

BiomaRt Demo

Marc-André Legault

September 19, 2014

Créer une connexion vers la BD

```
library("biomaRt")
ensembl = useMart("ensembl")

# On peut ensuite voir les jeux de données
# On utilise head pour afficher seulement quelques entrées.
datasets = listDatasets(ensembl)
head(datasets)
```

```
##               dataset
## 1      oanatinus_gene_ensembl
## 2      cporcellus_gene_ensembl
## 3      gaculeatus_gene_ensembl
## 4      lafricana_gene_ensembl
## 5 itridecemlineatus_gene_ensembl
## 6      choffmanni_gene_ensembl
##               description version
## 1  Ornithorhynchus anatinus genes (OANA5)  OANA5
## 2      Cavia porcellus genes (cavPor3) cavPor3
## 3  Gasterosteus aculeatus genes (BROADS1) BROADS1
## 4    Loxodonta africana genes (loxAfr3) loxAfr3
## 5 Ictidomys tridecemlineatus genes (spetri2) spetri2
## 6    Choloepus hoffmanni genes (choHof1) choHof1
```

```
# On peut remarquer que le dataset #31 est l'humain:
datasets[31,]
```

```
##               dataset               description    version
## 31 hsapiens_gene_ensembl Homo sapiens genes (GRCh37.p13) GRCh37.p13
```

```
# Choisissons donc ce dataset.
ensembl = useDataset("hsapiens_gene_ensembl", mart=ensembl)
# Alternativement, nous aurions pu le faire dès le début.
# ensembl = useMart("ensembl", dataset="hsapiens_gene_ensembl")
```

```
class(ensembl)
```

```
## [1] "Mart"
## attr(,"package")
## [1] "biomaRt"
```

Maintenant, nous avons un objet R de type “Mart” qui nous permet de rechercher dans la base de donnée sélectionnée. Pour ce faire, nous utiliserons:

- Attribute: Les champs que nous voulons sélectionner. Comme dans un **SELECT** en SQL.
- Filter: Un champ sur lequel nous voulons établir une contrainte. Comme un **WHERE** en SQL.
- Values: Les valeurs correspondant aux *filters*.

```
attributes = listAttributes(ensembl)
# Affichons certains attributs.
head(attributes, n=10)
```

```
##           name           description
## 1  ensembl_gene_id  Ensembl Gene ID
## 2  ensembl_transcript_id  Ensembl Transcript ID
## 3  ensembl_peptide_id  Ensembl Protein ID
## 4  ensembl_exon_id  Ensembl Exon ID
## 5  description      Description
## 6  chromosome_name  Chromosome Name
## 7  start_position  Gene Start (bp)
## 8  end_position    Gene End (bp)
## 9  strand          Strand
## 10 band            Band
```

Nous pouvons effectuer des requêtes avec la fonction getBM.

```
chromYGenes <- getBM(
  attributes=c("ensembl_gene_id", "external_gene_id", "chromosome_name", "start_position", "end_position"),
  filters=c("chromosome_name"),
  values=c("Y"),
  mart=ensembl
)
head(chromYGenes)
```

```
##  ensembl_gene_id external_gene_id chromosome_name start_position
## 1 ENSG00000226555      AGKP1           Y      16750976
## 2 ENSG00000228787      NLGN4Y-AS1        Y      16905522
## 3 ENSG00000236131      MED13P1          Y      17019778
## 4 ENSG00000227949      CYCSP46           Y      17053427
## 5 ENSG00000224518      AC006989.2        Y      17053626
## 6 ENSG00000234620      HDHD1P1           Y      17460542
##  end_position
## 1      16752238
## 2      16915913
## 3      17019948
## 4      17053630
## 5      17054595
## 6      17567954
```

Vous ferez la même chose avec la liste de SNPs fournis.

Essayez d'aller chercher les scores SIFT et PolyPhen pour ces variants.