Gene and Genome Annotation Resources in Bioconductor

Martin Morgan

2012-07-05 Thu

Contents

1 Introduction: Gene and genome annotations 1

2 Gene-centric discovery & selection 2

3 Genomic discovery & selection 3

4 biomaRt 4

5 Under the hood 5

6 Resources 5

1 Introduction: Gene and genome annotations

626 ‘Annotation’ packages

1.1 Gene-centric packages

- Organism: org.Dm.eg.db
- Platform: ...
- Homology: hom.Dm.imp.db
- System biology: GO.db, KEGG.db, Reactome.db

1.2 Genome-centric packages

- GenomicFeatures: TxDb.Dmelanogaster.UCSC.dm3.ensGene
- BSgenome: BSgenome.Dmelanogaster.UCSC.dm3

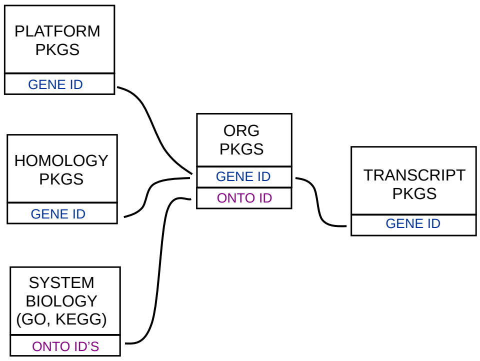

Figure 1: Types of gene-centric annotation packages

1.3 Web-based resources

- biomaRt
- rtracklayer: gff, bed, wig, etc. See `import`, `export`

2 Gene-centric discovery & selection

2.1 Context: DESeq 'top table'

```
## library( "DESeq" )
## ...
## cds = newCountDataSet( countTable, condition )
## cds = estimateSizeFactors( cds )
## cds = estimateDispersions( cds )
## res = nbinomTest( cds, "untreated", "treated" )
## topTable = res[ order(res$pval), ]
## save(topTable,
##      file="2012-07-05-pasilla-DESeq-topTable.rda")
load("2012-07-05-pasilla-DESeq-topTable.rda")
head(topTable, 3)
```

2.2 Discover & select

Discover

- data base reference, org.Dm.eg.db
- functions: keytypes, cols, keys

```
library(org.Dm.eg.db)
org.Dm.eg.db
keytypes(org.Dm.eg.db)  # types of keys to query with
cols(org.Dm.eg.db)      # available columns for results
head(keys(org.Dm.eg.db, keytype="FLYBASE"))
```

Select – arguments database, keys, cols, keytype

```
fbids <- topTable$id[1:3]
cols <- c("ENTREZID", "SYMBOL")
anno <- select(org.Dm.eg.db, fbids, cols, "FLYBASE")
anno
## some maps are 1:many
select(org.Dm.eg.db, fbids, "GO", "FLYBASE")
```

Between packages

```
fbids <- topTable$id[1]
anno1 <- select(org.Dm.eg.db, fbids, "GO", "FLYBASE")
anno1
## ?org.Dm.egGO

library(GO.db)
keytypes(GO.db)
cols(GO.db)
goAnno <- select(GO.db, anno1$GO[1], "TERM")
goAnno
```

2.3 Merging top table and annotation

Merge

```
fbids <- topTable$id
anno <- select(org.Dm.eg.db, fbids, cols, "FLYBASE")
topTableAnno <- merge(topTable, anno,
                      by.x="id", by.y="FLYBASE",
                      all.x=TRUE)
head( topTableAnno[ order(topTableAnno$padj), ], 3 )
```

3 Genomic discovery & selection

3.1 Discover and select transcripts

```
library(TxDb.Dmelanogaster.UCSC.dm3.ensGene)
TxDb.Dmelanogaster.UCSC.dm3.ensGene
txdb <- TxDb.Dmelanogaster.UCSC.dm3.ensGene
keytypes(txdb)
cols(txdb)

fbids <- topTable$id[1:3]
txnm <- select(txdb, fbids, "TXNAME", "GENEID")
```

3.2 Exons, transcripts, genes

Coding sequence ranges of differentially expressed gene transcripts?

```
cds0 <- cdsBy(txdb, "tx", use.names=TRUE)
cds <- cds0[ txnm$TXNAME ]
cds[[1]]
```

also: exonsByOverlaps, transcriptsByOverlaps, cdsByOverlaps

3.3 Sequences

```
library(BSgenome.Dmelanogaster.UCSC.dm3)
Dmelanogaster
txx <- extractTranscriptsFromGenome(Dmelanogaster, cds)
translate(txx)
```

4 biomaRt

4.1 Discovery and query

```
library(biomaRt)
head(listMarts(), 3)              # list the marts, 63 total
##                biomart                           version
## 1              ensembl      ENSEMBL GENES 67 (SANGER UK)
## 2                  snp  ENSEMBL VARIATION 67 (SANGER UK)
## 3 functional_genomics ENSEMBL REGULATION 67 (SANGER UK)
head(listDatasets(useMart("ensembl")), 3)
                                  # mart datasets, 58 total
##                   dataset
## 1  oanatinus_gene_ensembl
## 2   tguttata_gene_ensembl
## 3 cporcellus_gene_ensembl
##                               description     version
## 1  Ornithorhynchus anatinus genes (OANA5)       OANA5
## 2 Taeniopygia guttata genes (taeGut3.2.4) taeGut3.2.4
## 3         Cavia porcellus genes (cavPor3)     cavPor3
ensembl <-                        # fully specified mart
    useMart("ensembl", dataset = "hsapiens_gene_ensembl")
head(listFilters(ensembl), 3)     # filters, 333 total
##              name     description
## 1 chromosome_name Chromosome name
## 2           start Gene Start (bp)
## 3             end   Gene End (bp)
myFilter <- "chromosome_name"
filterOptions(myFilter, ensembl)  # possible values
myValues <- c("21", "22")
head(listAttributes(ensembl), 3)  # attributes, 1647 total
##                     name           description
## 1        ensembl_gene_id       Ensembl Gene ID
## 2 ensembl_transcript_id Ensembl Transcript ID
## 3     ensembl_peptide_id    Ensembl Protein ID
myAttributes <- c("ensembl_gene_id","chromosome_name")
```

```
## assemble and query the mart
res <- getBM(attributes =  myAttributes,
             filters =  myFilter, values =  myValues,
             mart = ensembl)
head(res, 3)
##   ensembl_gene_id chromosome_name
## 1 ENSG00000241945              21
## 2 ENSG00000248354              21
## 3 ENSG00000160221              21
```

5 Under the hood

sqlite data bases

- queried directly from R
- used by other software
- e.g., TranscriptDb objects are reference classes with ‘conn’ fields

```
dbListTables(txdb$conn)
dbListFields(txdb$conn, "chrominfo")
sql <- "SELECT chrom, length FROM chrominfo"
dbGetQuery(txdb$conn, sql)     # similar to seqinfo(txdb)
```

6 Resources

```
vignette(package="AnnotationDbi", "IntroToAnnotationPackages")
vignette(package="GenomicFeatures", "GenomicFeatures")
vignette(package="biomaRt", "biomaRt")
```