Zebrafish BiomaRt package tutorial

```
# the biomaRt package isn't installed on the VMs
# This checks for the package and installs it if
# it isn't already installed
if (!require("biomaRt", quietly = TRUE)) {
    BiocManager::install("biomaRt")
}
```

Load the `biomaRt` package.

```
library(tidyverse)
library(biomaRt)
```

To connect to Ensembl you need to select which mart you want and which dataset.

This is the equivalent of setting the Database and choosing a dataset when using BioMart on the Ensembl website.

Generally you will want ‘genes’, but there is also variation data (‘snps’) and regulation data (‘regulation’).

```
# you can list the different biomarts with
listEnsembl()
```

```
##         biomart                version
## 1         genes      Ensembl Genes 107
## 2 mouse_strains      Mouse strains 107
## 3          snps  Ensembl Variation 107
## 4    regulation Ensembl Regulation 107
```

To find the available datasets, you can use the `listDatasets()` function.

```
# this just shows the top 5
# If you want to look through the full list replace
# magrittr::extract(1:5,) with View()
listDatasets(useEnsembl(biomart = "genes")) %>%
  magrittr::extract(1:5,)
```

```
##                        dataset                           description
## 1 abrachyrhynchus_gene_ensembl Pink-footed goose genes (ASM259213v1)
## 2     acalliptera_gene_ensembl      Eastern happy genes (fAstCal1.2)
## 3   acarolinensis_gene_ensembl       Green anole genes (AnoCar2.0v2)
## 4    acchrysaetos_gene_ensembl       Golden eagle genes (bAquChr1.2)
## 5    acitrinellus_gene_ensembl        Midas cichlid genes (Midas_v5)
##       version
## 1 ASM259213v1
## 2  fAstCal1.2
## 3 AnoCar2.0v2
## 4  bAquChr1.2
## 5    Midas_v5
```

The example below connects to the zebrafish BioMart.

```
ensembl_dr <- useEnsembl(biomart = "genes",
                         dataset = "drerio_gene_ensembl")
```

It's also possible to connect to archive versions. (More info **here**)

```
# access an older version
ensembl_dr_102 <-
  useEnsembl(biomart = "genes",
             dataset = "drerio_gene_ensembl",
             version = 102,
             host = "https://feb2021.archive.ensembl.org")
```

If you have human Ensembl ids to search with you will want to load the Human dataset.

```
# load human dataset
ensembl_hs <- useEnsembl(biomart = "genes",
                         dataset = "hsapiens_gene_ensembl")
```

To do a BioMart query, you need a set of filters and a set of attributes which mirror the options in the web version of BioMart.

To get data frames of all possible filters and attributes do this:

```
filters = listFilters(ensembl_dr)
View(filters)

attributes = listAttributes(ensembl_dr)
View(attributes)
```

Here are some examples of how to look up specific type of attributes. For example, we could look for all the available attributes on the Homologs page or all the terms that start with "hsapiens_".

```
# find attributes from the homologs page
filter(attributes, page == "homologs") %>% View()

# find attributes that contain hsapiens_
filter(attributes, grepl("hsapiens_", name))

# biomaRt provides the searchFilters function to search
# filters as well
# searchFilters(mart = ensembl, pattern = "ensembl.*id")
```

Here I've made a vector of attributes about Human homologs.

```
required_cols <- c(
  'ensembl_gene_id', 'external_gene_name',
  'chromosome_name', 'start_position',
  'end_position', 'strand', 'description',
  'hsapiens_homolog_ensembl_gene',
  'hsapiens_homolog_associated_gene_name',
  'hsapiens_homolog_orthology_type',
  'hsapiens_homolog_perc_id',
  'hsapiens_homolog_perc_id_r1',
  'hsapiens_homolog_orthology_confidence')
```

Here I'm using the Amp dataset as a list of genes to search with.

```
zf_genes <- read_tsv('data/Amp.sig.tsv',
                     show_col_types = FALSE) %>%
  pull(Gene)
```

We run the query with the `getBM` function. The attributes are from the required_cols vector from above, the filters argument is 'ensembl_gene_id' and the Ensembl gene values to use are the Ensembl ids from the Amp dataset (zf_genes), `getBM` also requires the mart object. In this case `ensembl_dr`.

`getBM` returns a data.frame.

Here I've also filtered the results for high confidence orthologs.

```
biomart_results <- getBM(attributes = required_cols,
      filters = c('ensembl_gene_id'),
      values = zf_genes,
      mart = ensembl_dr)

# subset to high confidence orthologs
biomart_results_filtered <-
  filter(biomart_results,
         hsapiens_homolog_orthology_confidence == 1) %>%
  as_tibble()
```

More information on BioMart can be found **here**