Making 3'UTR files for additional species

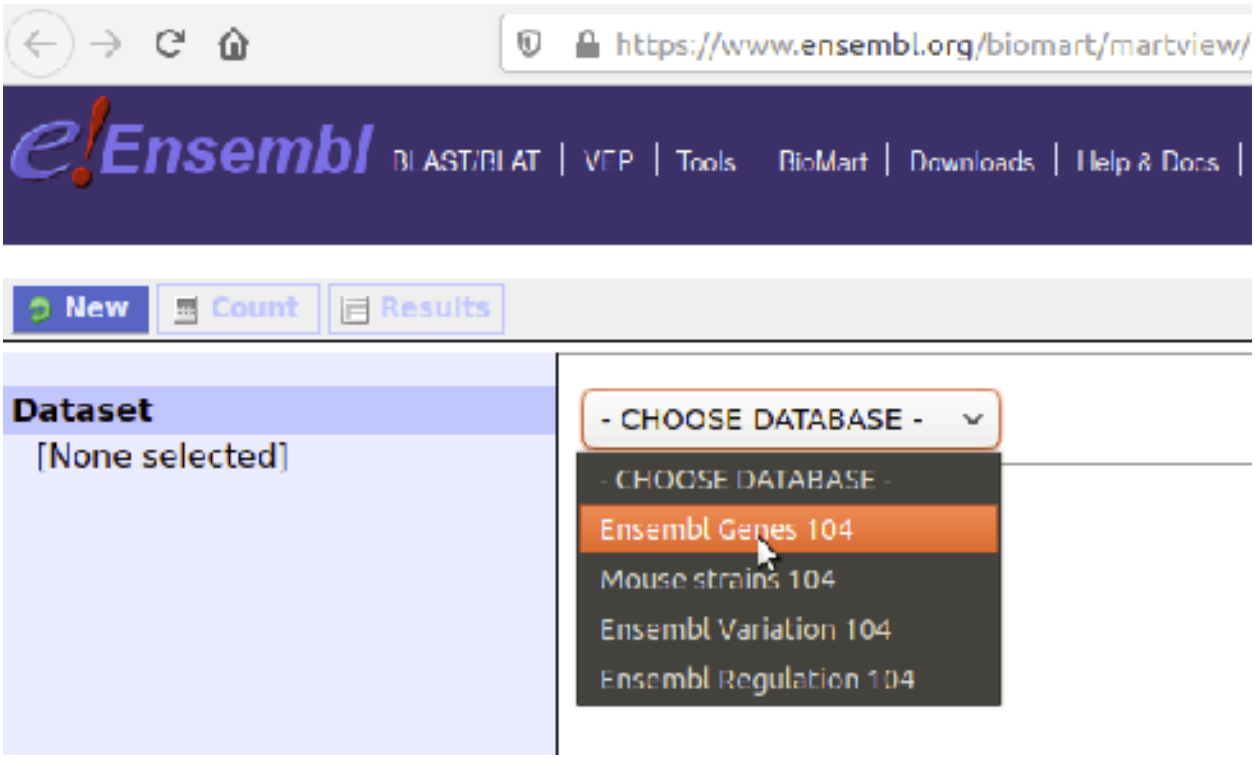
Morten Muhlig Nielsen

5/6/2021

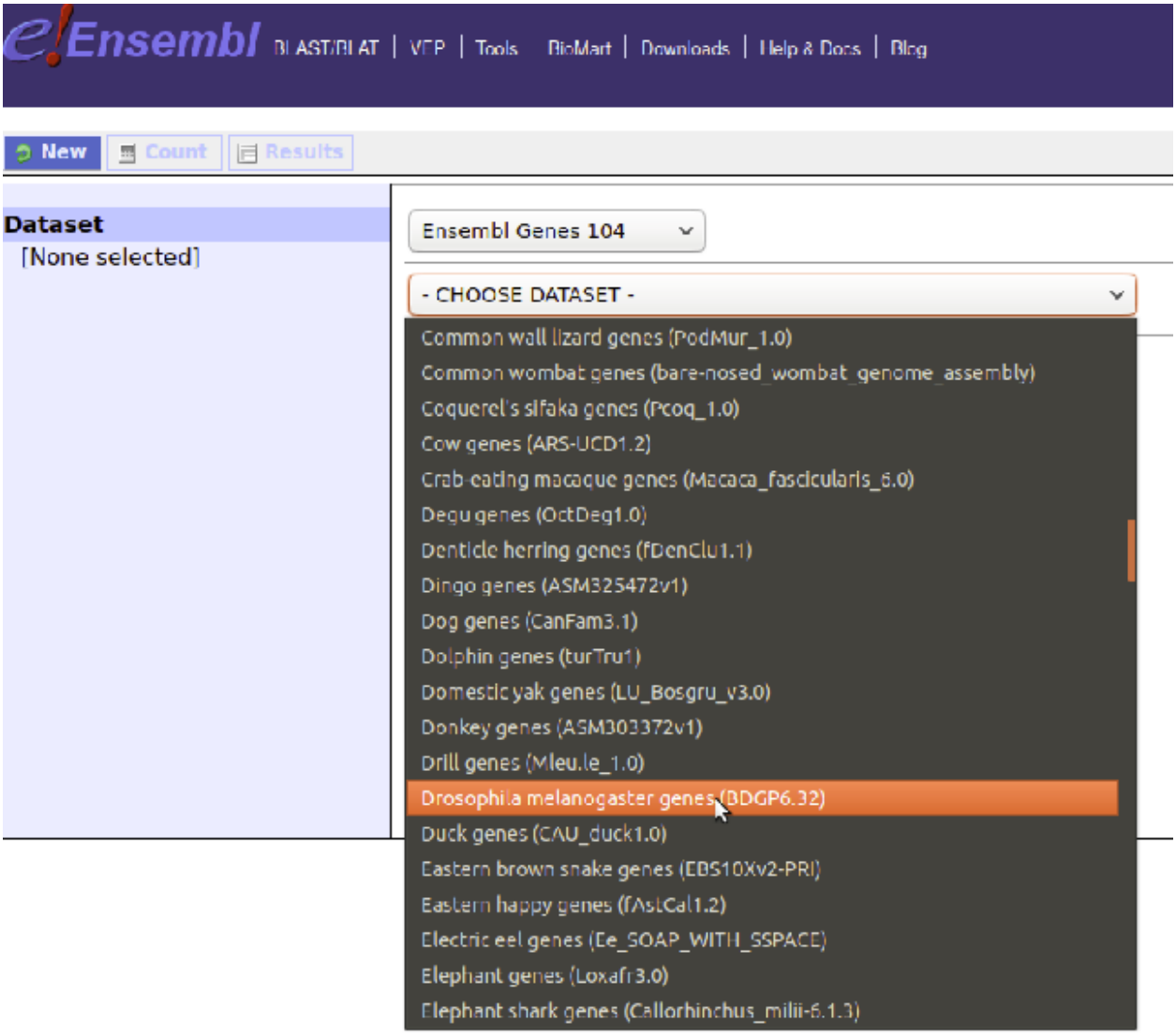
Obtain sequences from Ensembl

In a browser, navigate to https://www.ensembl.org/biomart/martview/.

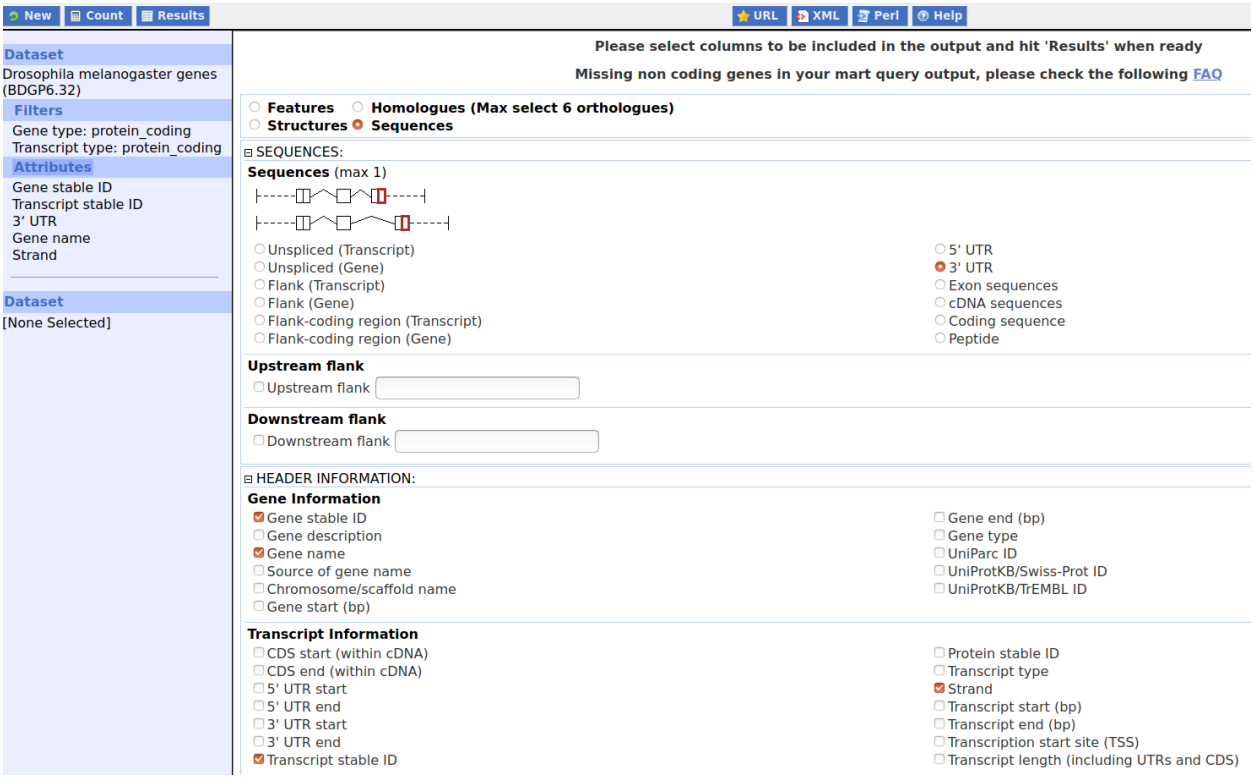
Choose 'Ensembl Genes' under 'CHOOSE DATABASE'. At the time of writing, the version was 104.



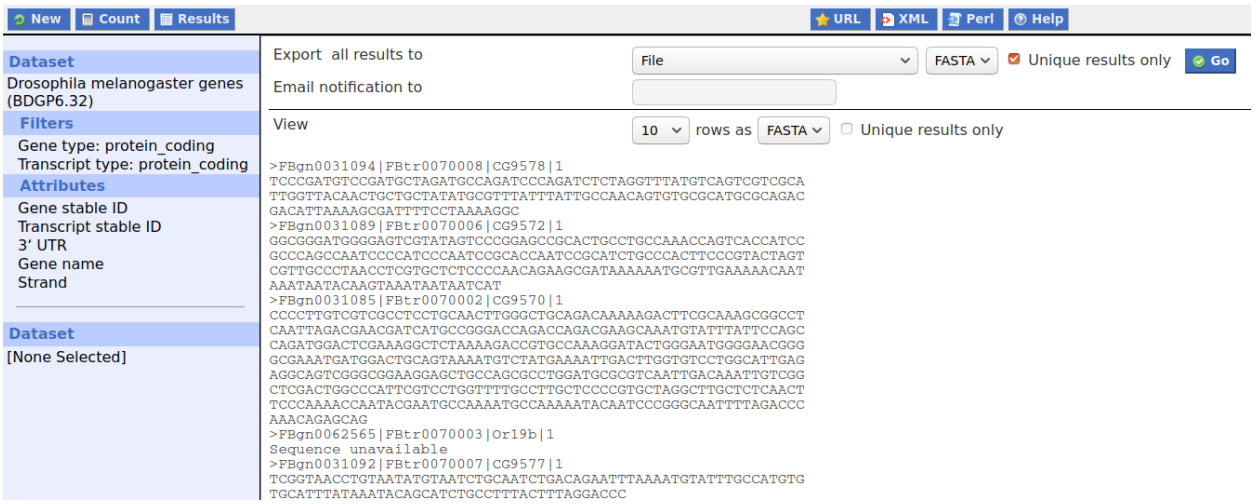
Now choose the species of interest:



Add a filter so that only protein coding genes are returned, and choose the attributes '3'UTR sequence', 'Gene Stable ID', 'Transcript Stable ID', 'Strand' and 'Gene Name'.



Export the results as a fasta file by pressing the 'Results' button. You can now download the fasta file by pressing 'Go'. You may choose to export a zipped file and get notified by mail when it is ready.



Save the file to an appropriate location. Now use the command line to create a one line fasta file:

```
cat /path/to/mart_export.txt | awk '/^>/&&NR>1{print "";}{ printf "%s",/^>/ ? $0" ":$0 }'
| awk '{print($1" "$2)}'> /desired/path/to/oneLineFasta.fa
# the cryptic step is to avoid three fields when sequence unavailable.
```

Now do the following in R to produce the seq dataframe and seqlist needed in miReact:

```
#################################
fastafile <- "/path/to/oneLineFasta.fa"
seqs <- read.table(as.is=TRUE, sep="|", comment.char = "", quote = "", header=F, fastafile)
dim(seqs)
```

```
# checkup that there are four columns,
# of which the last contains the sequence as well as the strand direction.
colnames(seqs) <- c("gid","tid","gsym","sequence")
head(seqs$gid) # check
seqs$gid<-sub(">","",seqs$gid) # remove the fasta '>' separator.
seqs$strand <- sub(" .*","",seqs$sequence) # Fetch the strand from the 'sequence' col.
table(seqs$strand) # should be '1' and '-1' only
seqs$sequence <- sub("^-","",seqs$sequence) # remove strand info from sequence field.
seqs$sequence <- sub("1 ","",seqs$sequence)
seqs$sequence<-toupper(seqs$sequence) # sequence to upper case.
seqs$nchar <- nchar(seqs$sequence) # sequence lengths
hist(seqs$nchar) # look at seq length distribution
min(seqs$nchar)
sum(seqs$nchar==1) # 44
head(seqs[grep("S",seqs$sequence),]) # See if some have 'SEQUENCE' dummy annotation.
sum(!grepl("S",seqs$sequence))
seqs$sequence <- sub("^SEQUENCE","",seqs$sequence) # make these have length 0
seqs$nchar <- nchar(seqs$sequence) # re-define sequence lengths accordingly.
hist(seqs$nchar) # checks...
min(seqs$nchar)
sum(seqs$nchar==0)
# get rid og seqs < 20 and > 10000
seqs <- seqs[seqs$nchar>20&seqs$nchar<10000,]
dim(seqs) # check reduction in size.
# order by gene ID and length
seqs <- seqs[order(seqs$gid,-seqs$nchar),]
# and save
saveRDS(seqs, file="/path/to/miReact/installation/folder/seqs/dm.utr3.seqs.rds")
# save to the miReact installation folder in the sub folder 'seqs'
# Note that the species name. here 'dm', for drosophila melanogaster,
# is needed as input in the 'species' parameter in miReact in the downstream process...

# Produce the seqlist, list of 3'UTR sequences with nucleotide probabilities
# This requires the Regmex package installed:
# install it with devtools::install_github("muhligs/Regmex")
require(Regmex)
require(parallel) # to speed up, run parallel by specifying cores > 1 below
seqlist <- Regmex::seq.list.con(seqlist = seqs$sequence, cores=1)
# it is imperative that the seqlist exactly corresponds to the seqs object above,
# i.e. the sequences must match.

# Now save to the seqs installation folder.
saveRDS(seqlist, file="/path/to/miReact/installation/folder/seqs/dm.utr3.seqlist.rds")
################################
```

And thats basically it...