Building and Using Ensembl-based Annotation Packages with ensembldb

Johannes Rainer[1]

June 26, 2016

[1]johannes.rainer@eurac.edu

Clone me @GitHub:
`http://github.com/jotsetung/Bioc2016-ensembldb`.

- `TxDb` objects from `GenomicFeatures` provide gene model annotations:
  - Used for RNA-seq, CHiP-seq, etc.
  - Providing mostly UCSC annotations.
- `ensembldb` package defines the `EnsDb` class:
  - Same functionality as `TxDb` objects, plus:
  - Designed for Ensembl: all genes, attributes *gene biotype* and *tx biotype.*
  - Allows to query specific annotations using a simple filter framework.
- Available methods to extract data:
  - `genes`
  - `transcripts`
  - `transcriptsBy`
  - exons

- exonsBy
- cdsBy
- fiveUTRsByTranscripts
- threeUTRsByTranscripts

- Example: get all genes' annotations.

```
## Load an EnsDb package matching Ensembl version 81
library(EnsDb.Hsapiens.v81)
edb <- EnsDb.Hsapiens.v81

## Now just get all genes
genes(edb)
```

```
  ...
ENSG00000185220          1 [248906196, 248919946]      + | ENSG00000185220
ENSG00000200495          1 [248912690, 248912795]      - | ENSG00000200495
ENSG00000233084          1 [248936581, 248937043]      + | ENSG00000233084
  gene_name    entrezid         gene_biotype seq_coord_system
<character> <character>          <character>      <character>
ENSG00000278806  AF065393.4                           miRNA         scaffold
ENSG00000210049       MT-TF                       Mt_tRNA       chromosome
ENSG00000211459     MT-RNR1                       Mt_rRNA       chromosome
    ...         ...         ...                     ...              ...
ENSG00000185220       PGBD2      267002       protein_coding       chromosome
ENSG00000200495  RNU6-1205P                           snRNA       chromosome
ENSG00000233084   RPL23AP25             processed_pseudogene       chromosome
-------
seqinfo: 338 sequences from GRCh38 genome
```

- Example: get all genes encoded on chromosome Y.

```
## Retrieve genes encoded on chromosome Y.
## Create a filter object
sf <- SeqnameFilter("Y")

## Retrieve the data.
genes(edb, filter=sf)
```

```
  ...
ENSG00000237917        Y [26594851, 26634652]      - | ENSG00000237917
ENSG00000231514        Y [26626520, 26627159]      - | ENSG00000231514
ENSG00000235857        Y [56855244, 56855488]      + | ENSG00000235857
  gene_name    entrezid           gene_biotype
<character> <character>            <character>
LRG_186     LRG_186        1438               LRG_gene
ENSG00000251841  RNU6-1334P                           snRNA
ENSG00000184895         SRY        6736         protein_coding
    ...         ...         ...                     ...
ENSG00000237917     PARP4P1             unprocessed_pseudogene
ENSG00000231514     FAM58CP               processed_pseudogene
ENSG00000235857     CTBP2P1               processed_pseudogene
seq_coord_system
     <character>
LRG_186       chromosome
ENSG00000251841       chromosome
ENSG00000184895       chromosome
    ...               ...
ENSG00000237917       chromosome
ENSG00000231514       chromosome
ENSG00000235857       chromosome
-------
seqinfo: 1 sequence from GRCh38 genome
```

- Use of filters can speed up queries.
- For genes:
  - `GeneidFilter`
  - `GenenameFilter`
  - `EntrezidFilter`
  - `GenebiotypeFilter`
- For transcripts:
  - `TxidFilter`
  - `TxbiotypeFilter`
- For exons:
  - `ExonidFilter`
  - `ExonrankFilter`
- *Generic* filters:
  - `SeqnameFilter`
  - `SeqstrandFilter`
  - `SeqstartFilter`

- SeqendFilter
- GRangesFilter: condition can be *within* or *overlapping*.

- Multiple filters are combined with a logical *AND*.
- Each filter supports 1:n values and also a *like* condition.
- Example: combine filters.

```
## Example for a GRangesFilter:
grf <- GRangesFilter(GRanges(17, IRanges(59000000, 59200000)),
                     condition="within")

## Get all genes encoded in this region.
genes(edb, filter=grf, columns=c("gene_name", "gene_biotype"))

## Combine with a GenebiotypeFilter to get all genes in the region
## EXCEPT pre-miRNAs and snRNAs.
genes(edb, filter=list(grf,
                       GenebiotypeFilter(c("miRNA", "snRNA"),
                                         condition="!=")))
```

```
GRanges object with 7 ranges and 3 metadata columns:
  seqnames              ranges strand |         gene_id
     <Rle>           <IRanges>  <Rle> |     <character>
  ENSG00000263558       17 [59059226, 59059493]      + | ENSG00000263558
  ENSG00000224738       17 [59106598, 59118267]      + | ENSG00000224738
  ENSG00000182628       17 [59109951, 59155269]      - | ENSG00000182628
  ENSG00000252212       17 [59129276, 59129458]      - | ENSG00000252212
  ENSG00000211514       17 [59137758, 59137872]      - | ENSG00000211514
  ENSG00000207996       17 [59151136, 59151221]      - | ENSG00000207996
```

```
 ENSG00000266537       17 [59174983, 59181787]      - | ENSG00000266537
   gene_name           gene_biotype
 <character>            <character>
 ENSG00000263558   RN7SL716P              misc_RNA
 ENSG00000224738  AC099850.1             antisense
 ENSG00000182628        SKA2         protein_coding
 ENSG00000252212    RNU2-58P                 snRNA
 ENSG00000211514      MIR454                 miRNA
 ENSG00000207996     MIR301A                 miRNA
 ENSG00000266537    SPDYE22P unprocessed_pseudogene
 -------
 seqinfo: 1 sequence from GRCh38 genome
GRanges object with 4 ranges and 5 metadata columns:
 seqnames               ranges strand |         gene_id
    <Rle>            <IRanges>  <Rle> |     <character>
 ENSG00000263558       17 [59059226, 59059493]      + | ENSG00000263558
 ENSG00000224738       17 [59106598, 59118267]      + | ENSG00000224738
 ENSG00000182628       17 [59109951, 59155269]      - | ENSG00000182628
 ENSG00000266537       17 [59174983, 59181787]      - | ENSG00000266537
   gene_name    entrezid           gene_biotype
 <character> <character>            <character>
 ENSG00000263558   RN7SL716P                           misc_RNA
 ENSG00000224738  AC099850.1                          antisense
 ENSG00000182628        SKA2      348235         protein_coding
 ENSG00000266537    SPDYE22P              unprocessed_pseudogene
 seq_coord_system
      <character>
 ENSG00000263558       chromosome
 ENSG00000224738       chromosome
 ENSG00000182628       chromosome
 ENSG00000266537       chromosome
 -------
 seqinfo: 1 sequence from GRCh38 genome
```

- EnsDb support all AnnotationDbi methods with filters.
- Example: use AnnotationDbi's select method to fetch annotations.

```
## Get all data for the gene SKA2
Res <- select(edb, keys="SKA2", keytype="GENENAME")
head(Res, n=3)

## Or: pass filters with keys parameter to have more control:
## For the gene SKA2: get all exons except exons 1 and 2
## for all tx targeted for nonsense mediated decay.
select(edb, keys=list(GenenameFilter("SKA2"),
                      TxbiotypeFilter("nonsense_mediated_decay"),
                      ExonrankFilter(1:2, condition="!=")))
```

```
  ENTREZID          EXONID EXONIDX EXONSEQEND EXONSEQSTART    GENEBIOTYPE
1   348235 ENSE00001324111       1   59155269     59155131 protein_coding
2   348235 ENSE00003636954       2   59131367     59131281 protein_coding
3   348235 ENSE00003478713       3   59119495     59119319 protein_coding
   GENEID GENENAME GENESEQEND GENESEQSTART ISCIRCULAR SEQCOORDSYSTEM
1 ENSG00000182628     SKA2   59155269     59109951          0     chromosome
2 ENSG00000182628     SKA2   59155269     59109951          0     chromosome
3 ENSG00000182628     SKA2   59155269     59109951          0     chromosome
  SEQLENGTH SEQNAME SEQSTRAND      TXBIOTYPE TXCDSSEQEND TXCDSSEQSTART
1  83257441      17        -1 protein_coding    59155163      59112277
2  83257441      17        -1 protein_coding    59155163      59112277
3  83257441      17        -1 protein_coding    59155163      59112277
     TXID          TXNAME TXSEQEND TXSEQSTART
1 ENST00000330137 ENST00000330137 59155269   59109951
2 ENST00000330137 ENST00000330137 59155269   59109951
3 ENST00000330137 ENST00000330137 59155269   59109951
  ENTREZID          EXONID EXONIDX EXONSEQEND EXONSEQSTART    GENEBIOTYPE
```

```
1   348235 ENSE00002710994       3   59124428     59124307 protein_coding
2   348235 ENSE00003552567       4   59119495     59119319 protein_coding
3   348235 ENSE00002729093       5   59112345     59111890 protein_coding
4   348235 ENSE00003594135       3   59119495     59119319 protein_coding
5   348235 ENSE00002695019       4   59112345     59112262 protein_coding
   GENEID GENENAME GENESEQEND GENESEQSTART ISCIRCULAR SEQCOORDSYSTEM
1 ENSG00000182628     SKA2   59155269     59109951          0     chromosome
2 ENSG00000182628     SKA2   59155269     59109951          0     chromosome
3 ENSG00000182628     SKA2   59155269     59109951          0     chromosome
4 ENSG00000182628     SKA2   59155269     59109951          0     chromosome
5 ENSG00000182628     SKA2   59155269     59109951          0     chromosome
  SEQLENGTH SEQNAME SEQSTRAND              TXBIOTYPE TXCDSSEQEND TXCDSSEQSTART
1  83257441      17        -1 nonsense_mediated_decay    59155163      59124363
2  83257441      17        -1 nonsense_mediated_decay    59155163      59124363
3  83257441      17        -1 nonsense_mediated_decay    59155163      59124363
4  83257441      17        -1 nonsense_mediated_decay    59155083      59119474
5  83257441      17        -1 nonsense_mediated_decay    59155083      59119474
     TXID          TXNAME TXSEQEND TXSEQSTART
1 ENST00000578519 ENST00000578519 59155182   59111890
2 ENST00000578519 ENST00000578519 59155182   59111890
3 ENST00000578519 ENST00000578519 59155182   59111890
4 ENST00000583976 ENST00000583976 59155177   59112262
5 ENST00000583976 ENST00000583976 59155177   59112262
```

- exonsBy: provide gene model information for feature counting.

- Example: feature counting using GenomicAlignments' summarizeOverlaps method.

```r
## Get exons by gene, for chromosomes 1:22, X, Y, excluding also locus reference
## genomic genes (LRG)
exns <- exonsBy(edb, by="gene", filter=list(SeqnameFilter(c(1:22, "X", "Y")),
                                            GeneidFilter("ENSG%", "like")))
exns

## Load the required libraries.
library(GenomicAlignments)
library(BiocParallel)

## Get the Bam files.
bfl <- BamFileList(dir("data/bam", pattern=".bam$", full.names=TRUE),
                   asMates=TRUE, yieldSize=1e+6, obeyQname=TRUE)
## Define a ScanBamParam with a mapping quality filter.
sbp <- ScanBamParam(mapqFilter=30)

## Do the gene counting
geneCounts <- bplapply(bfl, FUN=summarizeOverlaps, features=exns,
                       mode="IntersectionStrict", ignore.strand=TRUE,
                       singleEnd=FALSE, fragments=TRUE, param=sbp)
geneCounts <- do.call(cbind, geneCounts)
```

- Example: gene models for Rsubread'2 `featureCount` function.

```
## Convert the exon list to SAF format
saf <- toSAF(exns)

head(saf)

####
##  Do the feature counting using the Rsubread package
library(Rsubread)
bamf <- dir("data/bam", pattern=".bam$", full.names=TRUE)
cnts <- featureCounts(files=bamf, annot.ext=saf, isPairedEnd=TRUE, nthreads=1)
```

- UCSC and Ensembl use different chromosome naming styles.

- Example: How to integrate Ensembl based annotation with UCSC data?

```r
## Get chromosome names
head(seqlevels(edb))
## Different from UCSC style: chr1...

## Get genes on chromosome Y, UCSC style.
genes(edb, filter=SeqnameFilter("chrY"))

## Solution: change the chromosome naming style:
seqlevelsStyle(edb) <- "UCSC"

## Get chromosome names
head(seqlevels(edb))

genes(edb, filter=SeqnameFilter("chrY"))


## Use case:
## Get mRNA sequences for SKA2 using BSgenome.
library(BSgenome.Hsapiens.UCSC.hg38)  ## <- UCSC based

## Get exons by transcript
ska2tx <- exonsBy(edb, by="tx", filter=GenenameFilter("SKA2"))

## Use GenomicFeatures' extractTranscriptSeqs
head(extractTranscriptSeqs(BSgenome.Hsapiens.UCSC.hg38, ska2tx))


## Alternative (preferred) way:
seqlevelsStyle(edb) <- "Ensembl"
## Using AnnotationHub:
## Get the genomic fasta file matching the package's genome version:
faf <- getGenomeFaFile(edb)
extractTranscriptSeqs(faf, exonsBy(edb, by="tx",
                                   filter=GenenameFilter("SKA2")))
```

```
[1] "1"  "10" "11" "12" "13" "14"
GRanges object with 0 ranges and 5 metadata columns:
   seqnames    ranges strand |     gene_id   gene_name    entrezid gene_biotype
      <Rle> <IRanges>  <Rle> | <character> <character> <character>  <character>
   seq_coord_system
<character>
  -------
  seqinfo: no sequences
[1] "chr1"  "chr10" "chr11" "chr12" "chr13" "chr14"
Warning message:
In .formatSeqnameByStyleFromQuery(x, sn, ifNotFound) :
  More than 5 seqnames with seqlevels style of the database (Ensembl) could not be mapped to t
    ...
  ENSG00000237917     chrY [26594851, 26634652]      - | ENSG00000237917
  ENSG00000231514     chrY [26626520, 26627159]      - | ENSG00000231514
  ENSG00000235857     chrY [56855244, 56855488]      + | ENSG00000235857
    gene_name    entrezid           gene_biotype
  <character> <character>            <character>
  LRG_186     LRG_186        1438              LRG_gene
  ENSG00000251841  RNU6-1334P                           snRNA
  ENSG00000184895         SRY        6736         protein_coding
      ...         ...         ...                   ...
  ENSG00000237917     PARP4P1            unprocessed_pseudogene
  ENSG00000231514     FAM58CP              processed_pseudogene
  ENSG00000235857     CTBP2P1              processed_pseudogene
  seq_coord_system
       <character>
  LRG_186       chromosome
  ENSG00000251841       chromosome
  ENSG00000184895       chromosome
      ...               ...
  ENSG00000237917       chromosome
  ENSG00000231514       chromosome
```

```
  ENSG00000235857       chromosome
  -------
  seqinfo: 1 sequence from GRCh38 genome
  A DNAStringSet instance of length 6
    width seq                                                names
[1]  2798 AATGAGTGCGAGATGTTGAGTGA...AACCTACAATCCTCTTTCTAAAA ENST00000330137
[2]   625 GCCGCGGTCTGCGGAATGTCAAC...AATGAGAATAAAACGATTTAAAT ENST00000437036
[3]   689 GCGGAATGTCAACTATTCAACAT...TGTACATTTCAGTCATTCGGTAT ENST00000578105
[4]   894 GGAATGTCAACTATTCAACATGG...TATGTACATTTCAGTCATTCGGT ENST00000578519
[5]   689 GCGGAATGTCAACTATTCAACAT...TACATTTCAGTCATTCGGTATGT ENST00000580541
[6]   595 GACAGCTGTCCAATGGAGGCCCT...TTGCATCTGTTTTCTTTTTCTAA ENST00000581068
snapshotDate(): 2016-06-06
loading from cache '/Users/jo/.AnnotationHub/55651'
    '/Users/jo/.AnnotationHub/55652'
  A DNAStringSet instance of length 10
     width seq                                               names
 [1]  2798 AATGAGTGCGAGATGTTGAGTGA...ACCTACAATCCTCTTTCTAAAA ENST00000330137
 [2]   625 GCCGCGGTCTGCGGAATGTCAAC...ATGAGAATAAAACGATTTAAAT ENST00000437036
 [3]   689 GCGGAATGTCAACTATTCAACAT...GTACATTTCAGTCATTCGGTAT ENST00000578105
 [4]   894 GGAATGTCAACTATTCAACATGG...ATGTACATTTCAGTCATTCGGT ENST00000578519
 [5]   689 GCGGAATGTCAACTATTCAACAT...ACATTTCAGTCATTCGGTATGT ENST00000580541
 [6]   595 GACAGCTGTCCAATGGAGGCCCT...TGCATCTGTTTTCTTTTTCTAA ENST00000581068
 [7]   583 AACTATTCAACATGGAGGCGGAG...GAAGGGCAGATAATATATGAAT ENST00000583380
 [8]   533 GAGATGTTGAGTGACAGCTGTCC...TTTTTCTAAGTCATGATAATAT ENST00000583927
 [9]   570 GTCAACTATTCAACATGGAGGCG...TTTATGAAGAAATGGACTTGGA ENST00000583976
[10]   229 CTTAGTAAACTCAAGATAAAAAG...GAGCAGAAAGAGAGTAAGAGCC ENST00000584089
```

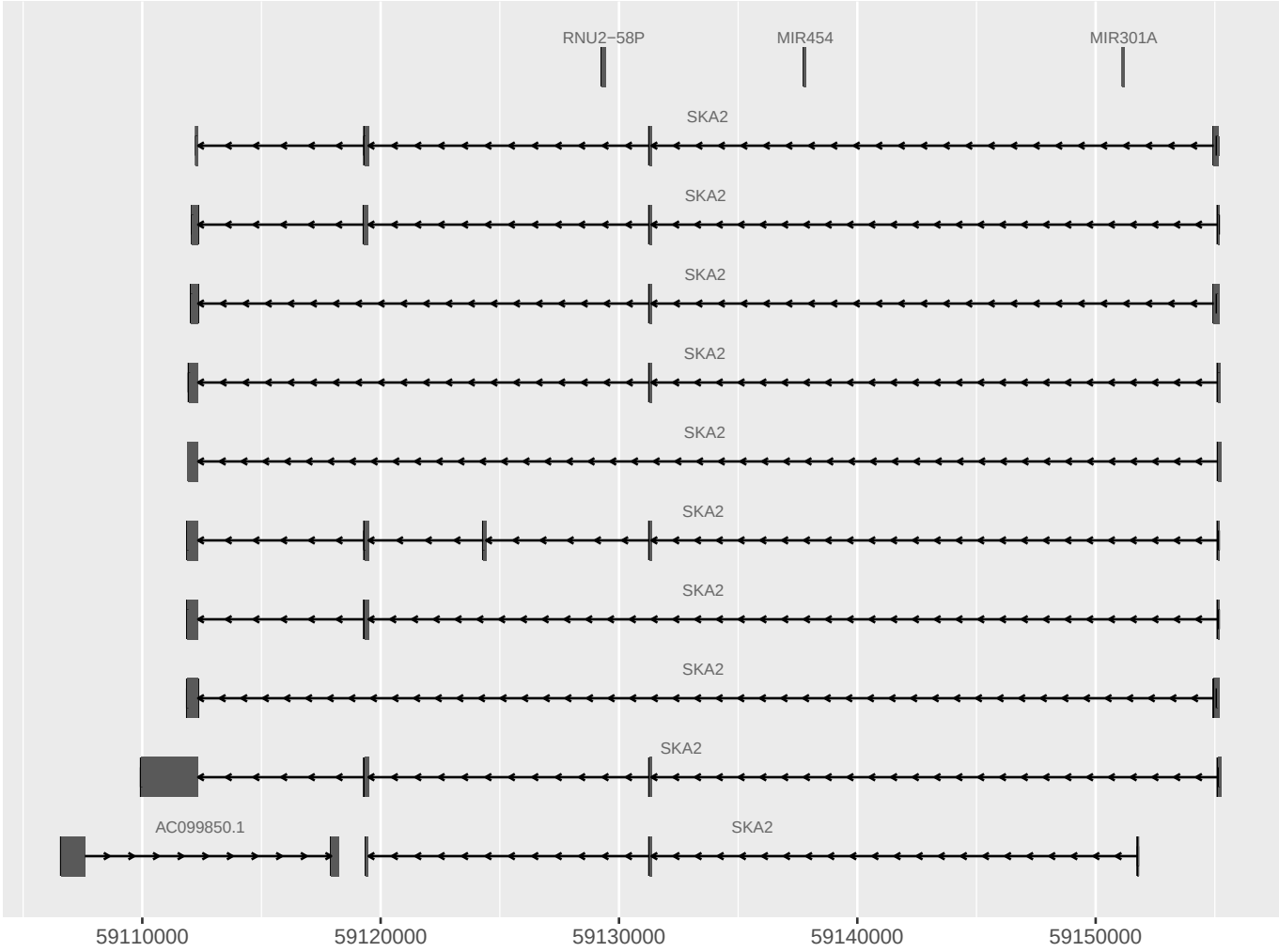
- Sequence names are mapped between *styles* using the `GenomeInfoDb` package.

- `ggbio` and `Gviz`: plot data along genomic coordinates.

- ggbio: support for EnsDb objects and filters integrated.

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

```
1  library(ggbio)
2
3  ## Plot the SKA2 gene model by passing a filter to the function.
4  autoplot(edb, GenenameFilter("SKA2"))
5
6  ## To plot all genes in the region:
7  ## Get the gene SKA2
8  ska2 <- genes(edb, filter=GenenameFilter("SKA2"))
9  strand(ska2) <- "*"
10
11 ## Plot the genomic region; we're using the gene name as labels for the transcripts.
12 autoplot(edb, GRangesFilter(ska2, condition="overlapping"),
13         names.expr="gene_name")
```



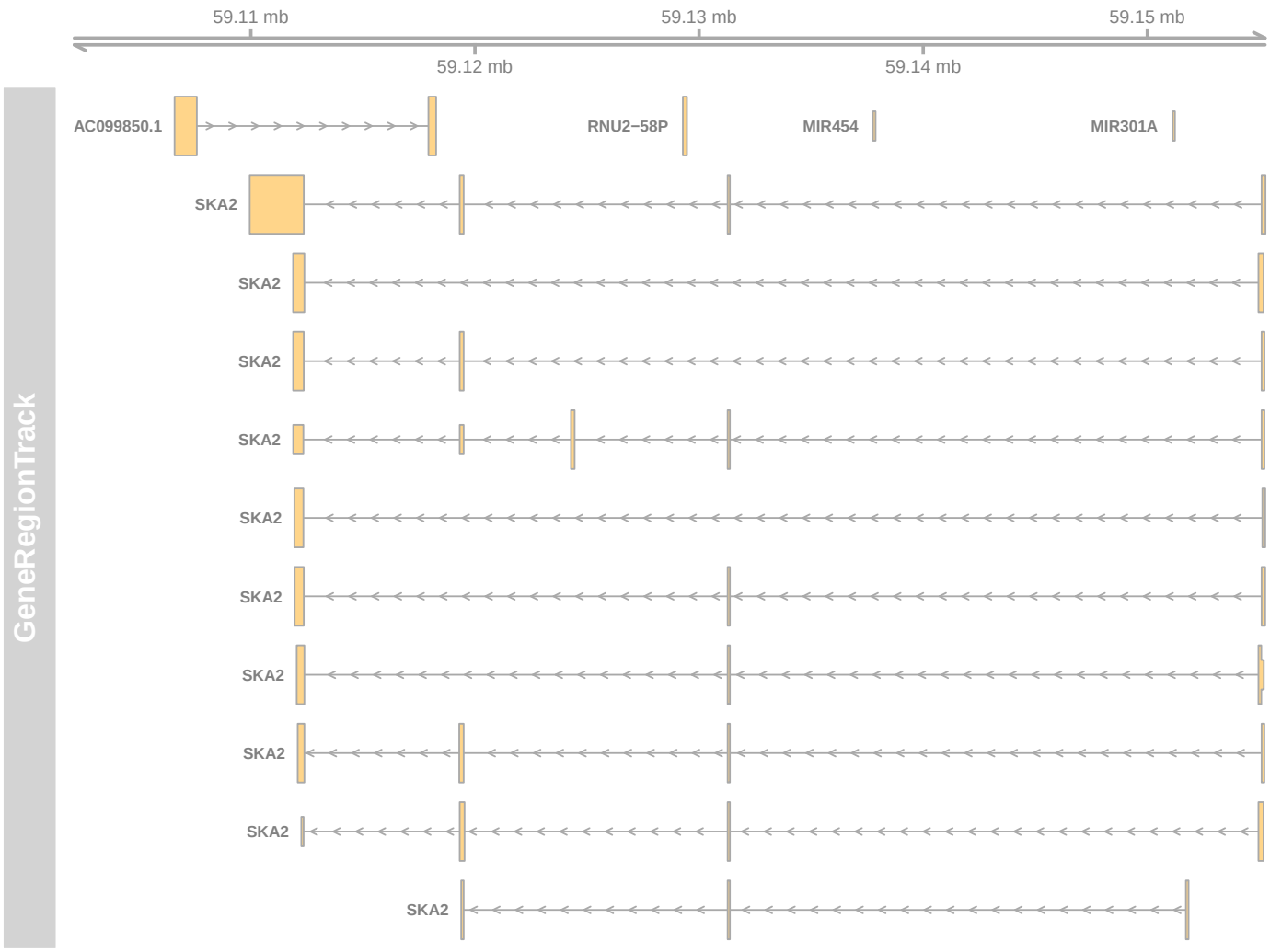
- `Gviz`: `getGeneRegionTrackForGviz` method to extract `Gviz`-formatted data.

- Example: plot genes encoded on a chromosomal region using `Gviz`.

```r
library(Gviz)

## Get all genes in the same genomic region and return as GRanges
## formatted for Gviz.
grt <- getGeneRegionTrackForGviz(edb, chromosome=seqlevels(ska2),
                                 start=start(ska2), end=end(ska2))
## Alternatively, using a GRangesFilter
strand(ska2) <- "*"
grt <- getGeneRegionTrackForGviz(edb, filter=GRangesFilter(ska2,
                                                           condition="overlapping"))

geneTrack <- GeneRegionTrack(grt)
## Plot the chromosomal region.
plotTracks(list(GenomeAxisTrack(), geneTrack), transcriptAnnotation="symbol",
           chromosome=seqlevels(ska2))
```



- The `ensembldb` shiny app allows interactive annotation look-up.

- Example: search for a gene using the shiny app and return the result to R.

```
## Run the shiny app:
Result <- runEnsDbApp()

## Inspect the result:
Result
```

- ensDbFromAH: build an EnsDb database from an AnnotationHub (gtf) resource.
- Example: create an EnsDb using AnnotationHub.

```
library(AnnotationHub)
ah <- AnnotationHub()

## Query for available Ensembl gtf files for release 83.
query(ah, pattern=c("ensembl", "release-83", "gtf"))

## Select one; in this case: Anolis carolinensis (lizard)
edbSql83 <- ensDbFromAH(ah=ah["AH7537"])

## Let's see what we've got.
db <- EnsDb(edbSql83)
genes(db, filter=SeqnameFilter("2"))

## Make a package.
makeEnsembldbPackage(ensdb=edbSql83, version="1.0.0",
                     maintainer="Johannes Rainer <johannes.rainer@eurac.edu>",
                     author="J Rainer")
```

- But: no NCBI Entrez Gene IDs available.
- ensDbFromGtf: create an EnsDb from a *gtf* or *gff* file.

- *Should* work with all gtf and gff files from Ensembl.
- But: gtf files don't provide NCBI Entrez Gene IDs.
- Example: create an `EnsDb` from a GTF file downloaded from `ftp://ftp.ensembl.org`.

```
## Create an EnsDb from an Ensembl GTF file.

## Create the SQLite database file:
##  o Eventually define 'organism' and 'genomeVersion'.
##  o Needs also an internet connection to retrieve the 'seqlengths'.
edbSql <- ensDbFromGtf("data/gtf/Canis_familiaris.CanFam3.1.84.gtf.gz")

edbSql

## Use the makeEnsembldbPackage to create a package, or load and use it.
dogDb <- EnsDb(edbSql)

dogDb

## Fully functional, except we don't have Entrez gene ids.
head(genes(dogDb, filter=SeqnameFilter("X")))
```

- Requires:
  - Perl.
  - Ensembl Perl API (and Bioperl).
- `fetchTablesFromEnsembl` to fetch the annotations from Ensembl.

- `makeEnsemblSQLiteFromTables` to create the SQLite database from the tables.
- `makeEnsembldbPackage` to create a package containing and providing the annotation.
- Example: create an `EnsDb` using the Perl API.

```
## Create an EnsDb using the Ensembl Perl API:

## This takes quite some time...
fetchTablesFromEnsembl(version="81",
                       ensemblapi="/Users/jo/ensembl/81/API/ensembl/modules",
                       species="dog")

## Create an SQLite database from the generated txt files
dbf <- makeEnsemblSQLiteFromTables()

## Finally, create the package
makeEnsembldbPackage(ensdb=dbf, version="1.0.0",
                     maintainer="Johannes Rainer <johannes.rainer@eurac.edu>",
                     author="Johannes Rainer")
```

Thank you for your attention!