One script to rule them all

Ugo Borello, Eleonora Sara Bacci

2025-09-14

If you work in bioinformatics, you must have realized that there are more genome browsers or repositories from which you can retrieve information about different organism's genomes.

The four more common database are `Ensembl`, `UCSC`, `NCBI Refseq` and `NCBI ENTREZ`.

First, let's load some libraries

```
library("biomaRt")
library(gplots)
```

In R is common to use the biomaRt package to access the data available in Ensembl, a freely available online service that allows you to download genomic annotations from several databases.

Let's connect to Ensembl's human genes.

```
ensembl <- useMart("ensembl",dataset="hsapiens_gene_ensembl")
ensembl
```

```
## Object of class 'Mart':
##   Using the ENSEMBL_MART_ENSEMBL BioMart database
##   Using the hsapiens_gene_ensembl dataset
```

In this way we now have an 'Mart' object that contains the genome of our interest and from which we can retrive some information.

Which are the attributes (column) that identifies this object? 'listAttributes' will shows all attributes of the data imported from enseml, but since there are a lot (3,170), we will just focus on the first 6, using head().

```
attributes <- listAttributes(ensembl)
head(attributes)
```

```
##                             name                 description         page
## 1              ensembl_gene_id              Gene stable ID feature_page
## 2      ensembl_gene_id_version      Gene stable ID version feature_page
## 3        ensembl_transcript_id        Transcript stable ID feature_page
## 4 ensembl_transcript_id_version Transcript stable ID version feature_page
## 5           ensembl_peptide_id           Protein stable ID feature_page
## 6   ensembl_peptide_id_version   Protein stable ID version feature_page
```

Of this attributes which are the ones containing the word 'entrez'?

```
grep(pattern="entrez", x=attributes$description, ignore.case=T)
```

```
## [1] 59 78 79 80
```

There are 4 attributes that have 'entrez' in their name. Which one of them is the gene ID for the entrez/NCBI repository?

```
lista_entrez <- list(attributes[59,], attributes[78,],
                     attributes[79,], attributes[80,])
do.call(rbind.data.frame, lista_entrez)
```

```
##                       name                                    description
## 59  entrezgene_trans_name              EntrezGene transcript name ID
## 78   entrezgene_accession   NCBI gene (formerly Entrezgene) accession
## 79          entrezgene_id          NCBI gene (formerly Entrezgene) ID
## 80 entrezgene_description NCBI gene (formerly Entrezgene) description
##            page
## 59 feature_page
## 78 feature_page
## 79 feature_page
## 80 feature_page
```

```
#it's this one!
attributes[80,]
```

```
##                       name                                    description
## 80 entrezgene_description NCBI gene (formerly Entrezgene) description
##            page
## 80 feature_page
```

Now that we have a match with the ensembl ID and the entrez ID, we can do the same for other two databases.

NCBI Refseq...

```
grep(pattern="refseq", x=attributes$description, ignore.case=T)
```

```
## [1] 23 24 85 86 87 88 89 90
```

```
lista <-list(attributes[23,], attributes[24,], attributes[85,],
             attributes[86,], attributes[87,], attributes[88,],
             attributes[89,], attributes[90,])
do.call(rbind.data.frame, lista)
```

```
##                             name                                      description
## 23        transcript_mane_select        RefSeq match transcript (MANE Select)
## 24 transcript_mane_plus_clinical RefSeq match transcript (MANE Plus Clinical)
## 85                    refseq_mrna                                  RefSeq mRNA ID
## 86         refseq_mrna_predicted                     RefSeq mRNA predicted ID
## 87                  refseq_ncrna                              RefSeq ncRNA ID
## 88        refseq_ncrna_predicted                   RefSeq ncRNA predicted ID
## 89                refseq_peptide                           RefSeq peptide ID
## 90      refseq_peptide_predicted                 RefSeq peptide predicted ID
##            page
## 23 feature_page
## 24 feature_page
## 85 feature_page
## 86 feature_page
## 87 feature_page
## 88 feature_page
## 89 feature_page
## 90 feature_page
```

```
#we need this one in this case
attributes[85,]
```

```
##           name    description         page
## 85 refseq_mrna RefSeq mRNA ID feature_page
```

. . . and UCSC.

```
grep(pattern="ucsc", x=attributes$description, ignore.case=T)
```

```
## [1] 96
```

```
attributes[95,] #This one was easy, since it was only one
```

```
##               name        description         page
## 95 hgnc_trans_name Transcript name ID feature_page
```

how many genes are in common in all databases?

First some preliminary data manipulation

We create a character vector, containing the number 1 to 22, and the letters M, X, Y. This vector allow us to names the human chromosomes.

```
my_chr <- c(1:22, 'M', 'X', 'Y')
```

```
my_chr
```

```
##  [1] "1"  "2"  "3"  "4"  "5"  "6"  "7"  "8"  "9"  "10" "11" "12" "13" "14" "15"
## [16] "16" "17" "18" "19" "20" "21" "22" "M"  "X"  "Y"
```

So we just created a vector that summarize the chromosomes that we can find in the genome!

To answer our question we use a specific biomaRt function called getBM, that is the main query function in biomaRt.

The getBM function has four main arguments:

- attributes: is a vector of attributes that one wants to retrieve (= the output of the query)
- filters: is a vector of filters that one will use as input to the query
- values: a vector of values for the filters. In case multiple filters are in use, the values argument requires a list of values where each position in the list corresponds to the position of the filters in the filters argument
- mart: is and object of class Mart, which is created by the useMart function.

With this function we can create new dataframe for each different genomic Database, that contains the list of all gene ID per database.

```
my_refseq_mrna <- getBM(attributes = 'refseq_mrna',
                        filters = 'chromosome_name',
                        values = my_chr,
                        mart = ensembl)
head(my_refseq_mrna)
```

```
##    refseq_mrna
## 1    NM_145177
## 2    NM_006883
## 3    NM_000451
## 4    NM_022148
## 5 NM_001012288
## 6 NM_001171038
```

```
my_entrez_gene <- getBM(attributes = 'entrezgene_id',
                        filters = 'chromosome_name',
                        values = my_chr,
                        mart = ensembl
```

```
)
```

```
head(my_entrez_gene)
```

```
##   entrezgene_id
## 1        207063
## 2     102464837
## 3     100359394
## 4          6473
## 5     100500894
## 6         64109
```

```
my_ucsc <- getBM(attributes = 'ucsc',
                 filters = 'chromosome_name',
                 values = my_chr,
                 mart = ensembl)
head(my_ucsc)
```

```
##         ucsc
## 1 uc317bog.1
## 2 uc317bof.1
## 3 uc317bri.1
## 4 uc317brh.1
## 5 uc317brg.1
## 6 uc317brf.1
```

```
my_ensembl_gene_id <- getBM(attributes = 'ensembl_gene_id',
                            filters = 'chromosome_name',
                            values = my_chr,
                            mart = ensembl
)
head(my_ensembl_gene_id)
```

```
##   ensembl_gene_id
## 1 ENSG00000290825
## 2 ENSG00000223972
## 3 ENSG00000310526
## 4 ENSG00000227232
## 5 ENSG00000278267
## 6 ENSG00000243485
```

How many records are stored in each database?

```
length(my_ensembl_gene_id[,1])
```

```
## [1] 78654
```

```
# in 2013 was 52322
```

```
length(my_entrez_gene[,1])
```

```
## [1] 28254
```

```
# in 2013 was 24231
```

```
length(my_refseq_mrna[,1])
```

```
## [1] 67036
```

```
# in 2013 was 35423
```

```
length(my_ucsc[,1])
```

```
## [1] 385199
```

```
# in 2013 was 53598
```

We can see that almost in all the repository the number of records incresead in the last 10 years!

But how many of this are in common?

To do so it's better to combine the four newly created dataframe, into one. So this new dataframe will have the gene IDs of the four repositories divided per column. Using the chromosome vector, generated before, we match their entries.

```
my_chr <- c(1:22, 'M', 'X', 'Y')

my_annotation <- getBM(attributes = c('ucsc', 'ensembl_gene_id',
                                      'refseq_mrna', 'entrezgene_id'),
                       filters = 'chromosome_name',
                       values = my_chr,
                       mart = ensembl
)

head(my_annotation)
```

```
##         ucsc ensembl_gene_id refseq_mrna entrezgene_id
## 1 uc317bog.1 ENSG00000292328                       NA
## 2 uc317bof.1 ENSG00000292329                       NA
## 3 uc317bri.1 ENSG00000292337                       NA
## 4 uc317brh.1 ENSG00000292338   NM_145177        207063
## 5 uc317brg.1 ENSG00000292338                    207063
## 6 uc317brf.1 ENSG00000292338                    207063
```

Let's substitute the blank lines with NAs and label the rows as sequential numbers.

```
my_annotation <- sapply(my_annotation,gsub,pattern="^$",replacement=NA)

row.names(my_annotation) <- 1:length(my_annotation[,1])


head(my_annotation)
```

```
##   ucsc          ensembl_gene_id   refseq_mrna entrezgene_id
## 1 "uc317bog.1" "ENSG00000292328" NA          NA
## 2 "uc317bof.1" "ENSG00000292329" NA          NA
## 3 "uc317bri.1" "ENSG00000292337" NA          NA
## 4 "uc317brh.1" "ENSG00000292338" "NM_145177" "207063"
## 5 "uc317brg.1" "ENSG00000292338" NA          "207063"
## 6 "uc317brf.1" "ENSG00000292338" NA          "207063"
```

```
#We now need a function to take in a row name and list of values.
#Also we need to check for NAs in the list of values.
#The function will return the row name or NA depending on the list of values.

my_function <- function(x, ...){
    to_return <- sapply(list(...), is.na)
    to_return <- gsub(pattern=FALSE, replacement=x, x=to_return)
    to_return <- gsub(pattern=TRUE, replacement=NA, x=to_return)
}
```

This is the function at work

```
my_venn <- my_annotation

for (i in 1:length(my_annotation[,1])){
    my_venn[i,] <- my_function(row.names(my_annotation)[i], my_annotation[i,])
}

head(my_venn)
```

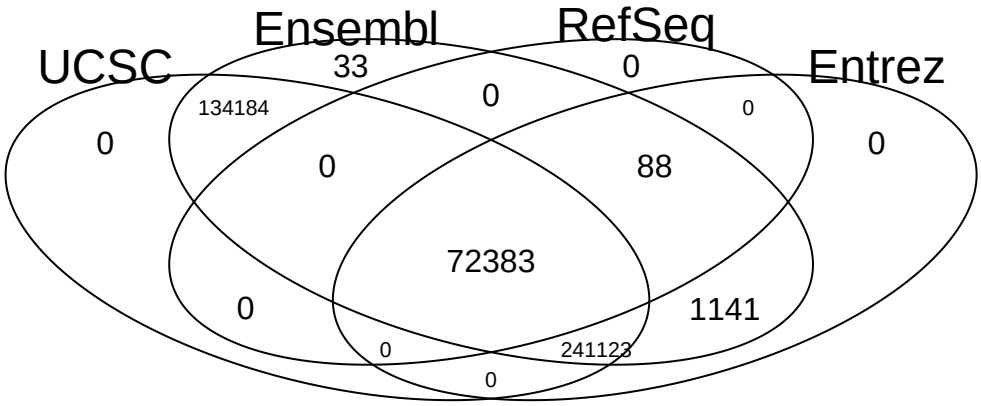
```
##   ucsc ensembl_gene_id refseq_mrna entrezgene_id
## 1 "1"  "1"             NA          NA
## 2 "2"  "2"             NA          NA
## 3 "3"  "3"             NA          NA
## 4 "4"  "4"             "4"         "4"
## 5 "5"  "5"             NA          "5"
## 6 "6"  "6"             NA          "6"
```

From this dataframe we can estract the rows for each of thee repositories.

```
my_venn_ucsc <- my_venn[,1]
my_venn_ucsc <- as.vector(na.omit(my_venn_ucsc))

my_venn_ensembl <- my_venn[,2]
my_venn_ensembl <- as.vector(na.omit(my_venn_ensembl))

my_venn_refseq <- my_venn[,3]
my_venn_refseq <- as.vector(na.omit(my_venn_refseq))

my_venn_entrez <- my_venn[,4]
my_venn_entrez <- as.vector(na.omit(my_venn_entrez))
```

We can finally plot a Venn diagram to find out how many genes are in common from the database or which are exclusively from that database.

```
VennList<-list(UCSC = my_venn_ucsc,
               Ensembl = my_venn_ensembl,
               RefSeq = my_venn_refseq,
               Entrez = my_venn_entrez
               )
venn(VennList)
```



The main observations from the Venn diagram are

- 3919 annotations are common between Ensembl and Entrez but missing in RefSeq and UCSC.(in 2013 were 17940)
- 166308 annotations are common between Ensembl, Entrez and UCSC but missing from RefSeq. (in 2013 were 33750)
- 71790 annotations are common to all databases.(in 2013 were 38115)

So when choosing which gene database to use for annotation, it's best to understand how the database is constructed and which type of information does it contain in order to get the best results!

N.B. this tutorial is adapted from a Dave Tang's blog post