

# Annotation

Martin Morgan ([mtmorgan@fhcrc.org](mailto:mtmorgan@fhcrc.org))  
Fred Hutchinson Cancer Research Center  
Seattle, WA, USA

July 15, 2014

# 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, ...
- ▶ 'Consortium' resources, TCGA, ENCODE, dbSNP, GTEx, ...

Other definitions (not covered here)

- ▶ SNP (and similar) consequences (*VariantAnnotation*, *VariantFiltering*, *ensemblVEP*)
- ▶ Assign function to novel sequences
- ▶ ...

# 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 – 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*.

# Background: Genomic Ranges

- ▶ Defined by chromosome, start, end, strand
  - ▶ *Bioconductor*: 1-based, closed interval
  - ▶ *GRanges*: Vector of genomic ranges
  - ▶ *GRangesList*: List, each element of which is a genomic range
- ▶ Describe data
  - ▶ *GRanges*: SNP locations, ungapped read alignments, ChIP peaks, copy number changes, ...
  - ▶ *GRangesList*: gapped or paired-end alignments, ...
- ▶ Describe annotations
  - ▶ *GRanges*: genes, exons, ...
  - ▶ *GRangesList*: transcripts, ...

# Genomic Ranges: GRanges

```
> gr = exons(TxDb.Hsapiens.UCSC.hg19.knownGene); gr
```

```
GRanges with 289969 ranges and 1 metadata column:
```

|          | seqnames | ranges               | strand | exon_id   |
|----------|----------|----------------------|--------|-----------|
|          | <Rle>    | <IRanges>            | <Rle>  | <integer> |
| [1]      | chr1     | [11874, 12227]       | +      | 1         |
| [2]      | chr1     | [12595, 12721]       | +      | 2         |
| [3]      | chr1     | [12613, 12721]       | +      | 3         |
| ...      | ...      | ...                  | ...    | ...       |
| [289967] | chrY     | [59358329, 59359508] | -      | 277748    |
| [289968] | chrY     | [59360007, 59360115] | -      | 277749    |
| [289969] | chrY     | [59360501, 59360854] | -      | 277750    |

```
---  
seqlengths:
```

| chr1      | chr2 ...      | chrUn_g1000249 |
|-----------|---------------|----------------|
| 249250621 | 243199373 ... | 38502          |

*GRanges*

```
length(gr); gr[1:5]  
seqnames(gr)  
start(gr)  
end(gr)  
width(gr)  
strand(gr)
```

*DataFrame*

```
mcols(gr)  
gr$exon_id
```

*Seqinfo*

```
seqlevels(gr)  
seqlengths(gr)  
genome(gr)
```

# Genomic Ranges: GRangesList

```
> gr1 = exonsBy(TxDb.Hsapiens.UCSC.hg19.knownGene, "tx", use.names=TRUE); gr1
```

```
GRangesList of length 82960:
```

```
$uc001aaa.3
```

```
GRanges with 3 ranges and 3 metadata columns:
```

|     | seqnames | ranges         | strand | exon_id   | exon_name   | exon_rank |
|-----|----------|----------------|--------|-----------|-------------|-----------|
|     |          | <IRanges>      | <Rle>  | <integer> | <character> | <integer> |
| [1] | chr1     | [11874, 12227] | +      | 1         | <NA>        | 1         |
| [2] | chr1     | [12613, 12721] | +      | 3         | <NA>        | 2         |
| [3] | chr1     | [13221, 14409] | +      | 5         | <NA>        | 3         |

```
GRangesList  
(list of GRanges)  
length(gr1)  
gr1[1:3]  
shift(gr1, 1)  
range(gr1)
```

```
$uc010nxq.1
```

```
GRanges with 3 ranges and 3 metadata columns:
```

|     | seqnames | ranges         | strand | exon_id | exon_name | exon_rank |
|-----|----------|----------------|--------|---------|-----------|-----------|
| [1] | chr1     | [11874, 12227] | +      | 1       | <NA>      | 1         |
| [2] | chr1     | [12595, 12721] | +      | 2       | <NA>      | 2         |
| [3] | chr1     | [13403, 14409] | +      | 6       | <NA>      | 3         |

```
GRanges  
gr1[[2]]  
gr1[["uc010nxq.1"]]
```

```
$uc010nxr.1
```

```
GRanges with 3 ranges and 3 metadata columns:
```

|     | seqnames | ranges         | strand | exon_id | exon_name | exon_rank |
|-----|----------|----------------|--------|---------|-----------|-----------|
| [1] | chr1     | [11874, 12227] | +      | 1       | <NA>      | 1         |
| [2] | chr1     | [12646, 12697] | +      | 4       | <NA>      | 2         |
| [3] | chr1     | [13221, 14409] | +      | 5       | <NA>      | 3         |

Two kinds of fun!

```
introns =  
psetdiff(range(gr1), gr1)
```

```
grr = unlist(gr1)  
## transform grr, then...  
gr1 = relist(grr, gr1)
```

'flesh'

'skeleton'

```
...  
<82957 more elements>
```

```
---
```

```
seqlengths:
```

|           |               |                |
|-----------|---------------|----------------|
| chr1      | chr2 ...      | chrUn_gl000249 |
| 249250621 | 243199373 ... | 38502          |



## Genomic Ranges: Range-Based Operations

- ▶ Within range: “I have a *GRangesList* instance *exByTx* of exons within transcripts. They use a 0-based, 1/2-open convention. I want them 1-based and closed.”

```
resize(shift(exByTx, 1), width(exByTx) - 1)
```

- ▶ Between ranges within instance: “I have a *GRanges* instance *reads* representing aligned reads. I want coverage.”

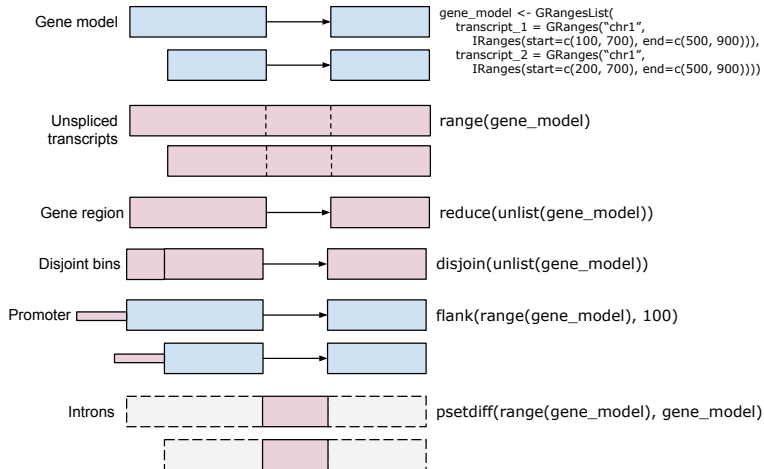
```
coverage(reads)
```

- ▶ Between instances: “How many reads overlap each gene?”

```
countOverlaps(exByTx, reads)
```

(Better: `GenomicAlignments::summarizeOverlaps` on the underlying BAM files)

# Genomic Ranges: Range-Based Operations



## TxDb.\* packages

- ▶ Gene models for common model organisms / 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, transcript, etc.

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

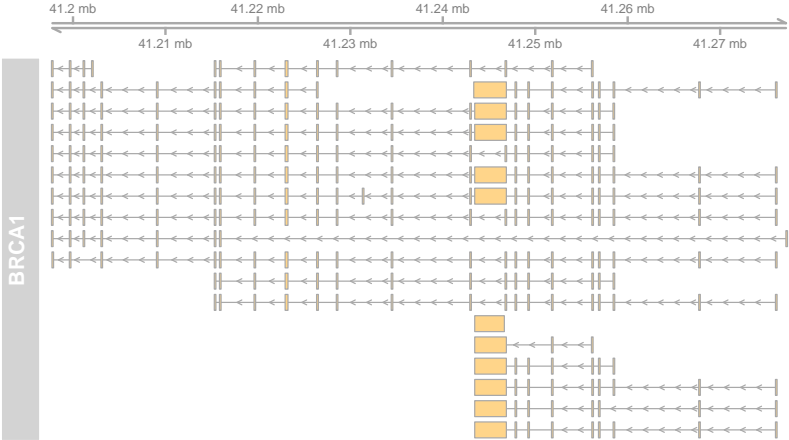
## Example: Visualize BRCA1 Transcripts

```
library(org.Hs.eg.db)
eid <- select(org.Hs.eg.db, "BRCA1", "ENTREZID",
  "SYMBOL") [["ENTREZID"]]

library(TxDb.Hsapiens.UCSC.hg19.knownGene)
txdb <- TxDb.Hsapiens.UCSC.hg19.knownGene
txid <- select(txdb, eid, "TXNAME", "GENEID") [["TXNAME"]]
cds <- cdsBy(txdb, by="tx", use.names=TRUE)
brca1cds <- cds[names(cds) %in% txid]

library(Gviz)
tx <- rep(names(brca1cds), elementLengths(brca1cds))
id <- unlist(brca1cds)$cds_id
grt <- GeneRegionTrack(brca1cds, name="BRCA1",
  gene="BRCA1", feature=tx, transcript=tx, id=tx, exon=id)
plotTracks(list(GenomeAxisTrack(), grt))
```

# Example: Visualize BRCA1 Transcripts



## BSgenome.\* Packages: Whole-Genome Sequences

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

```
library(BSgenome.Hsapiens.UCSC.hg19)
extractTranscriptSeqs(Hsapiens, brca1cds)

##      A DNASTringSet instance of length 20
##      width seq                                names
## [1]  2280 ATGGATTTATCTG...AGCCACTACTGA uc010whl.2
## [2]  5379 ATGAGCCTACAAG...AGCCACTACTGA uc002icp.4
## [3]   522 ATGGATGCTGAGT...AGCCACTACTGA uc010whm.2
## ...    ...    ...
## [18] 3954 ATGCTGAAACTTC...GATTCAAACCTTA uc010cyz.2
## [19] 4017 ATGGATTTATCTG...GATTCAAACCTTA uc010cza.2
## [20] 3207 ATGAATGTAGAAA...GATTCAAACCTTA uc010wht.1
```

# Bioconductor Annotation Resources – Web-based

## Rich web resources

- ▶ *biomaRt* (<http://biomart.org>), *rtracklayer* (UCSC genome browser)
- ▶ *ArrayExpress*, *GEOquery*, *SRADB*
- ▶ *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)
```



# PSICQUIC

- ▶ Proteomics **S**tandard **I**nitiative **C**ommon **Q**Uery **I**nterfa**C**e
- ▶ Programmatic access to molecular interaction data bases.
- ▶ <https://code.google.com/p/psicquic/>

```
library(PSICQUIC)
## Query web service for available providers
psicquic <- PSICQUIC()
providers(psicquic)           # 25 available providers
## interactions between TP53 and MYC
tbl <-
  interactions(psicquic, c("TP53", "MYC"), "9606")
nrow(tbl)                     # 7 interactions
```

See the package vignette for additional detail.

## 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      ## 10511 resources
```

## AnnotationHub: Example

- ▶ Evoln'arily conserved enhancer SNPs near genes on chr17

### Resources

- ▶ SNPs from dbGAP
- ▶ Enhancers from ENCODE ChromHMM
- ▶ Conservation track, from UCSC

### Steps

1. Retrieve enhancers, SNPs from *AnnotationHub*, gene coordinates from *TxDb.\**; harmonize chromosome and genome names
2. Download (large!) conservation track as BED file from UCSC, query for chr17 using *rtracklayer*
3. `subsetByOverlaps` SNPs and enhancers
4. Annotate enhancer SNPs with evolutionary conservation score
5. Find `nearest` and `distanceToNearest` genes to each SNP

# 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

## Facile manipulation of genomic ranges

- ▶ Many data munging and research questions very easy to answer
- ▶ Integrative analysis across data types


# Resources

## Additional resources

- ▶ Annotation, VariantAnnotation and other work flows
- ▶ AnnotationDbi, AnnotationHub and other package landing pages, including links to vignettes.
- ▶ Previous course material, including an Annotation walk-through from *useR!* 2014.

Bioc2014 Annual Conference<sup>1</sup>, July 30 – August 1, Boston

---

<sup>1</sup><https://register.bioconductor.org/BioC2014/> 

# Acknowledgements

- ▶ The *Bioconductor* team, Sonali Arora, Marc Carlson, Nate Hayden, Valerie Obenchain, Hervè Pagés, Paul Shannon, Dan Tennenbaum
- ▶ NIH / NHGRI U41HG004059; NSF 1247813.
- ▶ And of course the *Bioconductor* community!