Annotation resources - ensembldb

Johannes Rainer (Eurac Research, Italy)[1]

June 12, 2017 @CSAMA2017

[1]email: johannes.rainer@eurac.edu, github/twitter: jotsetung

Annotation of genomic regions

- Annotations for genomic features (genes, transcripts, exons) are provided by `TxDb` (`GenomicFeatures`) and `EnsDb` (`ensembldb`) databases.
- `EnsDb`:
  - Designed for Ensembl-based annotations.
  - One database per species and Ensembl release.
- Extract data using methods: `genes`, `transcripts`, `exons`, `txBy`, `exonsBy`, ...
- Results are returned as `GRanges` or `GRangesList` objects.
- Parameter `columns` to specify which additional attributes to return.

Annotation of genomic regions

- Example: get all gene annotations from an EnsDb:

```
## Load the database for human genes, Ensembl release 86.
library(EnsDb.Hsapiens.v86)
edb <- EnsDb.Hsapiens.v86
## Get all genes from the database.
gns <- genes(edb)
gns
```

```
GRanges object with 63970 ranges and 6 metadata columns:
  seqnames               ranges strand |         gene_id
     <Rle>            <IRanges>  <Rle> |     <character>
  ENSG00000223972        1       [11869, 14409]      + | ENSG00000223972
  ENSG00000227232        1       [14404, 29570]      - | ENSG00000227232
  ENSG00000278267        1       [17369, 17436]      - | ENSG00000278267
      ...      ...                   ...    ... .            ...
  ENSG00000237917        Y [26594851, 26634652]      - | ENSG00000237917
  ENSG00000231514        Y [26626520, 26627159]      - | ENSG00000231514
  ENSG00000235857        Y [56855244, 56855488]      + | ENSG00000235857
    gene_name                        gene_biotype
  <character>                         <character>
  ENSG00000223972     DDX11L1 transcribed_unprocessed_pseudogene
  ENSG00000227232      WASH7P             unprocessed_pseudogene
  ENSG00000278267   MIR6859-1                             miRNA
      ...         ...                                    ...
  ENSG00000237917     PARP4P1             unprocessed_pseudogene
  ENSG00000231514     FAM58CP               processed_pseudogene
  ENSG00000235857     CTBP2P1               processed_pseudogene
  seq_coord_system      symbol                          entrezid
       <character> <character>                            <list>
```

Annotation of genomic regions

- Example: get all gene annotations from an EnsDb (continued):

```
## Access start/end coordinates
head(start(gns))
head(end(gns))
```

```
[1] 11869 14404 17369 29554 34554 52473
[1] 14409 29570 17436 31109 36081 53312
```

```
## chromosome name
head(seqnames(gns))
```

```
factor-Rle of length 6 with 1 run
  Lengths: 6
  Values : 1
Levels(357): 1 10 11 12 13 14 15 16 ... LRG_311 LRG_721 LRG_741 LRG_93 MT X Y
```

```
## Metadata columns; gene name, gene biotype
head(gns$gene_name)
```

```
[1] "DDX11L1"   "WASH7P"    "MIR6859-1" "MIR1302-2" "FAM138A"   "OR4G4P"
```

```
head(gns$gene_biotype)
```

```
[1] "transcribed_unprocessed_pseudogene" "unprocessed_pseudogene"
[3] "miRNA"                              "lincRNA"
[5] "lincRNA"                            "unprocessed_pseudogene"
```

AnnotationFilter: basic classes for filtering annotation resources

- Extracting the full data not always required: filter databases.
- AnnotationFilter provides basic classes and concepts for filtering.
- One filter class for each annotation type/database attribute.
- Filter properties:
  - value: the *value* of the filter (e.g. *"BCL2"*).
  - condition: the filter condition (e.g. ==).
  - field: the default database table attribute (e.g. *"gene_id"*).

AnnotationFilter: basic classes for filtering annotation resources

- Filter categories:
  - CharacterFilter: e.g. SymbolFilter, GeneIdFilter.
    - condition: *"=="*, *"!="*, *"startsWith"*, *"endsWith"*, *"contains"*.
  - IntegerFilter: e.g. GenestartFilter.
    - condition: *"=="*, *"!="*, *">"*, *">="*, *"<"*, *"<="*.
  - GRangesFilter.
- Filter classes can be created with constructor functions or using *filter expressions* written as formulas.

AnnotationFilter: basic classes for filtering annotation resources

- Example: create filters

```
## Create filter using the constructor function
gnf <- GenenameFilter("BCL2", condition = "!=")
gnf
```

```
class: GenenameFilter
condition: !=
value: BCL2
```

```
## Create using a filter expression
gnf <- AnnotationFilter(~ genename != "BCL2")
gnf
```

```
class: GenenameFilter
condition: !=
value: BCL2
```

AnnotationFilter: basic classes for filtering annotation resources

- Example: create filters (continued)

```
## Combine filters
afl <- AnnotationFilterList(GenenameFilter("BCL2"),
                            TxBiotypeFilter("protein_coding"))
afl
```

```
class: AnnotationFilterList
length: 2
filters:

class: GenenameFilter
condition: ==
value: BCL2

 &

class: TxBiotypeFilter
condition: ==
value: protein_coding
```

Filtering EnsDb databases

- Pass filter(s) to `EnsDb` methods with the `filter` parameter.
- Example: get all transcripts for the gene *BCL2*.

```
transcripts(edb, filter = ~ genename == "BCL2")
```

```
GRanges object with 4 ranges and 7 metadata columns:
  seqnames               ranges strand |           tx_id
     <Rle>            <IRanges>  <Rle> |     <character>
  ENST00000398117       18 [63123346, 63320128]      - | ENST00000398117
  ENST00000333681       18 [63127035, 63319786]      - | ENST00000333681
  ENST00000590515       18 [63128212, 63161869]      - | ENST00000590515
  ENST00000589955       18 [63313802, 63318812]      - | ENST00000589955
    tx_biotype tx_cds_seq_start tx_cds_seq_end
   <character>        <integer>      <integer>
  ENST00000398117       protein_coding         63128625       63318666
  ENST00000333681       protein_coding         63128625       63318666
  ENST00000590515 processed_transcript             <NA>           <NA>
  ENST00000589955       protein_coding         63318049       63318666
  gene_id         tx_name   gene_name
      <character>     <character> <character>
  ENST00000398117 ENSG00000171791 ENST00000398117        BCL2
  ENST00000333681 ENSG00000171791 ENST00000333681        BCL2
  ENST00000590515 ENSG00000171791 ENST00000590515        BCL2
  ENST00000589955 ENSG00000171791 ENST00000589955        BCL2
  -------
  seqinfo: 1 sequence from GRCh38 genome
```

Filtering EnsDb databases

- Example: get all transcripts for the gene *BCL2* (continued)

```
## Combine filters: only protein coding tx for the gene
transcripts(edb, filter = ~ genename == "BCL2" &
                     tx_biotype == "protein_coding")
```

```
GRanges object with 3 ranges and 7 metadata columns:
  seqnames              ranges strand |           tx_id
     <Rle>           <IRanges>  <Rle> |     <character>
  ENST00000398117       18 [63123346, 63320128]      - | ENST00000398117
  ENST00000333681       18 [63127035, 63319786]      - | ENST00000333681
  ENST00000589955       18 [63313802, 63318812]      - | ENST00000589955
      tx_biotype tx_cds_seq_start tx_cds_seq_end
     <character>       <integer>      <integer>
  ENST00000398117 protein_coding         63128625       63318666
  ENST00000333681 protein_coding         63128625       63318666
  ENST00000589955 protein_coding         63318049       63318666
  gene_id         tx_name   gene_name
      <character>     <character> <character>
  ENST00000398117 ENSG00000171791 ENST00000398117        BCL2
  ENST00000333681 ENSG00000171791 ENST00000333681        BCL2
  ENST00000589955 ENSG00000171791 ENST00000589955        BCL2
  -------
  seqinfo: 1 sequence from GRCh38 genome
```

- Filters speed up queries.

Getting annotation resources

- Dedicated packages:
  - `TxDb.Hsapiens.UCSC.hg38.knownGene`: UCSC based.
  - `EnsDb.Hsapiens.v86`: based on Ensembl (version 86).
- `AnnotationHub`:
  - Central repository for annotation objects.
  - Downloaded resources cached locally.
  - Use `query` to search for entries, fetch them using `[[`.

Getting annotation resources

- Example: query `AnnotationHub` for available resources:

```
library(AnnotationHub)
ah <- AnnotationHub()
## List available EnsDb objects
query(ah, "EnsDb")
```

```
snapshotDate(): 2017-06-08
AnnotationHub with 136 records
# snapshotDate(): 2017-06-08
# $dataprovider: Ensembl
# $species: Ailuropoda Melanoleuca, Anas Platyrhynchos, Anolis Carolinensis,...
# $rdataclass: EnsDb
# additional mcols(): taxonomyid, genome, description,
#   coordinate_1_based, maintainer, rdatadateadded, preparerclass, tags,
#   rdatapath, sourceurl, sourcetype
# retrieve records with, e.g., 'object[["AH53185"]]'

    title
  AH53185 | Ensembl 87 EnsDb for Anolis Carolinensis
  AH53186 | Ensembl 87 EnsDb for Ailuropoda Melanoleuca
  AH53187 | Ensembl 87 EnsDb for Astyanax Mexicanus
  ...       ...
  AH53754 | Ensembl 88 EnsDb for Vicugna Pacos
  AH53755 | Ensembl 88 EnsDb for Xiphophorus Maculatus
  AH53756 | Ensembl 88 EnsDb for Xenopus Tropicalis
```

Getting annotation resources

- Example: query AnnotationHub for available resources (continued):

```
## Get one EnsDb database
edb_acor <- query(ah, c("EnsDb", "Anolis Carolinensis", "87"))[[1]]
edb_acor
```

```
loading from cache '/Users/jo//.AnnotationHub/59923'
EnsDb for Ensembl:
|Backend: SQLite
|Db type: EnsDb
|Type of Gene ID: Ensembl Gene ID
|Supporting package: ensembldb
|Db created by: ensembldb package from Bioconductor
|script_version: 0.3.0
|Creation time: Fri May 19 09:10:20 2017
|ensembl_version: 87
|ensembl_host: localhost
|Organism: anolis_carolinensis
|taxonomy_id: 28377
|genome_build: AnoCar2.0
|DBSCHEMAVERSION: 2.0
| No. of genes: 25920.
| No. of transcripts: 27172.
|Protein data available.
```

Getting annotation resources

- Example: query `AnnotationHub` for available resources (continued):

```
genes(edb_acor)
```

```
GRanges object with 25920 ranges and 6 metadata columns:
     seqnames            ranges strand |            gene_id
<Rle>        <IRanges>  <Rle> |        <character>
  ENSACAG00000032885        1 [ 44897,  47358]      - | ENSACAG00000032885
  ENSACAG00000009394        1 [ 77380, 183510]      - | ENSACAG00000009394
  ENSACAG00000030292        1 [222702, 230087]      + | ENSACAG00000030292
 ...      ...              ...    ... .                ...
  ENSACAG00000028244       MT   [14068, 15207]      + | ENSACAG00000028244
  ENSACAG00000028245       MT   [15208, 15276]      + | ENSACAG00000028245
  ENSACAG00000028246       MT   [15281, 15347]      - | ENSACAG00000028246
      gene_name   gene_biotype seq_coord_system      symbol
     <character>    <character>      <character> <character>
  ENSACAG00000032885                   lincRNA       chromosome
  ENSACAG00000009394        JAG2 protein_coding       chromosome        JAG2
  ENSACAG00000030292                   lincRNA       chromosome
 ...         ...              ...               ...         ...
  ENSACAG00000028244        CYTB protein_coding       chromosome        CYTB
  ENSACAG00000028245                   Mt_tRNA       chromosome
  ENSACAG00000028246                   Mt_tRNA       chromosome
      entrezid
<list>
  ENSACAG00000032885        NA
  ENSACAG00000009394 100552963
  ENSACAG00000030292        NA
 ...       ...
  ENSACAG00000028244   6385978
```

ensembldb: protein annotations

- EnsDb contain also protein annotation data:
  - Protein sequence.
  - Annotation to Uniprot ID identifiers.
  - Annotation of all protein domains within the protein sequences.
- To get data: proteins method or pass protein attributes to columns parameter.

ensembldb: protein annotations

- Example: get all proteins for the gene *BCL2*.

```
## Get protein annotations
prts <- proteins(edb, filter = ~ symbol == "BCL2", return.type = "AAStringSet")

## Result is returned as an AAStringSet
prts
```

```
  A AAStringSet instance of length 3
    width seq                                                 names
[1]   239 MAHAGRTGYDNREIVMKYIHYKL...LKTLLSLALVGACITLGAYLGHK ENSP00000381185
[2]   239 MAHAGRTGYDNREIVMKYIHYKL...LKTLLSLALVGACITLGAYLGHK ENSP00000329623
[3]   205 MAHAGRTGYDNREIVMKYIHYKL...RHLHTWIQDNGGWVGALGDVSLG ENSP00000466417
```

```
## Access the metadata columns
mcols(prts)
```

```
DataFrame with 3 rows and 3 columns
            tx_id      protein_id      symbol
      <character>     <character> <character>
1 ENST00000398117 ENSP00000381185        BCL2
2 ENST00000333681 ENSP00000329623        BCL2
3 ENST00000589955 ENSP00000466417        BCL2
```

Map coordinates within proteins to the genome

- `Pbase`: (Laurent Gatto and Sebastian Gibb): provides classes and functions for the analysis of protein sequence data in proteomics experiments.
- The `Proteins` object: container for proteins and peptide ranges within the AA sequence.

Map coordinates within proteins to the genome

- Example: fetch a `Proteins` object for the gene *BCL2* from an `EnsDb`.

```
library(Pbase)
bcl2 <- Proteins(edb, filter = ~ symbol == "BCL2")
bcl2
```

```
S4 class type     : Proteins
Class version     : 0.2
Created           : Fri Jun  9 08:37:13 2017
Number of Proteins: 3
Sequences:
  [1] ENSP00000381185 [2] ENSP00000329623 ... [2] ENSP00000329623 [3] ENSP00000466417
Protein ranges:
  ProteinDomains
```

```
## Amino acid sequence:
aa(bcl2)
```

```
  A AAStringSet instance of length 3
    width seq                                                 names
[1]   239 MAHAGRTGYDNREIVMKYIHYKL...LKTLLSLALVGACITLGAYLGHK ENSP00000381185
[2]   239 MAHAGRTGYDNREIVMKYIHYKL...LKTLLSLALVGACITLGAYLGHK ENSP00000329623
[3]   205 MAHAGRTGYDNREIVMKYIHYKL...RHLHTWIQDNGGWVGALGDVSLG ENSP00000466417
```

Map coordinates within proteins to the genome

- Example: fetch a `Proteins` object for the gene *BCL2* from an `EnsDb` (continued).

```
## Peptide features: the protein domains
pranges(bcl2)[, "ProteinDomains"]
```

```
IRangesList of length 3
$ENSP00000381185
IRanges object with 19 ranges and 3 metadata columns:
      start       end     width |      protein_id protein_domain_source
  <integer> <integer> <integer> |     <character>           <character>
  PS50062        97       197       101 | ENSP00000381185                pfscan
  PS50063        11        30        20 | ENSP00000381185                pfscan
  PS01260        10        30        21 | ENSP00000381185            scanprosite
      ...       ...       ...       ... .                ...                ...
  PR01862       143       171        29 | ENSP00000381185                prints
  PR01862       172       196        25 | ENSP00000381185                prints
  PR01862       130       142        13 | ENSP00000381185                prints
  interpro_accession
 <character>
  PS50062          IPR002475
  PS50063          IPR003093
  PS01260          IPR020731
      ...                ...
  PR01862          IPR026298
  PR01862          IPR026298
  PR01862          IPR026298

...
<2 more elements>
```

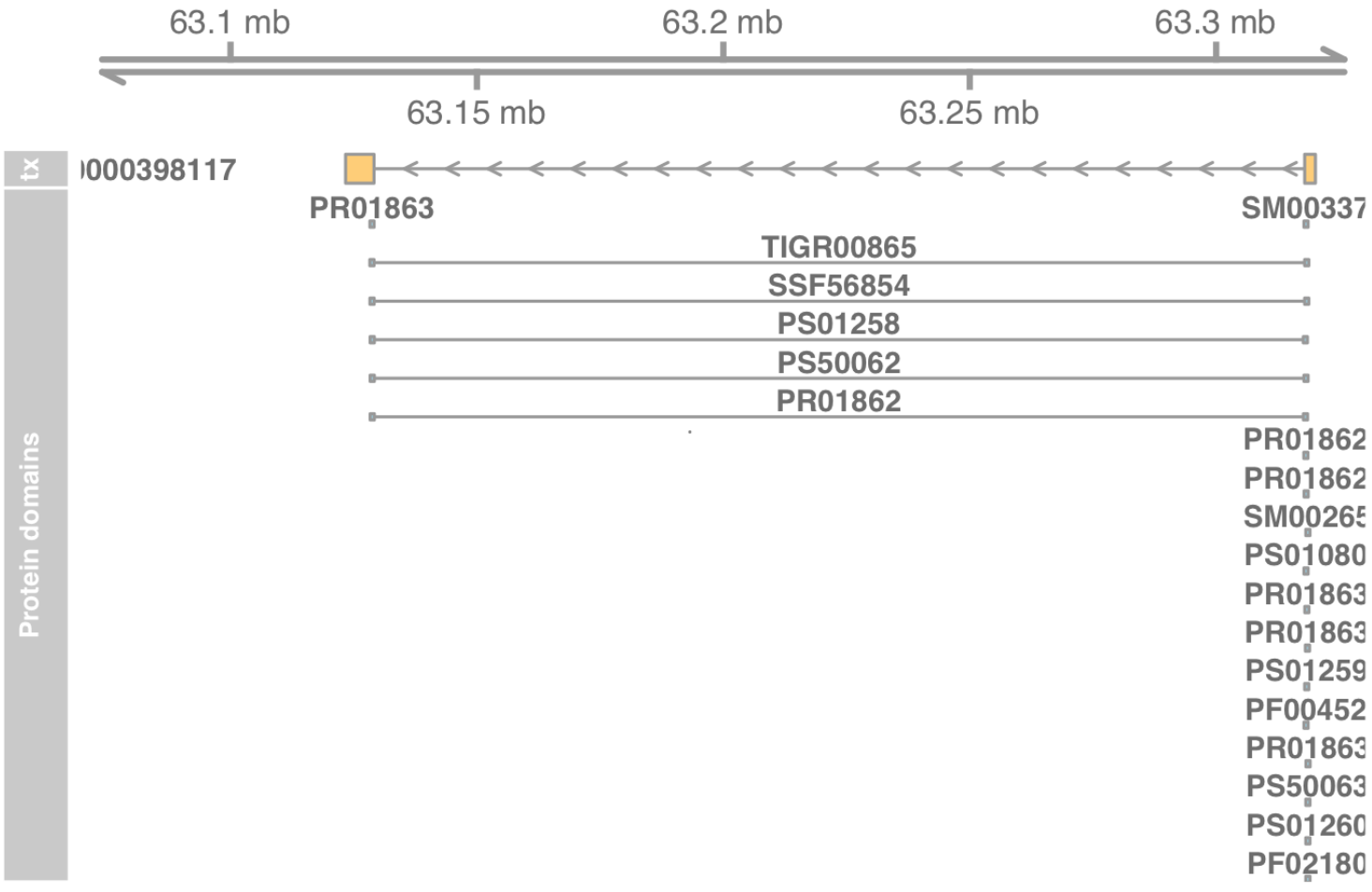
Map coordinates within proteins to the genome

- Example: use `ensembldb` to map peptide features within a protein to the genome:

```
## Map all protein domains from each protein/tx to the genome
gen_map <- mapToGenome(bcl2, edb)

## Plot the results for the first protein (transcript)
txid <- gen_map[[1]]$tx_id

## Get the gene region track for the first transcript
tx <- getGeneRegionTrackForGviz(edb, filter = ~ tx_id == txid)

## Extract the mapping for the first protein and add a protein ID column
map_1 <- gen_map[[1]]
map_1$id <- names(map_1)
```

Map coordinates within proteins to the genome

```
plotTracks(list(GenomeAxisTrack(), GeneRegionTrack(tx, name = "tx"),
                AnnotationTrack(map_1, groupAnnotation = "id", just.group = "above",
                                name = "Protein domains")),
           transcriptAnnotation = "transcript")
```



Getting annotations for feature counting

- Example: feature counting using `GenomicAlignments`' `summarizeOverlaps`:

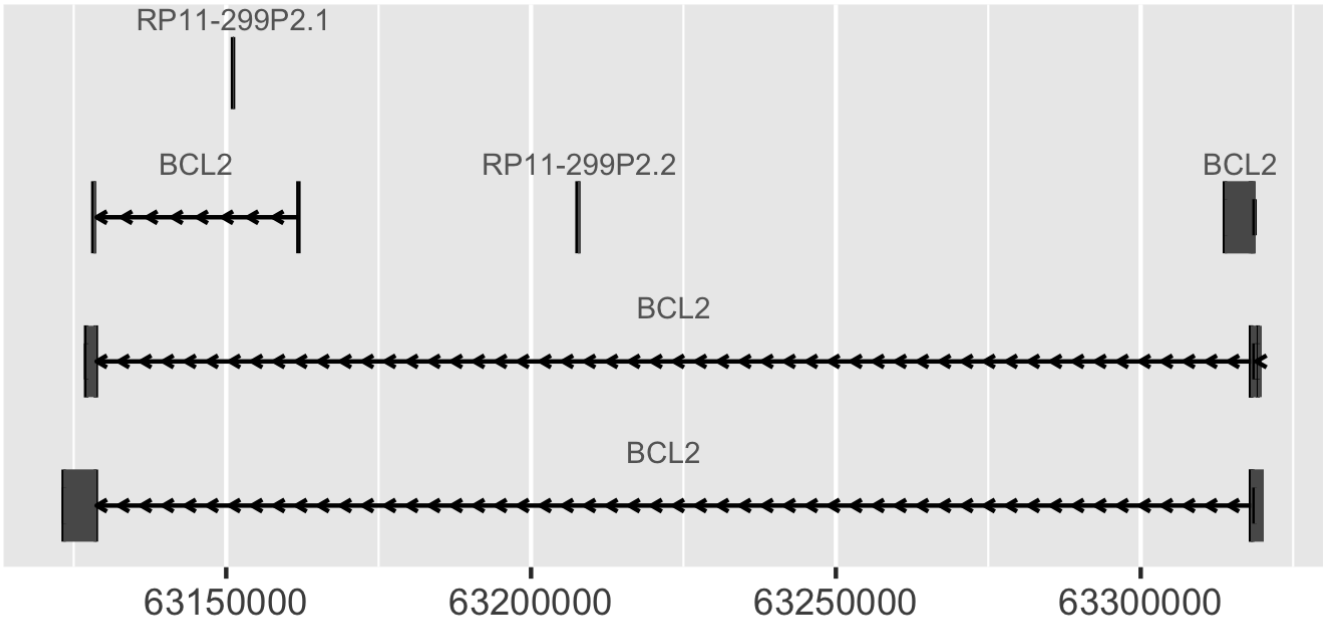
```
## Need a GRangesList of GRanges, one per gene.
## Get exons for all lincRNA genes.
exns <- exonsBy(edb, filter = ~ gene_biotype == "lincRNA", by = "gene")
exns
```

```
GRangesList object of length 7842:
$ENSG00000115934
GRanges object with 2 ranges and 2 metadata columns:
      seqnames              ranges strand |         exon_id gene_biotype
 <Rle>            <IRanges>  <Rle> |     <character>  <character>
  [1]       12 [23251461, 23251499]      - | ENSE00002222490      lincRNA
  [2]       12 [23181334, 23182623]      - | ENSE00002300099      lincRNA

$ENSG00000122043
GRanges object with 7 ranges and 2 metadata columns:
      seqnames              ranges strand |         exon_id gene_biotype
  [1]       13 [29935905, 29936224]      + | ENSE00001543395      lincRNA
  [2]       13 [29936516, 29936701]      + | ENSE00001543394      lincRNA
  [3]       13 [29937202, 29937281]      + | ENSE00001485711      lincRNA
  [4]       13 [29937906, 29938009]      + | ENSE00003595600      lincRNA
  [5]       13 [29942097, 29942567]      + | ENSE00001857117      lincRNA
  [6]       13 [29947603, 29947771]      + | ENSE00003521124      lincRNA
  [7]       13 [29950317, 29950488]      + | ENSE00000827436      lincRNA

$ENSG00000122548
GRanges object with 2 ranges and 2 metadata columns:
      seqnames              ranges strand |         exon_id gene_biotype
```

Plotting annotation data

- EnsDb integrated into ggbio.
- Example: use ggbio and ensembldb to plot a chromosomal region.

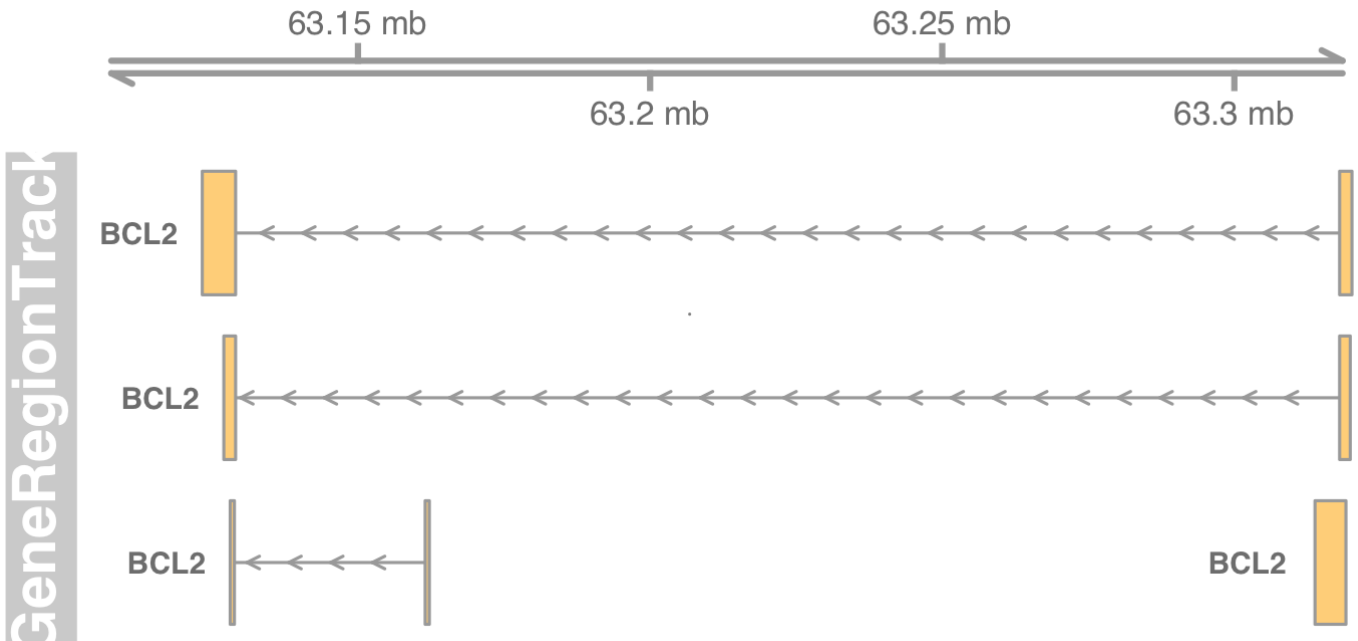
```
library(ggbio)
## Define the chromosomal region
gr <- GRanges(seqnames = 18, ranges = IRanges(63123000, 63320300))
autoplot(edb, GRangesFilter(gr), names.expr = "gene_name")
```



Plotting annotation data

- Gviz: use the getGeneRegionTrackForGviz.

```
library(Gviz)
grt <- getGeneRegionTrackForGviz(edb, filter = ~ genename == "BCL2")
plotTracks(list(GenomeAxisTrack(), GeneRegionTrack(grt)),
           transcriptAnnotation = "symbol")
```



AnnotationDbi integration

- EnsDb databases support keys, select, mapIds.
- Methods support passing filter expressions or filter objects with parameter keys.
- Example: Retrieve all data for the gene *BCL2*:

```
head(select(edb, keys = ~ genename == "BCL2"))
```

```
  ENTREZID          EXONID EXONIDX EXONSEQEND EXONSEQSTART    GENEBIOTYPE
1      596 ENSE00001531678       1   63320128     63318082 protein_coding
2      596 ENSE00001531678       1   63320128     63318082 protein_coding
3      596 ENSE00001531678       1   63320128     63318082 protein_coding
4      596 ENSE00001531678       1   63320128     63318082 protein_coding
5      596 ENSE00001531678       1   63320128     63318082 protein_coding
6      596 ENSE00001531678       1   63320128     63318082 protein_coding
   GENEID GENENAME GENESEQEND GENESEQSTART INTERPROACCESSION ISCIRCULAR
1 ENSG00000171791     BCL2   63320128     63123346         IPR002475          0
2 ENSG00000171791     BCL2   63320128     63123346         IPR002475          0
3 ENSG00000171791     BCL2   63320128     63123346         IPR003093          0
4 ENSG00000171791     BCL2   63320128     63123346         IPR003093          0
5 ENSG00000171791     BCL2   63320128     63123346         IPR020731          0
6 ENSG00000171791     BCL2   63320128     63123346         IPR020731          0
  PROTDOMEND PROTDOMSTART PROTEINDOMAINID PROTEINDOMAINSOURCE       PROTEINID
1        197           97         PS50062               pfscan ENSP00000381185
2        197           97         PS50062               pfscan ENSP00000381185
3         30           11         PS50063               pfscan ENSP00000381185
4         30           11         PS50063               pfscan ENSP00000381185
5         30           10         PS01260          scanprosite ENSP00000381185
```

Finally. . .

Thank you for your attention!