

# Practical: Annotations

Martin Morgan ([mtmorgan@fhcrc.org](mailto:mtmorgan@fhcrc.org))

3 February 2014

## Contents

---

<b>1 Gene annotation</b>	<b>1</b>
1.1 Data packages	1
1.2 Internet resources	5
<b>2 Genome annotation</b>	<b>6</b>
2.1 Transcript annotation packages	6
2.2 <i>rtracklayer</i>	8

## 1 Gene annotation

---

### 1.1 Data packages

Organism-level ('org') packages contain mappings between a central identifier (e.g., Entrez gene ids) and other identifiers (e.g. GenBank or Uniprot accession number, RefSeq id, etc.). The name of an org package is always of the form `org.<Sp>.<id>.db` (e.g. [org.Hs.eg.db](#)) where `<Sp>` is a 2-letter abbreviation of the organism (e.g. Hs for *Homo sapiens*) and `<id>` is an abbreviation (in lower-case) describing the type of central identifier (e.g. eg for ENTREZ gene identifiers). The "How to use the '.db' annotation packages" vignette in the [AnnotationDbi](#) package (org packages are only one type of ".db" annotation packages) is a key reference. The '.db' and most other *Bioconductor* annotation packages are updated every 6 months.

Annotation packages usually contain an object named after the package itself. These objects are collectively called *AnnotationDb* objects, with more specific classes named *OrgDb*, *ChipDb* or *TranscriptDb* objects. Methods that can be applied to these objects include `cols`, `keys`, `keytypes` and `select`. Common operations for retrieving annotations are summarized in Table~1.

**Exercise 1** *This exercise illustrates basic use of the 'select' interface to annotation packages.*

- What is the name of the org package for Homo sapiens? Load it. Display the *OrgDb* object for the [org.Hs.eg.db](#) package. Use the `keytypes` and `columns` methods to discover which sorts of annotations can be queried and extracted.
- Here are some ENTREZID values.

```
egids <- c("3183", "91828", "81537", "4776", "283624", "4053", "85446",  
          "10484", "55701", "1112")
```

*These are the most strongly differentially expressed genes from a subset of an RNA-seq differential expression analysis that you will encounter later in the course. The biological background is provided in [1]; see the ArrayExpress entry for [E-MTAB-1147](#). The data are from chromosome 14 only*

*Use the ENTREZIDs in the `select` method in such a way that you extract the SYMBOL (gene symbol) and GENENAME information for each. To what extent do the differentially expressed genes make biological sense?*

Table 1: Common operations for retrieving and manipulating annotations.

Category	Function	Description
Discover	columns	List the kinds of columns that can be returned
	keytypes	List columns that can be used as keys
	keys	List values that can be expected for a given keytype
	select	Retrieve annotations matching <code>keys</code> , <code>keytype</code> and <code>columns</code>
Manipulate	setdiff, union, intersect	Operations on sets
	duplicated, unique	Mark or remove duplicates
	%in%, match	Find matches
	any, all	Are any TRUE? Are all?
<i>GRanges*</i>	merge	Combine two different <code>data.frames</code> based on shared keys
	transcripts, exons, cds	Features (transcripts, exons, coding sequence) as <i>GRanges</i> .
	transcriptsBy, exonsBy, cdsBy	Features group by gene, transcript, etc., as <i>GRangesList</i> .

**Solution:** The 'org' package for humans (*Homo sapiens*) is [org.Hs.eg.db](#). Load the [org.Hs.eg.db](#) package.

```
library(org.Hs.eg.db)
```

Discover the key types and columns in the annotation package.

```
keytypes(org.Hs.eg.db)
```

```
## [1] "ENTREZID" "PFAM" "IPI" "PROSITE" "ACCNUM"
## [6] "ALIAS" "CHR" "CHRLOC" "CHRLOCEND" "ENZYME"
## [11] "MAP" "PATH" "PMID" "REFSEQ" "SYMBOL"
## [16] "UNIGENE" "ENSEMBL" "ENSEMBLPROT" "ENSEMBLTRANS" "GENENAME"
## [21] "UNIPROT" "GO" "EVIDENCE" "ONTOLOGY" "GOALL"
## [26] "EVIDENCEALL" "ONTOLOGYALL" "OMIM" "UCSCKG"
```

```
columns(org.Hs.eg.db)
```

```
## [1] "ENTREZID" "PFAM" "IPI" "PROSITE" "ACCNUM"
## [6] "ALIAS" "CHR" "CHRLOC" "CHRLOCEND" "ENZYME"
## [11] "MAP" "PATH" "PMID" "REFSEQ" "SYMBOL"
## [16] "UNIGENE" "ENSEMBL" "ENSEMBLPROT" "ENSEMBLTRANS" "GENENAME"
## [21] "UNIPROT" "GO" "EVIDENCE" "ONTOLOGY" "GOALL"
## [26] "EVIDENCEALL" "ONTOLOGYALL" "OMIM" "UCSCKG"
```

```
cols <- c("SYMBOL", "GENENAME")
```

```
select(org.Hs.eg.db, keys=egids, columns=cols, keytype="ENTREZID")
```

```
## ENTREZID SYMBOL
## 1 3183 HNRNPC
## 2 91828 EXOC3L4
## 3 81537 SGPP1
## 4 4776 NFATC4
## 5 283624 LINC00641
## 6 4053 LTBP2
## 7 85446 ZFH2
## 8 10484 SEC23A
## 9 55701 ARHGEF40
## 10 1112 FOXN3
##
## GENENAME
## 1 heterogeneous nuclear ribonucleoprotein C (C1/C2)
## 2 exocyst complex component 3-like 4
```

```
## 3 sphingosine-1-phosphate phosphatase 1
## 4 nuclear factor of activated T-cells, cytoplasmic, calcineurin-dependent 4
## 5 long intergenic non-protein coding RNA 641
## 6 latent transforming growth factor beta binding protein 2
## 7 zinc finger homeobox 2
## 8 Sec23 homolog A (S. cerevisiae)
## 9 Rho guanine nucleotide exchange factor (GEF) 40
## 10 forkhead box N3
```

**Exercise 2** This exercise annotates a larger selection of the differential expression results, merging the statistics of differential expression with annotation.

- Read the comma-separate value file `E-MTAB-1147-toptable.csv` in to R using `read.csv`; include the argument `row.names=1` to name the rows of the data (these are ENTREZIDs). Perform basic R operations to discover the object dimensions, and to view the head (first 6 rows) of the data frame. The columns of this data frame are statistics of differential representation, to be discussed later in the course.
- Use `select` to map all ENTREZIDs (`rownames`) to their `CHR` (chromosome), `SYMBOL` and `GENENAME`. Verify that in fact all genes are located on chromosome 14.
- Use `merge` to add the `SYMBOL` and `GENENAME` annotations to the differential expression statistics.
- Print out the 10 rows of the merged data frame with highest absolute log fold change. To do this, you'll need to (a) take the absolute value of the log fold change column; (b) determine the order, decreasing from largest to smallest, of the absolute values; (c) select the rows of the merged data frame in decreasing order of fold change; and (d) select, using `head`, just the first 10 rows of the ordered data frame.

### Solution:

- Find the file `E-MTAB-1147-toptable.csv` of differentially expressed genes.

```
f1 <- file.choose()
```

Verify that the file exists!

```
file.exists(f1)
## [1] TRUE
```

Input the file, specifying that the the first column should be used for row names; look at basic properties of the data.

```
csv <- read.csv(f1, row.names=1)
class(csv) # data.frame
## [1] "data.frame"
dim(csv) # 528 genes x 6 columns
## [1] 528 6
head(csv) # log (base 2) fold change, # adjusted P-values
## baseMean log2FoldChange lfcSE stat pvalue padj
## 3183 1432.40 -4.523 0.1426 -31.710 1.114e-220 4.992e-218
## 91828 58.28 -3.478 0.3757 -9.259 2.071e-20 8.434e-19
## 81537 1106.57 -2.758 0.1301 -21.195 1.061e-99 2.377e-97
## 4776 97.46 -2.246 0.2557 -8.785 1.568e-18 4.943e-17
## 283624 29.54 -2.240 0.4058 -5.520 3.394e-08 4.344e-07
## 4053 135.52 -2.114 0.2408 -8.779 1.655e-18 4.943e-17
```

- Select `CHR`, `SYMBOL`, and `GENENAME` annotations for each ENTREZID, and verify that all genes are from `CHR 14`.

```
cols <- c("CHR", "SYMBOL", "GENENAME")
anno <- select(org.Hs.eg.db, rownames(csv), cols)
class(anno)
```

```
## [1] "data.frame"
dim(anno)
## [1] 528 4
head(anno)
##   ENTREZID CHR   SYMBOL
## 1    3183  14   HNRNPC
## 2   91828  14  EXOC3L4
## 3   81537  14   SGPP1
## 4    4776  14   NFATC4
## 5  283624  14 LINC00641
## 6   4053  14   LTBP2
##
##                                     GENENAME
## 1                heterogeneous nuclear ribonucleoprotein C (C1/C2)
## 2                        exocyst complex component 3-like 4
## 3                sphingosine-1-phosphate phosphatase 1
## 4 nuclear factor of activated T-cells, cytoplasmic, calcineurin-dependent 4
## 5                        long intergenic non-protein coding RNA 641
## 6                latent transforming growth factor beta binding protein 2
all(anno$CHR %in% "14")      # all on CHR 14?
## [1] TRUE
```

- c. Merge the differential expression and annotation data frames, specifying that rows are to be matched based on row names in the first data frame (by .x=0) and by the ENTREZID in the second data frame (by .y="ENTREZID"). Verify that the result is as expected using standard *R* commands.

```
annotated <- merge(csv, anno, by.x=0, by.y="ENTREZID")
class(annotated)
## [1] "data.frame"
dim(annotated)
## [1] 528 10
head(annotated)
##   Row.names baseMean log2FoldChange lfcSE  stat  pvalue  padj CHR  SYMBOL
## 1    10001  141.222      0.4709 0.1891  2.490 1.276e-02 2.962e-02 14  MED6
## 2 100128927 236.118     -0.4525 0.1706 -2.652 8.007e-03 2.062e-02 14  ZBTB42
## 3 100129075  31.483      0.3801 0.3206  1.186 2.357e-01 3.418e-01 14  KTN1-AS1
## 4 100129794  21.091     -0.5833 0.4009 -1.455 1.457e-01 2.282e-01 14  SLC25A21-AS1
## 5 100288846   6.135     -1.3658 0.6248 -2.186 2.880e-02 5.865e-02 14  LOC100288846
## 6 100289511  87.824     -1.5688 0.2793 -5.617 1.948e-08 2.645e-07 14  LOC100289511
##
##                                     GENENAME
## 1                mediator complex subunit 6
## 2 zinc finger and BTB domain containing 42
## 3                KTN1 antisense RNA 1
## 4                SLC25A21 antisense RNA 1
## 5                uncharacterized LOC100288846
## 6                uncharacterized LOC100289511
```

- d. Re-organize the annotated data frame by (a) taking the absolute value of the log fold change column and (b) determine the order, decreasing from largest to smallest, of the absolute values of the fold change. Here is a 'one-liner'; explain it to your neighbor. What is o?

```
o <- order(abs(annotated$log2FoldChange), decreasing=TRUE)
```

Finally, order the annotated data frame and view the 10 genes with largest differential expression.

```
annotated[head(o),]
##   Row.names baseMean log2FoldChange lfcSE  stat  pvalue  padj CHR  SYMBOL
## 216    3183  1432.40      -4.523 0.1426 -31.710 1.114e-220 4.992e-218 14  HNRNPC
```



```

useMart("ensembl", dataset = "hsapiens_gene_ensembl")

head(listFilters(ensembl), 3)           ## filters
myFilter <- "chromosome_name"
head(filterOptions(myFilter, ensembl), 3) ## return values
myValues <- c("21", "22")
head(listAttributes(ensembl), 3)       ## attributes
myAttributes <- c("ensembl_gene_id", "chromosome_name")

## assemble and query the mart
res <- getBM(attributes = myAttributes, filters = myFilter,
             values = myValues, mart = ensembl)

```

Use `head(res)` to see the results.

**Exercise 4** As an optional exercise, annotate the genes that are differentially expressed in the DESeq2 laboratory, e.g., find the *GENENAME* associated with the five most differentially expressed genes. Do these make biological sense? Can you merge the annotation results with the ‘top table’ results to provide a statistically and biologically informative summary?

**Using PSICQUIC** PSICQUIC is a really useful effort to provide programmatic access to molecular interaction data bases. The *PSICQUIC* package provides an *R* / *Bioconductor* interface to PSICQUIC.

1. Follow instructions on the PSICQUIC package landing page<sup>1</sup> to install the package.
2. Work through section 2 ‘Quick Start’ of the PSICQUIC vignette<sup>2</sup>, discovering documented interactions between Myc and TP53.
3. If interested, explore more of the PSICQUIC vignette. Save yourself typing by using the R script from the package landing page.

## 2 Genome annotation

---

There are a diversity of packages and classes available for representing large genomes. Several include:

***TxDb***.\* For transcript and other genome / coordinate annotation.

***BSgenome*** For whole-genome representation. See `available.packages` for pre-packaged genomes, and the vignette ‘How to forge a BSgenome data package’ in the

***Homo.sapiens*** For integrating *TxDb*\* and *org*.\* packages.

***SNPlocs***.\* For model organism SNP locations derived from dbSNP.

**FaFile** (*Rsamtools*) for accessing indexed FASTA files.

***SIFT***.\*, ***PolyPhen***, ***ensemblVEP*** Variant effect scores.

### 2.1 Transcript annotation packages

Genome-centric packages are very useful for annotations involving genomic coordinates. It is straight-forward, for instance, to discover the coordinates of coding sequences in regions of interest, and from these retrieve corresponding DNA or protein coding sequences. Other examples of the types of operations that are easy to perform with genome-centric annotations include defining regions of interest for counting aligned reads in RNA-seq

<sup>1</sup><http://bioconductor.org/packages/release/bioc/html/PSICQUIC.html>

<sup>2</sup><http://bioconductor.org/packages/release/bioc/vignettes/PSICQUIC/inst/doc/PSICQUIC.pdf>

experiments and retrieving DNA sequences underlying regions of interest in ChIP-seq analysis, e.g., for motif characterization.

**Exercise 5** *This exercise uses annotation packages to go from gene identifiers to coding sequences.*

- Map from an informal gene SYMBOL, e.g., BRCA1, to ENTREZID gene identifiers using the [org.Hs.eg.db](#) package and the `select` function, use the [TxDb.Hsapiens.UCSC.hg19.knownGene](#) package and a second map to go from ENTREZID to TXNAME.
- Extract the coding sequence grouped by transcript using the [TxDb.Hsapiens.UCSC.hg19.knownGene](#) package and `cdsBy` function; select just those transcripts we are interested in.
- Retrieve the nucleotide sequence from the [BSgenome.Hsapiens.UCSC.hg19](#) package using the function `extractTranscriptsFromGenome`.
- Verify that the coding sequences are all multiples of 3, and translate from nucleotide to amino acid sequence.

**Solution:** Map from gene SYMBOL to ENTREZID, and from ENTREZID to TXNAME

```
library(org.Hs.eg.db)
egid <- select(org.Hs.eg.db, "BRCA1", "ENTREZID", "SYMBOL")$ENTREZID
library(TxDb.Hsapiens.UCSC.hg19.knownGene)
txdb <- TxDb.Hsapiens.UCSC.hg19.knownGene
egToTx <- select(txdb, egid, "TXNAME", "GENEID")

## Warning: 'select' resulted in 1:many mapping between keys and return rows
```

Extract the relevant coding sequence, grouped by transcript

```
cdsByTx <- cdsBy(txdb, "tx", use.names=TRUE)[egToTx$TXNAME]
```

Retrieve the sequence

```
library(BSgenome.Hsapiens.UCSC.hg19)
txx <- extractTranscriptsFromGenome(Hsapiens, cdsByTx)
```

Translate to amino acid sequence

```
all(width(txx) %% 3 == 0) # sanity check

## [1] TRUE

translate(txx) # amino acid sequence

## A AAStringSet instance of length 20
## width seq
## [1] 760 MDLSALRVEEVQNVINAMQKILECPICLELIKEPVSTK...MCEAPVVTREWVLD SVALYQCQELDTYLIPQIPHSHY*
## [2] 1793 MSLQESTRFSQLVEELKIIICAFQLDTGLEYSYNFA...MCEAPVVTREWVLD SVALYQCQELDTYLIPQIPHSHY*
## [3] 174 MDAEFVCERTLKYFLGIAGGKVVSYFWVTQSIKERKM...MCEAPVVTREWVLD SVALYQCQELDTYLIPQIPHSHY*
## [4] 700 MDLSALRVEEVQNVINAMQKILECPICLELIKEPVSTK...CCYGPFTNMPTGCPPNCGCAARCLDRGQWLPCNWADV*
## [5] 1817 MLKLLNQKKGPSQCPLCKNDITKRSQESTRFSQLVEE...MCEAPVVTREWVLD SVALYQCQELDTYLIPQIPHSHY*
## ... ..
## [16] 1365 MDLSALRVEEVQNVINAMQKILECPICLELIKEPVSTK...SESGVGLSDKELVSDDEERTGLEENNQEEQSMDSNL
## [17] 1365 MDLSALRVEEVQNVINAMQKILECPICLELIKEPVSTK...SESGVGLSDKELVSDDEERTGLEENNQEEQSMDSNL
## [18] 1318 MLKLLNQKKGPSQCPLCKNDITKRSQESTRFSQLVEE...SESGVGLSDKELVSDDEERTGLEENNQEEQSMDSNL
## [19] 1339 MDLSALRVEEVQNVINAMQKILECPICLELIKEPVSTK...SESGVGLSDKELVSDDEERTGLEENNQEEQSMDSNL
## [20] 1069 MNVEKAEFCKSKQPGLARSQLHNRWAGSKETCNDRTP...SESGVGLSDKELVSDDEERTGLEENNQEEQSMDSNL
```

## 2.2 rtracklayer

The `rtracklayer` package allows us to query the UCSC genome browser, as well as providing `import` and `export` functions for common annotation file formats like GFF, GTF, and BED.

### Exercise 6 *warning: This exercise requires INTERNET ACCESS*

Here we use `rtracklayer` to retrieve estrogen receptor binding sites identified across cell lines in the ENCODE project. We focus on binding sites in the vicinity of a particularly interesting region of interest.

- Define our region of interest by creating a `GRanges` instance with appropriate genomic coordinates. Our region corresponds to 10Mb up- and down-stream of a particular gene.
- Create a session for the UCSC genome browser
- Query the UCSC genome browser for ENCODE estrogen receptor `ERalphaa` transcription marks; identifying the appropriate track, table, and transcription factor requires biological knowledge and detective work.
- Visualize the location of the binding sites and their scores; annotate the mid-point of the region of interest.

**Solution:** Define the region of interest

```
roi <- GRanges("chr10", IRanges(92106877, 112106876, names="ENSG00000099194"))
```

Create a session

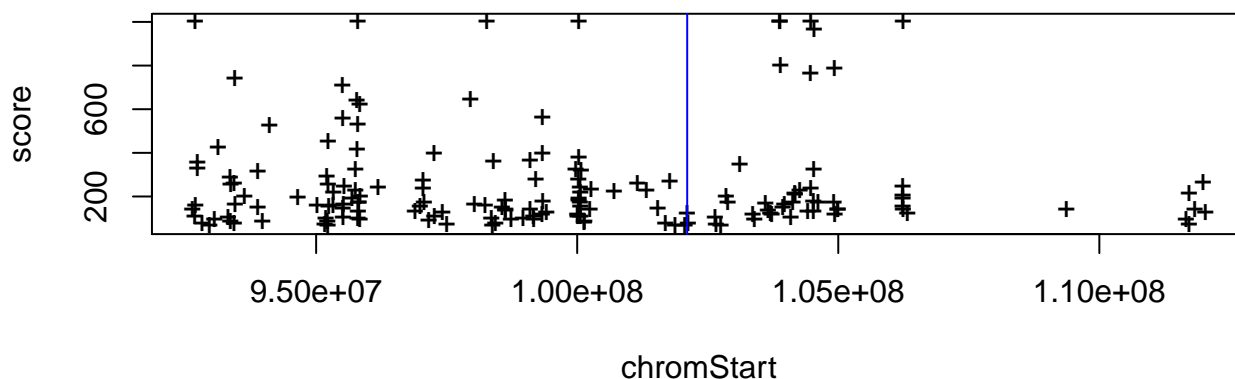
```
library(rtracklayer)
session <- browserSession()
```

Query the UCSC for a particular track, table, and transcription factor, in our region of interest

```
trackName <- "wgEncodeRegTfbsClusteredV2"
tableName <- "wgEncodeRegTfbsClusteredV2"
trFactor <- "ERalpha_a"
ucscTable <- getTable(ucscTableQuery(session, track=trackName,
  range=roi, table=tableName, name=trFactor))
```

Visualize the result

```
plot(score ~ chromStart, ucscTable, pch="+")
abline(v=start(roi) + (end(roi) - start(roi) + 1) / 2, col="blue")
```





## References

---

- [1] K. Zarnack, J. König, M. Tajnik, I. Martincorena, S. Eustermann, I. Stevant, A. Reyes, S. Anders, N. M. Luscombe, and J. Ule. Direct competition between hnRNP C and U2AF65 protects the transcriptome from the exonization of Alu elements. *Cell*, 152(3):453–466, Jan 2013.