Annotation and Data Integration

Martin Morgan[1]
Fred Hutchinson Cancer Research Center
Seattle, WA

18 November 2013

[1] mtmorgan@fhcrc.org

Outline

What is 'Annotation'?

- Genes – classification schemes (e.g., Entrez, Ensembl), pathway membership, ...
- Genomes – reference genomes; exons, transcripts, coding sequence; coding consequences
- System / network biology – pathways, biochemical reactions, ...

Other defintions (not covered here): assigning function to novel sequence assemblies, ...

Bioconductor Annotation Resources – Packages

Model organism annotation packages

- *org.** – gene names and pathways
- *TxDb.** – gene models
- *BSgenome.** – whole-genome sequences

*org.** packages

The 'select' interface:

- Discovery: `keytypes`, `columns`, `keys`
- Retrieval: `select`

```
> library(org.Hs.eg.db)
> keytypes(org.Hs.eg.db)
> columns(org.Hs.eg.db)
> egid <-
+     select(org.Hs.eg.db, "BRCA1", "ENTREZID", "SYMBOL")
```

*org.** packages – Useful *R* commands

Within-*vector* or *data.frame*

- Finding and removing duplicates: `duplicated`, `unique`
- `any`, `all`

Between-*vector* or *data.frame*

- Matching `%in%`, `match`
- Set operations: `setdiff`, `union`, `intersect`
- `merge` Join two *data.frame*s based on shared column.

*org.** pacakges – Under the hood...

SQL (sqlite) data bases

- `org.Hs.eg_dbconn()` to query using *RSQLite* package
- `org.Hs.eg_dbfile()` to discover location and query outside *R*.

*TxDb.** packages

- Gene models for common model organsisms / genome builds / known gene schemes
- Supports the 'select' interface (`keytypes`, `columns`, `keys`, `select`)
- 'Easy' to build custom packages when gene model exist

Retrieving genomic ranges

- `transcripts`, `exons`, `cds`,
- `transcriptsBy` , `exonsBy`, `cdsBy` – group by gene, transcirpt, etc.

```
> library(TxDb.Hsapiens.UCSC.hg19.knownGene)
> txdb <- TxDb.Hsapiens.UCSC.hg19.knownGene
> cdsByTx <- cdsBy(txdb, "tx")
```

*BSgenome.** packages

Whole-genome sequences

- 'Masks' when available, e.g., repeat regions
- Load chromosomes, range-based queries: `getSeq`, `extactTranscriptsFromGenome`

```
> library(BSgenome.Hsapiens.UCSC.hg19)
> library(GenomicFeatures)
> dna <- extractTranscriptsFromGenome(Hsapiens, cdsByTx)
```

Bioconductor Annotation Resources – Web-based

Rich web resources

- *biomaRt* (`http://biomart.org`), *rtracklayer* (UCSC genome browser)
- *ArrayExpress*, *GEOquery*, BiocpkgSRAdb
- *PSICQUIC*, *KEGGREST*, *uniprot.ws*, ...
- *AnnotationHub*

biomaRt

- http://biomart.org
- Drill-down discovery: `listMarts`, `listDatasets`, `listFilters`, `listAttributes`
- Retrieval: `getBM`

```
> library(biomaRt)
> ensembl <-                              ## discover & use
+    useMart("ensembl", dataset="hsapiens_gene_ensembl")
> head(listFilters(ensembl), 3)
> myFilter <- "chromosome_name"
> myValues <- c("21", "22")
> myAttributes <- c("ensembl_gene_id","chromosome_name")
> res <- getBM(attributes=myAttributes, filters=myFilter,
+                values=myValues, mart=ensembl)
```

AnnotationHub

- Large-scale genome resources, lightly curated for easy access from *R*.
- Supports tab-completion, `metadata` discovery, selection and filtering.

```
> library(AnnotationHub)
> hub <- AnnotationHub()
> hub           ## 9150 resources
```

Outline

Advantages of integrated data containers

We could separately define a features×samples *matrix* of expression values, a *data.frame* describing samples, and a *GRanges* object describing the ranges of interest, but. . .

- Difficult and error prone to manipulate, e.g., subset, in a coordinated fashion.
- Different pacakges might follow different conventions for representing data, e.g., samples×features representation of expression values.

Instead. . .

- Create a class that integrates different data types
- Re-use established classes as much as possible

SummarizedExperiment

- `assays`: feature×sample matricies
- `colData`: *DataFrame* of sample attributes
- `rowData`: *GRanges* / *GRangesList* of features
- Coordination between assays, colData and rowData

```
> library(GenomicRanges)
> ?SummarizedExperiment
> example(SummarizedExperiment)
> sset
> dim(assays(sset)[[1]])
> colData(sset)
> rowData(sset)
```

SummarizedExperiment – manipulation

- Use `$` to access colData
- Use range-based operations, e.g., `%over%` (does the left-hand side overlap the right-hand side?) for row-based queries

```
> sset$Treatment
> sset[, sset$Treatment == "ChIP"]
> roi <- GRanges("chr1", IRanges(1, 249250621))
> sset[sset %over% roi, ]
```

Outline

Conclusions

Rich annotation resources

- Model organism and custom *org.**, *TxDb.**, *BSgenome.** packages
- Web-based access to public (e.g., *biomaRt* and *Bioconductor*-specific (e.g., *AnnotationHub*) resources

Flexible integrated data containers

- Less error-prone
- Convenient
- Interoperability between packages