

# Making 3'UTR files for additional species

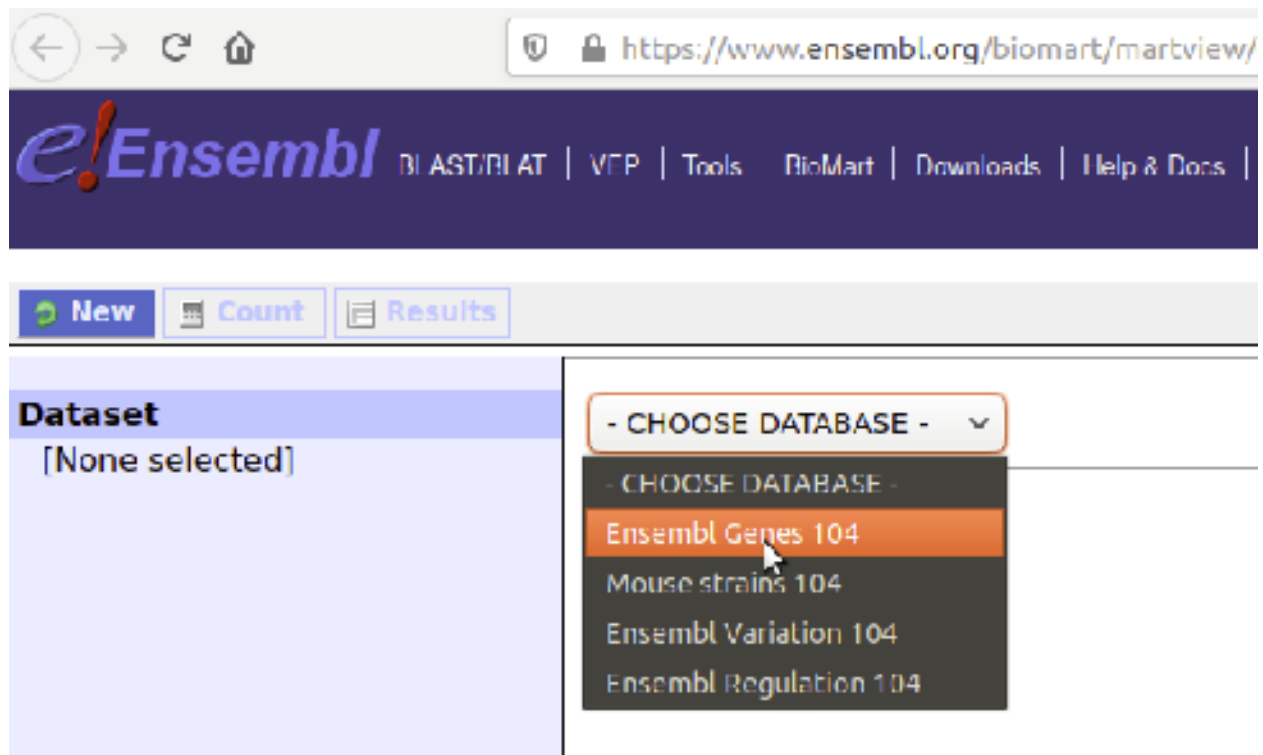
Morten Muhlig Nielsen

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:

**e!Ensembl** | [BI AST/BI AT](#) | [VFP](#) | [Tools](#) | [BioMart](#) | [Downloads](#) | [Help & Docs](#) | [Blog](#)

**New** **Count** **Results**

**Dataset**  
[None selected]

Ensembl Genes 104

- CHOOSE DATASET -

- Common wall lizard genes (PodMur\_1.0)
- Common wombat genes (bare-nosed\_wombat\_genome\_assembly)
- Coquerel's sifaka genes (Pcoq\_1.0)
- Cow genes (ARS-UCD1.2)
- Crab-eating macaque genes (Macaca\_fascicularis\_6.0)
- Deagu genes (OctDeg1.0)
- Denticle herring genes (fDenclu1.1)
- Dingo genes (ASM325472v1)
- Dog genes (CanFam3.1)
- Dolphin genes (turTru1)
- Domestic yak genes (LU\_Bosgru\_v3.0)
- Donkey genes (ASM303372v1)
- Drill genes (Mleu.le\_1.0)
- Drosophila melanogaster genes (BDGP6.32)**
- Duck genes (CAU\_duck1.0)
- Eastern brown snake genes (EB510Xv2-PR1)
- Eastern happy genes (fAslCal1.2)
- Electric eel genes (Ee\_SOAP\_WITH\_SSSPACE)
- Elephant genes (Loxafi 3.0)
- Elephant shark genes (Callorhinchus\_milii-6.1.3)

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'.

**New** **Count** **Results** **URL** **XML** **Perl** **Help**

Please select columns to be included in the output and hit 'Results' when ready  
Missing non coding genes in your mart query output, please check the following [FAQ](#)

☐ Features ☐ Homologues (Max select 6 orthologues)  
☐ Structures ☒ Sequences

**SEQUENCES:**

**Sequences (max 1)**

☐ Unspliced (Transcript)  
☐ Unspliced (Gene)  
☐ Flank (Transcript)  
☐ Flank (Gene)  
☐ Flank-coding region (Transcript)  
☐ Flank-coding region (Gene)

☐ 5' UTR  
☒ 3' UTR  
☐ Exon sequences  
☐ cDNA sequences  
☐ Coding sequence  
☐ Peptide

**Upstream flank**  
☐ Upstream flank

**Downstream flank**  
☐ Downstream flank

**HEADER INFORMATION:**

**Gene Information**

☒ Gene stable ID  
☐ Gene description  
☒ Gene name  
☐ Source of gene name  
☐ Chromosome/scaffold name  
☐ Gene start (bp)

☐ Gene end (bp)  
☐ Gene type  
☐ UniParc ID  
☐ UniProtKB/Swiss-Prot ID  
☐ UniProtKB/TrEMBL ID

**Transcript Information**

☐ CDS start (within cDNA)  
☐ CDS end (within cDNA)  
☐ 5' UTR start  
☐ 5' UTR end  
☐ 3' UTR start  
☐ 3' UTR end  
☒ Transcript stable ID

☐ Protein stable ID  
☐ Transcript type  
☒ Strand  
☐ Transcript start (bp)  
☐ Transcript end (bp)  
☐ Transcription start site (TSS)  
☐ Transcript length (including UTRs and CDS)

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.

**New** **Count** **Results** **URL** **XML** **Perl** **Help**

**Dataset**  
Drosophila melanogaster genes (BDGP6.32)

**Filters**  
Gene type: protein\_coding  
Transcript type: protein\_coding

**Attributes**  
Gene stable ID  
Transcript stable ID  
3' UTR  
Gene name  
Strand

**Dataset**  
[None Selected]

Export all results to   ☒ Unique results only **Go**

Email notification to

View  rows as  ☐ Unique results only

```
>FBgn0031094|FBtr0070008|CG9578|1
TCCGATGTCCTCCGATGCTAGATGCCAGATCCGAGTCTCTAGGTTTATGTCACTGCTCGCA
TTGGTTTAACTGCTGCTATATGCGTTTATTATTGCGCAACAGTGTGGCATGCGCAGAC
GACATTAACGCGATTTTCCTAAAGGC
>FBgn0031089|FBtr0070006|CG9572|1
GGCGGATGGGAGTCTGATAGTCCCGGAGCCGCACTGCTGCCAAACAGTCCACCATCC
GCCAGCCAATCCCATCCCAATCCGACCAATCCGCACTGCGCACTTCCGCTACTAGT
CGTGGCCTAACCTCGTGTCTCCCAACAGAGCGATAAAAAATGCGTTGAAAAACAAT
AAATAATACAGTAATAATATCAT
>FBgn0031085|FBtr0070002|CG9570|1
CCCTTGTGCTGCTCCTGCAACTTGGGCTGCAGACAAAAAGACTTCGCAAGCGGCCT
CAATTAGACGACGATCATGCGGGACCGACAGACGAAAGCAATGTATTATTCAGC
CAGATGGACTCGAAAGGCTCTAAAGACCGTGCCAAAGGATACTGGGAATGGGGAACGGG
GCGAATGATGGACTGCAGTAAATGTCTATGAAATTTGACTTGGTGTCTGGCATTGAG
AGCGAGTCGGGCGAAGGAGCTGCCAGGCGCTGGATGCGCTCAATTGACAAATTGTCCG
CTGACTGGGCCATTGCTCCTGTTTGGCTTGGCTCCCGTCTAGGCTTGTCTCAACT
TCCCAAAACCAATACGAATGCCAAATGCCAAATACAAATCCCGGGCAATTTTAGACCC
AAACAGACGAC
>FBgn0062565|FBtr0070003|Or19b|1
Sequence unavailable
>FBgn0031092|FBtr0070007|CG9577|1
TCGGTAACCTGTAATATGTAATCTGCAATCTGACAGAATTTAAATGTATTGCCATGTG
TGCATTTATAAATACAGCATCTGCCTTACTTTAGGACCC
```

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 of 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...