACA...GGTGGGTGTGGTGTGTGTG III dna:chromosom...
 [4] 1531933 ACACCACACCCACACCACA...GGTAGTAAGTAGCTTTTGG IV dna:chromosome...
 [5]  439888 CACACACACCACACCCACA...GTGGGTGTGGTGTGTGTGT IX dna:chromosome...
 ...     ... ...
[13] 1078177 CACACACACACACCACCCA...ACGTACATGAGGGCTATTT XII dna:chromosom...
[14]  924431 CCACACACACACCACACCC...GTGTGGTGTGTGTGTGGGG XIII dna:chromoso...
[15]  784333 CCGGCTTTCTGACCGAAAT...GTGTGGGTGTGGTGTGGGT XIV dna:chromosom...
[16] 1091291 ACACCACACCCACACCACA...GTGTGTGGGTGTGGTGTGT XV dna:chromosome...
[17]  948067 AAATAGCCCTCATGTACGT...TTTAATTTCGGTCAGAAAN XVI dna:chromosom...
```

The last item in the list of available *S. cerevisiae* annotation is the GTF file containing the genetic features and we can retrieve this into a `GenomicRanges` object

```
genes <- sc[["AH50407"]]
head(genes)
```

```
GRanges object with 6 ranges and 19 metadata columns:
      seqnames       ranges strand |   source        type     score
         <Rle>    <IRanges>  <Rle> | <factor>    <factor> <numeric>
  [1]       IV [1802, 2953]      + |  ensembl        gene      <NA>
  [2]       IV [1802, 2953]      + |  ensembl  transcript      <NA>
  [3]       IV [1802, 2953]      + |  ensembl        exon      <NA>
```

```
  [4]       IV [1802, 2950]      + |  ensembl         CDS      <NA>
  [5]       IV [1802, 1804]      + |  ensembl start_codon      <NA>
  [6]       IV [2951, 2953]      + |  ensembl  stop_codon      <NA>
         phase     gene_id gene_version   gene_name gene_source
     <integer> <character>  <character> <character> <character>
  [1]      <NA>     YDL248W            1        COS7     ensembl
  [2]      <NA>     YDL248W            1        COS7     ensembl
  [3]      <NA>     YDL248W            1        COS7     ensembl
  [4]         0     YDL248W            1        COS7     ensembl
  [5]         0     YDL248W            1        COS7     ensembl
  [6]         0     YDL248W            1        COS7     ensembl
       gene_biotype transcript_id transcript_version transcript_name
        <character>   <character>        <character>     <character>
  [1] protein_coding          <NA>               <NA>            <NA>
  [2] protein_coding       YDL248W                  1            COS7
  [3] protein_coding       YDL248W                  1            COS7
  [4] protein_coding       YDL248W                  1            COS7
  [5] protein_coding       YDL248W                  1            COS7
  [6] protein_coding       YDL248W                  1            COS7
     transcript_source transcript_biotype exon_number     exon_id
          <character>        <character> <character> <character>
  [1]              <NA>               <NA>        <NA>        <NA>
  [2]           ensembl     protein_coding        <NA>        <NA>
  [3]           ensembl     protein_coding           1   YDL248W.1
  [4]           ensembl     protein_coding           1        <NA>
  [5]           ensembl     protein_coding           1        <NA>
  [6]           ensembl     protein_coding           1        <NA>
     exon_version  protein_id protein_version
      <character> <character>     <character>
  [1]         <NA>        <NA>            <NA>
  [2]         <NA>        <NA>            <NA>
  [3]            1        <NA>            <NA>
  [4]         <NA>     YDL248W               1
  [5]         <NA>        <NA>            <NA>
  [6]         <NA>        <NA>            <NA>
  -------
  seqinfo: 17 sequences (1 circular) from Saccharomyces_cerevisiae.R64-1-1.83.gtf genome; no seqlengths
```

It is possible to retrieve the sequences for a gene of interest using these two annotations

```
# get the GenomicRanges item that has the gene name of interest and is of type gene
ser3 <- genes[which(genes$gene_name=="SER3" & genes$type=="gene"),]
# pass this as a parameter to the getSeq function
seqSer3 <- getSeq(scFaFile,ser3)
names(seqSer3) <- ser3$gene_id
seqSer3
```

```
  A DNAStringSet instance of length 1
    width seq                                               names
[1]  1410 ATGACAAGCATTGACATTAAC...CAATTAGATTGCTATATTAA YER081W
```

Exercise 2.1

- Which chromosome is the SER3 gene on.

- In the example above what other types genetic features are listed for the SER3 gene?
- How many different bio-types are there in the genetic features data and what are they?
- Find the name, length and sequence for the longest snoRNA.

NGS Data Processing

While it is possible to complete all steps of an RNA-seq analysis with tools from bioconductor there are many other alternatives. Web interface tools such as Galaxy provide form driven interfaces to many of the command line tools including several bioconductor tools and might be an easier introduction to NGS data analysis. Commercial tools such as CLC Bio NGS Workbench also exist that if you have access to might also be an attractive alternative. However bioconductor offers great flexibility and a huge depth in downstream analysis of results that there are certainly benefits from learning how to do an RNA-seq analysis with bioconductor even if you choose to use other software subsequently.

QC

I am not necessarily going to recommend doing QC in R since FastQC introduced earlier in the course is a fast and relatively painless why to summaries quality information on FASTQ sequence files. However it can be done and one way the task can be performed with in R using the ShortRead package.

```
require(ShortRead)
filePath <- system.file(package="ShortRead", "extData","E-MTAB-1147")
qualAnalysis <- qa(dirPath,"fastq.gz", BPPARAM=SerialParam())
report(qualAnalysis,dest="myQual",type="html")
system("open myQual/index.html")
```

The package EDASeq (Risso, Schwartz, Sherlock, et al., 2011) also provides tools for the exploration of unaligned (fastq) data and aligned (bam) data.

```
require(EDASeq)
# Make a list of the files of interest
files = list.files(dataDir,pattern=".*bam$",full=T)

# prettify the names
names(files) <- gsub("[.]bam$", "", basename(files))

# make a list of the meta data in this case the sample condition
cond <- gsub("[0-9]$","",names(files))

# create a data frame with the meta data in
meta <- DataFrame(cond,row.names=names(files))

# create a BamFileList object and set the meta data
bfs <- BamFileList(files)
elementMetadata(bfs) <- meta

# Call the EDASeq function to plot the qualities
plotQuality(bfs)

# Plot the perbase quality for a bamfile
# WARNING this function hangs my version of R
```

```
plotQuality(bfs[[1]])

# Plot the total read counts
barplot(bfs)

# Plot the nucleotide frequencies for a sample
plotNtFrequency(bfs[[1]])
```

Exercise 2.2

Using the EDASeq function `plotQuality()` to examine the per cycle base quality for the file in the directory '/usr/local/share/BS32010/pschofield/data/fastq'

Alignment

While you might choose other alignment tools such as TopHat or STAR for speed and flexibility there are aligners available in Bioconductor, chiefly

gmapR an interface to the GMAP/GSNAP/GSTRUCT suite Rsubread which provides an interface onto the subread/subjunc/featureCounts tool set

For simplicities sake and to show the process all with-in the Bioconductor environment this workshop uses the `Rsubread` package.

Build Index

As with all aligners there is a two step process to aligning reads with Rsubread and at least two files are necessary. First an index is needs to be built from a FASTA format genomic sequence reference, aligners tend to have their own index structure so indexes for one aligner will not work for an other in fact often indexes for one version release of an aligner will not work for another version.

There are two options at this point.

- Use the genomic sequence, all the chromosomal DNA sequence to build the index, then at a subsequent step use an annotation file that hold information on the location of genetic features to count the reads that fall in the regions of DNA covered by each feature of interest.
- Use the transcriptome (cDNA) sequence only, so instead of a sequence per chromosome there is a sequence per feature. This way when the alignment is performed the assignment to features is performed in the same step.

Thought Exercise

What are some advantages and disadvantages of the two alignment refernce strategies?

```
require(Rsubread)
ref <- system.file("extdata","reference.fa",package="Rsubread")
# Build an index using the reference fasta file
buildindex(basename="./myorganism_index",reference=ref)
```

Align

Once the reference is built the FASTQ files can be aligned. In two similar ways you can perform the alignment with either the basic `align()` function that calls the subread aligner, or with the `subjunc()` function that calls the more sophisticated `subjunc` aligner that performs alternative splicing junction detection using donor/receptor site sequence detection. For simple gene-wise differential expression `subread()` should be sufficient.

```
reads <- system.file("extdata","reads.txt.gz",package="Rsubread")
align(index="./myorganism_index",readfile1=reads,
      output_file="./Rsubread_alignment.BAM",phredOffset=64)
```

Both `align()` and `subjunc()` have a lot of parameters for fine tuning the alignment process. For example they can perform read trimming, and set limits on numbers of mismatches and fragmemt size. The output from both aligners are a BAM file, a file of insert-deletion locations

Feature Assignment

The feature assignment function in the Rsubread package is the `featureCounts()` function. It takes a list of BAM files and a genomic features annotation file and produces a `featureCounts` object that contains a count matrix and some statistics on the numbers of no assigned reads.

Exercise 2.3

Using the paired end fastq files available in the directory '/usr/local/share/BS32010/pschofield/data/fastq' obtain a featureCounts object containing the reads that align to the genes on chromosomes 5 (V) of the *Saccharomyces cerevisiae* genome. Report the number of assigned and unassigned reads.

You will need to

- Obtain the sequence for the chromosomes and build an index
- Align the reads to the reference
- Obtain the genomic features annotation for the features on those chromosomes
- Assign the aligned reads to the featureRegions

Bibliography

[1] M. Lawrence, W. Huber, H. Pagès, et al. "Software for Computing and Annotating Genomic Ranges". In: PLoS Computational Biology 9 (2013).

[2] D. Risso, K. Schwartz, G. Sherlock, et al. "GC-content normalization for RNA-Seq data." In: BMC Bioinformatics 12 (2011), p. 480.

Session Info

```
R version 3.2.3 (2015-12-10)
Platform: x86_64-apple-darwin13.4.0 (64-bit)
Running under: OS X 10.11.3 (El Capitan)
```

```
locale:
[1] en_GB.UTF-8/en_GB.UTF-8/en_GB.UTF-8/C/en_GB.UTF-8/en_GB.UTF-8

attached base packages:
[1] stats4    parallel  stats     graphics  grDevices utils     datasets
[8] methods   base

other attached packages:
 [1] Rsamtools_1.22.0     Biostrings_2.38.4    XVector_0.10.0
 [4] GenomicRanges_1.22.4 GenomeInfoDb_1.6.3   IRanges_2.4.8
 [7] S4Vectors_0.8.11     AnnotationHub_2.2.3  BiocGenerics_0.16.1
[10] knitr_1.12.3         RefManageR_0.10.6    pietalib_0.1

loaded via a namespace (and not attached):
 [1] Rcpp_0.12.3                 futile.logger_1.4.1
 [3] BiocInstaller_1.20.1        formatR_1.2.1
 [5] plyr_1.8.3                  futile.options_1.0.0
 [7] bitops_1.0-6                tools_3.2.3
 [9] zlibbioc_1.16.0             digest_0.6.9
[11] lubridate_1.5.0             evaluate_0.8
[13] RSQLite_1.0.0               bibtex_0.4.0
[15] shiny_0.13.1                DBI_0.3.1
[17] curl_0.9.6                  yaml_2.1.13
[19] stringr_1.0.0               httr_1.1.0
[21] Biobase_2.30.0              R6_2.1.2
[23] AnnotationDbi_1.32.3        BiocParallel_1.4.3
[25] XML_3.98-1.3                rmarkdown_0.9.5
[27] RJSONIO_1.3-0               lambda.r_1.1.7
[29] magrittr_1.5                htmltools_0.3
[31] mime_0.4                    interactiveDisplayBase_1.8.0
[33] xtable_1.8-2                httpuv_1.3.3
[35] stringi_1.0-1               RCurl_1.95-4.7
```